Attack Mechanisms

A Taxonomy of Attacks

S&B: Chapter 7*

Some Well Know Types of Attacks

Denial of Service Attacks

DoS and DDoS

- *Floods*
- *Spoofs*
- *Smurfs*

Examples

- *ICMP flooding*
- *Source address spoofing*

Some Other Attacks

IP fragmentation

Man in the Middle

Replays

TCP hijacking

ARP poisoning

...

Buffer Overflows

Denial of Service

Both DoS and DDoS

DoS Definition
Applies to DoS & DDoS

An action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bit rate capacity, network resources and disk space

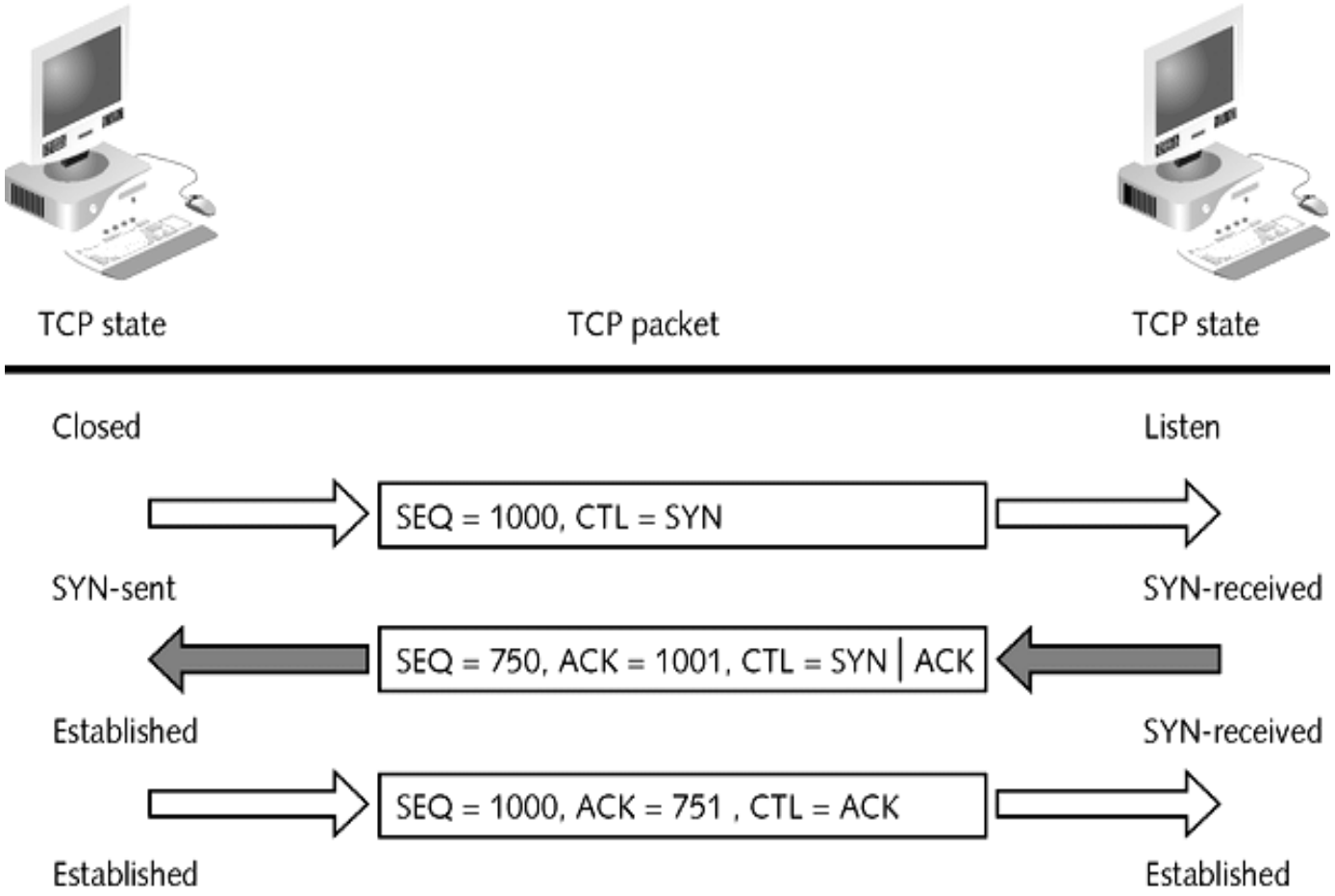
Usually overloads

- *Network and network bit rate capacity*
- *Computer system resources*
- *Application software resources*

Sync Attacks

Sync attacks exploit the TCP three-way handshake

Normal TCP 3-way Handshake



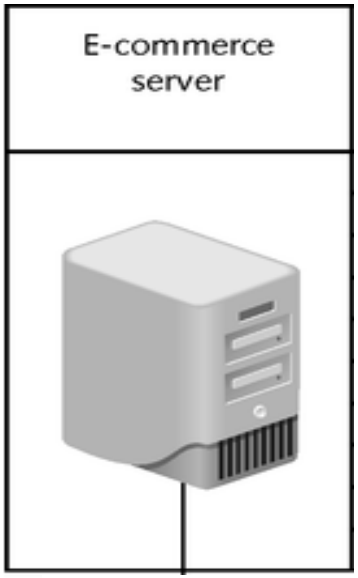
Sync Spoof

A Sync Spoof saturates the target computer's TCP connection tables with half-open connections

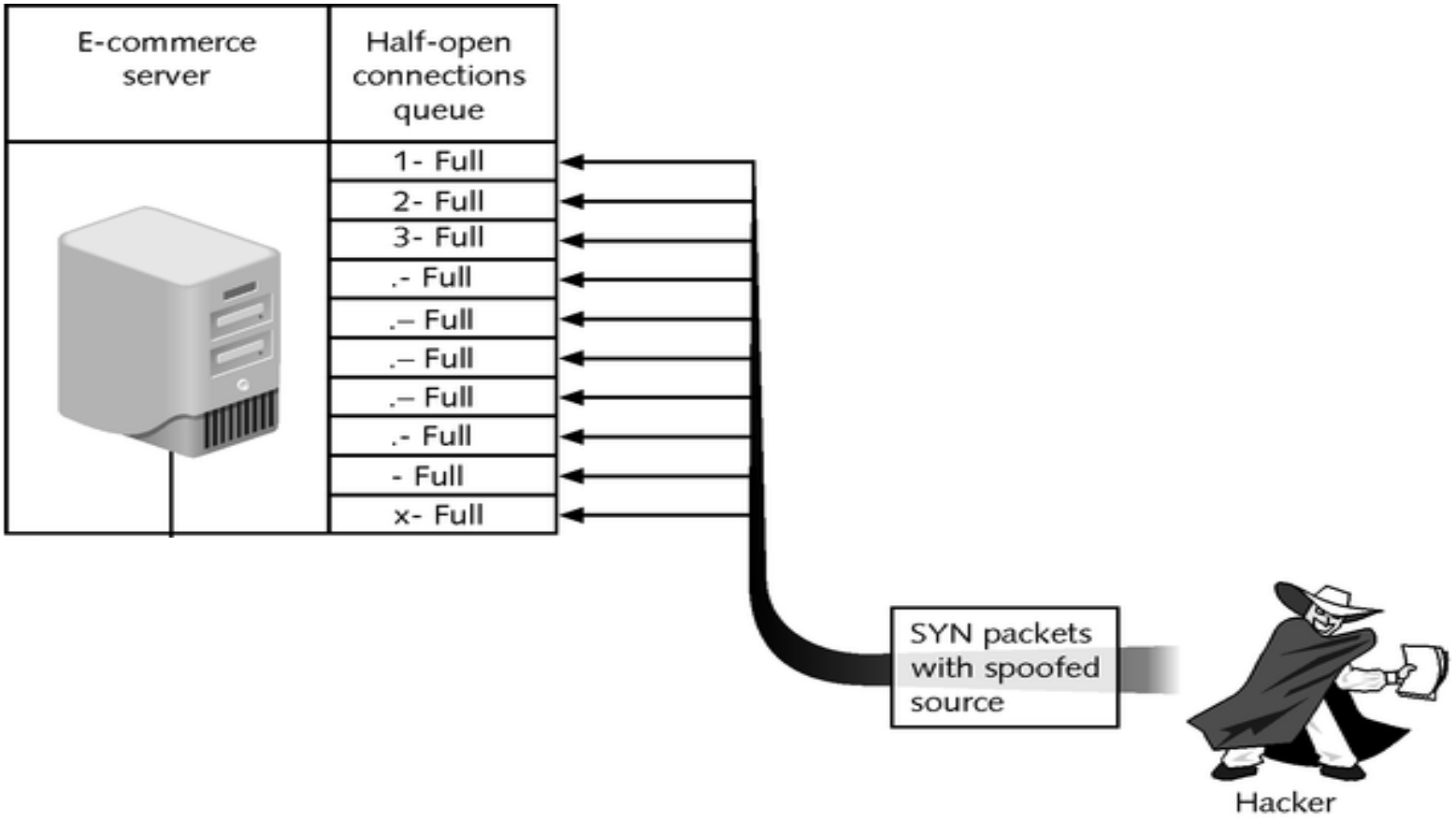
The target TCP will keep half-open connections and send sync/ack responses several times before clearing the partial connection

This inhibits server's ability to accept new TCP connections from other legitimate clients

