Name: Atmika Sharma

SBU ID: 111464371

--------------------- Assignment 4-------------------------------

Problem:

To develop:

1) an on-path DNS packet injector

which

2) a passive DNS poisoning attack detector

which conforms to the following specification:

dnsdetect [-i interface] [-r tracefile] expression

-----------------------------------------------------------------

Requirements:

Python 2.7.12

ubuntu 16.04 LTS

$ pip install scapy

-----------------------------------------------------------------

Part 1: dnsinject

Conforms to the following specification:

dnsinject [-i interface] [-h hostnames] expression

1. -i

-> Injection occurs live on the network device <interface> (e.g., wlp2s0). If not specified, the program dnsinject automatically selects a default interface to inject on.

2. -f

-> Reads a list of IP addresses and hostnames which specificy the hostname to inject on. If no file option is specified the local machines IP is sent as the answer.

3. <expression>

-> A BPF filter that specifies which packets will be monitored.

----------------------------------------------------------------

Part 2: dnsdetect

Conforms to the following specification:

dnsdetect [-i interface] [-r tracefile] expression

1. -i

-> Detection occurs on the network device <interface> (e.g., wlp2s0). If not specified, the program dnsinject automatically selects a default interface to monitor.

2. -r

-> Reads a tracefile and detects any DNS poisoning attacks on in that file.

3. <expression>

-> A BPF filter that specifies which packets will be monitored.

-----------------------------------------------------------------

Implementation details:

For injection, the program will listen on a particular interface in promiscous mode and sniffs DNS query packets which are tested with nslookup. For every nslookup, scapy injects a packet and then keeps on listening further for any similar queries.

For detection, the program will sniff for any DNS query packets on the interface specified and check whether any response matches with stored mylist dictionary value. If it does not, then it is a poisoning attempt. In case of offline poisoning check, dnsdetect reads a pcap file and follows the same logic.

----------------------------------------------------------------

Run and Execute:

DNSINJECT:

sudo python dnsinject.py -i wlxc025e9107cb2 -f hostnames

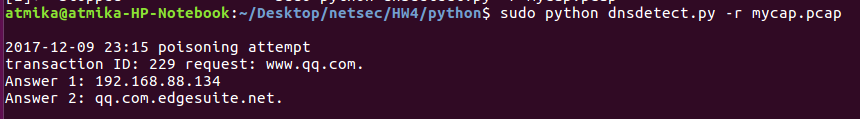
sudo python dnsinject.py -i wlxc025e9107cb2

DNSDETECT:

sudo python dnsdetect.py -i wlxc025e9107cb2

sudo python dnsdetect.py -r mycap.pcap

----------------------------------------------------------------



---------------------------------------------------------------

References and Help:

https://pymotw.com/2/getopt/

http://scapy.readthedocs.io/en/latest/usage.html

https://docs.python.org/2/library/os.path.html

https://stackoverflow.com/questions/28292224/scapy-packet-sniffer-triggering-an-action-up-on-each-sniffed-packet

https://github.com/ksasmit/NS--DNS-injector-and-detector

https://github.com/waytoalpit/ManOnTheSideAttack-DNS-Spoofing/blob/master/README.txt

https://thepacketgeek.com/scapy-sniffing-with-custom-actions-part-1/

https://packettotal.com/app/analysis?id=ed08c585a943f12203ea747ab072c1da