Sam Holroyd

CYBE 467

PenTest2

4/26/24

Pen Test 2 Report

Table of Contents

Executive Summary

This Pen Test was conducted on the target machine cybe467ctf2.1 from a kali VM. The goal of the test was to discover security weaknesses and uncover as many user credentials as possible.

Findings and Recommendations

The usernames across all four of the machines are the same. I recommend having unique usernames for each machine. Multiple passwords are the same across machines and must be changed. Passwords should be changed regularly and follow standard password rules. I also recommend implementing Least privilege as it will limit what an attacker can view if a user is compromised.

The telnet service currently running on the linux machines is a security risk. Telnet is known for multiple high level vulnerabilities that can compromise the machines. Open SSH is already running and is far superior to telnet.

The OpenSSH service needs extra layers of security. I recommend changing the port that ssh runs on, limiting the number of login attempts, and disable the use of passwords for logging in.

The webserver running on Target1 has a setup.html file that should not be in the html directory. This allows anyone to see the credentials of an admin. It also is a script to add a new user as an administrator. This script should be secured in a restricted file so that potential attackers cannot take advantage of the script.

Scope details

The scope for this PenTest will be limited to two linux machines and two windows machines. The machines that the pen test will be conducted on are: cybe467target1, cybe467target2, cybe467target3, cybe467target4. I will conduct the pen test from a kali linux instance. The pen test aims to identify vulnerabilities and potential security weaknesses within the target machines. All services will remain active during the duration of the test. All web applications hosted on the target machines will be scanned and could be exploited. All accessible network services, such as ssh, https, apache web servers, etc. will be scanned and analyzed for vulnerabilities. It is intended to thoroughly explore the file systems on each machine and attempt to escalate privileges, including attempting to log in as the root user if the password can be discovered. Password cracking software can be used to gain an initial foothold on the machines. Existing user passwords will not be changed or modified in any way. Additional users can be created to help aid in the testing process. Any files in the system will be looked at but not modified.

The tools used to assess vulnerabilities in the systems include Metasploit framework, Nmap, Hydra, and John the ripper. The Metasploit framework is used to gather information using the built-in Nmap scanner. Once the services running on each machine have been discovered, Hydra and John the ripper will be used to attempt a brute force login. To help aid with finding usernames or passwords, each machine's web server will be thoroughly scanned and exploited if able.

Methodology

Throughout the duration of the pen test all four machines will be scanned using the Metasploit framework. To begin reconnaissance on the target machines, I performed an nmap scan using the db_nmap scanner. This will allow me to view the name of the machine and any services that may be running. After the initial scan of the systems I visited each machine's website to begin looking for any vulnerabilities. Starting with the target1 machine, I used a directory enumeration tool to discover hidden directories in the website that I could then visit for additional information. A setup directory was found and the page was then explored for data that could aid in the pen test. Inside the directory there is a setup.html file that displays administrator credentials for the Target1 machine. With the credentials I was able to login as an administrator using the active ssh service, granting access to all files on the machine.

Once access to the Target1 machine was obtained, I could extract a list of users for hydra to use on the Target2 machine. Hydra was able to find a password for the user "ebrown" on the Target2 machine using the provided wordlist "kroyou.txt". Once logged into the Target2 machine with the user "ebrown", which is an administrator on the machine. I now have access to the /etc/passwd file. I can now craft another list of users to attempt a brute force attack on the Target3 machine. The users found on the Target1 machine are the same as the Target2 machine, making crafting the username list easy. Using hydra and the "kroyou.txt" wordlist on the Target3 machine, I was able to find the credentials for the user "jdoe". Now that I am logged in on the Target3 machine I can again use the userlist from earlier to crack a user's password on the Target4 machine. Now that I have access to all four machines, I can conduct a more intensive scan of the services currently running on each machine. This will allow me to find out what version of services are in use. On the linux machines systemctl commands were used to find out what services are running and what version. For the windows machines the Get-Service command is used. Now that I have gathered all the information from the machines I can begin to crack other users passwords, locate out of date applications, and find vulnerable services that could be a risk to the security of the system.