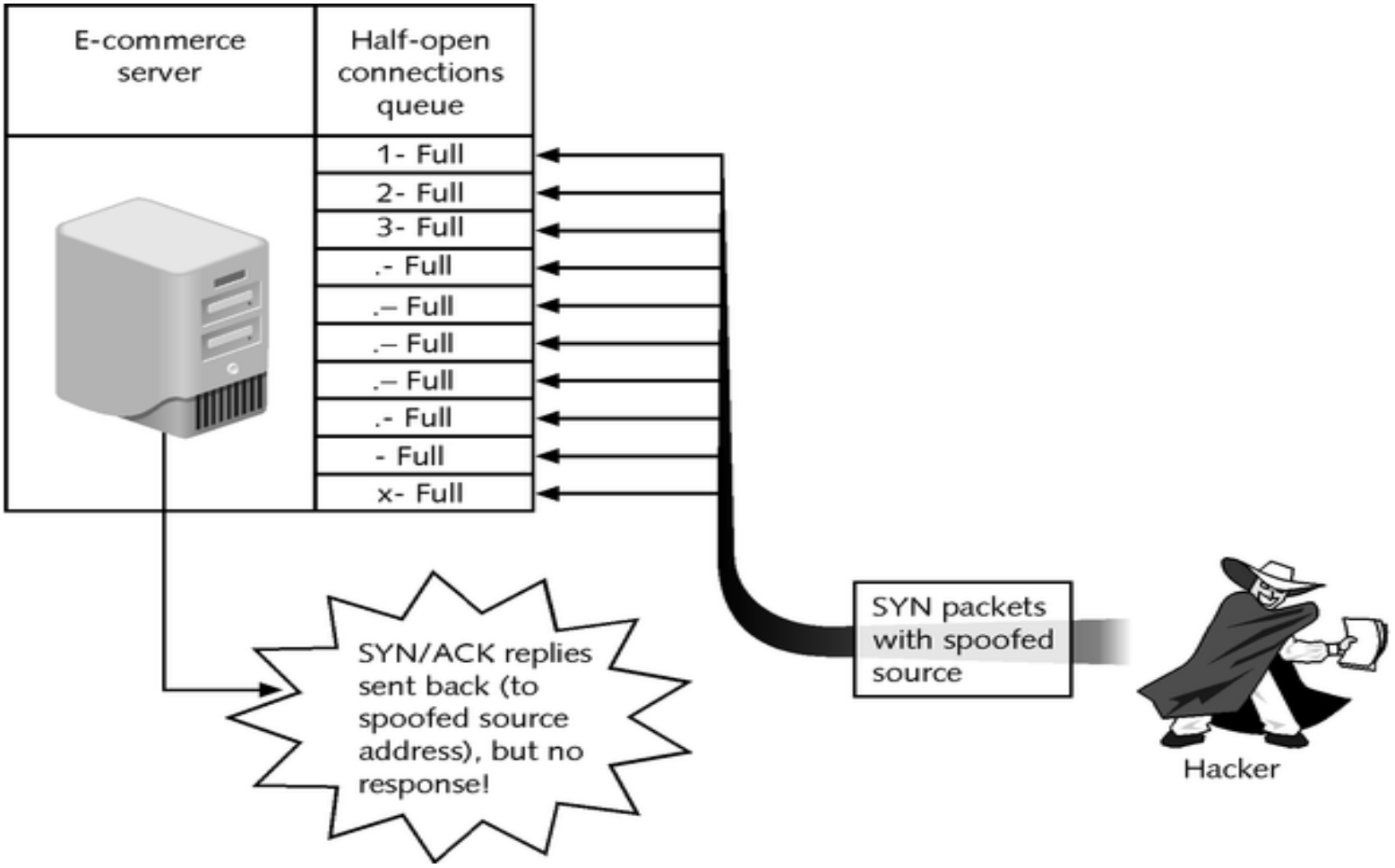
SYN Spoof Example-1



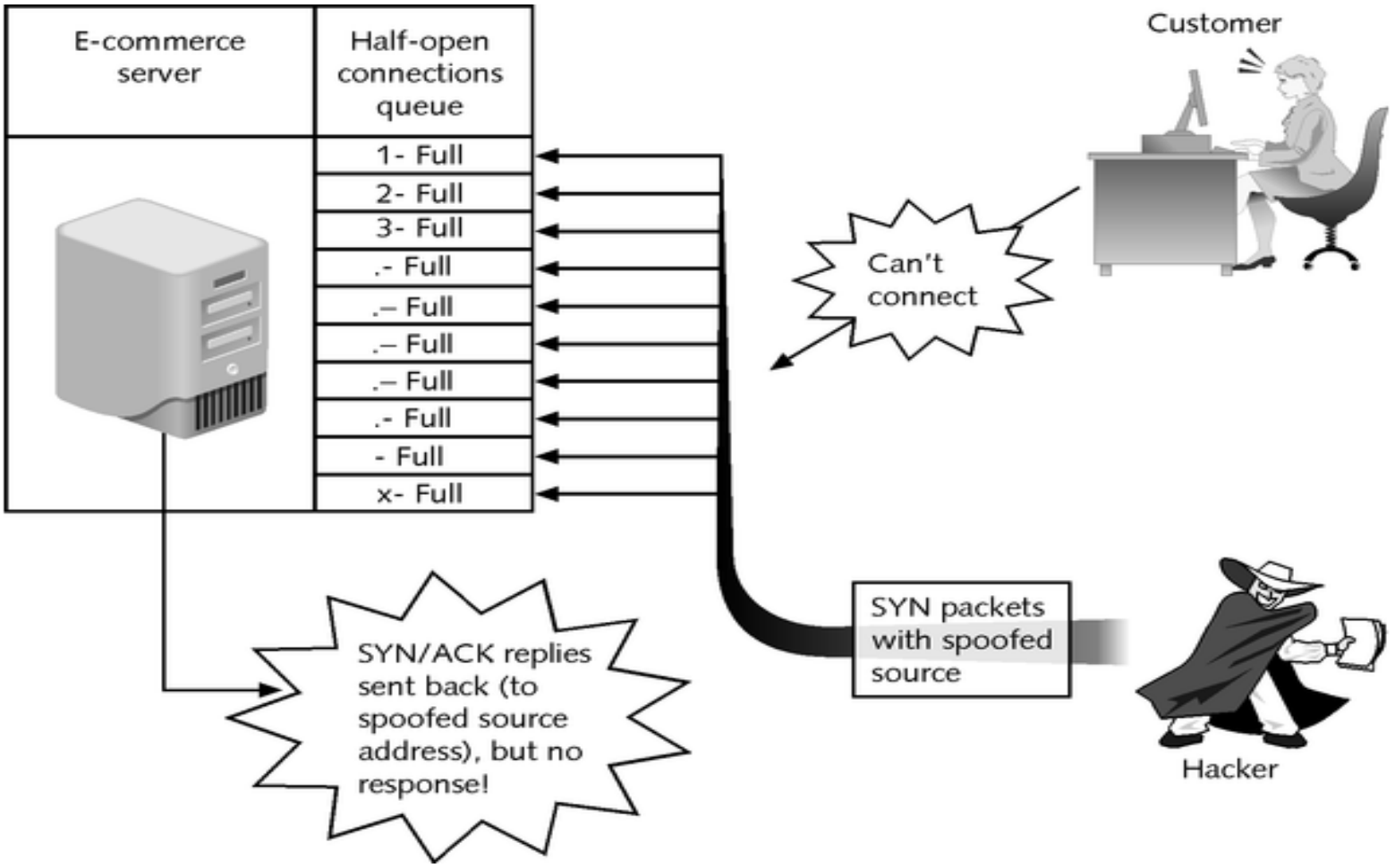
SYN Spoof Example-2



SYN Spoof Example-3



SYN Spoof Example-4



SYN Spoof Observations

Spoof traffic volume can be low relative to the bit rate of the target computer's connection

The server tables are not usually too extensive

An attack computer with a low bit rate connection can effectively attack a target connected to the Internet at a much higher bit rate

All you need to do is keep the target's TCP connection tables filled

You could do it from you home if you ISP allowed it

Very popular type of attack in the early days

Defenses

Rate limit the number of TCP connections that a server will accept

But this will limit the performance of network server systems

Sync Flood

Similar to Sync Spoof, but…

Difference is that a Sync Flood saturates the target's network connection by sending a large number of TCP sync segments

Here the segments can be rejected by the target

Attacker's purpose is NOT to saturate the TCP connection tables

Attacker needs high bit rate to do this attack

Alternately can employ multiple attacker computers

A DDoS attack

SYN Flood Observations

Target computer cannot defend against this. Why?

- *Attacker is not explicitly attacking the target computer*
- *Instead is attacking the network*

Defenses

- *Since the network connection is being attacked, the defense must be at the network level*
- *An IPS can be at least partially effective*

ICMP Flood

- Saturate a target's network connection with ICMP packets
  - *Use spoofed source addresses*
- Originally used PING packets
  - *Use to be allowed to pass firewalls for diagnostic purposes*
  - *Now many systems (including IIT) either totally or partially block PINGs -- at least pings from outside*
- Now the bad guys use other ICMP packet types
  - *These are usually needed for correct network operation*
  - *e.g., Destination & port unreachable, time exceeded…*

UDP Flood

Saturate a target's network connection with UDP packets to some port

Use spoofed source addresses

If the port is unreachable, often the target replies with a ICMP port unreachable, thus increasing the network traffic

Smurf

Non-OS specific attack that uses the network to amplify its effect on the victim

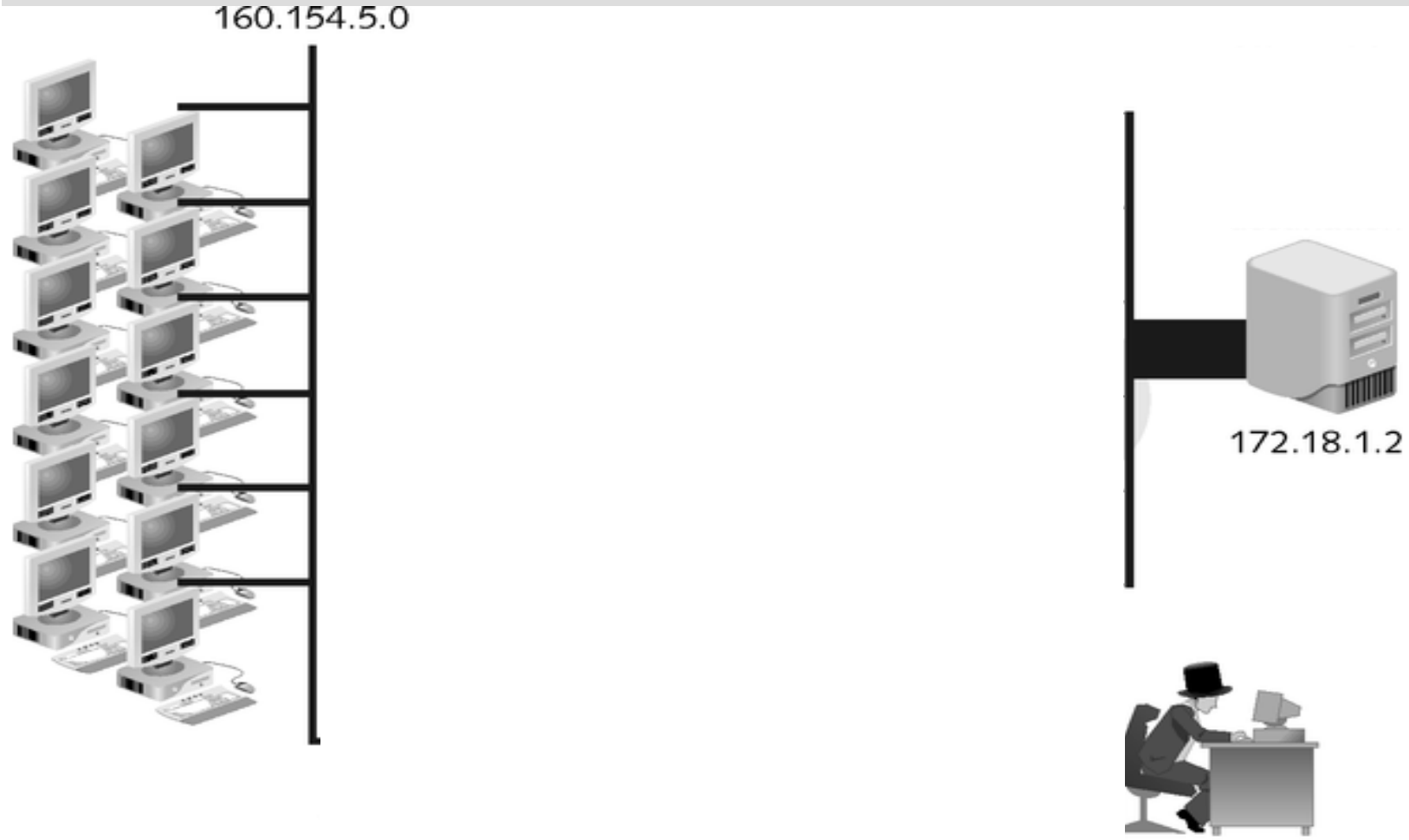
Indirectly floods a host with ICMP packets

Different from an ICMP Flood in that the target is not directly attacked by the attacker

