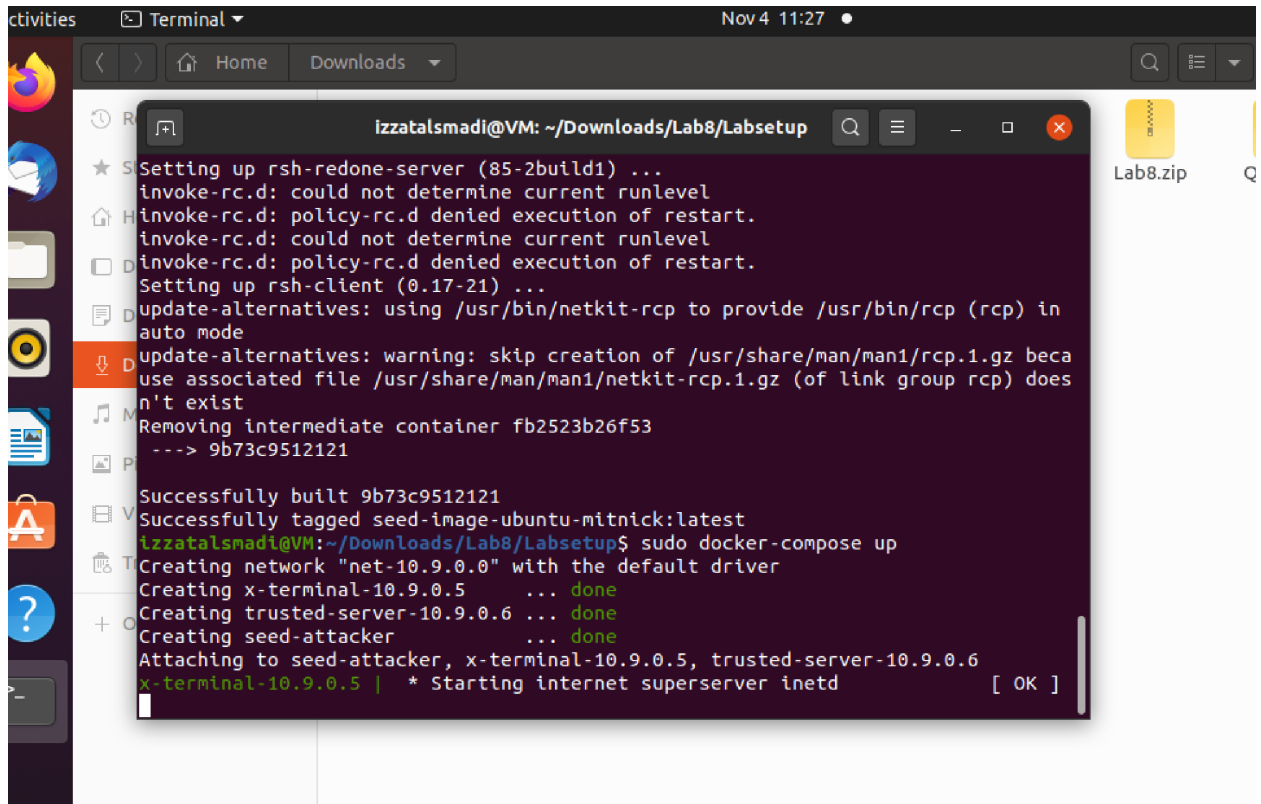


Lab8 Demo 2024

- Build and start the container



The screenshot shows a terminal window titled "Terminal" with the date "Nov 4 11:27". The user is logged in as "izzatalsmadi@VM" and is in the directory "~/Downloads/Lab8/Labsetup". The terminal output shows the following steps:

```
Setting up rsh-redone-server (85-2build1) ...
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of restart.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of restart.
Setting up rsh-client (0.17-21) ...
update-alternatives: using /usr/bin/netkit-rcp to provide /usr/bin/rcp (rcp) in
auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/rcp.1.gz beca
use associated file /usr/share/man/man1/netkit-rcp.1.gz (of link group rcp) does
n't exist
Removing intermediate container fb2523b26f53
--> 9b73c9512121
Successfully built 9b73c9512121
Successfully tagged seed-image-ubuntu-mitnick:latest
izzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ sudo docker-compose up
Creating network "net-10.9.0.0" with the default driver
Creating x-terminal-10.9.0.5 ... done
Creating trusted-server-10.9.0.6 ... done
Creating seed-attacker ... done
Attaching to seed-attacker, x-terminal-10.9.0.5, trusted-server-10.9.0.6
x-terminal-10.9.0.5 | * Starting internet superserver inetd [ OK ]
```

