DNS Spoofing

Comp 8505 Assignment 4

Shane Spoor

Mat Siwoski

Introduction

The purpose of this assignment was to become familiar with DNS spoofing and to implement an application that would do basic web site spoofing. The basic application is command-line with the appropriate switches to perform the various functions.

Constraints

The assignment had the following requirements:

- Your application will simply sense an HTML DNS Query and respond with a crafted Response answer, which will direct the target system to a your own web site.
- You will test this Proof Of Concept on a LAN on your own systems only. This means that you are not to carry out any DNS spoofing activity on unsuspecting client systems.
- You are required to handle any arbitrary domain name string and craft a spoofed Response.

Dependencies:

The application requires the following Python packages to be installed:

- NetfilterQueue
- Scapy

In the event that these packages are not installed, run the following commands as root to install them:

```
dnf install libnetfilter-queue-devel
dnf install libnetfilter-queue
pip install scapy
pip install netfilterqueue
```

Running the Application

asd

Design

DNS Spoofing

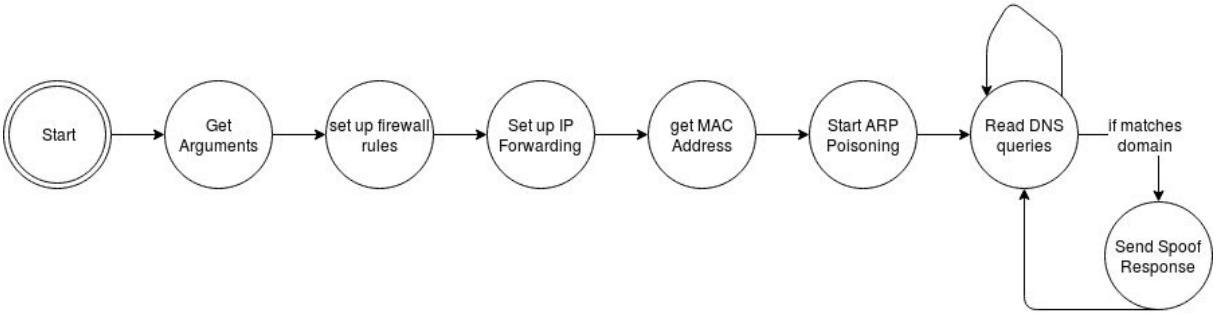

Fig. 1: DNS Spoof state transition diagram

Pseudocode

```
Parse Arguments
    Take in and set arguments

NetfilterQueueCallback
    Get Payload
    Get IP Info from Scapy
        Get packet layer
            If
                Accept Packet
            Else
                Spoof packet

Main
    Set up iptables rules
    Get Mac Addresses
    Start Threading

    Poison Threads
        While loop
            ARP Poison Victim
            Sleep
    Start Poison threads

Run Main

Spoof Packet
    Receive IP Info from Scapy
    Set payload with spoofed packet
    Accept the packet

Arp Poison Victim
    Send to Target IP
    Send to Host IP
```

Testing

Test #	Test Description	Result
1	Help screen with all available arguments	Passed (Fig. 2)
2	Original MAC Address	Passed (Fig. 3)
3	Host MAC Address	Passed (Fig. 4)
4	Spoofed MAC Address	Passed (Fig. 5)
5	Spoofed Website	Passed (Fig. 6)