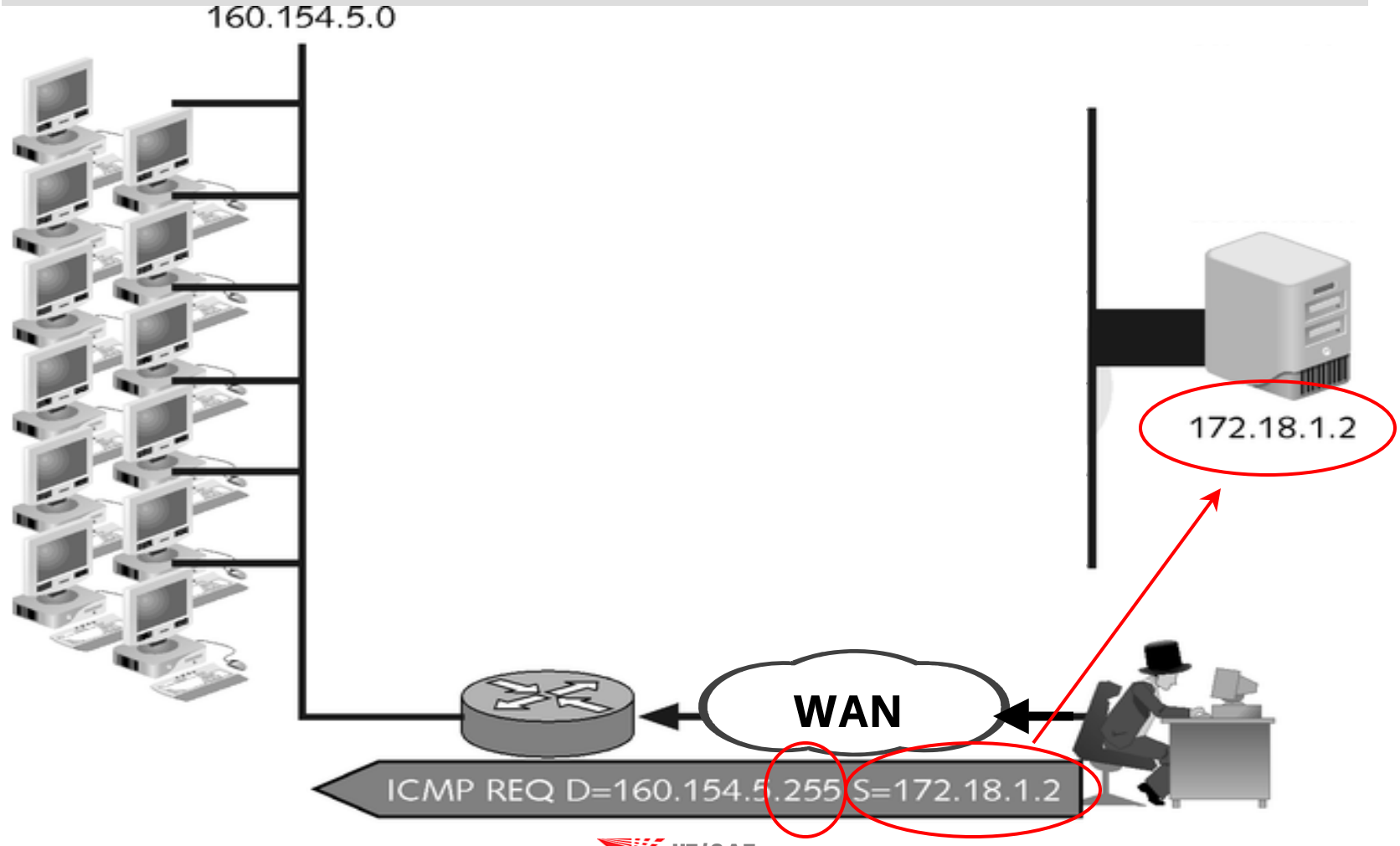
The attacker uses the network to amplify her attack

Saturates the target's Internet connection with bogus traffic and delays/prevents legitimate traffic from reaching its destination