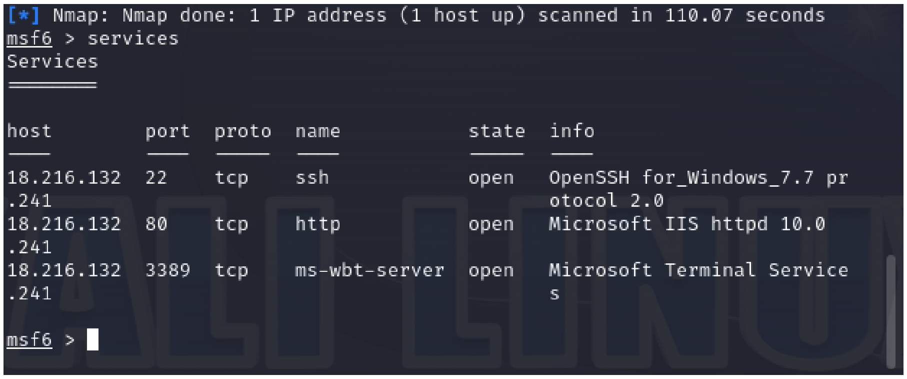
Findings

Target1

- **Services**

  The Target1 machine is a windows machine and is running three services, an OpenSSH server, a httpd windows server, and a remote desktop service. Figure 1 shows the result of the db_nmap -sV command being run from the msfconsole.

  - ➔ OpenSSH: version 7.7 for windows, running protocol 2.0
  - ➔ Microsoft IIS httpd 10.0 service
  - ➔ Microsoft Terminal Service



```
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 110.07 seconds
msf6 > services
Services
========

host         port  proto  name           state  info
----         ----  -----  ----           -----  ----
18.216.132   22    tcp    ssh            open   OpenSSH for_Windows_7.7 pr
.241                                            otocol 2.0
18.216.132   80    tcp    http           open   Microsoft IIS httpd 10.0
.241
18.216.132   3389  tcp    ms-wbt-server  open   Microsoft Terminal Service
.241                                            s

msf6 > 
```

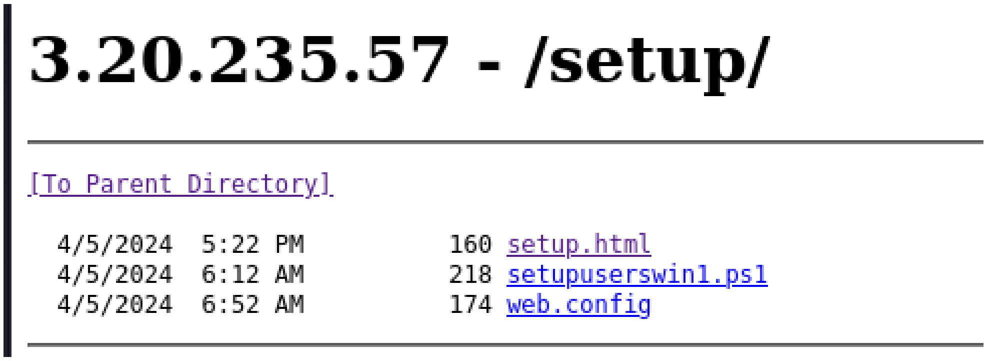
Figure 1: Target1 Nmap scan.

- **Vulnerabilities**

The OpenSSH server allows logins with passwords and is running on the default port of 22. The SSH server also does not limit the number of login attempts from a single IP address.

The httpd server that is currently running on the machine allows the user to browse directories. The setup page is available to see from the website if the user types in /setup after the IP address they will be redirected to a directory listing as seen in figure 2. Once at the directory page, a page called “setupt.html” can be accessed. From that page is html code that creates a user “landerson” with the password “Let2019!Me!In” and adds them to the administrators group. The html code can be seen in figure 3.

A remote desktop procedure should only be running when and only when it needs to be used. Leaving the service running all the time can leave the machine vulnerable to attackers.

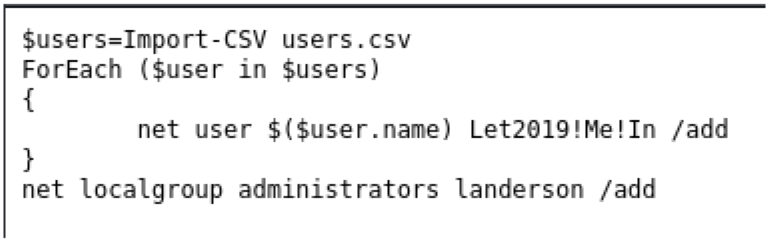


3.20.235.57 - /setup/

```
[To Parent Directory]

  4/5/2024  5:22 PM          160 setup.html
  4/5/2024  6:12 AM          218 setupuserswin1.ps1
  4/5/2024  6:52 AM          174 web.config
```

