**CS526 Information Security- Project 2**

[**hegde12@purdue.edu**](mailto:hegde12@purdue.edu)

**Problem 1**

1. HTTPS Session

Successfully visited ips:

|  |
| --- |
| 235.51.104.218 |
| 221.196.14.135 |
| 171.87.25.167 |
| 171.87.25.154 |
| 171.87.25.134 |
| 171.54.85.171 |
| 171.38.206.203 |
| 171.38.206.134 |
| 171.38.194.75 |
| 171.118.85.218 |
| 169.231.16.199 |
| 169.215.245.198 |
| 169.215.245.165 |
| 169.102.199.170 |
| 169.100.175.166 |
| 169.100.167.250 |
| 167.87.119.134 |
| 167.86.245.153 |
| 167.70.230.107 |
| 167.119.105.170 |
| 167.103.51.138 |
| 157.54.163.75 |
| 157.54.163.135 |
| 137.79.61.203 |
| 137.54.75.155 |
| 137.245.108.202 |
| 135.54.222.170 |
| 109.85.130.234 |
| 109.85.130.199 |
| 107.70.210.154 |
| 107.70.10.71 |
| 107.39.238.234 |
| 105.71.216.186 |
| 105.55.89.170 |
| 105.55.3.171 |
| 103.245.209.202 |

(Count – 36)

*Filter used – “http and http.response”*

What it does – Displays only the http packets where the server has sent back some response code

Rationale – If servers are ‘successfully visited’, they must have sent some response code

*Graphical user interface, application, table

Description automatically generated*

1. Directory Traversal

Host - 169.68.172.139

*Filter used – ‘frame contains “../../..” ‘*

What it does – Displays only the packets that would have the particular string in it’s frame

Rationale – For directory traversal, the attacker must have sent traversal string “../../..” at some point.

Graphical user interface, application

Description automatically generated

1. Password Guessing

Host – ‘167.71.8.234’

*Filter used – ‘ftp and ftp contains "Password" ‘*

What it does – Displays only the ftp packets that have the word ‘Password’ in it

Rationale – For password guessing, there must be requests/responses with the word “Pass” in it

Table

Description automatically generated

1. Unencrypted Usernames and Passwords

Username : ‘calrules’

Password: ‘thisissosecure’

*Filter used – ‘ftp and (frame contains USER or frame contains PASS or* [*ftp.response.code==230*](ftp://ftp.response.code==230)*)’*

What it does – Displays only the ftp packets that have either ‘USER’ or ‘PASS’ in it or have an ftp response code of 230

Rationale – We want all packets that either send username and password or give a success response

Graphical user interface, text, application, email

Description automatically generated

1. Service Versions

Host ip – 169.100.167.250

Version – Apache/1.3.29

*Filter Used – ‘http and http.server ~ "Apache/1.3.2"’*

What it does –Displays only the http packets that have an Apache server whose name matches a part of the string ‘*Apache/1.3.2’*

Rationale – Oldest version of Apache is 1.3

Graphical user interface, application, table

Description automatically generated

1. DNS and Source Port Randomization

Host ip - 169.100.167.155 and 169.100.167.250

Src Port – 32927 and 33814 respectively

*Filter Used – ‘dns and udp.dstport == 53’*

What it does – Displays only the dns packets that have their udp destination port address as 53

Rationale – DNS queries are sent with destination port set to 53

Graphical user interface, text, application

Description automatically generated

1. TCP Sequence Numbers

*Filter Used – ‘tcp and tcp.flags.syn ==1 and tcp.flags.ack==0’*

What it does – Displays only the tcp packets that are sending SYN and not ACK

Rationale – We want to find the Ips that participate in 5 or more connections. So it is enough to get the packets participating in the first step of the TCP/IP handshake.

The question says “Find the IP addresses of the 2 TCP endpoints that participate in 5 connections or more”. If this is interpreted as “Two different hosts that connect to atleast 5 servers each”

Ips - 167.215.167.186 and 167.86.245.153

If it is interpreted as, “Two hosts that connect with each other atleast 5 times”

Ips - 167.86.245.153 and 169.100.167.155

Graphical user interface, text, application

Description automatically generated

1. Traceroute Scanning

Ip host – 43.231.148.107

Ip destination – 141.231.148.107

*Filter Used – ‘icmp’*

What it does – Displays only the packets that use ICMP protocol

Rationale – Traceroute packets are sent out using ICMP protocol

Graphical user interface, text, application

Description automatically generated

1. Cross-Site Scripting

Server with vulnerability – 171.87.25.167

*Filter used – ‘http and http.request.uri contains "<script>"’*

What it does – Displays only the http packets that have the string “<script>” in their url