- Notice the network has 3 machines
 1. Attacker-10.9.0.105
 2. Trusted-server-10.9.0.6
 3. X-terminal-10.9.0.5Before the attack, we need to set up the trusted relationship between X-Terminal (10.9.0.5) and Trusted Server (10.9.0.6).
- 2. Login to x-terminal and trusted server
- Start the VMs

The image displays a series of terminal windows from a Kali Linux VM, documenting the setup of a Docker-based lab environment. The primary terminal window shows the user running `sudo docker exec -it x-terminal-10.9.0.5 bash`, which results in a `root@15b16e9794d1:/#` prompt. Overlaid windows show the following steps and messages:

- An error message: `'docker exec' requires at least 2 arguments. See 'docker exec --help'.`
- A usage message: `Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]`
- A message about removing an intermediate container: `Removing intermediate container fb2523b26f53 --> 9b73c9512121`
- A success message: `Successfully built 9b73c9512121`
- A success message: `Successfully tagged seed-image:izzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ sudo docker exec -it x-terminal-10.9.0.5 bash`
- A message about creating a network: `Creating network "net-10.9.0.0"`
- A message about creating a container: `Creating x-terminal-10.9.0.5`
- A message about creating a trusted server: `Creating trusted-server-10.9.0.`
- A message about creating a seed attacker: `Creating seed-attacker`
- A message about attaching to the seed attacker: `Attaching to seed-attacker, x-t`
- A message about starting the x-terminal container: `x-terminal-10.9.0.5 | * Starti`

- On X-Terminal: Set up the trust relationship

```

+ | izzatalsmadi@VM: ~/Downloads/Lab8/Labsetup
izzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ sudo docker exec -it x-terminal-10.9.0.5 bash
root@15b16e9794d1:/# su seed
seed@15b16e9794d1:/$ cd
$ seed@15b16e9794d1:~$ touch .rhosts
seed@15b16e9794d1:~$ echo 10.9.0.6 > .rhosts
def: seed@15b16e9794d1:~$ chmod 644 .rhosts
seed@15b16e9794d1:~$

```

- On Trusted Server: Verify the trust relationship

```
Run a command in a running container
izzatalsmadi@VM: ~/Downloads/Lab8/Labsetup$ sudo docker exec -it trusted
-server-10.9.0.6 bash
root@ea7c61ac50b8:/# su seed
seed@ea7c61ac50b8:/# rsh 10.9.0.5 date
Mon Nov  4 16:38:29 UTC 2024
seed@ea7c61ac50b8:/#
```

- Task 1: Simulated SYN flooding

```
izzatalsmadi@VM: ~/Downloads/Lab8/Labsetup
izzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ sudo docker exec -it x-terminal-10.9.0.5
root@15b16e9794d1:/# su seed
seed@15b16e9794d1:/# cd
seed@15b16e9794d1:~$ touch .rhosts
seed@15b16e9794d1:~$ echo 10.9.0.6 > .rhosts
seed@15b16e9794d1:~$ chmod 644 .rhosts
seed@15b16e9794d1:~$ exit
exit
root@15b16e9794d1:/# arp -s 10.9.0.6 aa:bb:cc:dd:ee:ff
root@15b16e9794d1:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   aa:bb:cc:dd:ee:ff   CM                    eth0
root@15b16e9794d1:/#
```