Figure 2: /setup on the apache 2 web server.



```
$users=Import-CSV users.csv
ForEach ($user in $users)
{
        net user $($user.name) Let2019!Me!In /add
}
net localgroup administrators landerson /add
```

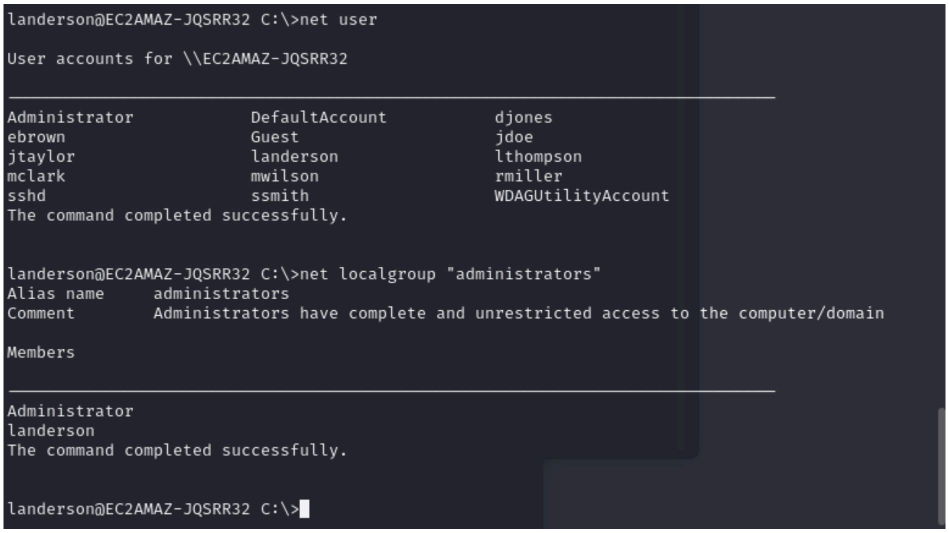
Figure 3: html code in /setup/setup.html.

- **Credentials**

Username and passwords that were cracked:

➔ landerson : “Let2019!Me!In

Users found on the machine can be seen in figure 4 and two administrator accounts were found.



```
landerson@EC2AMAZ-JQSRR32 C:\>net user

User accounts for \\EC2AMAZ-JQSRR32

-------------------------------------------------------------------------------
Administrator            DefaultAccount           djones
ebrown                   Guest                    jdoe
jtaylor                  landerson                lthompson
mclark                   mwilson                  rmiller
sshd                     ssmith                   WDAGUtilityAccount
The command completed successfully.


landerson@EC2AMAZ-JQSRR32 C:\>net localgroup "administrators"
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
landerson
The command completed successfully.


landerson@EC2AMAZ-JQSRR32 C:\>
```

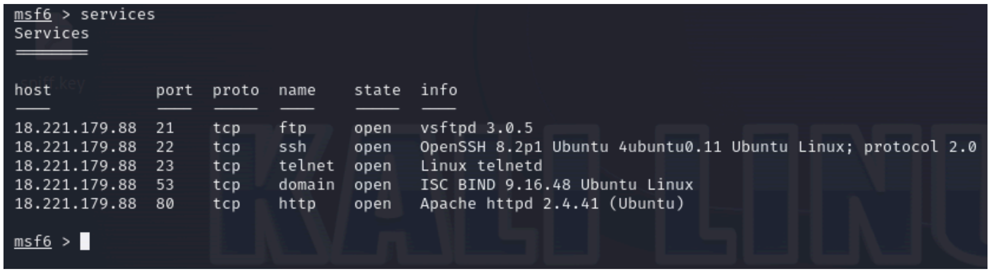
Figure 4: User list found on Target1.

Target2

- **Services**

The Target2 machine is a linux machine and is running five services: vsftpd, OpenSSH, telnet, ISC Bind and an Apache 2 web server. Figure 5 shows the results from scanning the Target2 machine.

- ➔ Vsftpd version 3.0.5
- ➔ OpenSSH for linux version 8.2 protocol 2.0
- ➔ Linux Telnet
- ➔ ISC Bind version 9.16.48
- ➔ Apache web server version 2.4.41



```
msf6 > services
Services
========

host           port  proto  name    state  info
----           ----  -----  ----    -----  ----
18.221.179.88  21    tcp    ftp     open   vsftpd 3.0.5
18.221.179.88  22    tcp    ssh     open   OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 Ubuntu Linux; protocol 2.0
18.221.179.88  23    tcp    telnet  open   Linux telnetd
18.221.179.88  53    tcp    domain  open   ISC BIND 9.16.48 Ubuntu Linux
18.221.179.88  80    tcp    http    open   Apache httpd 2.4.41 (Ubuntu)

msf6 > █
```

Figure 5: Target2 services.

