Student ID: 111483647
CSE 508 Network Security Fall-2017
Homework 4 -DNS Injection and Detection.

1] IMPLEMENTATION:
a) Test environment:
Distributor ID: Ubuntu
Description: Ubuntu 16.04.3 LTS
Release: 16.04
Codename: xenial
Linux 4.10.0-28-generic x86_64

b) Language used: Python 2.7

2] DNS INJECTION:

Commands for execution:
python disinfect.py -i enp0s3
python dnsinject.py -h hostname

Used arg to take input from the user. The flag is set to 1 if the input contains the interface value, else the flag is set to 0. A function spoof is defined to send the actual spoofed packet. The source ip address and port of the spoofed packed is same as the destination ip address and port of the request packet respectively and vice versa. The transaction id of the spoofed packet is the same as the transaction id of the request packet. The spoofed packet has a different value of 'rdata' which is equal to the attacker's ip address if the file containing the hostnames(-h) is not specified in the input; else the 'rdata' takes the value from the hostname file.

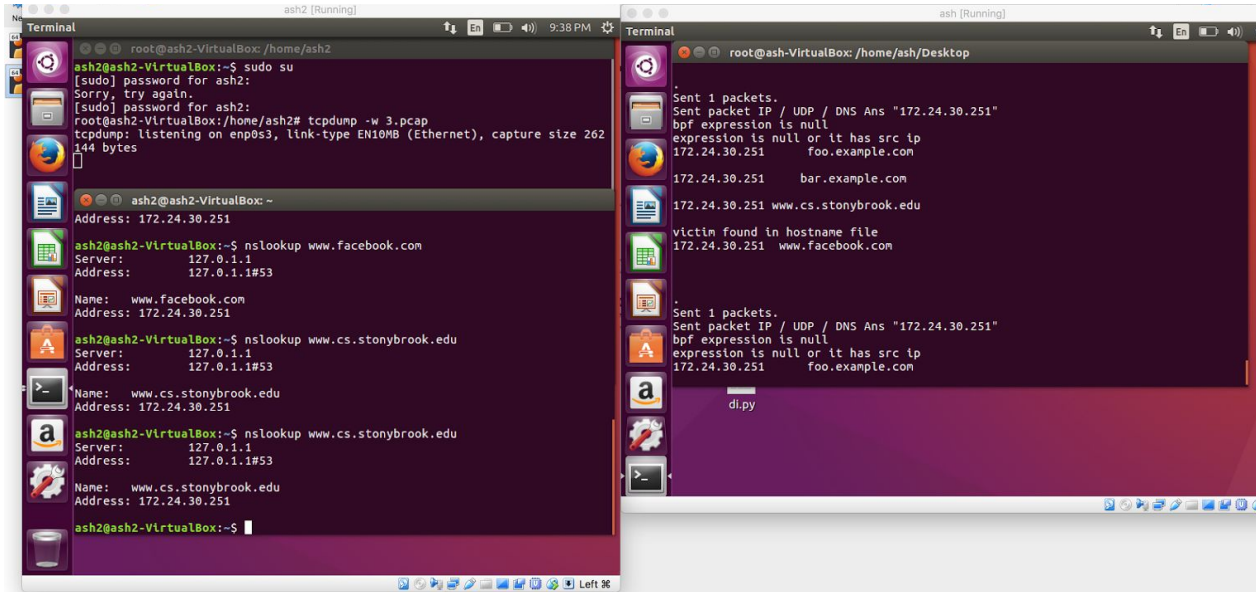
3] DNS DETECTION:

Commands for execution:
python dnsdetect.py -r mycap.pcap

If a single query receives 2 dns response then there is a chance of dns spoofing.To check spoofing :
1-Check if the source ip, destination ip, source port address, destination port address, transaction ID and domain name of the two packets are the same. If they are same, check if the payload and the 'rdata' value of those packets are different. If the above conditions are satisfied, then it is a case of dns spoofing.
2- To test for false positive check if the MAC address of the two responses are the same. Is so the packets are valid and there is no spoofing. Else if MAC address of the two responses are different , dns spoofing has occurred and is detected.

