Network Security

- Firewall (Network Intrusion Detection)
- IPsec
- Web Security
- IP Traceback

Firewall

Professor Yeali S. Sun

Information Management Department

National Taiwan University

Outline

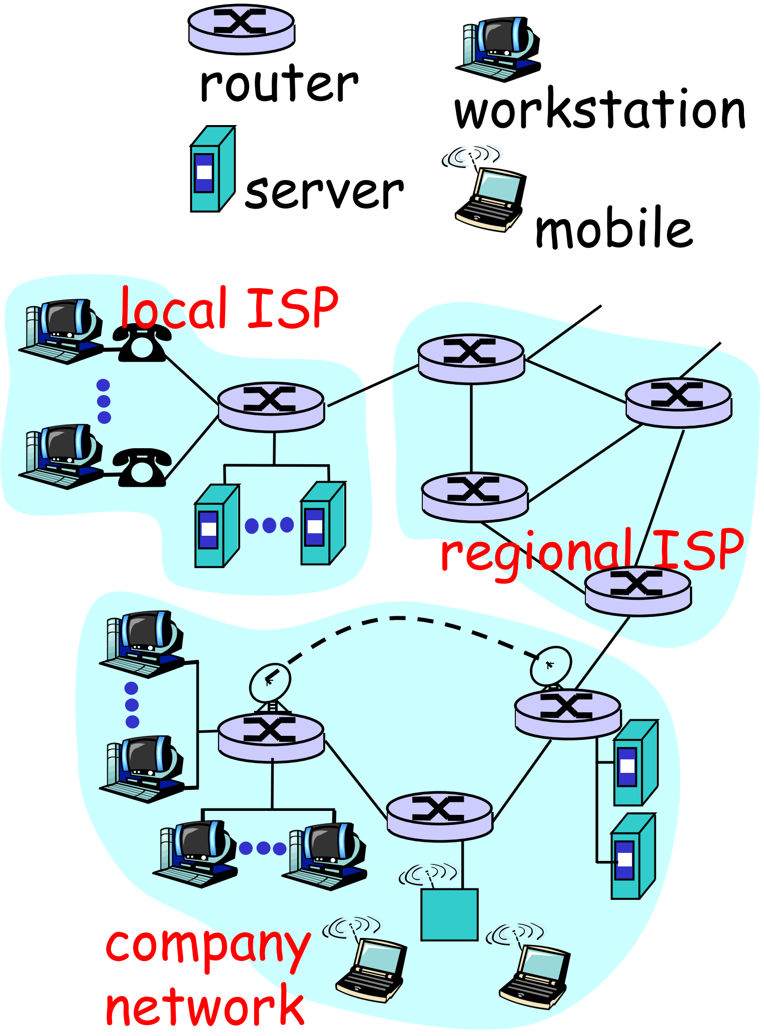
- Network Security Management
  - What are you trying to protect? (norm)
  - What are you trying to protect against? (threats)
- Firewalls
- Techniques for Secure Communications

Security Management

- “What ***resources*** are we trying to protect?”
  - data, files, storage device, computers, etc.
  - AAA (authentication, authorization, and accounting), identity management, access control - privileges for resource including Internet access
  - passwords, encryption
- “*Against* ***who***, must the computer systems be defended?
  - Attacker/hacker, hacking software, insider, outsider, etc.
- Network Security, Computer Security (including storage) and Information Security

Network Security

- To protect network components (hardware and software)
  - *Internet:* "network of networks"
  - *communication links*
    - fiber, copper, radio, satellite
  - *routers:* forward packets (chunks of data)
  - *protocols* control sending, receiving of msgs
    - e.g., TCP, IP, HTTP, FTP, PPP
- To protect network services
- To protect the content delivery over networks



To err is Human

- The techniques attacks used were technical in nature (and human natures and behaviors nowadays).
- They exploited weakness in the implementations of many network protocols (e.g., TCP) and systems (and humans).

