CMSC 626 Principles of Computer Security

Project

Exercise 3

Faisal Rasheed Khan Shrenik PolySetty

VB02734 AZ61492

vb02734@umbc.edu

1.

a. Team Information:

   i. Name: **Faisal Rasheed Khan**

   University Id: **VB02734**

   ii. Name: **Shrenik PolySetty**

   University Id: **AZ61492**

b. Secret Key: “Principles of Computer Security”

c.

```
nc -l 12345
nc 130.85.220.34 12345
python3 rc4s.py -k 'Principles of Computer Security' -m 'Hello, it is a nice sunny day and we should enjoy the weather'
python3 rc4s.py <vm1pipe | nc 130.85.220.34 12345 >vm1pipe
nc -l 12345 <vm2pipe | python rc4r.py >vm2pipe
mkfifo vm1pipe
python3 rc4s.py -k 'Principles of Computer Security' -m 'Hello, it is a nice sunny day and we should enjoy the weather' | nc 130.85.220.34 12345
nc -l 12345 | python3 rc4r.py
python3 rc4s.py
python3 diffehellman.py
cat tcpdumpcapture.cap
nano rc4s.py
ls
ifconfig -a
tcpdump -D
tcpdump -n -i ens160 -w tcpdumpcapture.cap host 130.85.121.106 and port 12345
tcpdump -n -i ens160 -w tcpdumpcapture.cap host 130.85.220.34 and port 12345
tcpdump -n -i ens160 -w tcpdumpcapture.cap host 130.85.121.106 and port 12345
tcpdump -r capture.pcap
tcpdump -n -i ens160 -w vm1capture2.pcap host 130.85.220.34
```

d. The challenges faced were:

- While encrypting and decrypting the pain text and cipher text, faced issue of unable to convert bytes to text

  Resolved this issue with the help of encoding and decoding with ‘utf-8’

- Faced issue while connecting to the other Virtual Machine to send the encrypted plain text using netcat command

  Resolved the issue by passing the command of netcat to subprocess.Popen()

- While encrypting and decrypting the pain text and cipher text, faced issue of unable to decode with ‘utf-8’ as it was in other format

  Resolved by removing unnecessary encoding.

- Faced challenges while using wireshark, tshark as access to install that was not there

  Resolved using tcpdump command

- The tcpdump command was capturing every data incoming

  Resolve by applying proper filters

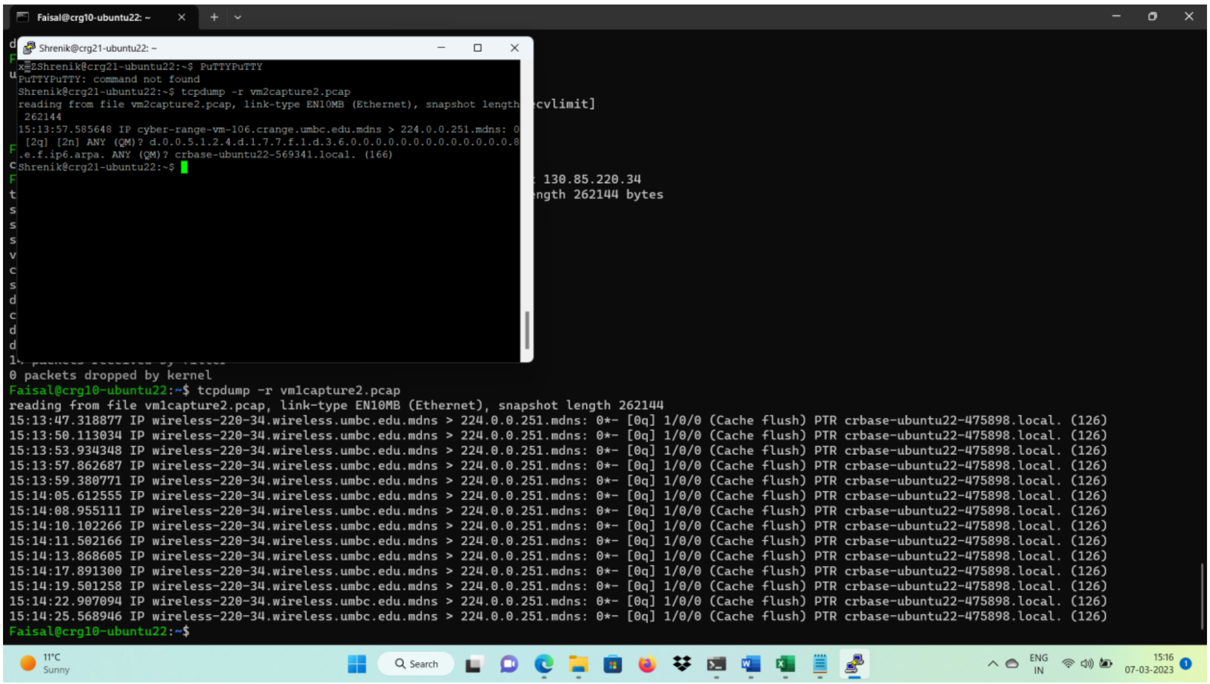
- For tcpdump command, which interface to use to capture the data was a problem

  Resolved by using this command, tcpdump -D

e. Successfully implemented the RC4 Algorithm to encrypt and decrypt the text over the communicating channels between two virtual machines. Learnt different ways of sending the data via command line arguments or incorporating everything in the python file. Successfully implemented the tcpdump/wireshark capture to capture the real time packets sent/received.