- Ping should work so I deleted ARP record to add ping first

```
izzatalsmadi@VM: ~/Downloads/Lab8/Labsetup
seed@15b16e9794d1:~$ echo 10.9.0.6 > .rhosts
seed@15b16e9794d1:~$ chmod 644 .rhosts
seed@15b16e9794d1:~$ exit
exit
root@15b16e9794d1:/# arp -s 10.9.0.6 aa:bb:cc:dd:ee:ff
root@15b16e9794d1:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   aa:bb:cc:dd:ee:ff   CM                    eth0
root@15b16e9794d1:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
^Z
[1]+  Stopped                  ping 10.9.0.6
root@15b16e9794d1:/# arp -d 10.9.0.6
root@15b16e9794d1:/# arp -n
root@15b16e9794d1:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.101 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.084 ms
64 bytes from 10.9.0.6: icmp_seq=5 ttl=64 time=0.069 ms
^Z
[2]+  Stopped                  ping 10.9.0.6
root@15b16e9794d1:/#
```

- Now ARP is added properly

```

root@15b16e9794d1:/# arp -n
root@15b16e9794d1:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.101 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.084 ms
64 bytes from 10.9.0.6: icmp_seq=5 ttl=64 time=0.069 ms
^Z
[2]+  Stopped                  ping 10.9.0.6
root@15b16e9794d1:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                  ether    02:42:0a:09:00:06    C                      eth0
root@15b16e9794d1:/#

```

- Now arp -s

```

10.9.0.6                  ether    02:42:0a:09:00:06    C                      eth0
root@15b16e9794d1:/# sudo arp -s 10.9.0.6 02:42:0a:09:00:06
bash: sudo: command not found
root@15b16e9794d1:/# arp -s 10.9.0.6 02:42:0a:09:00:06
root@15b16e9794d1:/#
superserver inetd

```

- Notice Flag mask change now

```

Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                  ether    02:42:0a:09:00:06    C                      eth0
root@15b16e9794d1:/# sudo arp -s 10.9.0.6 02:42:0a:09:00:06
bash: sudo: command not found
root@15b16e9794d1:/# arp -s 10.9.0.6 02:42:0a:09:00:06
root@15b16e9794d1:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                  ether    02:42:0a:09:00:06    CM                     eth0
root@15b16e9794d1:/#

```

- Task 2: Spoof TCP Connections and rsh Sessions

To launch the attack, we need to do the following:

