**Report – HW4**

* **Introduction**

We have created two applications – dnsInject and dnsDetect.  
**1. dnsInject**

* This application will capture DND query packets and send a spoofed response to the victim. It will take three arguments:
* -i (Interface) :

It will listen to this interface if specified. If not, it will listen to the default interface.

* -h (Hostnames file) :

Whenever a DNS Query packet is captured, it will see if the hosname is from this list. If yes, it will insert the corresponding IP address. If the hostname is not present or the file is not provided, it will insert the IP address of the machine on which it is running.

* [expression] :

It will only sniff the packets satisfying the BPF filter expression.

**How to Run?**

* This is a python file. It should be run as follows:  
  python dnsInject.py [-i interface] [-h file] [exp]
* It requires scapy, fcntl, argparse, socket and struct packages of python.
* We will need to install scapy.

A pcap extract of the packets injected by this application is provided in the submission.

**Design Details**

This application has following components:

* Main

It extracts the input and calls the sniff function which sniffs every packet and calls the function ‘dnsInject’ which injects a spoofed reply.

* readHostInfo  
  It reads the hostfile and creates a hashmap with key – Hostname and value – IP address

**2. dnsDetect**

* This application will sniff packets which victim receives and detect an packet injection.
* It takes 3 arguments:
* -i – The interface to listen to. The symantics are same as dnsInject
* -r – The pcap to sniff on. Will listen to this file, if provided and not any other interface
* Exp – BPF filter expression to filter the packets.

**How to Run?**

* This is a python file. It should be run as follows:  
  python dnsDetect.py [-i interface] [-r file] [exp]
* It requires scapy, argparse packages of python.
* We will need to install scapy.

**Design Details**

This application has following components:

* Main

It extracts the input and calls the sniff function which sniffs every packet and calls the function ‘dnsDetect’ which detects any spoofed packet by the following method:  
1. Put the packet in hashmap with key as packet.id if the packet is not present.

2. If present, check the MAC address of the already present packet and the current packet.

If they are same, it’s a false positive. Don’t do anything.

If they are different, it’s a packet injection. Report the packet details.

* getIPsFromDNS  
  It gives a list of IP addresses, reported for the given packet.  
    
  The output of dnsDetection is also provided in the submission.
* **Major References:**
* <https://thepacketgeek.com/scapy-p-06-sending-and-receiving-with-scapy/>
* <https://en.wikipedia.org/wiki/Ethernet_frame>
* <https://stackoverflow.com/questions/415511/how-to-get-current-time-in-python>
* <http://www.zytrax.com/books/dns/ch15/#qrbit>