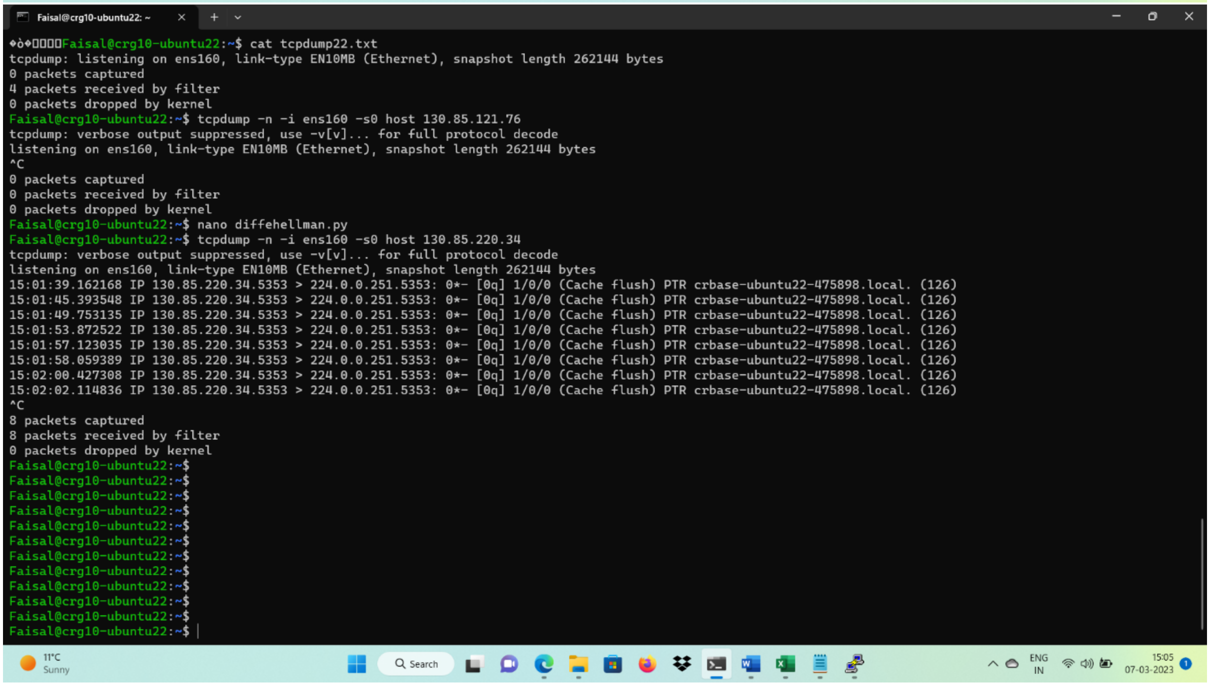
f. References:

CH02-CompSec4e_accessible_L03 (blackboardcdn.com)
How To Use Netcat to Establish and Test TCP and UDP Connections | DigitalOcean
L07-CH21-CompSec4e_accessible (blackboardcdn.com)
Primitive Root - Algorithms for Competitive Programming (cp-algorithms.com)
tcp_client.py
tcp_server.py
RC4-KeyGeneration(1).py
subprocess — Subprocess management — Python 3.11.2 documentation
Tcpdump Command in Linux | Linuxize



```
x▯ZShrenik@crg21-ubuntu22:~$ PuTTYPuTTY
PuTTYPuTTY: command not found
Shrenik@crg21-ubuntu22:~$ tcpdump -r vm2capture2.pcap
reading from file vm2capture2.pcap, link-type EN10MB (Ethernet), snapshot length
 262144
15:13:57.585648 IP cyber-range-vm-106.crange.umbc.edu.mdns > 224.0.0.251.mdns: 0
 [2q] [2n] ANY (QM)? d.0.0.5.1.2.4.d.1.7.7.f.1.d.3.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8
.e.f.ip6.arpa. ANY (QM)? crbase-ubuntu22-569341.local. (166)
Shrenik@crg21-ubuntu22:~$
```

```
ecvlimit]
130.85.220.34
ngth 262144 bytes
0 packets dropped by kernel
Faisal@crg10-ubuntu22:~$ tcpdump -r vm1capture2.pcap
reading from file vm1capture2.pcap, link-type EN10MB (Ethernet), snapshot length 262144
15:13:47.318877 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:13:50.113034 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:13:53.934348 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:13:57.862687 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:13:59.380771 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:05.612555 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:08.955111 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:10.102266 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:11.502166 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:13.868605 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:17.891300 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:19.501258 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:22.907094 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:14:25.568946 IP wireless-220-34.wireless.umbc.edu.mdns > 224.0.0.251.mdns: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
Faisal@crg10-ubuntu22:~$
```



```
◆ò◆▯▯▯▯Faisal@crg10-ubuntu22:~$ cat tcpdump22.txt
tcpdump: listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144 bytes
0 packets captured
4 packets received by filter
0 packets dropped by kernel
Faisal@crg10-ubuntu22:~$ tcpdump -n -i ens160 -s0 host 130.85.121.76
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
Faisal@crg10-ubuntu22:~$ nano diffehellman.py
Faisal@crg10-ubuntu22:~$ tcpdump -n -i ens160 -s0 host 130.85.220.34
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144 bytes
15:01:39.162168 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:01:45.393548 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:01:49.753135 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:01:53.872522 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:01:57.123035 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:01:58.059389 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:02:00.427308 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
15:02:02.114836 IP 130.85.220.34.5353 > 224.0.0.251.5353: 0*- [0q] 1/0/0 (Cache flush) PTR crbase-ubuntu22-475898.local. (126)
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$
Faisal@crg10-ubuntu22:~$ |
```