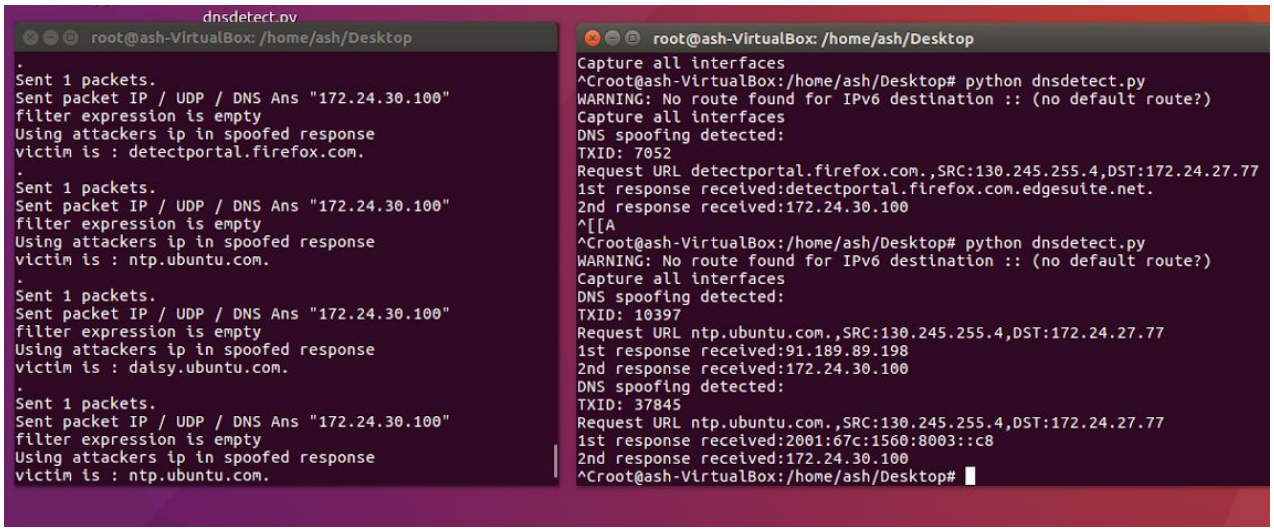
4] EXAMPLE OUTPUT:

A. DNSINJECT: Configure 2 VMs in Bridged mode. Set one of them as an attacker and the other as the victim. Run dnsinject.py file on the attacker VM. Run nslookup on the victim machine. The spoofed packet will be sent from the attacker VM as soon as the query is fired at the victim VM. You can check whether the dns injection is successful by observing the output of nslookup at the victim VM.



B. DNSDETECT:

Scenario 1: Configure 2 VMs as described above. Run dnsdetect.py on victim VM and dnsinject.py on attacker VM.



Scenario 2: Run dnsinject.py with the .pcap file captured during dns injection on the victim side.

```
ash@ash-VirtualBox:~$ sudo su
[sudo] password for ash:
root@ash-VirtualBox:/home/ash# cd Desktop
root@ash-VirtualBox:/home/ash/Desktop# python dnsdetect.py -r mycap.pcap
WARNING: No route found for IPv6 destination :: (no default route?)
TraceFile: mycap.pcap
DNS spoofing detected:
TXID: 17958
Request URL www.google.com.,SRC:192.168.217.2,DST:192.168.217.132
1st response received:216.58.219.196
2nd response received:192.168.217.129
1st packet: ###[ Ethernet ]###
  dst       = 00:0c:29:78:17:46
  src       = 00:50:56:fd:4e:2b
  type      = 0x800
###[ IP ]###
     version   = 4L
     ihl       = 5L
     tos       = 0x0
     len       = 76
     id        = 13984
     flags     =
     frag      = 0L
     ttl       = 128
     proto     = udp
     chksum    = 0xd028
     src       = 192.168.217.2
     dst       = 192.168.217.132
     \options   \
###[ UDP ]###
        sport     = domain
        dport     = 36593
        len       = 56
        chksum    = 0x7318
###[ DNS ]###
```

5] CITATIONS:

For dns injection:
http://www.cs.dartmouth.edu/~sergey/netreads/local/reliable-dns-spoofing-with-python-scapy-nfqueue.html

For dns detection:
https://witestlab.poly.edu/blog/redirect-traffic-to-a-wrong-or-fake-site-with-dns-spoofing-on-a-lan/
https://github.com/lianke123321/CSE508-project/blob/master/hw4/quantumdetect.py