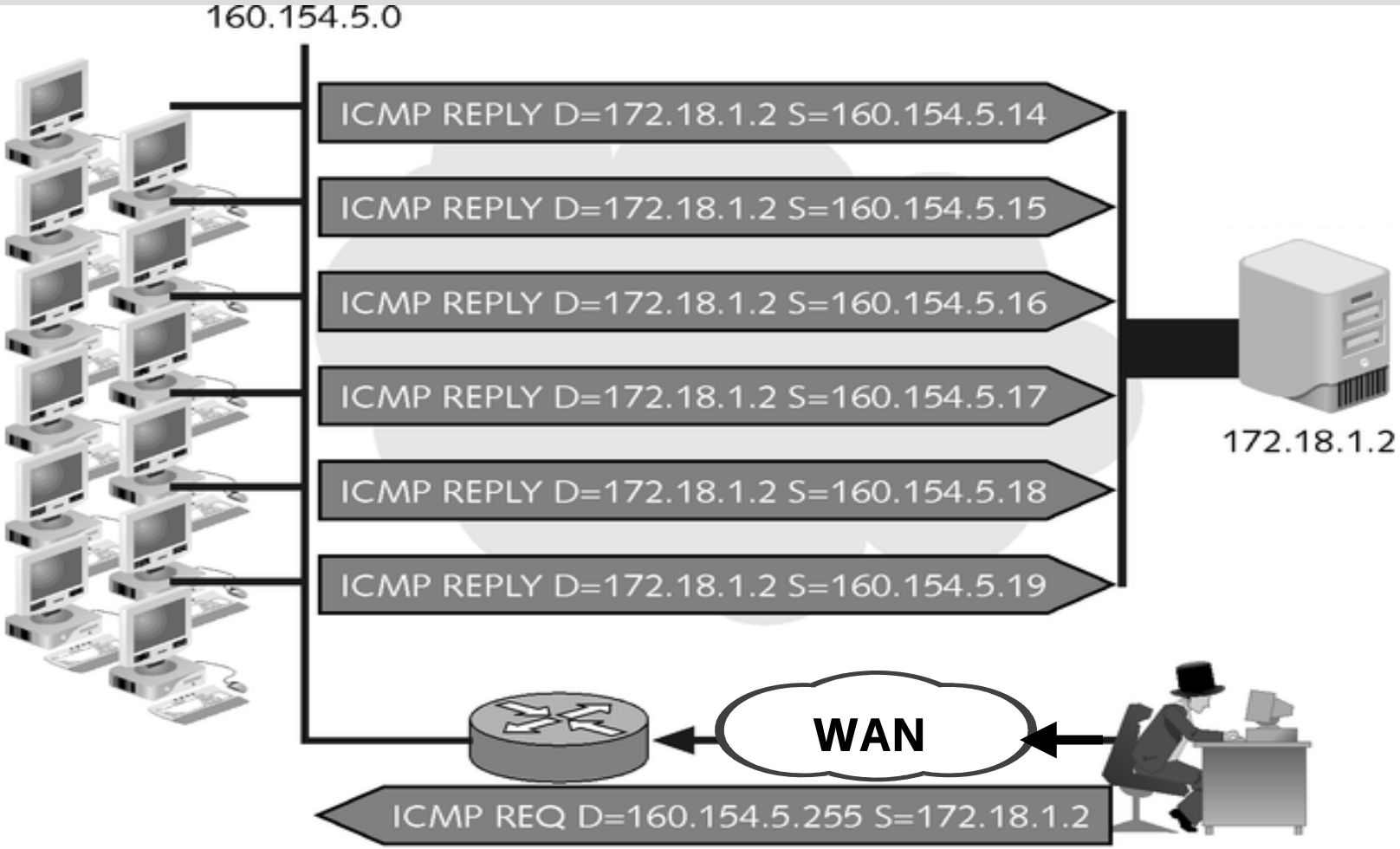
Smurf Example-1



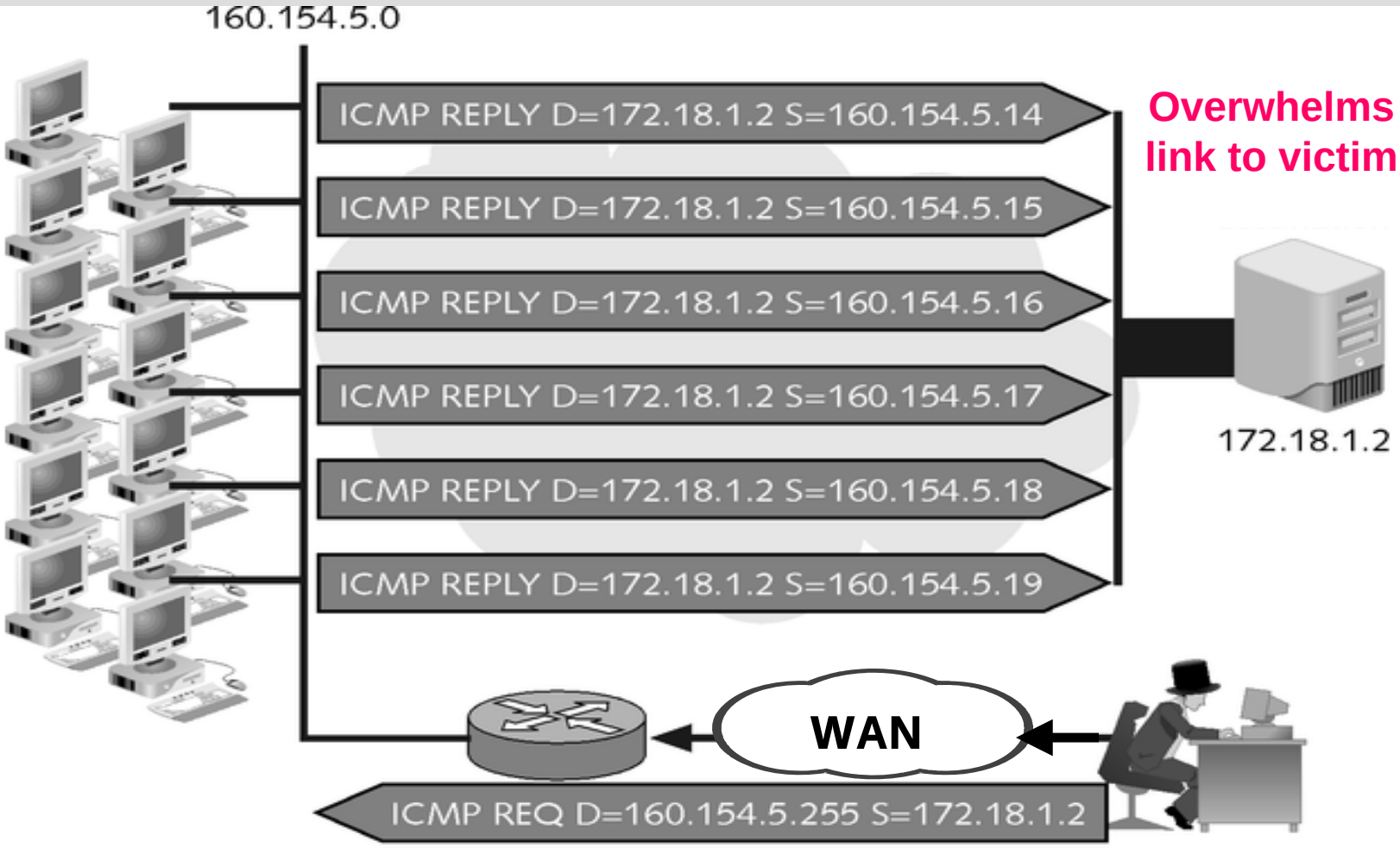
Smurf Example-2



Smurf Example-3



Smurf Example-4



IP Fragmentation

Some TCP/IP stacks in operating systems don‘t process large IP packets correctly

Large IP packets must be fragmented in order to traverse LANs

Ethernet MTU = 1500 bytes

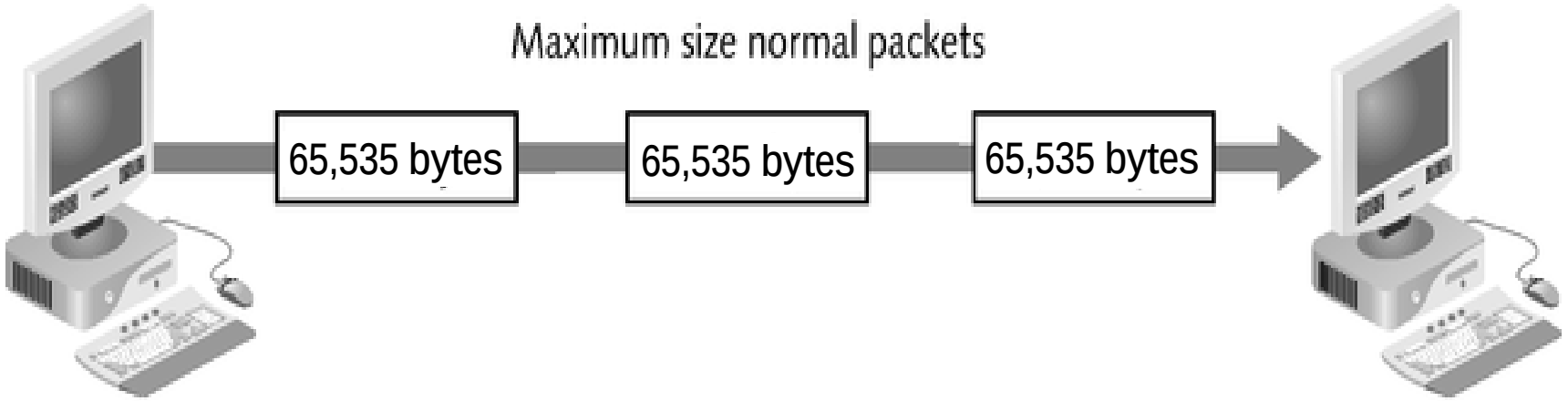
802.11 MTU = 2312 bytes

Some attacks use IP to fragments to crash remote systems

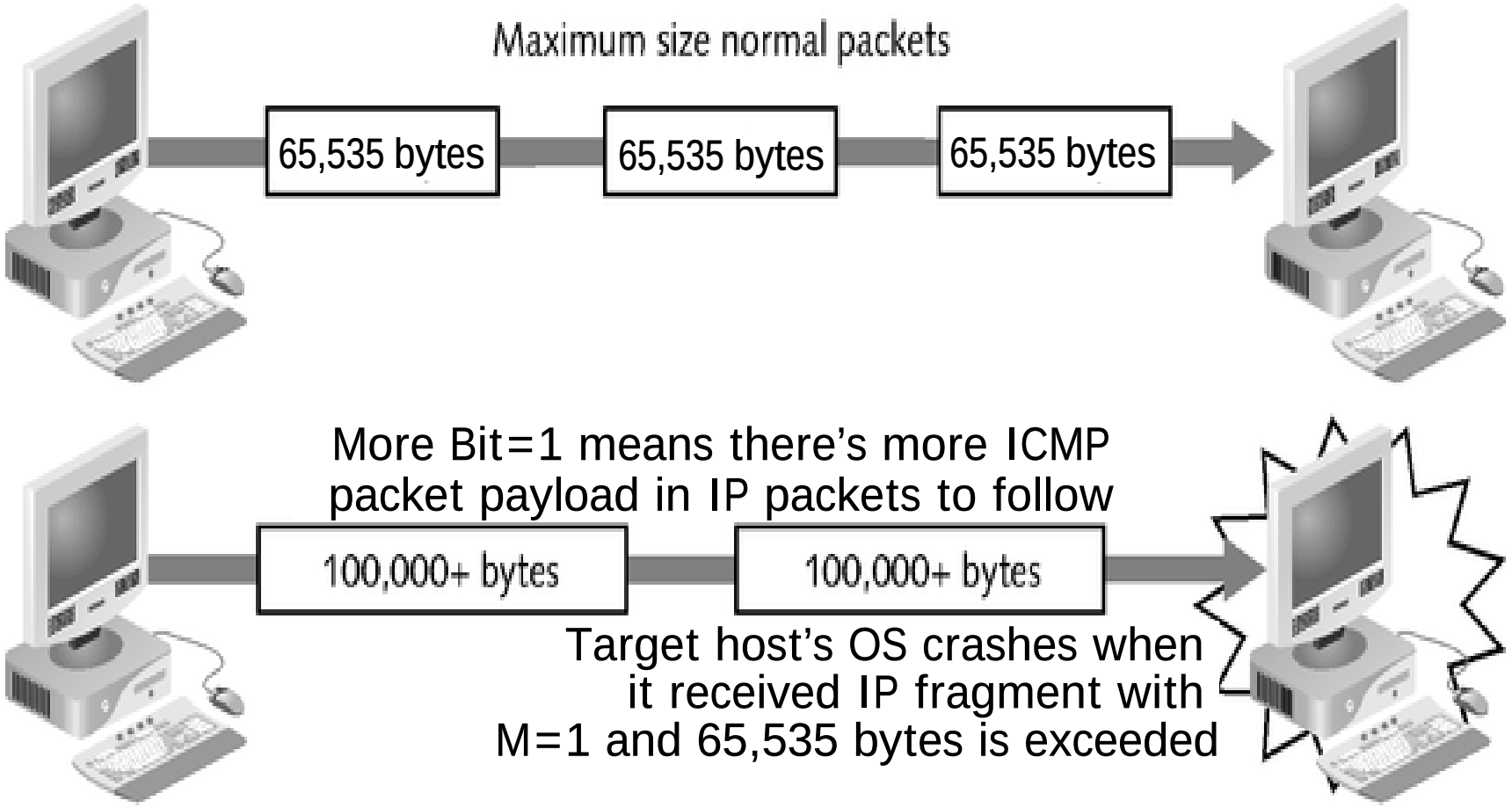
Very simple; all hacker needs is an IP address

PING of Death

IP Fragmentation Example



PING of Death
IP Fragmentation Example



DDoS

DDoS

DoS attacks are limited if a single attacker is used

Multiple systems allow much higher traffic volumes to form a Distributed Denial of Service (DDoS) Attack

Zombies

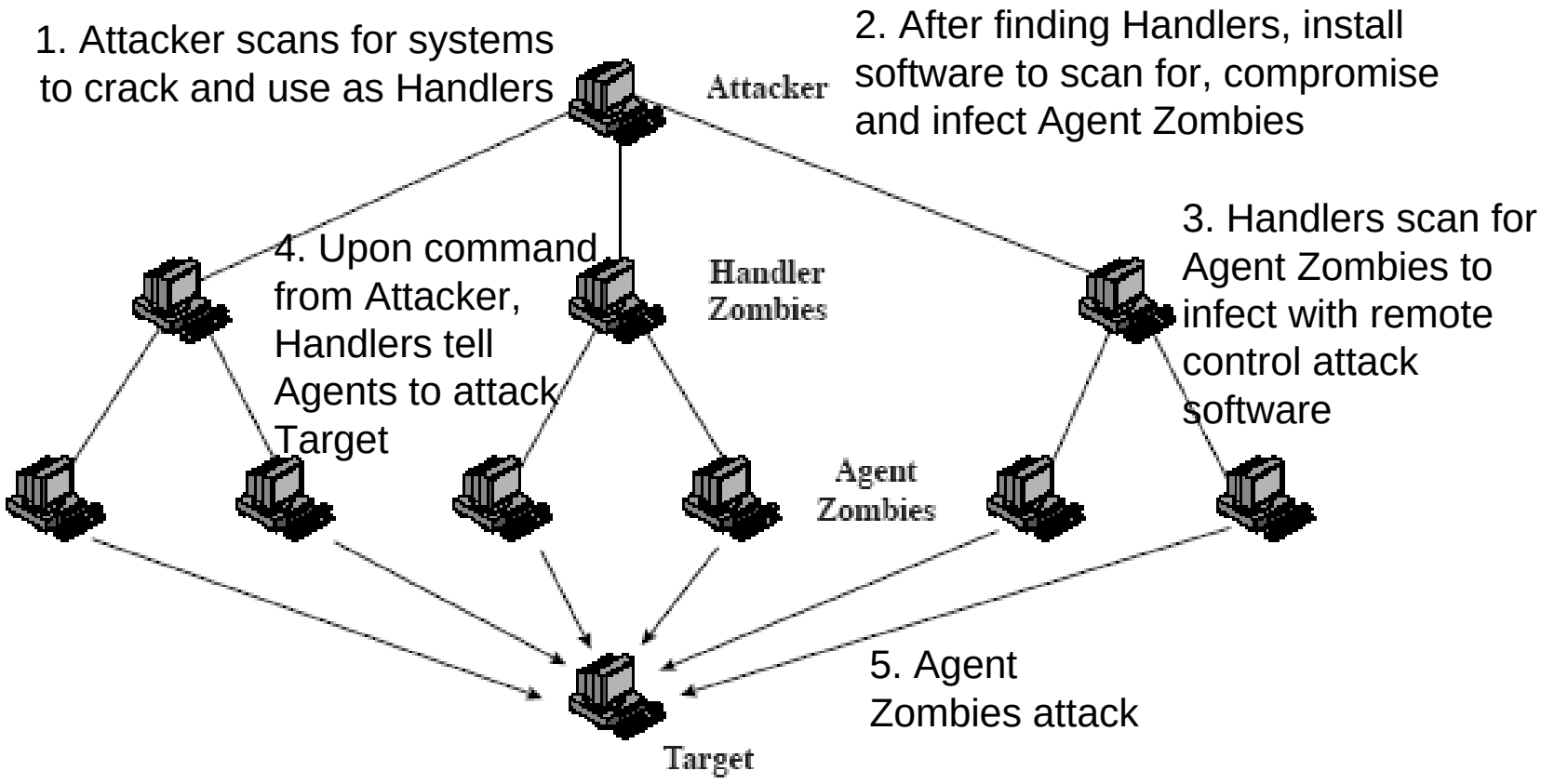
Compromised PC's / workstations

Backdoor programs installed

forming a botnet

e.g. Tribe Flood Network (TFN), TFN2K

DDoS Scheme


1. Attacker scans for systems to crack and use as Handlers
Attacker
2. After finding Handlers, install software to scan for, compromise and infect Agent Zombies
3. Handlers scan for Agent Zombies to infect with remote control attack software
4. Upon command from Attacker, Handlers tell Agents to attack Target
Handler Zombies
Agent Zombies
5. Agent Zombies attack
Target

DDoS Observations

Attacker is removed from attack by at least 2 layers of computers

Attack can be initiated automatically

Can use hundreds of hosts on the Internet to attack the victim

Flood victim's link to the Internet or deprives it of resources

Automated tools are available

They can be executed by script kiddies

DDoS Defense

Attack Prevention and Preemption

Provide alternate servers on different LANs and with different IP addresses

Rate limit the network

But this reduces access to very popular servers

Attack Detection and Filtering

Use an Intrusion Detection System (IDS) such as SNORT

Must have good rule sets

Upon detection of an attack, filter it

Tie the IDS to the firewall to create an Intrusion Prevention System (IPS)

DDoS Defense

Try to Identify the Attack Source

Will need to work with multiple Internet service providers (ISP) and others

Only useful for preventing future attacks

More Spoofing

Spoofing

Falsely identifies a packet's IP address or MAC address or URL or...

