COMP 8505 Assignment 4 Design Doc

Alex Zielinski – A00803488

04/11/2018

Table of Contents

User Guide

Purpose

The purpose of this software is to demonstrate a proof of concept for a DNS spoofer. This POC is written in Python and makes use of the ***scapy*** python library. The POC has two components. The first is an ARP poisoner where the software ARP poisons the target machine as well as the router. This way the attacking machine places itself in the middle of the target and the router in order to view all traffic occurring between the two. The second component is the DNS spoofer itself. Once the attacker is able to view all traffic traveling between the target and the router then it can start spoofing DNS packets. The software reads incoming DNS queries from the target machine. If the website URL within the DNS query that the target machine sends matches any whitelisted sites defined by the attacker, then the software will craft a DNS response containing the IP address the attacker wants to redirect the target machine to and will then send that DNS response to the target machine as to redirect them.

Note: one limitation of this POC is speed. Because this POC makes use of python, it is too slow at forging a DNS response and sending it. The issue is a race condition. The real response from the DNS server will arrive first at the target machine before the forged dns response created by this program. This means the target machine will read the real dns response from the server and will ignore the forged one. Due to this issue firewall IP forwarding on the attacker's machine is turned off as not to forward and DNS responses from the DNS server. As a result the only DNS responses the target machine will receive are the ones created by this program.

Project Folder Directory Listings

Captures
|---- test case 2 – captures from attacker machine
|---- test case 3 – captures from attacker machine
|---- test case 4 – captures from attacker machine
|---- test case 5 – captures from attacker machine
dns_spoofer.py
Testing Doc
Design Doc

Running the Code

Some steps must be taken before running the python script. First ensure that Python is installed on your machine. Next you need to install the python library ***scapy***. To do this follow these steps (for Fedora):

1. Open up terminal and type ***dnf install git python-devel***
2. Next go into the /tmp directory via the command ***cd /tmp***
3. Next clone scapy's github repo via the command ***git clone https://github.com/secdev/scapy***
4. Go into the directry scapy via the command ***cd scapy***

5. Install the library via the command ***python setup.py install***

Now that scapy is installed you are almost ready to run the program. First open up the python script called ***dns_spoofer.py*** found inside the project directory folder. At the top of the script you will find a 2d array called ***sites*** as seen in the screenshot below.

```
sites = [["milliways.bcit.ca", "192.168.0.18"],
         ["bcit.ca", "192.168.0.18"],
         ["sfu.ca", "192.168.0.18"],
         ["ubc.ca", "192.168.0.18"],
         ["cbc.ca", "192.168.0.18"],
         ["sd43.bc.ca", "192.168.0.18"]]
```

This is where you can specify a website query to look for and the IP to redirect to. You can add your own entry with the following syntax: ***<Website> , <IP to redirect>***. So, for example in the case of the screenshot, the first entry says any DNS query that contains the website ***"milliways.bcit.ca"*** then redirect the target to the IP ***"192.168.0.18"***. Once you have setup the 2d array as you wish then you can start the program.

To start the program, open up terminal and navigate to the project directory. In order to run the script, you have to provide 3 command line arguments. The program usage is as follows:

./dns_spoofer.py <ATTACKER IP> <TARGET IP> <ROUTER IP>

ATTACK IP: IP address of the man in the middle (attacker)
TARGET IP: IP address of the target machine
ROUTER IP: IP address of the router

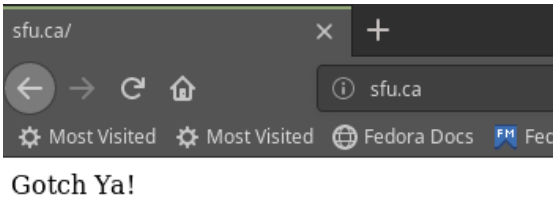
Once you start the script you will get some basic information regarding the MAC and IP address of all the machines that are part of the attack. The screenshot below illustrates this information.

```
Target ARP Response: 192.168.0.19 -> 98:90:96:dc:ef:dc
Router ARP Response: 192.168.0.100 -> 44:d9:e7:95:e4:9f

Attacker: 192.168.0.18 -> 98:90:96:dc:f0:63
Target  : 192.168.0.19 -> 98:90:96:dc:ef:dc
Router  : 192.168.0.100 -> 44:d9:e7:95:e4:9f
```

At this point the attacker machine is listening for any DNS queries from the target machine that contain any whitelisted sites. On the target machine, within a browser, enter a site that is whitelisted. When the DNS query for the whitelisted site is sent out then the DNS spoofer software will output the following.

