CS241 Coursework 2021-2022

Operating Systems and Computer Networks

The Task:
For this coursework, you will implement a basic intrusion detection system. This will test your understanding of TCP/IP protocols (networks) and threading (OS) as well as your ability to develop a non-trivial program in C. The coursework contributes 20% to your total marks in the module.

I. INTRODUCTION

The goal of this project is to detect potentially malicious traffic in high-throughput networks. It implements multithreading and detects the SYN flooding attack, ARP cache poisoning, and Blacklisted URLs.

II. CRITICAL DESIGN DECISIONS

A. Packet breakdown

```
struct ether_header *linklayer = (struct ether_header *)packet;
struct iphdr *iplayer = (struct iphdr *) (packet+14);
struct tcphdr *tcplayer = (struct tcphdr *) (packet+14 + iplayer->ihl*4);
```

(Figure 1: Define the ethernet, ip and tcp layer)

The location of IP is ethernet packet location plus the size of ethernet header, which is 14 bytes(fixed), SIZE_ ETHERNET = 14. The location of TCP is packet location plus ethernet header plus the length of IP header. However, the IP header does not have a fixed length. The length of the IP header depends on the value stored in IHL(stores the length of the header in 4 bits words). Therefore, the location of TCP equals to packet plus 14 plus IHL * 4. [1]

Variable	Location (in bytes)
`sniff_ethernet`	X
`sniff_ip`	X + `SIZE_ETHERNET`
`sniff_tcp`	X + `SIZE_ETHERNET` + {IP header length}
`payload`	X + `SIZE_ETHERNET` + {IP header length} + {TCP header length}

(Figure 2: Location of ethernet, ip and tcp)[2]

B. Function of each file

The main code part is divided into 4 files.

1. analysis.c: detect syn flooding attack, arp cache poisoning and blacklisted url.
2. sniff.c: continuously capture packets and debug.
3. dispatch.c: realize multithreading
4. vecs.c: use vector to count the number of different ips.

C. Intrusion detection

1. SYN flooding attack

TCP 3-way handshake process can be seen as a way of how TCP connection is established. [3] In a SYN flooding attack, the malicious sender sends overwhelming numbers of SYN requests and does not send the ACK. This occupies a lot of resources at the server. Therefore, if syn is true and other control bits(urg, ack, psh, rst, syn, fin)are false, then can be considered a syn attack. [4] Then, record different IP addresses by adding the IP addresses to the vector. If there are already duplicate

values, there is no need to add.

```
if(tcplayer->syn){
  if(!(tcplayer->urg && tcplayer->ack && tcplayer->psh && tcplayer->rst && tcplayer->fin)){
    g_nums_syn = g_nums_syn + 1;
    if(IsContainValue(iplayer->saddr) == 0)
      g_nums_ips = g_nums_ips + 1;
  }
}
```

(Figure 3: Detect syn attack and add the ip addresses to the vector)

2. ARP cache poisoning

ARP poisoning is a MitM attack that allows an attacker to intercept communications between network devices. [4] If the same MAC address but different IP addresses are detected, it means an ARP attack is taking place.

```
if(ntohs(linklayer->ether_type) == ETHERTYPE_ARP){
  const unsigned char *linkPackets = packet + ETH_HLEN;
  struct struct_arp_packet *arp_Packet = (struct struct_arp_packet *) linkPackets;
  struct arphdr *arp_Header = (struct arphdr *) &arp_Packet->ea_hdr;
  if(ntohs(arp_Header->ar_op) == ARPOP_REPLY){
      g_nums_arp = g_nums_arp +1;
  }
}
```

(Figure 4: Detect arp cache poisoning)

3. Blacklisted URLs

According to the requirements of coursework, www.google.co.uk and www.bbc.com are identified as suspicious domains that we wish to monitor. Need to determine whether it is suspicious domains by detecting the domain name, port number, the string of this web address in the HTTP header in the Data offset byte.

D. Multithreading

The benefits of multithreaded programming include increased responsiveness, shared resources, more economical, and increased scalability. [5] The code of multithreading is mainly realized in dispatch.c and sniff.c.

```
void *thread_func(void *arg)
{
  pthread_mutex_lock(&mut);
  struct info *base = (struct info *)(arg);
  analyse(base->header, base->packet, base->verbose);
  pthread_mutex_unlock(&mut);
  pthread_exit(NULL);
}
```

(Figure 5: *thread_func in dispatch.c)

Unlimited threads may could cause excessive consumption of resources, therefore, thread pool is used here to achieve multithreading. The thread pool is to create a fixed number of threads when the process startup, and then put them into a thread pool. [6]

```
pthread_t thread[2];
pthread_mutex_t mut;
```

(Figure 6: Thread pool model ---- 2 threads are used)