Primary types

- *ARP poisoning*
- *Web spoofing*
- *DNS spoofing*
- *IP spoofing*

ARP Poisoning

Each host operates an ARP table

Contains IPaddress/MACaddress tuples

10.1.2.3 *12:2c:46:e5:ef:7a*

ARP protocol is very simple (gullible)

Remember from ITMD 440 that there is a process called ***Gratuitous ARP***

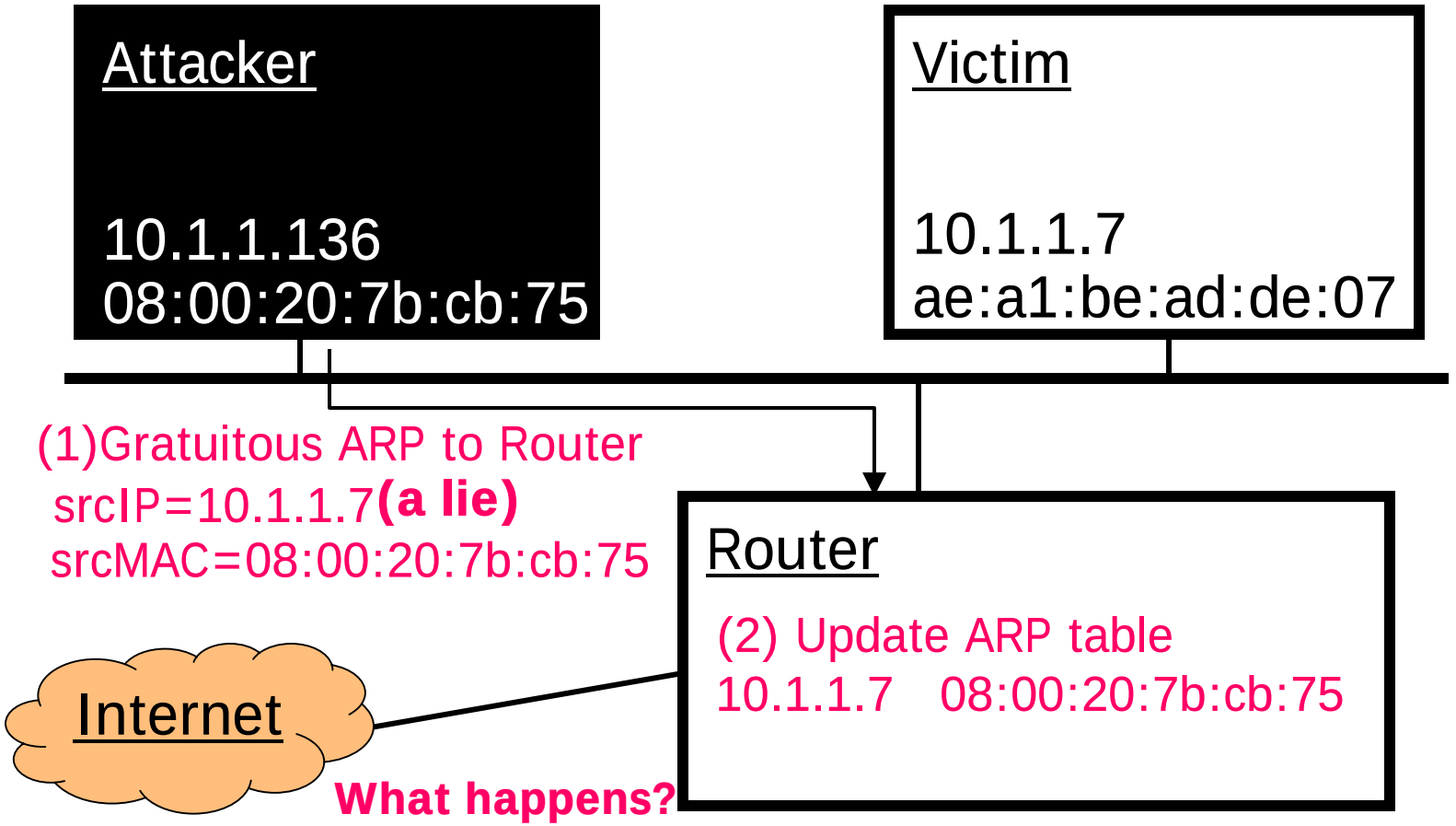
Purpose is to update the ARP tables of the computers on a LAN

Host send ARP request with own MAC address

All hosts on LAN update their ARP tables

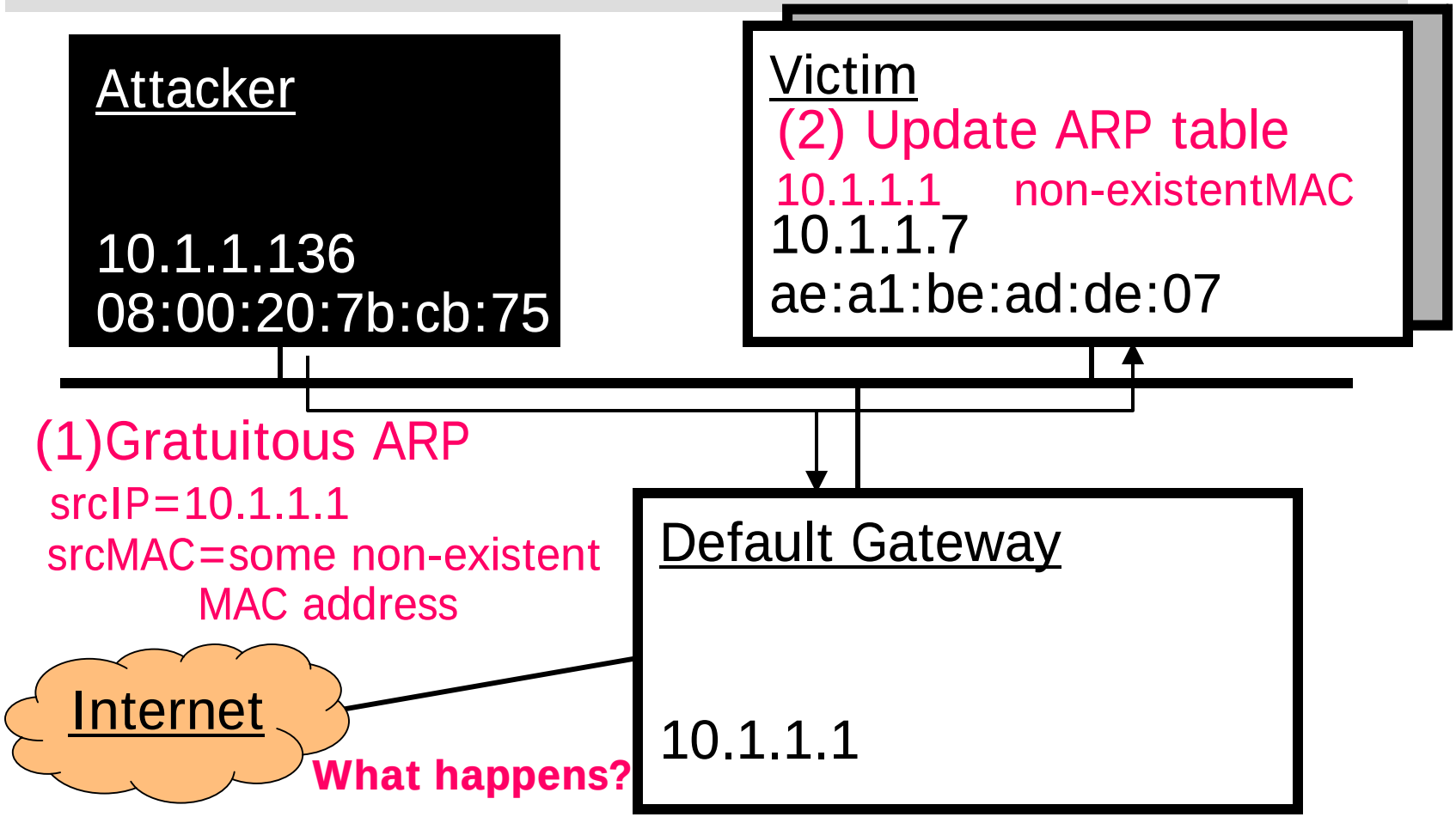
This can be used as an attack

ARP Poisoning Example 1*



Now router will forward all incoming IP packets with destIP=10.1.1.7 to Attacker

ARP Poisoning Example 2*



Web Spoofing

Must first lure victim into visiting attacker's web site

- *Email with link in it*
- *URL similar to well known URL*
- *False link on well know (but compromised) web site*

Result

- *Attacker's Web site gets in between Victim and legitimate Web site*
- *Attacker can monitor all interactions*