Picking a Security Policy

- A *security policy* is a set of decisions that collectively determines an organization’s posture toward security
  - to decide what is and is not **permitted**
  - driven by the business needs of the organization
    - guard against employees to exporting valuable data or importing software (licensing, *insider intrusion*)
    - specific protocols/services can not be used because of administratively being unsecured

Stance

- The *stance* is the attitude of the corporate network security designers
  - cost of the failure of the firewall
    - a fail-safe design
      - if we have overlooked a security hole or installed a broken program, we believe our firewalls are still safe
  - designers' estimate of that likelihood
  - designers' abilities

Security Risk Analysis

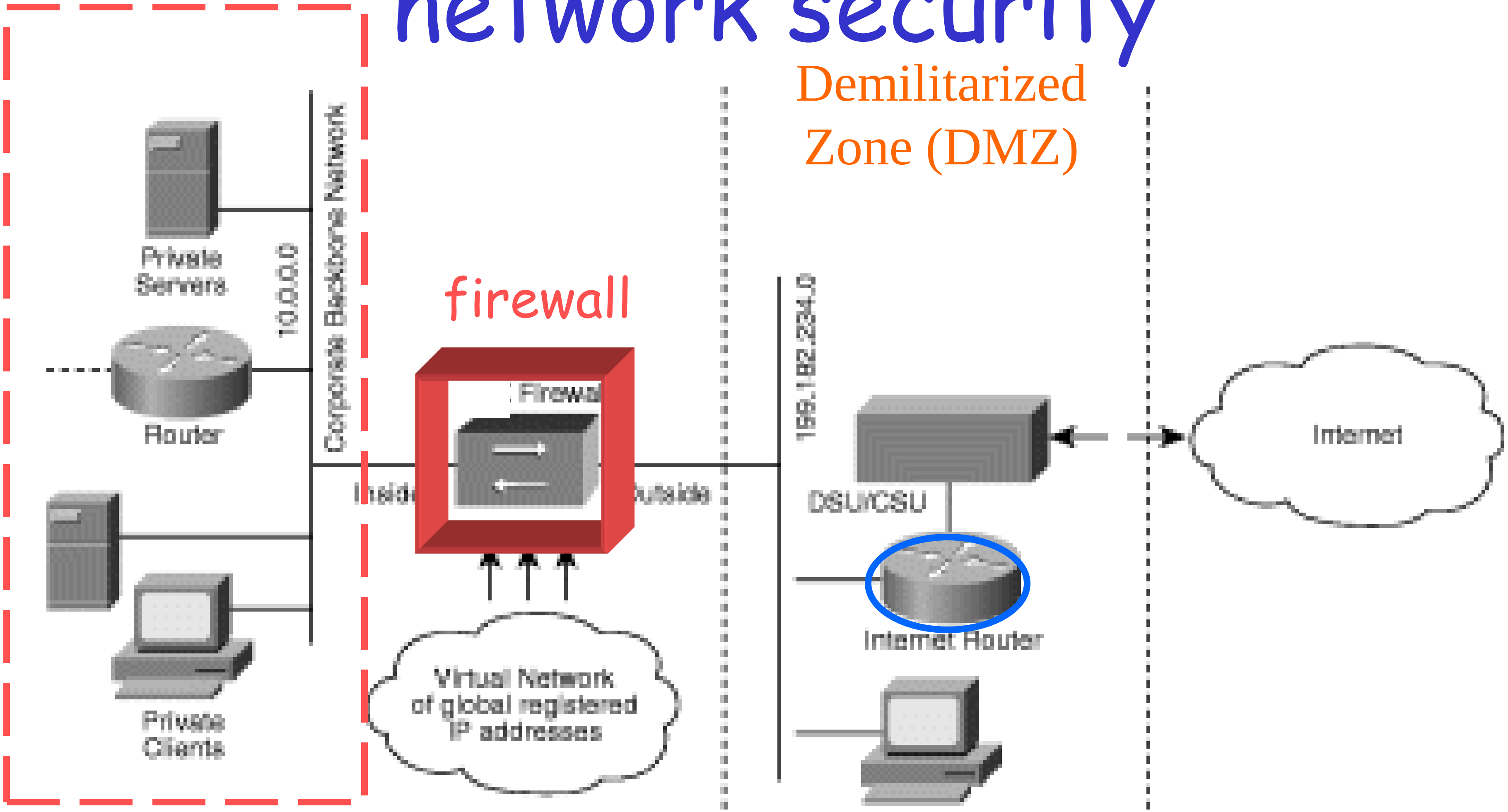
- "*Why would a company risk losing its secrets for the benefits of network connection?*"

Firewall

Typical Corporate Security Concerns

- How can a company prevent users who access their public Web site from accessing other highly sensitive private network resources?
- What about *internal* employees who wish to transmit highly sensitive data from the corporate intranet to the outside world?