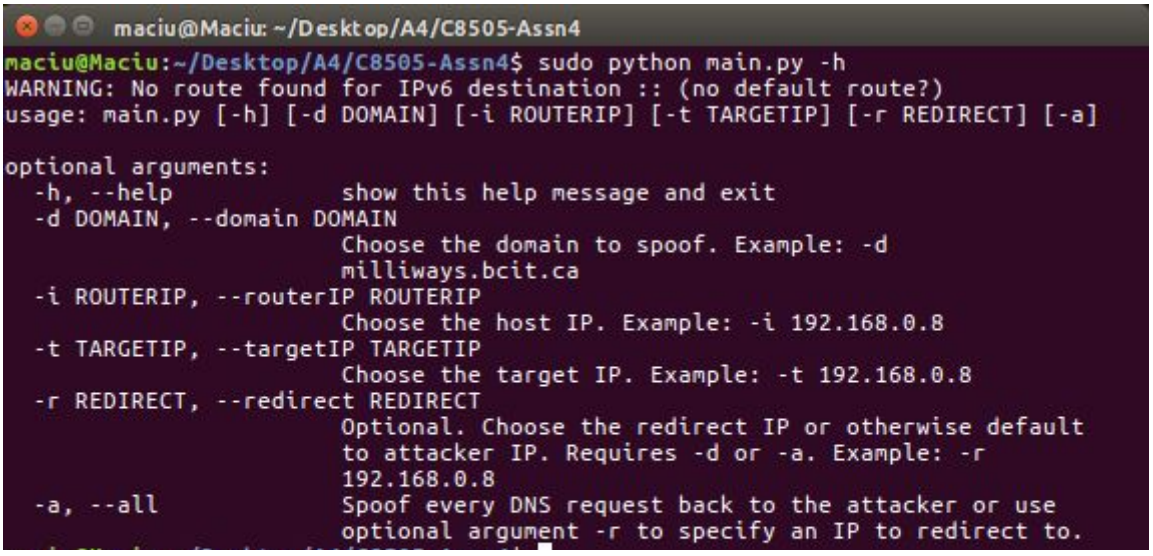

Fig. 2: Help screen with all available arguments

```
File  Edit  Tabs  Help
pi@raspberrypi ~ $ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.1.254            ether   20:76:00:e0:cc:88   C                     eth0
pi@raspberrypi ~ $
```

Fig. 3: Original MAC Address

```
root@datacomm:~/Downloads/c8505-assn3
File  Edit  View  Search  Terminal  Help
[root@datacomm c8505-assn3]# python main.py 8001 8000 -p testtest -k test
usage: main.py [-h] [-p PASSWORD] [-k KEY] [-s SERVER] [-m MASK]
               {client,server} lport dport
main.py: error: argument mode: invalid choice: '8001' (choose from 'client', 'server')
[root@datacomm c8505-assn3]# python main.py client 8001 8000 -p testtest -k test
-s/--server is required in client mode.
[root@datacomm c8505-assn3]# python main.py client 8001 8000 -s 192.168.0.8 -p testtest -k
 test
len: 4; start: 0
.
Sent 1 packets.
Enter a command to execute on the server: ls
.
Sent 1 packets.
```