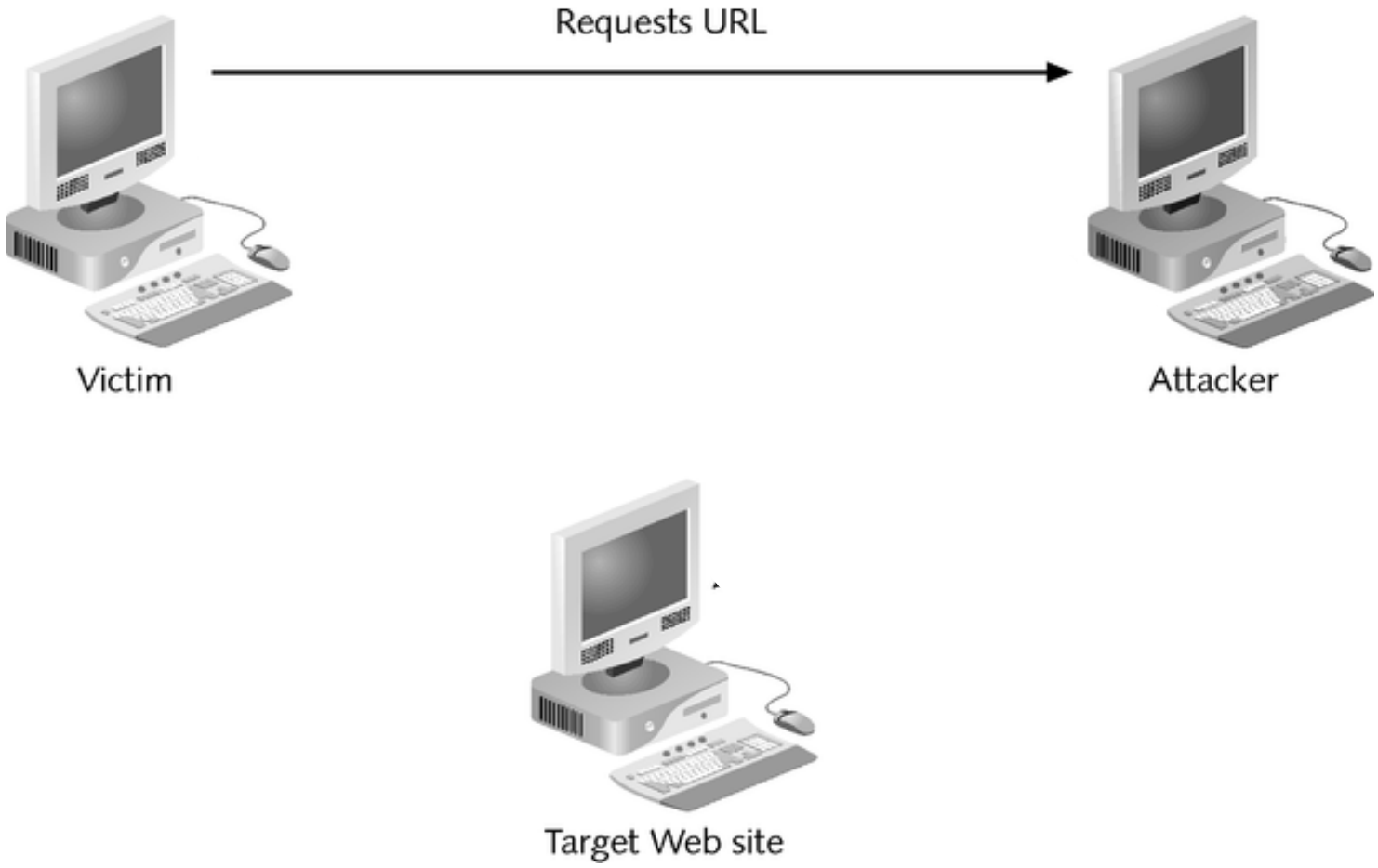
Web Spoofing

Victim

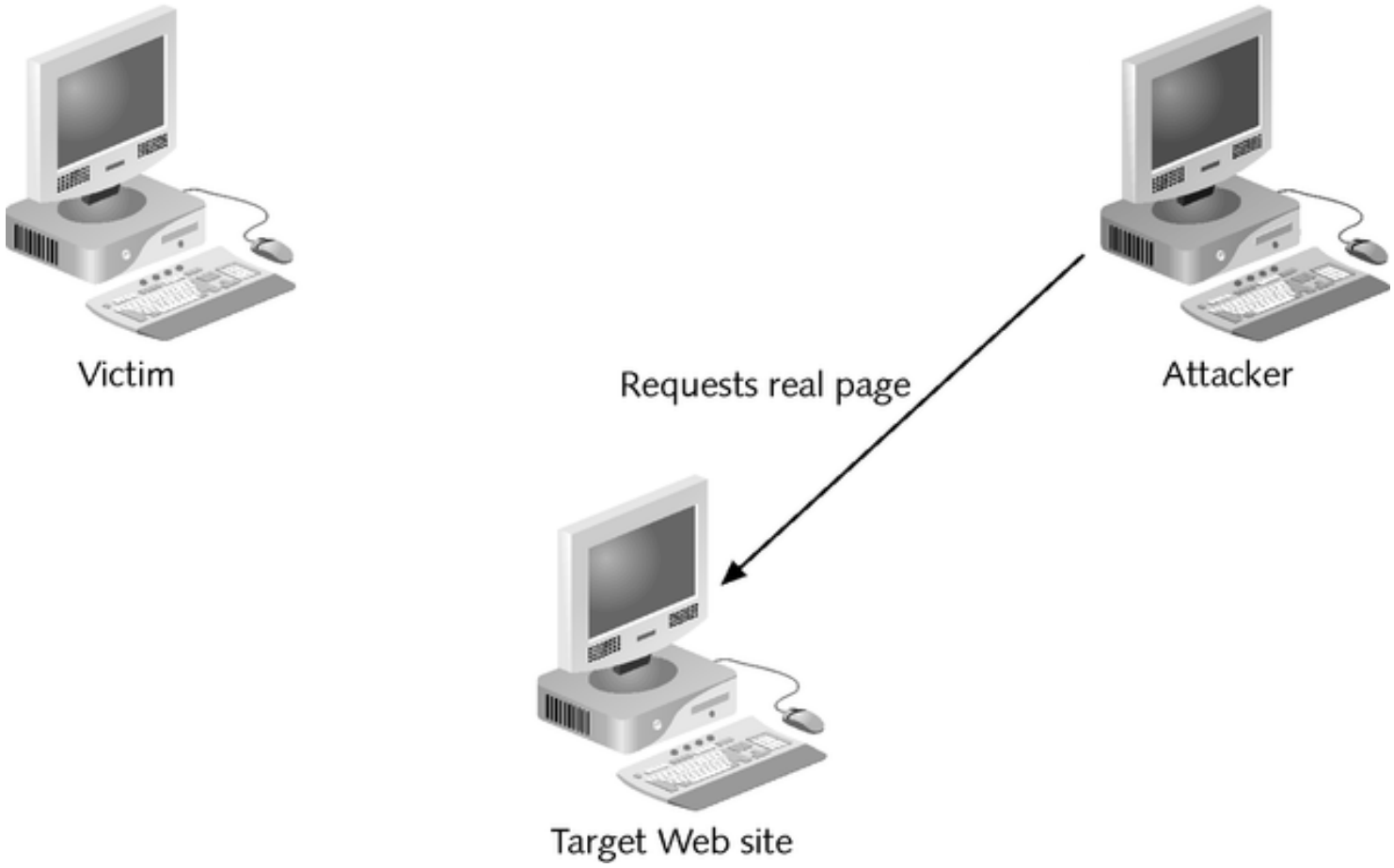
Attacker

Target Web site

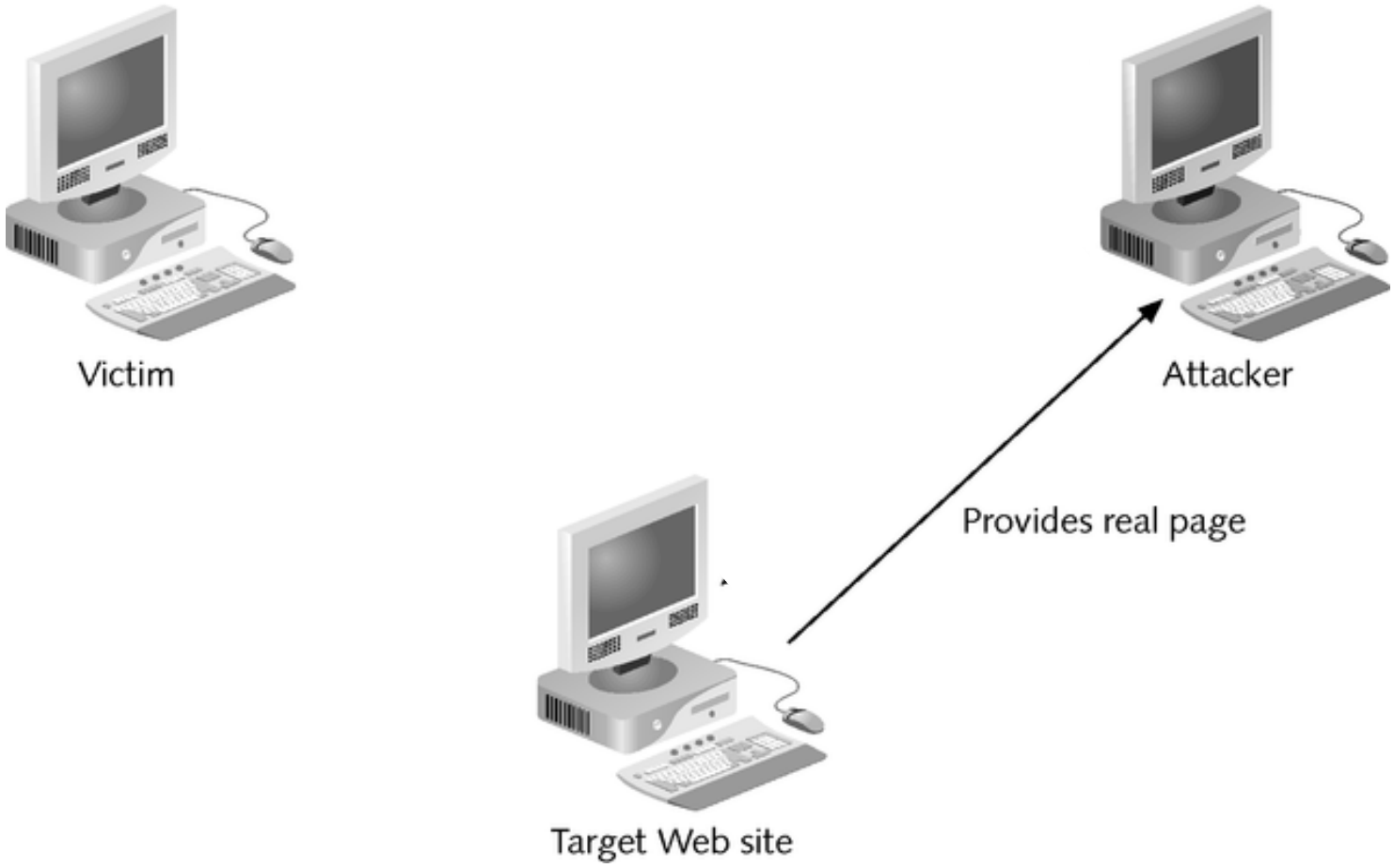
Web Spoofing



Web Spoofing



Web Spoofing



Web Spoofing

Victim thinks that she/he is communicating with legitimate *Target Web Site*

But attacker can see and record everything

DNS Spoofing

Using IP spoofing, Attacker poisons victim hosts DNS caches

Hosts start using the Attacker's DNS server instead of the legitimate DNS server

To intercept local DNS server queries to an authoritative DNS server

Attacker poisons the DNS cache on the local DNS server using IP spoofing

Queries then go to Attacker's DNS server

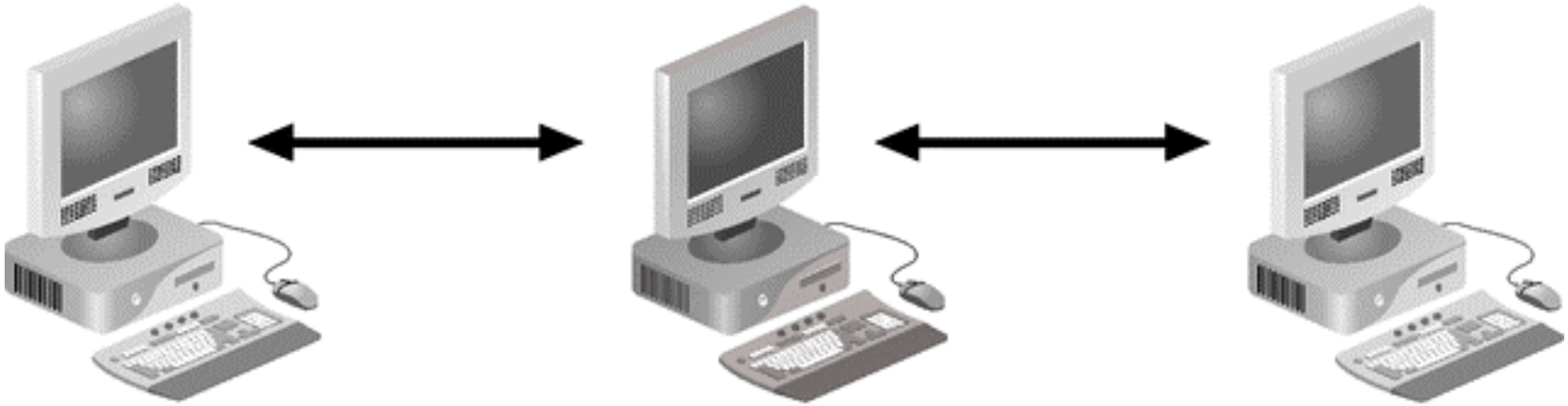
Man in the Middle

Class of attacks in which the attacker places himself between two communicating hosts and listens in on their session

To help protect against

Configure routers to ignore ICMP redirect packets

Man in the Middle



Host A

Thinks it's communicating with Host B

Attacker

Sits between the two hosts and is able to monitor and modify their transmissions

Host B

Thinks it's communicating with Host A

Man-in-the-Middle Applications

Web spoofing

TCP session hijacking

- *We'll do a homework assignment on this*

Information theft

Other attacks

- *Denial-of-service*
- *Corruption of transmitted data*
- *Traffic analysis to gain information about victim's network)*

Man-in-the-Middle Methods

ARP poisoning

ICMP redirects

DNS poisoning

Replay Attacks

Attempts to circumvent authentication mechanisms

General replay scheme

- *Record authentication messages from a legitimate user even if encrypted*
  - For instance, could use WireShark to record
- *Later reissue those messages in order to impersonate the user and gain access to systems*

Replay Attacks & The Web

There are two related classes of vulnerabilities associated with the Web

- *Vulnerabilities because of proxy Web servers*
- *Vulnerabilities because of the nature of http combined with the use of authentication tokens*

Replay Attacks & The Web
Proxy Web Servers

Proxy Web servers cache logins, passwords, and other authentication tokens

Sometimes these proxy servers are not too secure

If an attacker can gain access to such a server, (s)he can

Replay a Web session startup using the cached authentication tokens to access a user's account on the real Web server

Replay Attacks & The Web
Http & Authentication Tokens

HTTP protocol has provisions to include name, password and session ID in the URL that is sent to a real Web server

Suppose such a URL is copied as it is legitimately sent from a browser to a server

Later the attacker can easily use it in her Web browser to gain restricted access

This is theoretically possible even if encryption is used

Replay Attack Prevention

Many security systems have anti-replay attack features

Common approaches

- *Incorporate time into the session ID*
  - e.g., Kerberos
- *Use a different encryption "seed" each time a session is established*
- *Use predetermined sequence of non-reusable keys*
  - e.g., one-time pads

Birthday Paradox

Assertion

> *If 57 people get together in the same room, the probability is ≥ 99% that two people will have the same birthday*

Is this correct?