- **Vulnerabilities**

The OpenSSH server allows logins with passwords and is running on the default port of 22. The SSH server also does not limit the number of login attempts from a single IP address.

The current version of vsftpd can be affected by a vulnerability found in a prior version 2.3.4 which allows backdoor command execution. This vulnerability can be found in CVE-2011-2523.

The telnet service is known for its vulnerabilities such as transmitting data in clear text and lacking encryption, amplifying the system's exposure to risks, leaving it susceptible to eavesdropping, session hijacking, and brute force attacks.

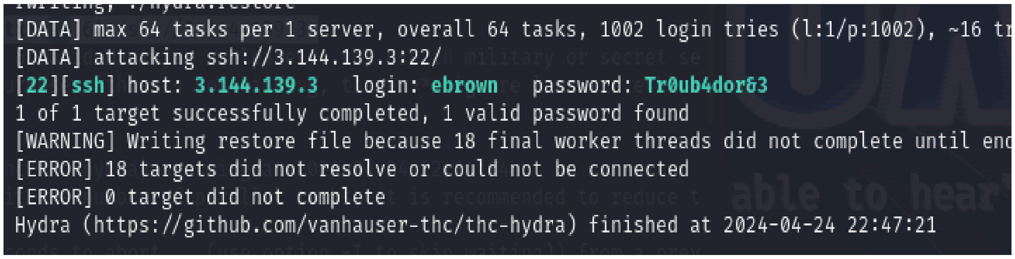
This machine also uses duplicate users from the Target1 machine. With a list of users an attacker can easily crack duplicate and insecure passwords.

The Apache 2 server that is currently running on the machine has two known vulnerabilities with version 2.4.41. CVE-2019-0211: This vulnerability affects Apache HTTP Server versions 2.4.17 to 2.4.38. It allows remote attackers to execute arbitrary code via a crafted request during file transfer using the mod_remoteip module. The second vulnerability is CVE-2019-10082: This vulnerability affects Apache HTTP Server versions 2.4.0 to 2.4.39. It involves URL normalization inconsistencies in mod_proxy_balancer that could allow remote attackers to bypass intended access restrictions.

- **Credentials**

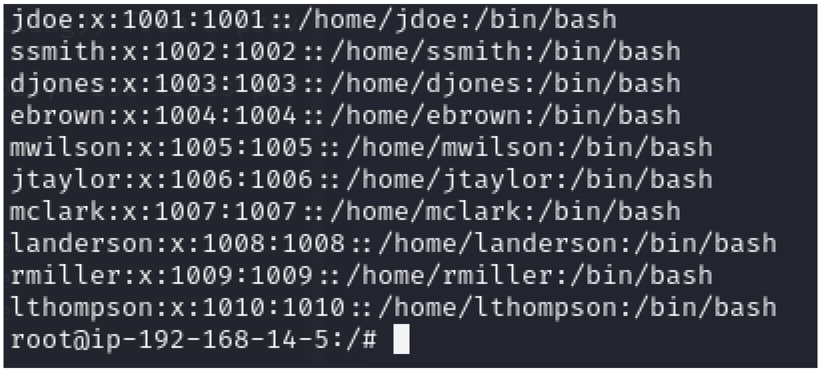
Username and passwords that were cracked:

➔ Ebrown : Tr0ub4dor&3



```
[DATA] max 64 tasks per 1 server, overall 64 tasks, 1002 login tries (l:1/p:1002), ~16 tr
[DATA] attacking ssh://3.144.139.3:22/
[22][ssh] host: 3.144.139.3   login: ebrown   password: Tr0ub4dor&3
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 18 final worker threads did not complete until end
[ERROR] 18 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-24 22:47:21
```

Figure 6: Hydra cracking username ebrown’s password.



```
jdoe:x:1001:1001::/home/jdoe:/bin/bash
ssmith:x:1002:1002::/home/ssmith:/bin/bash
djones:x:1003:1003::/home/djones:/bin/bash
ebrown:x:1004:1004::/home/ebrown:/bin/bash
mwilson:x:1005:1005::/home/mwilson:/bin/bash
jtaylor:x:1006:1006::/home/jtaylor:/bin/bash
mclark:x:1007:1007::/home/mclark:/bin/bash
landerson:x:1008:1008::/home/landerson:/bin/bash
rmiller:x:1009:1009::/home/rmiller:/bin/bash
lthompson:x:1010:1010::/home/lthompson:/bin/bash
root@ip-192-168-14-5:/#
```

Figure 7: Users on Target2 machine.