Two-tiered approach to network security



Demilitarized Zone (DMZ)

firewall

Protected Internal network

External DNS, WWW, Mail Relay, public FTP

Firewall: Basic Requirements

- Commonly used to protect a local system or network of systems from network-based security threats.

- At the same time it should allow access from the inside to the outside world via wide area networks and the Internet.

Firewall: Design Principles

- *All* traffic from inside to outside, and vice versa, MUST pass through the firewall.
  - One point of control
  - Often at the gateway router

- Only authorized traffic as defined by the local security policy, will be allowed to pass.
  - Different features for different purposes.

- The firewall itself MUST be immune to penetration.

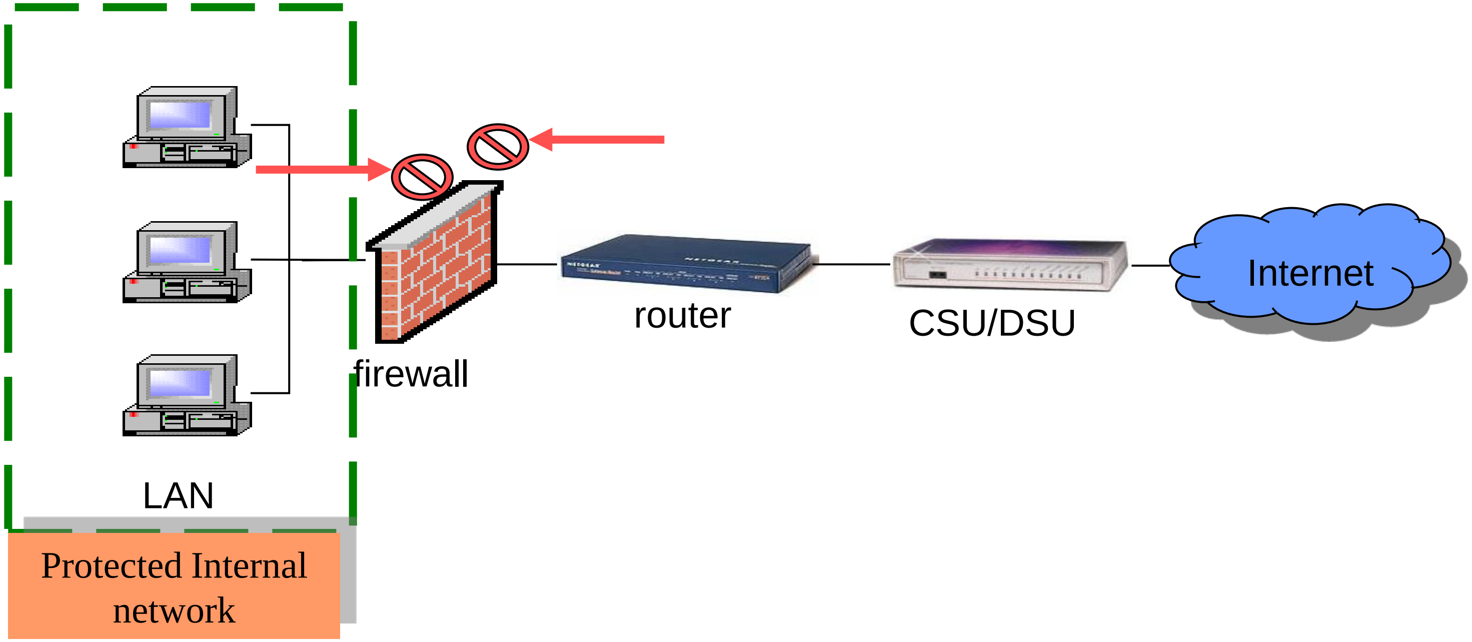
Firewall: Other Popular Services

- Security-related events monitoring, auditing and logging, event reporting
  - watch over traffic (or content) to ensure proper conduct is maintained
- Network address translator (NAT)
  - Maps private addresses to Internet addresses

Firewall - Service Characteristics

- Service Control
- Direction Control
- User Control
- Behavior Control

Firewall Service #1: Service Control



- Determine the type of services: Denial vs. Permitted.
- Inbound (ingress) and/or outbound (egress)
- Packet/Content filtering based on some criteria
  - e.g., IP addresses, Layer 4 port numbers, protocol numbers, application contents, etc.
- Deep Packet Inspection (DPI)