Why?

Called the ***Birthday Paradox***

Birthday Paradox Proof

Let $p_d(n)$ be the probability that the birthdays of n randomly selected people will all be the different

Then

$$p_d(n) = 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{n-1}{365}\right)$$

$$= \frac{365 \times 364 \cdots (365 - n + 1)}{365^n} = \frac{365!}{365^n (365 - n)!}$$

Very small

And

$$p_s(n) = 1 - p_d(n) \leq 0.99 \leq 99\% \text{ (for } n \geq 57)$$

$p_s(n)$ is the probability that two or more out of n people will have the same birthday

Birthday Attacks

Birthday attack:

An attack on a cryptographic system that exploits the mathematics underlying the Birthday Paradox

Works well on items such as short passwords and PINs

Birthday Attacks

Suppose a password is comprised of 4 alpha-numeric characters

The number of possible passwords is 36^4 = 1,679,616

Seems really big

Suppose I choose 10,000 (10^4) passwords randomly

Now I pick one of the 10,000

What's the probability that one of the 10,000 will be the same as the one that I pick?

$p_d(n) = [1679616!] / [1679616^{10000}\ (1679616-10000)!\ ]$ = ***almost zero***

$p_s(n) = (1 - p_d(n))$ = ***almost 1***

Now 1679616 / 10000 = ~168

What's the strategy for trying to find the password quickly?

Birthday Attacks
4-digit PIN

A 4-digit PIN has 10^4 possible values

Suppose I choose 10^2 values randomly

Then I pick one of the 10^2 values

What is the probability that the value that I pick will be the same as one of the 10^2 values?

$$p_d(n) = [(10^4)!][/ (10^4)^2 (10^4 - 10^2)!] = \text{very small}$$

$$p_s(n) = (1 - p_d(n)) = \text{almost } 1$$

Now

$$10^4 / 10^2 = 100$$

So create 100 groups of random numbers

Pick one out of each group

Try the 100 PINs.

This is a much faster way to crack the PIN than brute force

Buffer Overflows

Buffer Overflow vulnerability are usually the result of:

- Address arithmetic (e.g., C, C++)
- Stack architectures
- Almost infinite amount of ***P*⁴**

  Poor Past and Present Programming

Buffer Overflows still account for over 50% of today's exploits

Buffer Overflows

Malicious code is written onto the stack beyond the return address and function calls

Return address is modified to point to the beginning of the malicious code

When a call returns, the program pointer is loaded with the malicious code

The malicious code is then executed

Simple Buffer Overflow Example

```
void f()
{
    int a[10];
    a[20] = 3;
}
```

This code will compile and may execute

Might result in a segmentation fault error

But if the attacker puts the right data into the array a[20] (instead of the number "3") then she can execute malicious code

And On and On...

There are other attack mechanisms

But many are combinations of some of those that I've described

Now one last attack: TCP Hijacking

TCP Hijacking

Attacker uses TCP hijacking

- *To make the victim believe he or she is connected to a trusted host, when in fact the victim is communicating with the attacker*
- *To make the trusted host believe that it is connected to a legitimate client when it is really communicating with the attacker*

Can also be used for a Man-in-the-Middle attack

TCP Hijacking

Attacker sniffs traffic looking for a session (e.g., ftp, telnet, http) in progress between a Client and a Server

Attacker takes over session from the Server using ARP poisoning

Poisons Router ARP table so that Attacker impersonates Router

Attacker takes over session from the Client using ARP poisoning

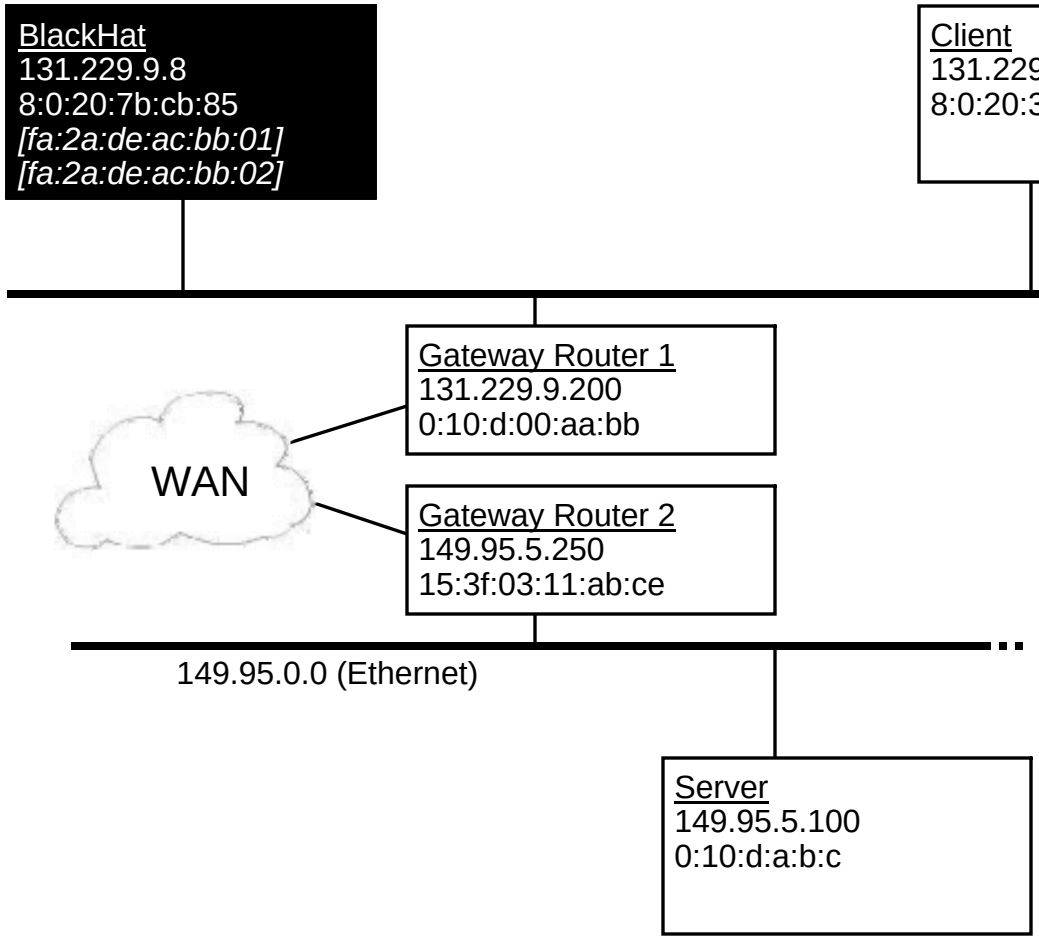
Poisons Victim ARP table so that Attacker impersonates Victim

Attacker now has access to Victim's session

Special Problem 05b

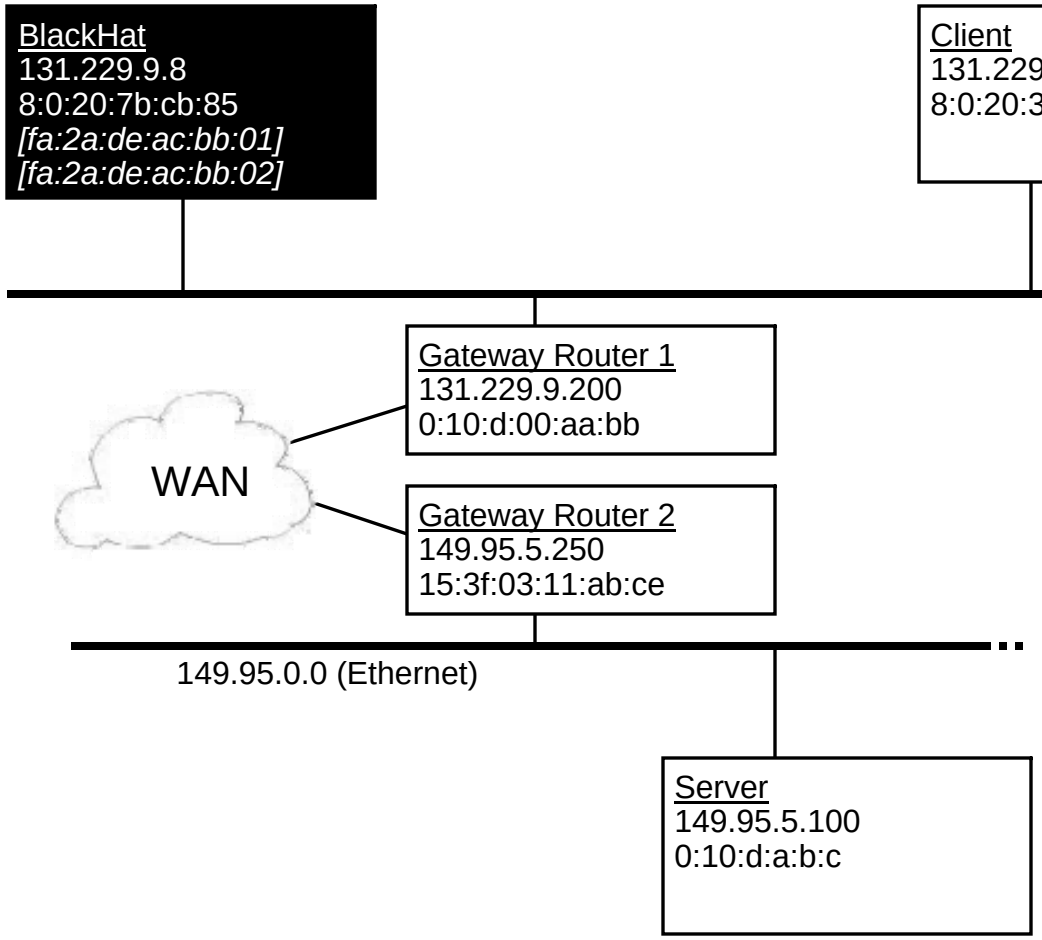
TCP Hijack

Special Problem 05b
Setup



1. The MAC and IP addresses for each host are shown.
2. *UnhappyUser* decides to get even with *Client*
   Becomes a BlackHat
3. *BlackHat* sniffs her network and as a result invents two MAC addresses that do not belong to any hosts on the network 131.229.0.0.
   fa:2a:de:ac:bb:01
   fa:2a:de:ac:bb:02

Special Problem 05b
Setup



4. The Client now establishes a TCP connection with the Server
   Encrypted name and password
5. BlackHat now hijacks the connection
   Client no longer is connected
   BlackHat now is connected as if she were the client

Show how BlackHat does this using the instructions on the next slide.