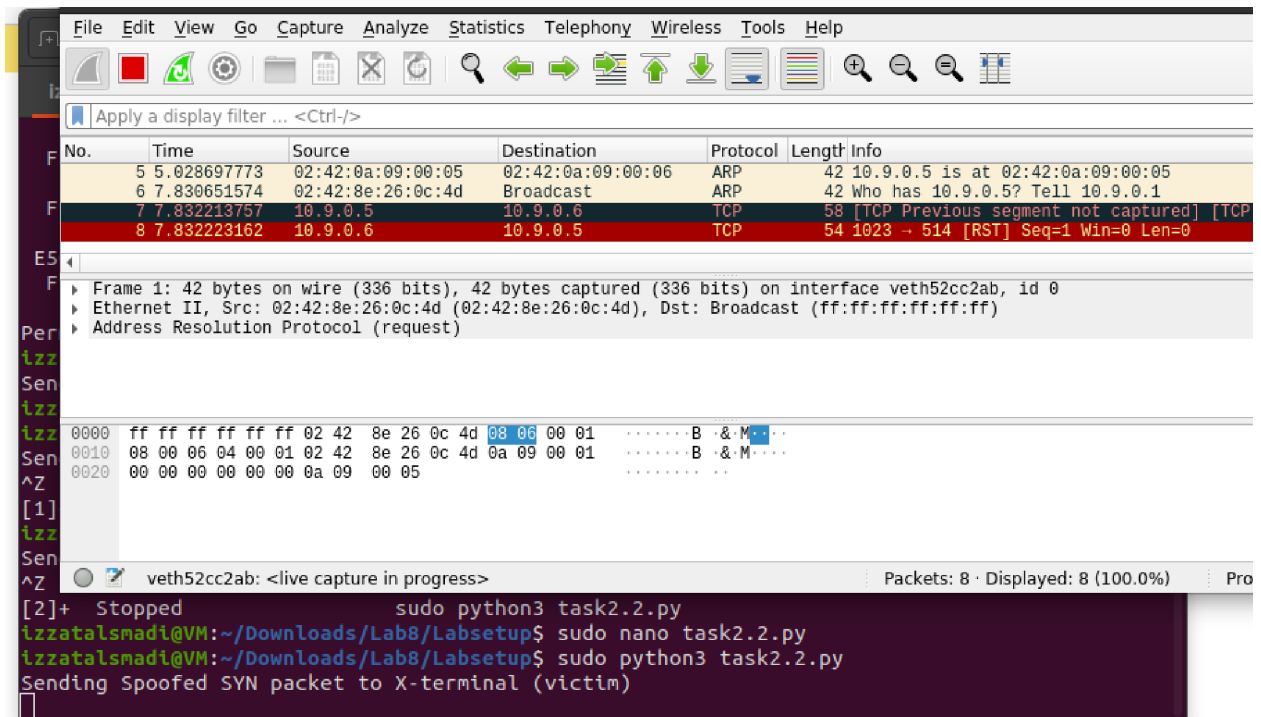
- Step 1: Spoof a SYN packet from Trusted server to X-terminal.

```

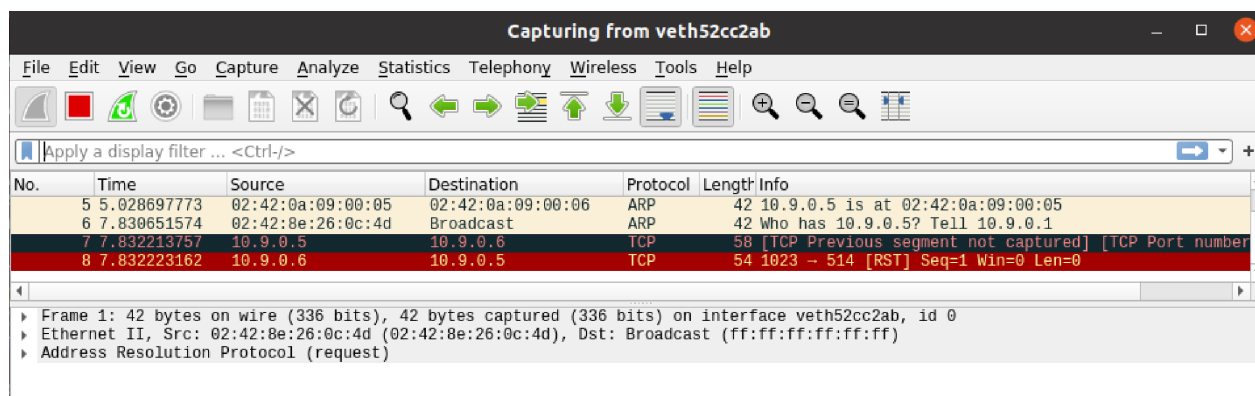
seed@lzzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ python3 task2.1.py
Mon N Sending Spoofed SYN packet to X-terminal (victim)
seed@lzzatalsmadi@VM:~/Downloads/Lab8/Labsetup$
Traceback (most recent call last):
  File "task2.1.py", line 9, in <module>
    send(pkt,verbose=0)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 345, in send
    socket = socket or conf.L3socket(*args, **kargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 398, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(type)) # noqa
  File "/usr/lib/python3.8/socket.py", line 231, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted
g netw lzzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ sudo python3 task2.1.py
g x-ter Sending Spoofed SYN packet to X-terminal (victim)
g trus lzzatalsmadi@VM:~/Downloads/Lab8/Labsetup$
g seec

```

- Step 2: Step 1 will trigger X-Terminal to send out a SYN+ACK. We need to spoof an ACK to finish the handshake protocol.



- Run task2.1 then task2.2 and check wireshark output (successful TCP connection)



- (Create a to the trusted server, then from that connection (while code is running), type: su seed, then rsh 10.9.0.5 date, then monitor output on the other trusted server terminal

```
Run a command in a running container
lzzataismadi@VM:~/Downloads/Lab8/Labsetup$ sudo docker exec -it trusted
-server-10.9.0.6 bash
root@ea7c61ac50b8:/# su seed
seed@ea7c61ac50b8:/# rsh 10.9.0.5 date
Mon Nov 4 16:38:29 UTC 2024
seed@ea7c61ac50b8:/# rsh 10.9.0.5 date
Mon Nov 4 17:08:08 UTC 2024
seed@ea7c61ac50b8:/#
```