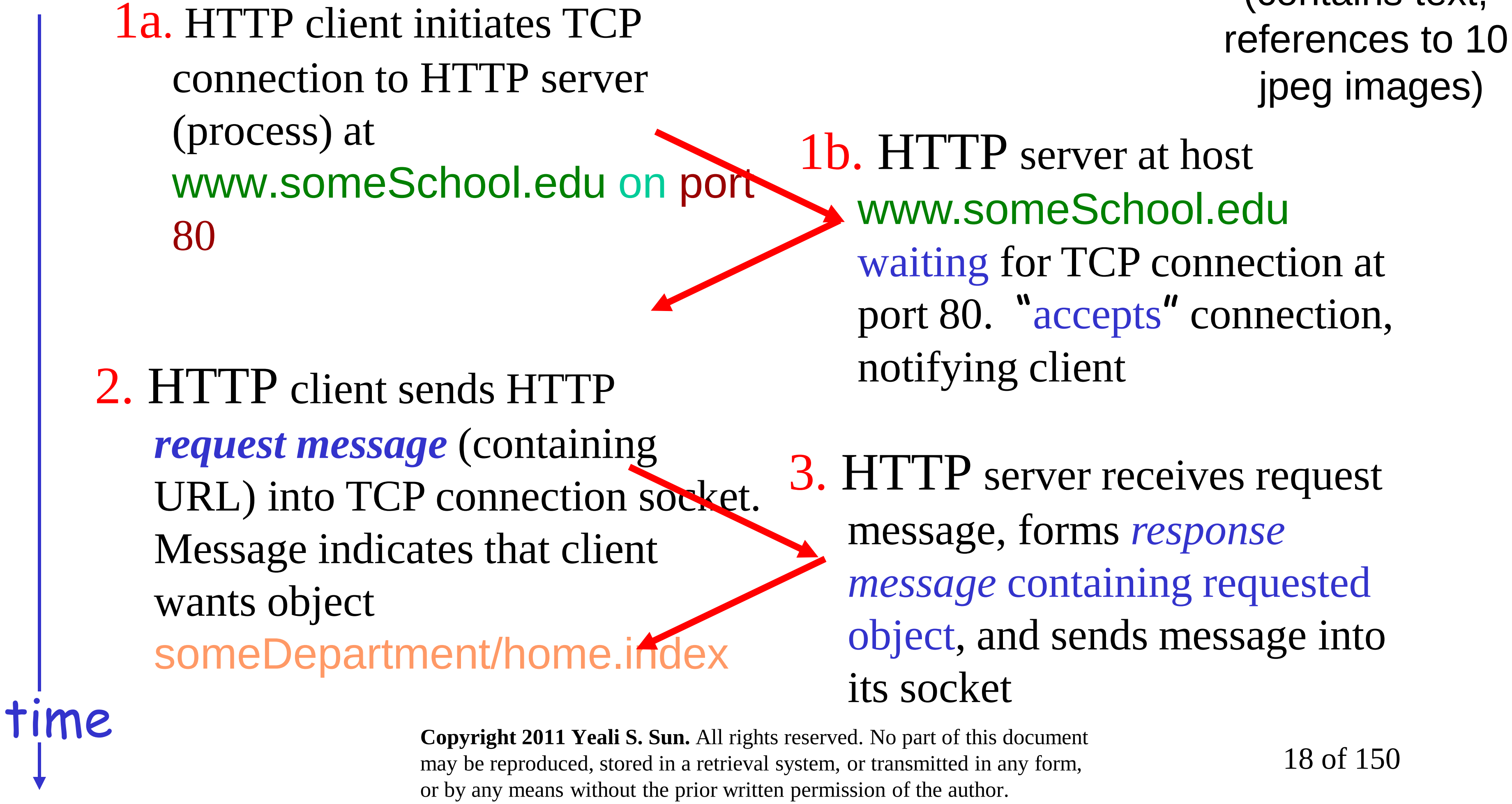
Content Filtering

Example #1: URI-based filtering (1/3)

Suppose user enters URL

www.someSchool.edu/someDepartment/home.index

(contains text, references to 10 jpeg images)