Target3

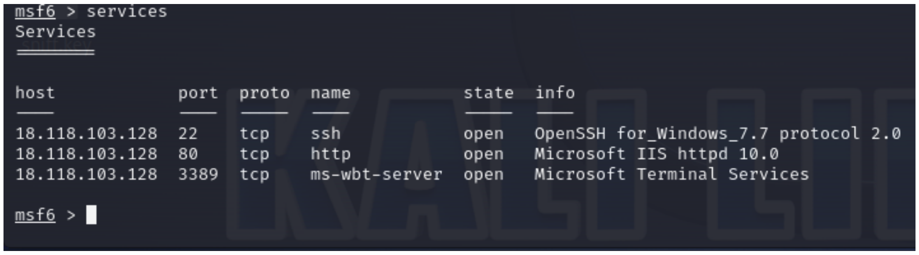
- **Services**

The Target3 machines are running the same services as the Target 1 machine. The services running are an OpenSSH server, a httpd windows server, and a remote desktop service.

➔ OpenSSH: version 7.7 for windows, running protocol 2.0

➔ Microsoft IIS httpd 10.0 service

➔ Microsoft Terminal Service



```
msf6 > services
Services
========

host            port  proto  name           state  info
----            ----  -----  ----           -----  ----
18.118.103.128  22    tcp    ssh            open   OpenSSH for_Windows_7.7 protocol 2.0
18.118.103.128  80    tcp    http           open   Microsoft IIS httpd 10.0
18.118.103.128  3389  tcp    ms-wbt-server  open   Microsoft Terminal Services

msf6 >
```

Figure 8: Target3 Services.

- **Vulnerabilities**

Similar to that of the Target1 machine the Open SSH server allows login with passwords and is running on the default port 22.

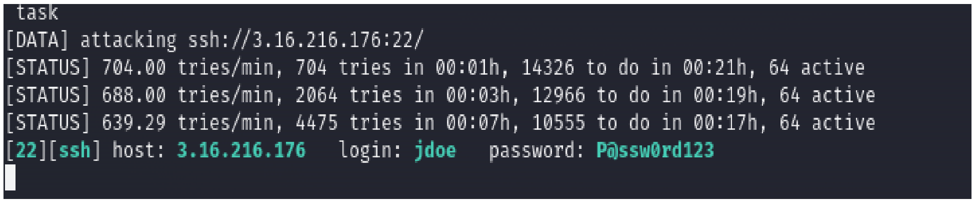
This machine also uses duplicate users from the Target1 and Target2 machines. With a list of users an attacker can easily crack duplicate and insecure passwords.

A remote desktop procedure should only be running when and only when it needs to be used. Leaving the service running all the time can leave the machine vulnerable to attackers.

- **Credentials**

Username and passwords that were cracked:

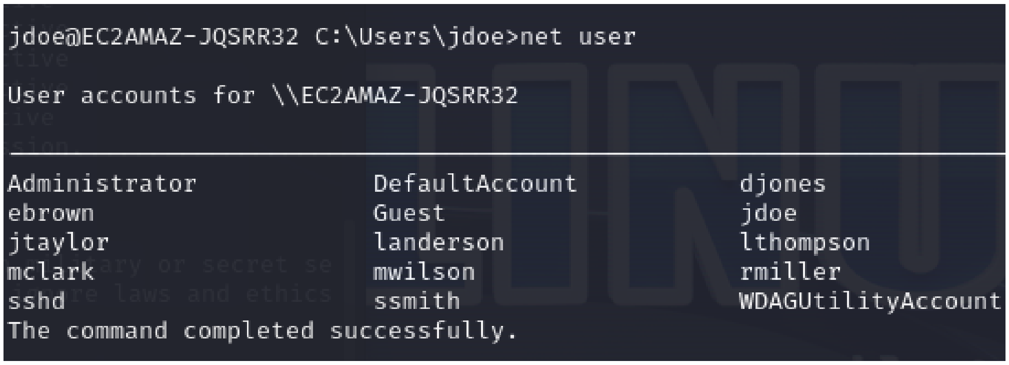
➔ Jdoe : P@ssw0rd123



```
 task
[DATA] attacking ssh://3.16.216.176:22/
[STATUS] 704.00 tries/min, 704 tries in 00:01h, 14326 to do in 00:21h, 64 active
[STATUS] 688.00 tries/min, 2064 tries in 00:03h, 12966 to do in 00:19h, 64 active
[STATUS] 639.29 tries/min, 4475 tries in 00:07h, 10555 to do in 00:17h, 64 active
[22][ssh] host: 3.16.216.176   login: jdoe   password: P@ssw0rd123
```

Figure 9: Cracking jdoe password on Target3 machine.