Rationale – reflected XSS attacks send script inside url

Graphical user interface, text, application, email

Description automatically generated

**Problem 2**

|  |  |
| --- | --- |
| **IP** | **MAC** |
| 10.0.2.1 | Apple\_e5:66:07 (00:26:08:e5:66:07) |
| 10.0.2.2 | Apple\_d8:0f:fa (04:0c:ce:d8:04:fa) |
| 10.0.2.3 | IntelCor\_50:f0:a6 (8c:a9:82:50:f0:a6) |

1. This IP range falls under the private IP ranges of Class A IP addresses. Private network addresses are not allocated to any specific organization. These addresses are commonly used for [local area networks](https://en.wikipedia.org/wiki/Local_area_network) (LANs) in residential, offices etc. The network shown is a small network with 3 hosts.
   1. Source system – 10.0.2.2, ftp server – 194.109.21.66

Ans - [ftp.mirror.nl](ftp://ftp.mirror.nl)

* 1. Active Connection
  2. When we follow a tcp stream we can see that the login details are sent in plaintext
  3. In place of FTP we can use SFTP or HTTPS for secure file transfer
  4. The browser is authenticated to facebook through tokens stored in cookies. This is insecure because anybody who gets hold of the cookies will be able to access the user’s account without username and password.
  5. The attacker can get the cookies, access user’s account and impersonate as the user.
  6. Users can protect themselves by logging out after every session so that the cookies are destroyed
  7. While on facebook, the user searched for users with names starting with ‘zak’, clicked on ‘zakirbpd’ and went to that user’s profile and visited the timeline. He then opened a new chat message and sent the message ‘Остановить нюхают My WiFi’. (Which when translated to English becomes ‘stop sniffing my wifi’)

**Problem 3**

Code:

import dpkt

import sys

import socket

from dpkt.compat import compat\_ord

#f = open('project2\_part3.pcap','rb')

f = open(sys.argv[-1],'rb')

pcap = dpkt.pcap.Reader(f)

host\_ips\_syn={}

host\_ips\_syn\_ack={}

for timestamp,buf in pcap:

try:

eth = dpkt.ethernet.Ethernet(buf)

except (dpkt.dpkt.UnpackError,IndexError):

continue

# print(':'.join('%02x' % compat\_ord(b) for b in eth.src))

if not isinstance(eth.data, dpkt.ip.IP):

#print('Non IP Packet type not supported %s\n' % eth.data.\_\_class\_\_.\_\_name\_\_)

continue

ip = eth.data

ip\_src = socket.inet\_ntoa(ip.src)

ip\_dst = socket.inet\_ntoa(ip.dst)

#print(ip\_src,ip\_dst)

# We are only interested in TCP

if ip.p != dpkt.ip.IP\_PROTO\_TCP:

continue

tcp = ip.data

if tcp.flags & dpkt.tcp.TH\_SYN and not (tcp.flags & dpkt.tcp.TH\_ACK): #If syn flag in the packet

if ip\_src in host\_ips\_syn: #if this ip exists in the src\_ips-syn map, add count

host\_ips\_syn[ip\_src] += 1

else:

host\_ips\_syn[ip\_src] = 1 #else add ip to the map and initialize

if (tcp.flags & dpkt.tcp.TH\_SYN) and (tcp.flags & dpkt.tcp.TH\_ACK): #If syn and ack flags in the packet

if ip\_dst in host\_ips\_syn\_ack: #if destination ip exists in the dest\_ip-syn/ack map, add count

host\_ips\_syn\_ack[ip\_dst] += 1

else:

host\_ips\_syn\_ack[ip\_dst] = 1 #else add ip to map initialize

ips = []

# #For every ip in the source ips, compare the count syn+ack it received vs syn it sent

for ip in host\_ips\_syn:

#print(ip,":",host\_ips\_syn[ip])

if ip in host\_ips\_syn\_ack and host\_ips\_syn[ip]>=3\*host\_ips\_syn\_ack[ip]: #if that ip exists in source ip list and if the number of syn packets >= 3\* number of destination syn+ack packets,

ips.append(ip)

elif ip not in host\_ips\_syn\_ack and host\_ips\_syn[ip]:

ips.append(ip)

for i in ips:

print(i)

References:

<https://dpkt.readthedocs.io/en/latest/_modules/dpkt/ethernet.html>

<https://dpkt.readthedocs.io/en/latest/api/api_auto.html#dpkt.ethernet.Ethernet.pack_hdr>

<https://github.com/kbandla/dpkt/issues/232>

<https://stackoverflow.com/questions/25370010/parsing-ip-address-with-dpkt>

<https://dpkt.readthedocs.io/en/latest/_modules/examples/print_packets.html>

<https://dpkt.readthedocs.io/en/latest/print_packets.html>