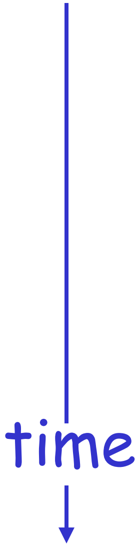
1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. HTTP client sends HTTP ***request message*** (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

Example #1: URI-based filtering (2/3)



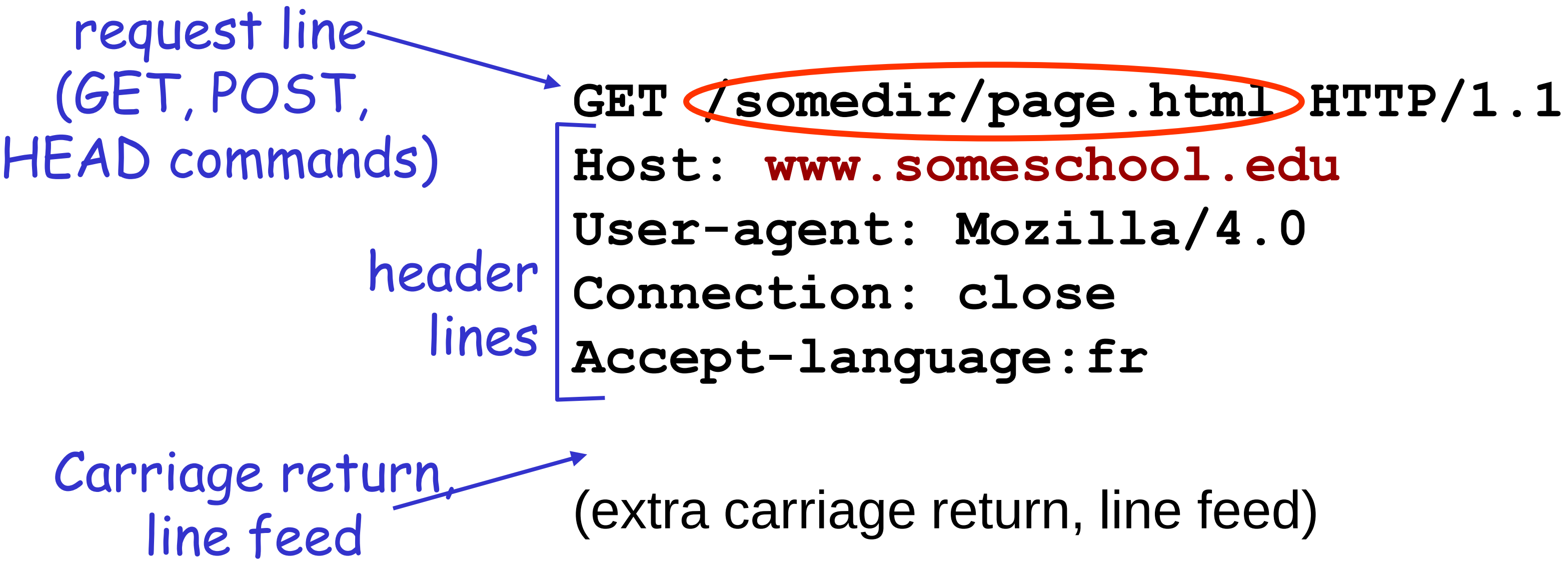
4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for **each** of 10 jpeg objects

Example #1: URI-based filtering (3/3)

- Two types of HTTP messages: *request, response*
- HTTP request message:
  - ASCII (human-readable format)



request line (GET, POST, HEAD commands)