- You can see that after getting the RSH data, X-Terminal will initiate the second connection and send it to the Trusted Server. We need to spoof an ACK. If this connection cannot be established, X-Terminal will abort. , then try the code below
- Then complete Task 3 (Task 3: Set Up a Backdoor)

Capturing from veth52cc2ab

No.	Time	Source	Destination	Protocol	Length	Info
5	5.028697773	02:42:0a:09:00:05	02:42:0a:09:00:06	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
6	7.830651574	02:42:8e:26:0c:4d	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.1
7	7.832213757	10.9.0.5	10.9.0.6	TCP	58	[TCP Previous segment not captured] [TCP Port number
8	7.832223162	10.9.0.6	10.9.0.5	TCP	54	1023 → 514 [RST] Seq=1 Win=0 Len=0
9	298.933800003	10.9.0.6	10.9.0.5	TCP	74	[TCP Port numbers reused] 1023 → 514 [SYN] Seq=0 Win
10	298.934340354	10.9.0.5	10.9.0.6	TCP	74	[TCP Port numbers reused] 514 → 1023 [SYN, ACK] Seq=
11	298.934395056	10.9.0.6	10.9.0.5	TCP	66	1023 → 514 [ACK] Seq=1 Ack=1 Win=502 Len=0 TSval=278
12	298.942392210	10.9.0.6	10.9.0.5	RSH	86	Session Establishment
13	298.942429230	10.9.0.5	10.9.0.6	TCP	66	[TCP ACKed unseen segment] [TCP Previous segment not
14	298.968687588	10.9.0.5	10.9.0.6	TCP	74	1023 → 1022 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
15	298.968787356	10.9.0.6	10.9.0.5	TCP	74	1022 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0
16	298.968806589	10.9.0.5	10.9.0.6	TCP	66	1023 → 1022 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=
17	298.988992518	10.9.0.5	10.9.0.6	RSH	67	[TCP ZeroWindowProbe] [TCP ACKed unseen segment] Seq
18	298.989114188	10.9.0.6	10.9.0.5	TCP	66	[TCP ACKed unseen segment] 1023 → 514 [ACK] Seq=21 A
19	299.005889747	10.9.0.5	10.9.0.6	RSH	95	[TCP ACKed unseen segment] [TCP Previous segment not
20	299.005909595	10.9.0.6	10.9.0.5	TCP	66	[TCP ACKed unseen segment] 1023 → 514 [ACK] Seq=21 A
21	299.008984098	10.9.0.5	10.9.0.6	TCP	66	1023 → 1022 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0
22	299.009106362	10.9.0.5	10.9.0.6	TCP	66	[TCP ACKed unseen segment] 514 → 1023 [FIN, ACK] Seq
23	299.009122360	10.9.0.6	10.9.0.5	TCP	66	[TCP ACKed unseen segment] 1023 → 514 [FIN, ACK] Seq
24	299.009195837	10.9.0.5	10.9.0.6	TCP	66	[TCP ACKed unseen segment] 514 → 1023 [ACK] Seq=375
25	299.009214109	10.9.0.6	10.9.0.5	TCP	66	1022 → 1023 [FIN, ACK] Seq=1 Ack=2 Win=65280 Len=0
26	299.009232674	10.9.0.5	10.9.0.6	TCP	66	1023 → 1022 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TSval=
27	304.036779346	02:42:0a:09:00:06	02:42:0a:09:00:05	ARP	42	Who has 10.9.0.5? Tell 10.9.0.6
28	304.036941453	02:42:0a:09:00:05	02:42:0a:09:00:06	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth52cc2ab, id 0

Ethernet II, Src: 02:42:8e:26:0c:4d (02:42:8e:26:0c:4d), Dest: Broadcast (ff:ff:ff:ff:ff:ff)

```

0000  ff ff ff ff ff 02 42 8e 26 0c 4d 08 06 00 01  .....B..&M....
0010  08 00 06 04 00 01 02 42 8e 26 0c 4d 0a 09 00 01  .....B..&M....
0020  00 00 00 00 00 00 0a 09 00 05  .....

```