Fig. 4: Host MAC Address

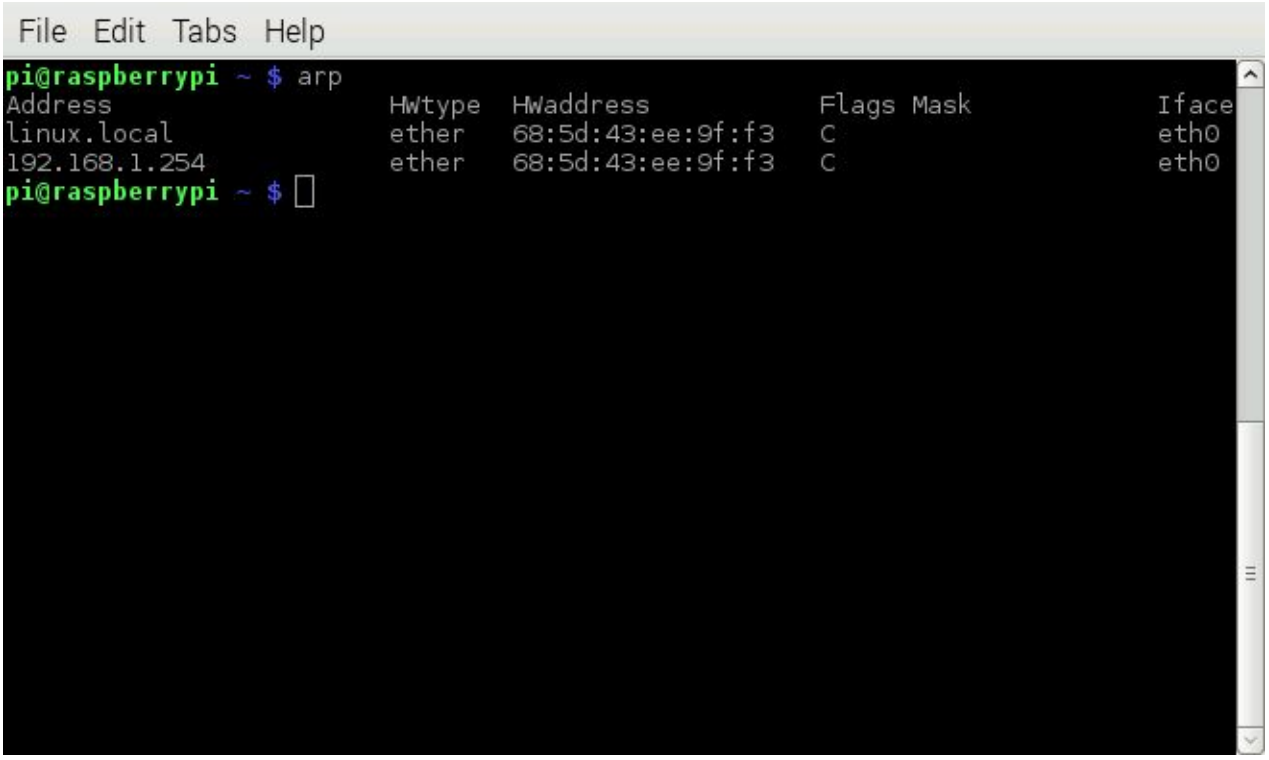

Fig. 5: Spoofed MAC Address

Fig. 6: Spoofed Website

```
root@datacomm:~/Downloads/C8505-Assn3
File  Edit  View  Search  Terminal  Help
[root@datacomm C8505-Assn3]# python main.py server 8000 8001 -m trustd -p testtest -k test

len: 4; start: 0
Waiting for client...
Client connected: 192.168.0.7

Command type: SHELL; command: ls
exit code: 0
stdout: backdoor.py
backdoor.pyc
command.py
command.pyc
main.py
README.md
utils.py
utils.pyc

stderr:

.
Sent 1 packets.
```

Fig. 7: Client sends a command

```
[root@datacomm ~]# pgrep trustd
2453
[root@datacomm ~]# pgrep trustd
2453
[root@datacomm ~]#
```

Fig. 8: Process found on the machine

```
root      2429  0.0  0.0      0     0 ?        S    17:20   0:00 [kworker/2:0]
root      2430  0.0  0.0      0     0 ?        S    17:20   0:00 [kworker/3:0]
root      2453  0.1  0.4 267900 35096 pts/0    S+   17:21   0:00 trustd
root      2465  0.0  0.0 308564  5968 ?        Ssl  17:21   0:00 /usr/libexec/gvfsd-metada
ta
root      2504  0.0  0.0      0     0 ?        S    17:23   0:00 [kworker/1:0]
root      2506  0.0  0.0      0     0 ?        S    17:23   0:00 [kworker/0:0]
root      2538  0.0  0.0 122708  4828 pts/2    Ss   17:24   0:00 bash
root      2576  0.0  0.0 151416  3744 pts/2    R+   17:24   0:00 ps -aux
root      2577  0.0  0.0 116060   948 pts/2    S+   17:24   0:00 less
(END)
```

Fig. 9: Process currently running