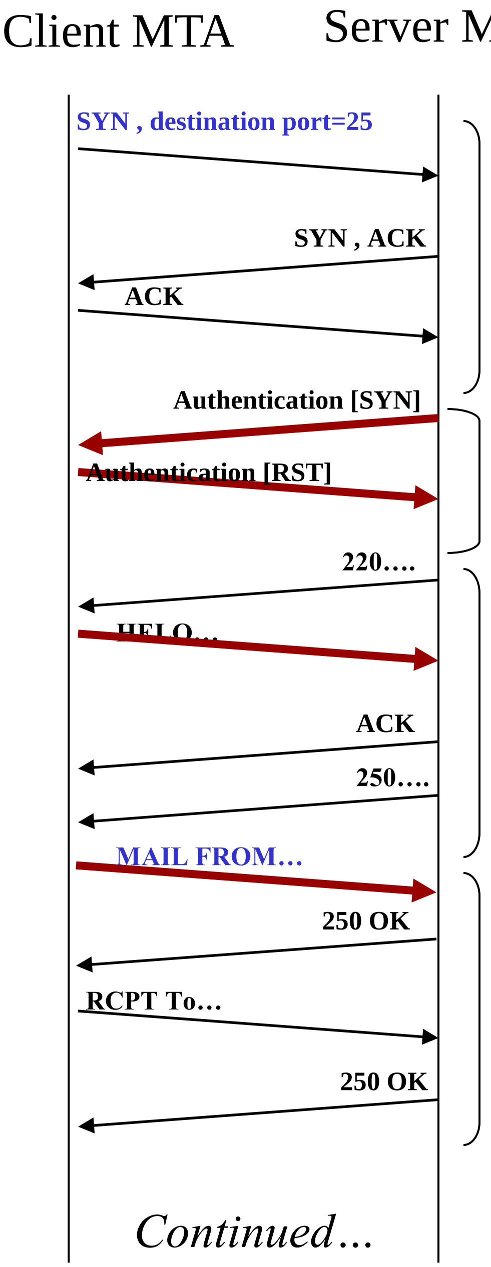
```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language:fr
```

header lines

Carriage return, line feed indicates the end of message

(extra carriage return, line feed)

Example #2: backlist based filtering (1/2)



TCP three-way handshaking (IP of Client MTA)

Mail server black list: IP address

Different ports (don't have the function of authentication now)

220: service ready

Mail server black list: domain name

HELO <domain> // Client MTA use it to identify itself

250 <Server MTA domain>

MAIL FROM: reversing path

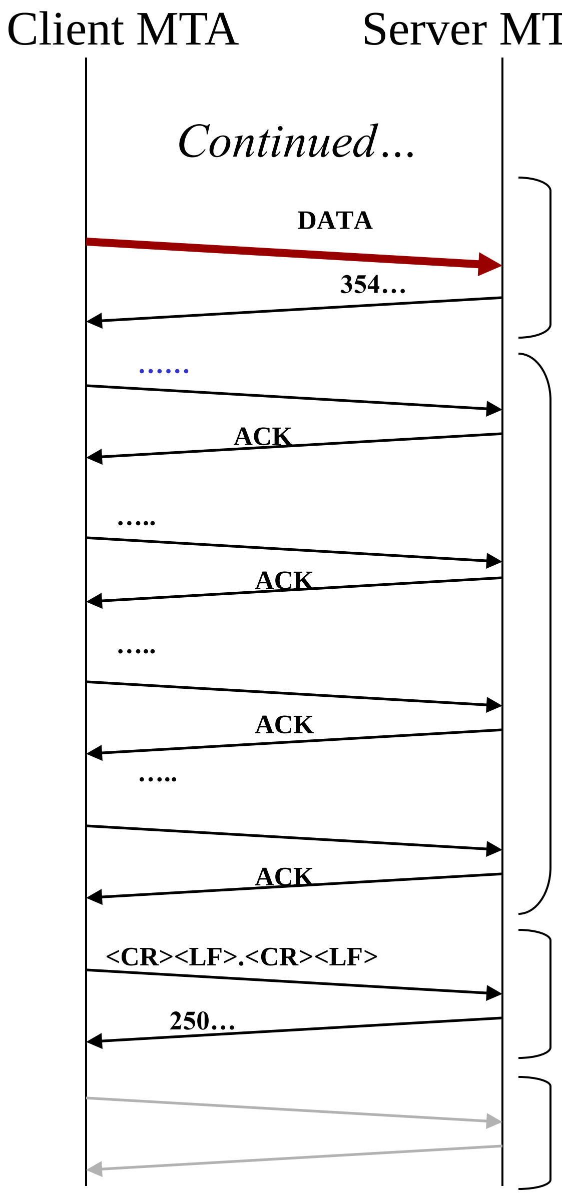
<- domain of relaying MTA, sender's mail account

RECP TO: forwarding path

<- receiver's mail account

Continued…

Example #2: backlist based filtering(2/2)



The receiver treats the lines following the “DATA” packet as mail data from the sender.