Usernames found on the machine:



```
jdoe@EC2AMAZ-JQSRR32 C:\Users\jdoe>net user

User accounts for \\EC2AMAZ-JQSRR32

-------------------------------------------------------------------------------
Administrator            DefaultAccount           djones
ebrown                   Guest                    jdoe
jtaylor                  landerson                lthompson
mclark                   mwilson                  rmiller
sshd                     ssmith                   WDAGUtilityAccount
The command completed successfully.
```

Figure 10: Users found on Target3

Target4

- **Services**

The Target4 machine is a linux machine and is running the four out of the five services that are on the Target2 machine: OpenSSH, telnet, ISC Bind and an Apache 2 web server. Figure 11 shows the results from scanning the Target4 machine.

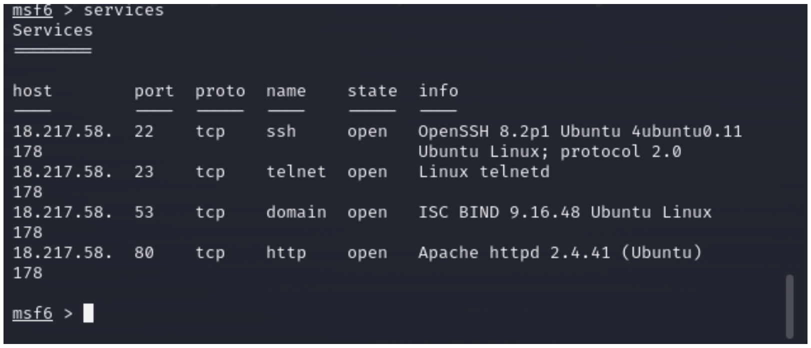
➔ Vsftpd version 3.0.5

➔ OpenSSH for linux version 8.2 protocol 2.0

➔ Linux Telnet

➔ ISC Bind version 9.16.48

➔ Apache web server version 2.4.41



```
msf6 > services
Services
========

host        port  proto  name    state  info
----        ----  -----  ----    -----  ----
18.217.58.  22    tcp    ssh     open   OpenSSH 8.2p1 Ubuntu 4ubuntu0.11
178                                     Ubuntu Linux; protocol 2.0
18.217.58.  23    tcp    telnet  open   Linux telnetd
178
18.217.58.  53    tcp    domain  open   ISC BIND 9.16.48 Ubuntu Linux
178
18.217.58.  80    tcp    http    open   Apache httpd 2.4.41 (Ubuntu)
178

msf6 > 
```

Figure 11: Services running on Target4.

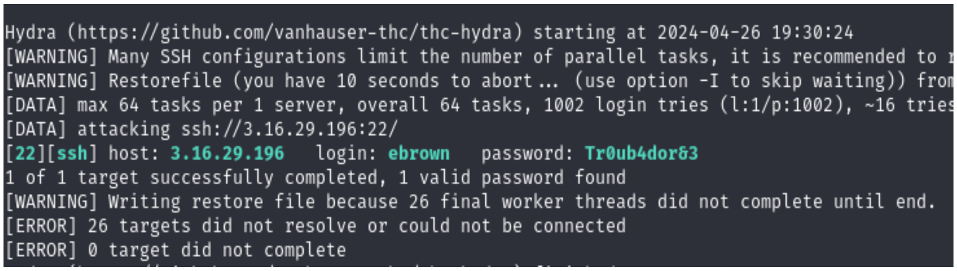
- **Vulnerabilities**

Like Target2 the OpenSSH server allows logins with passwords and is running on the default port of 22. The SSH server also does not limit the number of login attempts from a single IP address. The telnet service running on the machine is vulnerable to the same CVE’s that were listed under the target2 machine. This machine uses duplicate users from the Target, Target2, and Target3 machines. With a list of users an attacker can easily crack duplicate and insecure passwords. In fact the user “ebrown” has the same password that was used on Target2.