The benefits of thread pools include using existing threads to be faster than creating threads, limiting the number of threads, and allowing different strategies to handle different tasks. [6]

III. TESTING

1. SYN flooding attack

Use the command hping3 -c 100 -d 120 -S -w 64 -p 80 -i u100 --rand-source localhost, hping3 -c 100 -d 120 -S -w 64 -p 80 -i u100 --rand-source localhost and send 100 and 1000 packets respectively.

```
root@cs241:~/cs241/skeleton/src# ../build/idsniff -i lo
../build/idsniff invoked. Settings:
        Interface: lo
        Verbose: 0
SUCCESS! Opened lo for capture
^CIntrusion Detection Report:
100 SYN packets detected from 100 different IPs (syn attack)
0 ARP responses (cache poisoning)
0 URL Blacklist violations
root@cs241:~/cs241/skeleton/src# ../build/idsniff -i lo
../build/idsniff invoked. Settings:
        Interface: lo
        Verbose: 0
SUCCESS! Opened lo for capture
^CIntrusion Detection Report:
1000 SYN packets detected from 1000 different IPs (syn attack)
0 ARP responses (cache poisoning)
0 URL Blacklist violations
```

2. ARP cache poisoning

Use the command python3 arp-poison.py to test the ARP cache poisoning.

```
root@cs241:~/cs241/skeleton/src# ../build/idsniff -i lo
../build/idsniff invoked. Settings:
        Interface: lo
        Verbose: 0
SUCCESS! Opened lo for capture
^CIntrusion Detection Report:
0 SYN packets detected from 0 different IPs (syn attack)
1 ARP responses (cache poisoning)
0 URL Blacklist violations
```

3. Blacklisted URLs

Send the wget www.google.co.uk, wget www.bbc.com, and wget www.google.co.uk and wget www.bbc.com. In the eth0 interface, the output is:

```
root@cs241:~/cs241/skeleton/src# ../build/idsniff -i eth0
../build/idsniff invoked. Settings:
        Interface: eth0
        Verbose: 0
SUCCESS! Opened eth0 for capture
==========================
Blacklisted URL violation detected
Source IP address:10.0.2.15
Destination IP address:172.217.16.227
==========================
^CIntrusion Detection Report:
2 SYN packets detected from 2 different IPs (syn attack)
0 ARP responses (cache poisoning)
1 URL Blacklist violations
root@cs241:~/cs241/skeleton/src# ../build/idsniff -i eth0
../build/idsniff invoked. Settings:
        Interface: eth0
        Verbose: 0
SUCCESS! Opened eth0 for capture
==========================
Blacklisted URL violation detected
Source IP address:10.0.2.15
Destination IP address:212.58.233.248
==========================
^CIntrusion Detection Report:
4 SYN packets detected from 2 different IPs (syn attack)
0 ARP responses (cache poisoning)
1 URL Blacklist violations
```

```
root@cs241:~/cs241/skeleton/src# ../build/idsniff -i eth0
../build/idsniff invoked. Settings:
        Interface: eth0
        Verbose: 0
SUCCESS! Opened eth0 for capture
==========================
Blacklisted URL violation detected
Source IP address:10.0.2.15
Destination IP address:172.217.16.227
==========================
==========================
Blacklisted URL violation detected
Source IP address:10.0.2.15
Destination IP address:212.58.233.248
==========================
^CIntrusion Detection Report:
6 SYN packets detected from 3 different IPs (syn attack)
0 ARP responses (cache poisoning)
2 URL Blacklist violations
```

IV. CONCLUSION

The project uses multi-threading to improve operating efficiency and can capture different attacks at the same time. However, the ability to capture blacklisted URLs needs further testing.

V. REFERENCES

[1] A. Mukhopadhyay, "CS 241: Selected topics in Networking", 2021. [Online]. Available: https://warwick.ac.uk/fac/sci/dcs/teaching/material/cs241/cn2021/cn_lec2_courseworktopics.pdf.

[2] T. Carstens, "Programming with pcap | TCPDUMP & LIBPCAP", *Tcpdump.org*. [Online]. Available: https://www.tcpdump.org/pcap.html.

[3] "TCP 3-Way Handshake Process - GeeksforGeeks", *GeeksforGeeks*, 2021. [Online]. Available: https://www.geeksforgeeks.org/tcp-3-way-handshake-process/.

[4] A. Mukhopadhyay, "CS241 Coursework 2021-2022", 2021. [Online]. Available: https://warwick.ac.uk/fac/sci/dcs/teaching/material/cs241/coursework21-22/.

[5] A. Silberschatz, P. Galvin and G. Gagne, *Applied operating system concepts*, 9th ed. John Wiley, pp. 163-164

[6] A. Silberschatz, P. Galvin and G. Gagne, *Applied operating system concepts*, 9th ed. John Wiley, pp. 177-178