<- 354: Start mail input; end with .

Client MTA sends the content of the mail object.

Server MTA replies with “ACK” packet
(IP of relaying MTAs)
(IP of original host)

Client MTA sends the end-of-mail command (.)

250: Requested mail action okay, completed

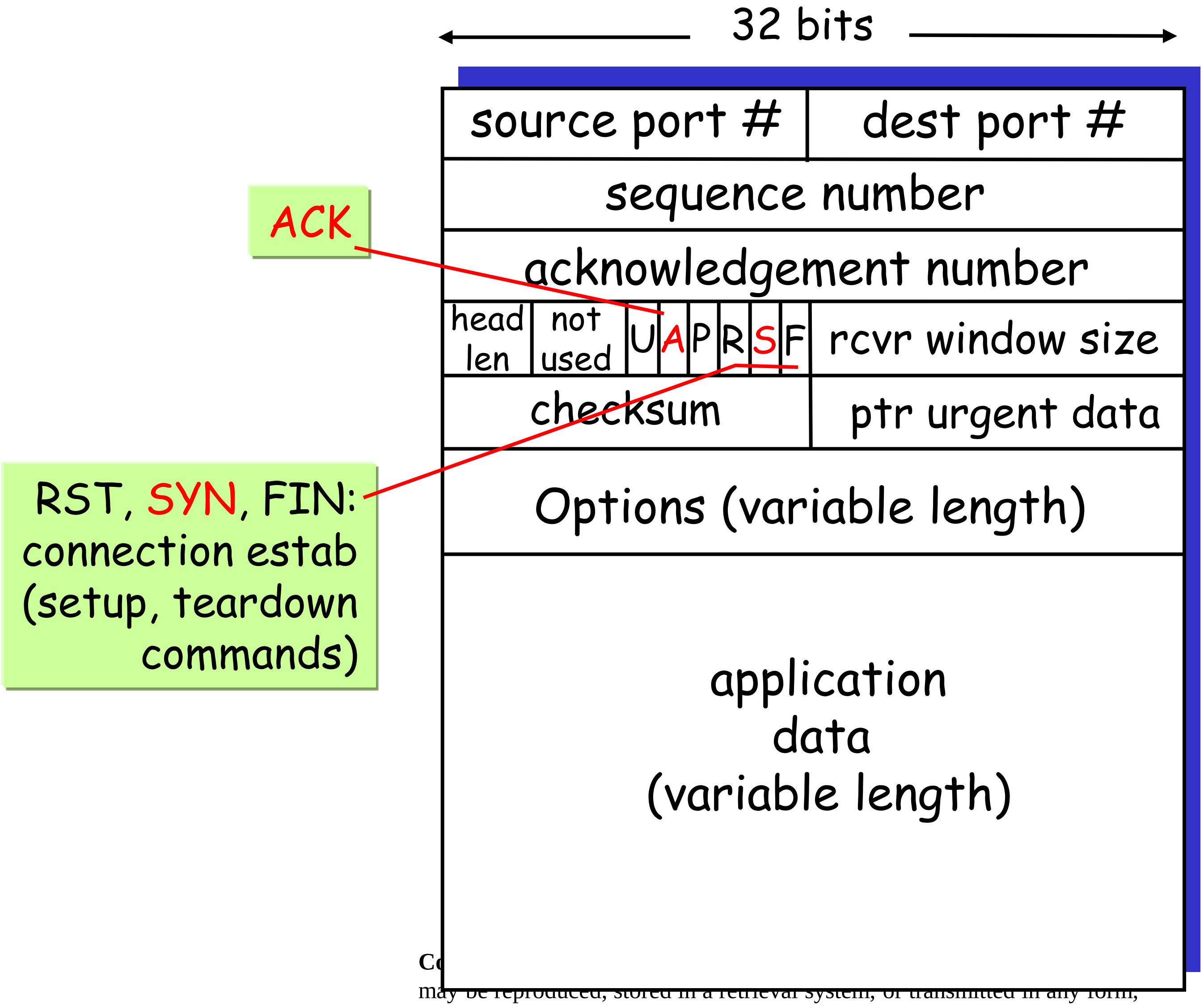
2 cases:

- Client MTA has more mails to send, repeat “MAIL FROM”
- Client MTA has NO mail to send, sends “QUIT” packet
- Server MTA replies with 221 and closes the connection