- **Credentials**

Username and passwords that were cracked:

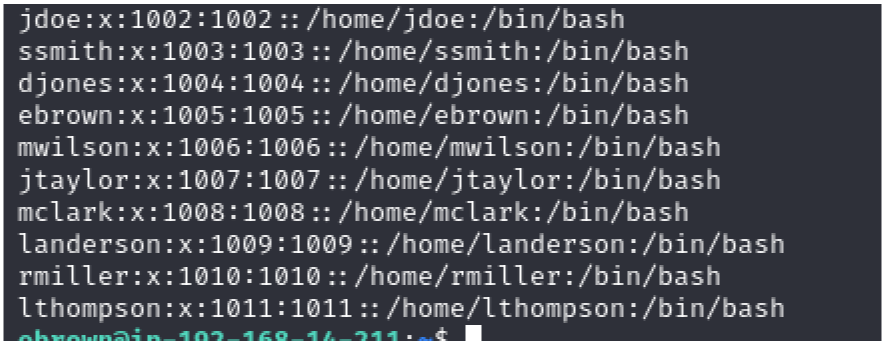
➔ Ebrown : Tr0ub4dor&3



```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-26 19:30:24
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to r
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from
[DATA] max 64 tasks per 1 server, overall 64 tasks, 1002 login tries (l:1/p:1002), ~16 tries
[DATA] attacking ssh://3.16.29.196:22/
[22][ssh] host: 3.16.29.196   login: ebrown   password: Tr0ub4dor&3
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 26 final worker threads did not complete until end.
[ERROR] 26 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
```

Figure 12: Cracking the ebrown password on Target4.

Usernames found on the machine:



```
jdoe:x:1002:1002::/home/jdoe:/bin/bash
ssmith:x:1003:1003::/home/ssmith:/bin/bash
djones:x:1004:1004::/home/djones:/bin/bash
ebrown:x:1005:1005::/home/ebrown:/bin/bash
mwilson:x:1006:1006::/home/mwilson:/bin/bash
jtaylor:x:1007:1007::/home/jtaylor:/bin/bash
mclark:x:1008:1008::/home/mclark:/bin/bash
landerson:x:1009:1009::/home/landerson:/bin/bash
rmiller:x:1010:1010::/home/rmiller:/bin/bash
lthompson:x:1011:1011::/home/lthompson:/bin/bash
ebrown@ip-192-168-14-211:~$
```

Figure 13: Usernames on Target4.

Remediation

Usernames and Passwords

The same usernames are used across all four of the target machines. Usernames should be specific to each machine to stop attackers from brute forcing passwords across machines.

There are instances of multiple reused passwords across machines. Similar to the usernames, passwords should be different for each machine and follow password rules. Rules include a character minimum usually eight or 12, cannot be a password that has been used in the past, have a special character and have at least one uppercase letter.

Services

All services across all machines should be updated to the latest release. Running old versions can leave the machines vulnerable to attacks, like the vsftpd backdoor exploit.

OpenSSH needs to be configured to increase security and mitigate the chances of a brute force attack being successful. In the SSH config file the port should be changed to one of your choosing, I would recommend staying away from ports 22, 222, and 2222 as they are common. Allowing login with a password should be disabled. The use of ssh keys should be the only way to login as a user. Open SSH should also limit the number of login attempts from a single IP. I would recommend 3-5 attempts.

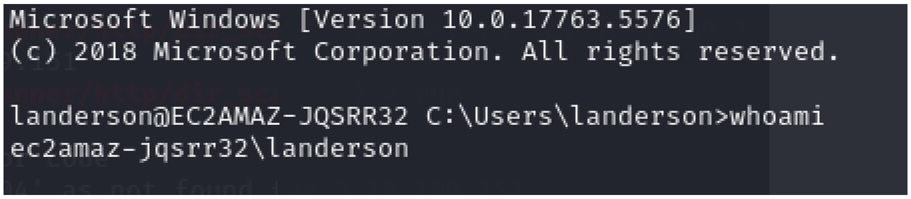
Telnet should not be running on target machines two and four. Telnet has multiple security vulnerabilities associated with its service and can leave the linux machines vulnerable to attacks. Since OpenSSH is running it's better to use that service for remote access as it is more secure, offering encryption and authentication.

Conclusion

Throughout the duration of this pentest I was able to gain knowledge on reconnaissance, privilege escalation, and experience with attacking vulnerable machines. The vulnerabilities in the four machines help me learn the importance of usernames and passwords on a machine. Changing your username across machines is an important rule to follow. Seeing first hand how easy it is to crack a password once you have a username you know exists helped me further understand cybersecurity. The importance of keeping your machine and its services up to date was also highlighted in this pentest. Overall I think this pen test will help me be better prepared for the work force and shed light on the kind of work i would be doing if I chose to be a penetration tester.