Special Problem 05b
Instructions on How to Work Problem

In the three diagrams following this slide, do the following:

For the Client, Server, *and both* Gateway Routers *show in the Partial ARP Table boxes the pertinent ARP Table entries in the boxes provided for each of the following states as shown in the following 3 slides.*

State1: After the Client has established a TCP connection with the Server but before the BlackHat has started her attack

State2: After the BlackHat has taken over the session from the Server but before State 3.

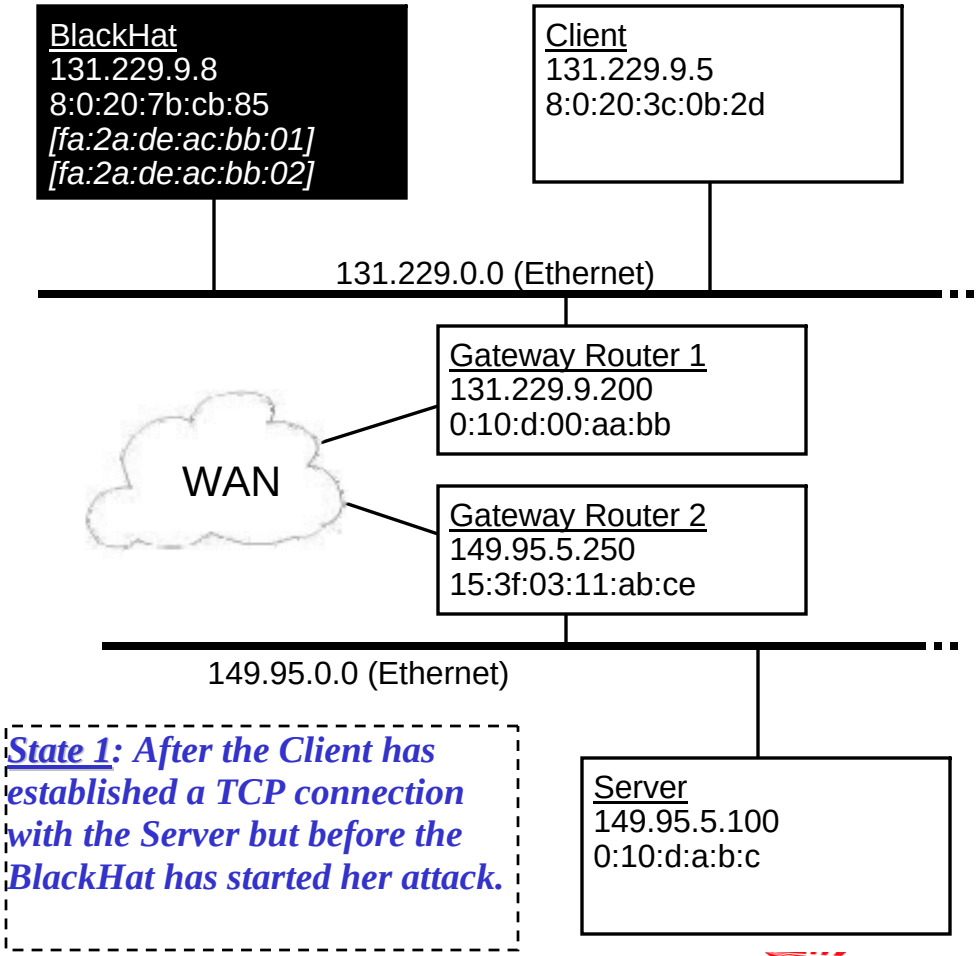
State3: After the BlackHat has taken over the session from both the Server and the Client.

The idea is to show how the BlackHat can hijack the TCP session by causing the ARP tables to change.

I refer to “pertinent ARP table entries” because we are only interested in the ARP entries that effect the TCP hijack. There may be other unrelated entries.

Special Problem 05b

Show Partial ARP Tables for State 1



***State 1**: After the Client has established a TCP connection with the Server but before the BlackHat has started her attack.*

Client Partial APR Table

IPaddress MACaddress

Gateway Router 1 Partial APR Table

IPaddress MACaddress

Gateway Router 2 Partial APR Table

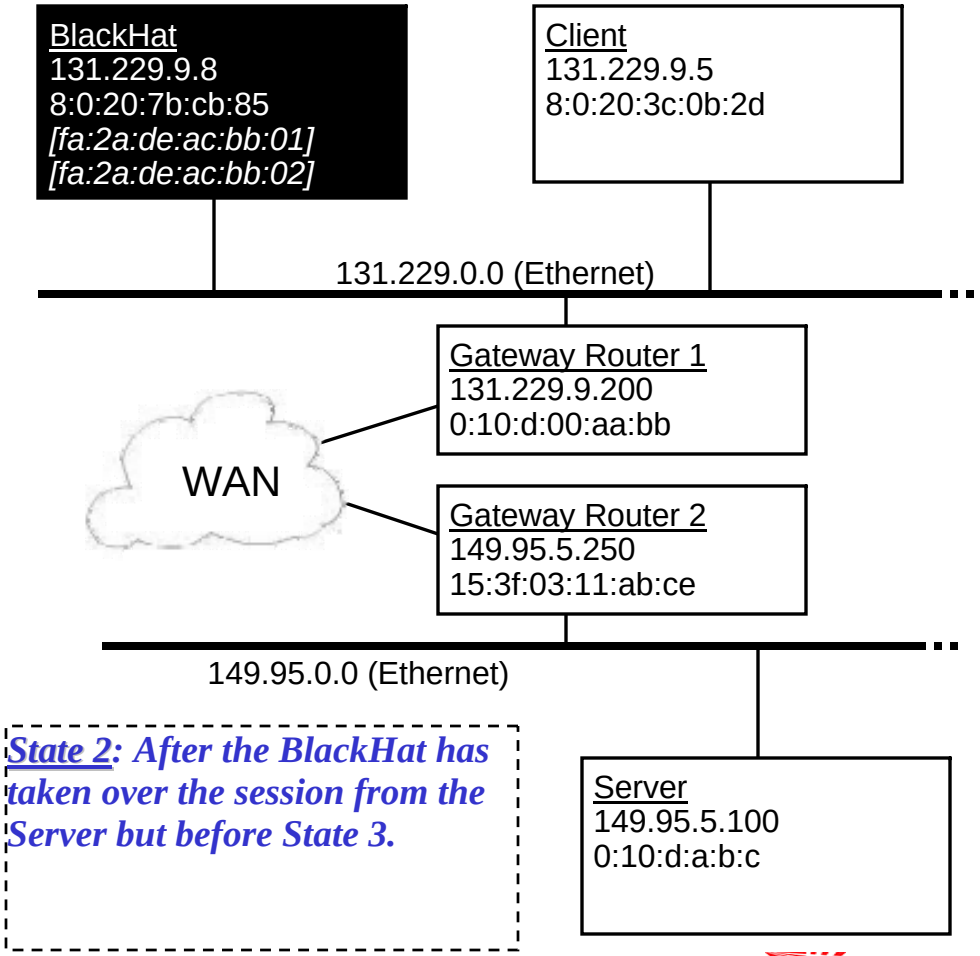
IPaddress MACaddress

Server Partial APR Table

IPaddress MACaddress

Special Problem 05b

Show Partial ARP Tables for State 2



***State 2**: After the BlackHat has taken over the session from the Server but before State 3.*

Client Partial APR Table

IPaddress MACaddress

Gateway Router 1 Partial APR Table

IPaddress MACaddress

Gateway Router 2 Partial APR Table

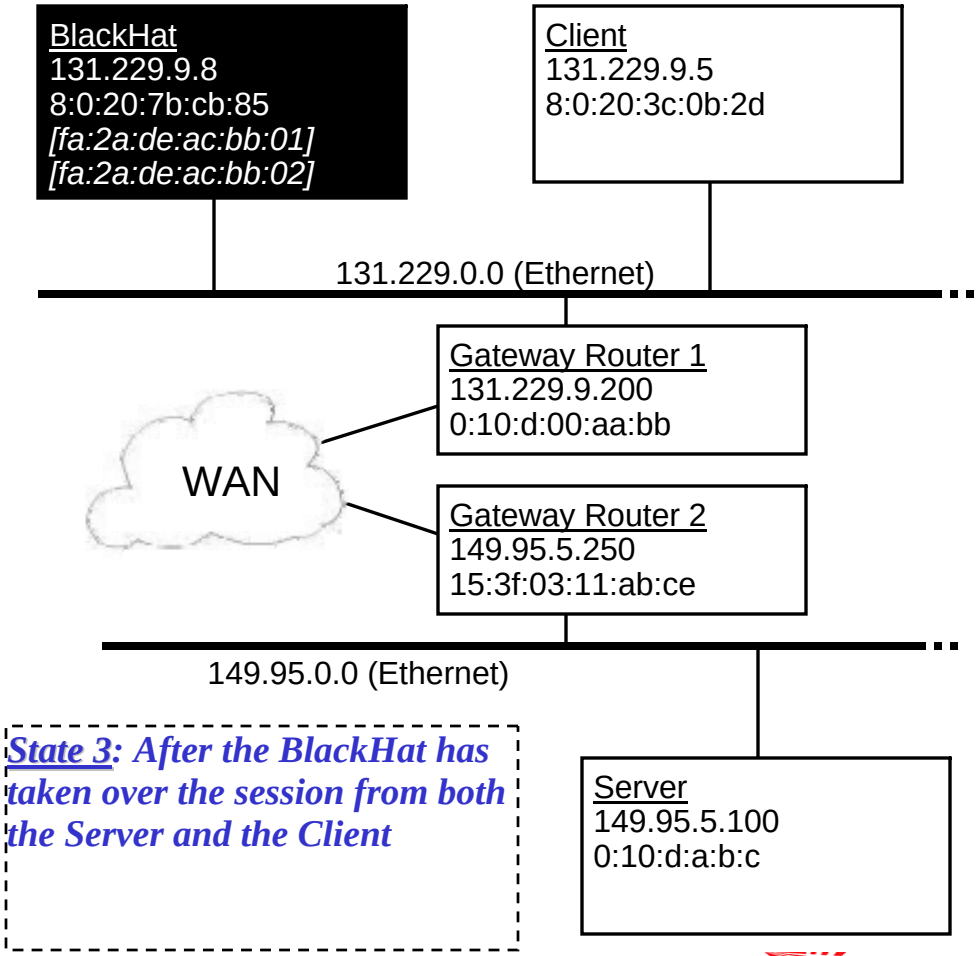
IPaddress MACaddress

Server Partial APR Table

IPaddress MACaddress

Special Problem 05b

Show Partial ARP Tables for State 3



***State 3**: After the BlackHat has taken over the session from both the Server and the Client*

Client Partial APR Table

IPaddress MACaddress

Gateway Router 1 Partial APR Table

IPaddress MACaddress

Gateway Router 2 Partial APR Table

IPaddress MACaddress

Server Partial APR Table

IPaddress MACaddress

Assignment 05b

1) Read S&B Chap 7
2) Do Review Questions 7.1-7.16 on p 245-246 of S&B.
3) Work Problems 7.1, 7.3 and 7.5 on pp 246-247 of S&B.
4) Work *Special Problem 05b*. Submit the results as the three slides with the four Partial ARP Tables in each slide completed with the pertinent entries. I will provide the three slides in PowerPoint format on Bb.
5) Try to find a user-level tool or a mechanism for Win7 or Win8.1 that is controlled by the user & that does ARP poisoning/spoofing. Verify that the tool works if you can. Document by providing:

   (a) a brief description of the tool and how to use it,

   (b) indicate how it can be downloaded & from what URL and

   (c) show your verification or explain why you can't verify.
6) Repeat item 5) for Linux.