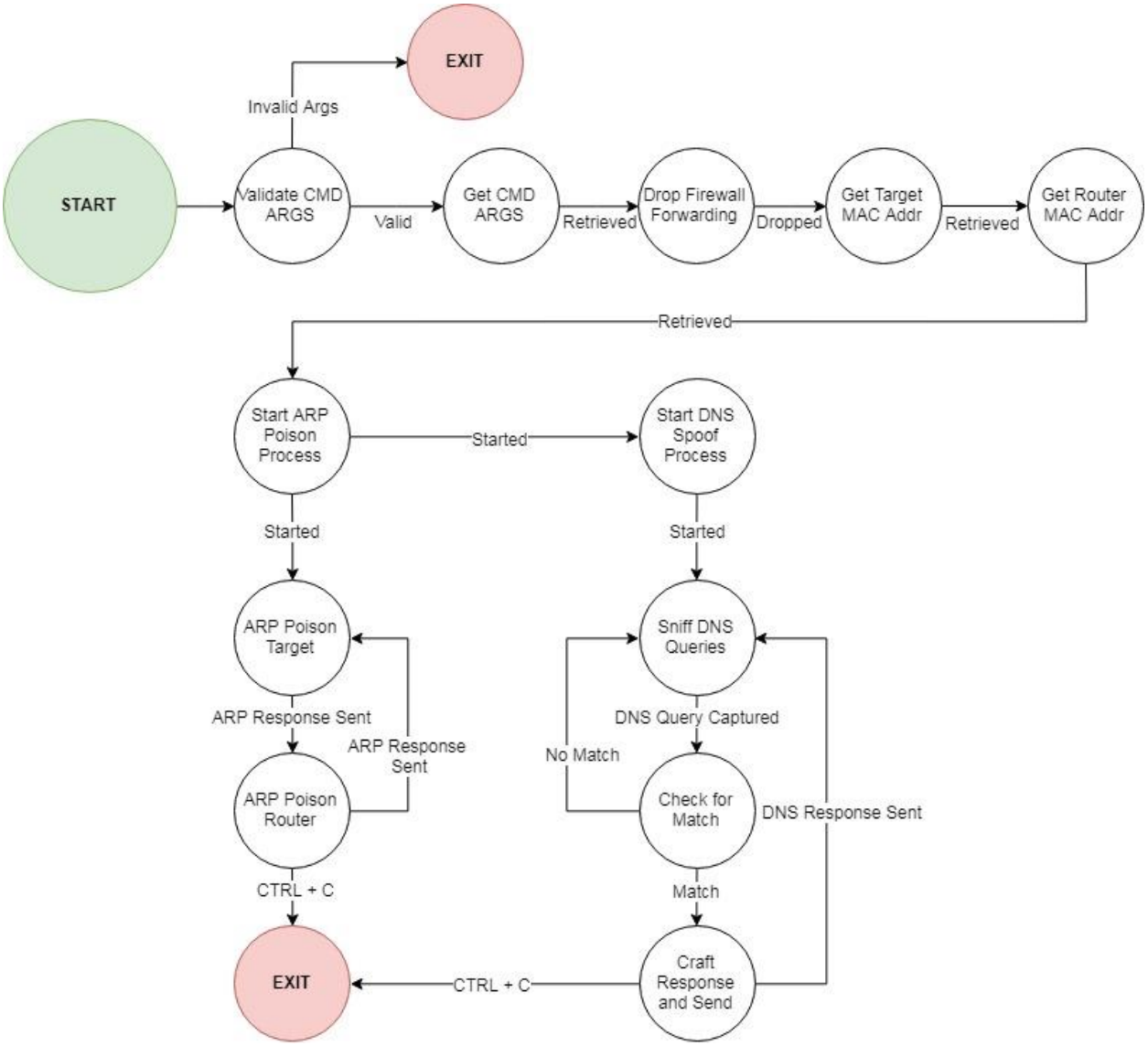
```
Target ARP Response: 192.168.0.19 -> 98:90:96:dc:ef:dc
Router ARP Response: 192.168.0.100 -> 44:d9:e7:95:e4:9f

Attacker: 192.168.0.18 -> 98:90:96:dc:f0:63
Target  : 192.168.0.19 -> 98:90:96:dc:ef:dc
Router  : 192.168.0.100 -> 44:d9:e7:95:e4:9f

sfu.ca. -> 192.168.0.18
sfu.ca. -> 192.168.0.18
```

This is saying that the attacker has received a query containing a whitelisted site ***sfu.ca***. As a result, the software crafts a DNS response containing the IP ***192.168.0.18*** in response to the query. On the target machine the browser in this case redirects the target to a web server running within the LAN. The following webpage is displayed on the target machine.



Finite State Machines

DNS Spoofer



Pseudo Code

Validate CMD ARGS

Check if there are 4 arguments
If not, then print usage message and **exit**
Otherwise go to **Get CMD ARGS**

Get CMD ARGS

Copy CMD ARGS into variables
Go to **Drop Firewall Forwarding**

Drop Firewall Forwarding

Run iptables command to drop all UDP port 53 forwarding
Go to **Get Target MAC Addr**

Get Target MAC Addr

Start a thread
In thread send ARP query to target machine
Start another thread
In thread read ARP response from target containing target MAC addr
Go to **Get Router MAC Addr**

Get Router MAC Addr

Start a thread
    In thread send ARP query to router
Start another thread
    In thread read ARP response from router containing router MAC addr
Go to **ARP Poisoning**

ARP Poisoning

Create new process
    LOOP
    Go to **ARP Spoof Target**
    Go to **ARP Spoof Router**
    Create an ARP response with attacker MAC addr and router IP
    Send ARP response to target
    Create an ARP response with attacker MAC addr and target IP
    Send ARP response to router
    LOOP
Go to **DNS Spoof**

ARP Spoof Target

Create an ARP response with attacker MAC addr and router IP
Send ARP response to target

ARP Spoof Router

Create an ARP response with attacker MAC addr and target IP
Send ARP response to router

DNS Spoof

Create new process
LOOP
Go to **Sniff DNS Queries**
LOOP

Sniff DNS Queries

Check if the site within DNS query sniffed from target matches any of the whitelisted sites
If it does match, then go to **Send DNS Response**

Send DNS Response

Create DNS response packet containing site from DNS query captures and IP to redirect to
Send DNS response packet to target