Appendix

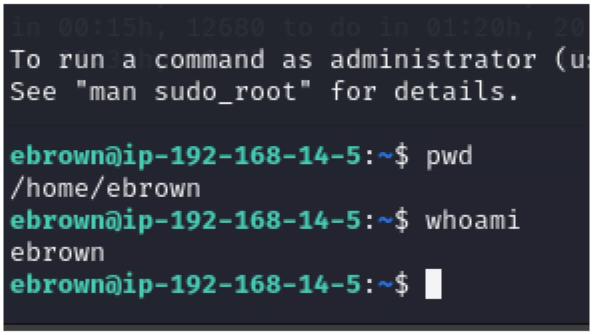
Logging into each machine via SSH.



```
Microsoft Windows [Version 10.0.17763.5576]
(c) 2018 Microsoft Corporation. All rights reserved.

landerson@EC2AMAZ-JQSRR32 C:\Users\landerson>whoami
ec2amaz-jqsrr32\landerson
```

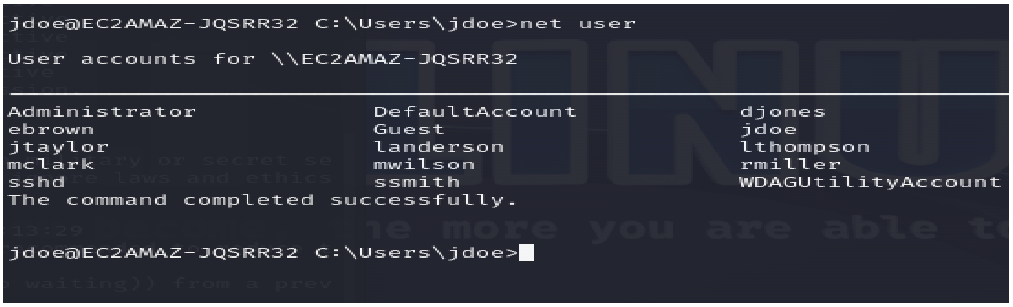
Figure14: SSH into Target1.



```
in 00:15h, 12680 to do in 01:20h, 20
To run a command as administrator (u
See "man sudo_root" for details.

ebrown@ip-192-168-14-5:~$ pwd
/home/ebrown
ebrown@ip-192-168-14-5:~$ whoami
ebrown
ebrown@ip-192-168-14-5:~$ 
```

Figure 15: SSH into Target2



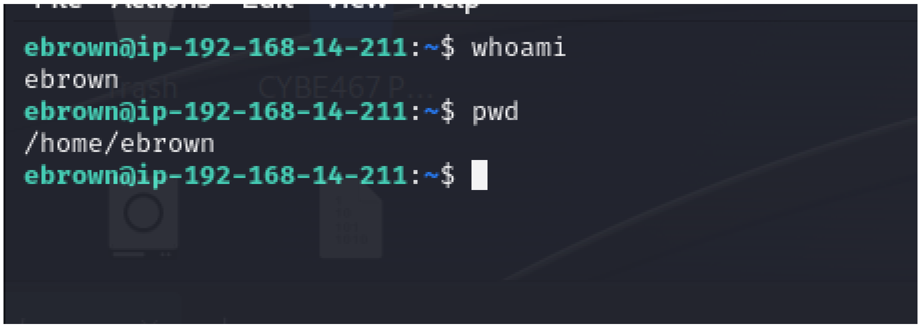
```
jdoe@EC2AMAZ-JQSRR32 C:\Users\jdoe>net user

User accounts for \\EC2AMAZ-JQSRR32

-------------------------------------------------------------------------------
Administrator            DefaultAccount           djones
ebrown                   Guest                    jdoe
jtaylor                  landerson                lthompson
mclark                   mwilson                  rmiller
sshd                     ssmith                   WDAGUtilityAccount
The command completed successfully.

jdoe@EC2AMAZ-JQSRR32 C:\Users\jdoe>
```

Figure 16: SSH into Target3



```
ebrown@ip-192-168-14-211:~$ whoami
ebrown
ebrown@ip-192-168-14-211:~$ pwd
/home/ebrown
ebrown@ip-192-168-14-211:~$
```

Figure 17: SSh into Target4