Firewall Service #2: Direction Control

- Determine the *direction* in which particular service requests may be *initiated* and allowed to *flow through* the firewall.
- Example: ftp via TCP connection blocking
  - TCP flags (8-bit)
  - TCP connection establishment – Three-way Handshake.
    - Syn, Syn/Ack and Ack

Example: Security control of TCP connections (1/3)



32 bits

source port #

dest port #

sequence number

acknowledgement number

head len

not used

U A P R S F

rcvr window size

checksum

ptr urgent data

Options (variable length)

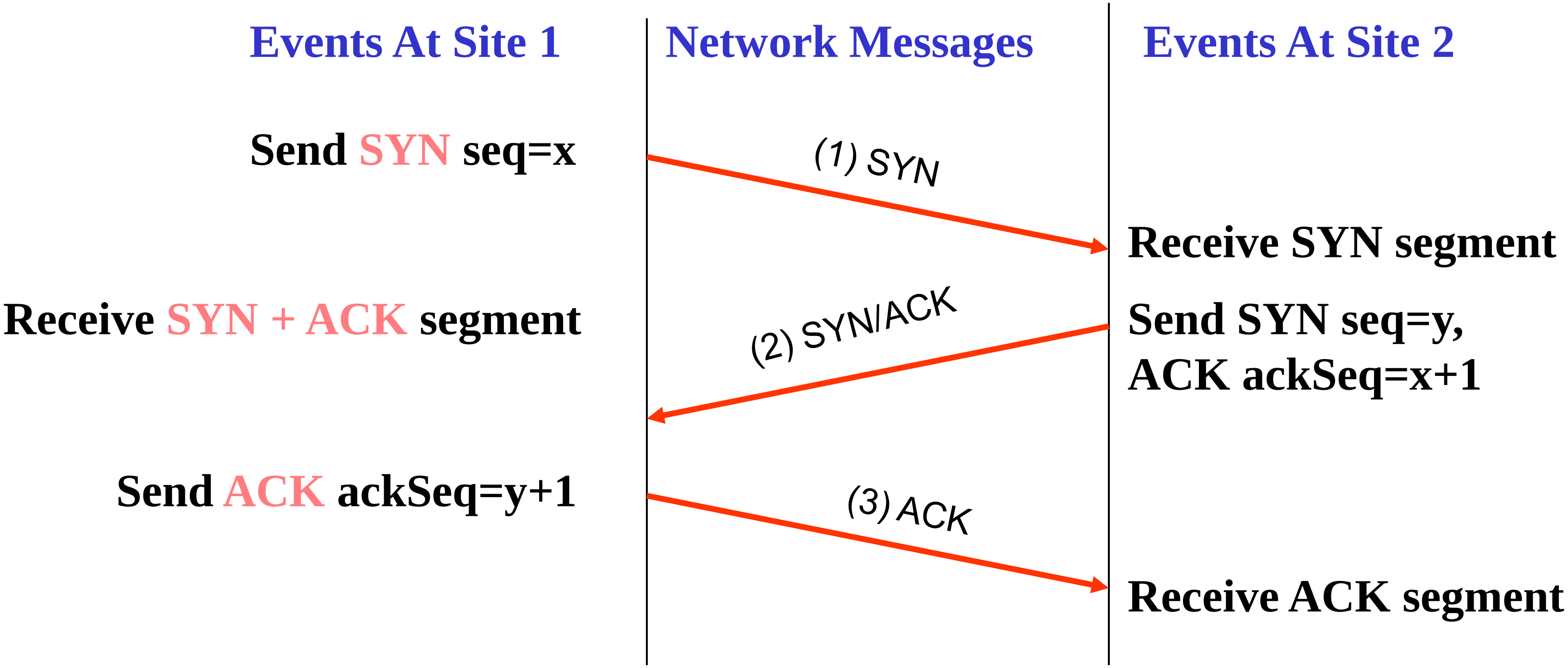
application data (variable length)

ACK

RST, SYN, FIN: connection estab (setup, teardown commands)

Example: Security control of TCP connections (2/3)

Connection Establishment using Three-Way Handshake



Events At Site 1

Network Messages

Events At Site 2

Send SYN seq=x

(1) SYN

Receive SYN segment

Receive SYN + ACK segment

(2) SYN/ACK

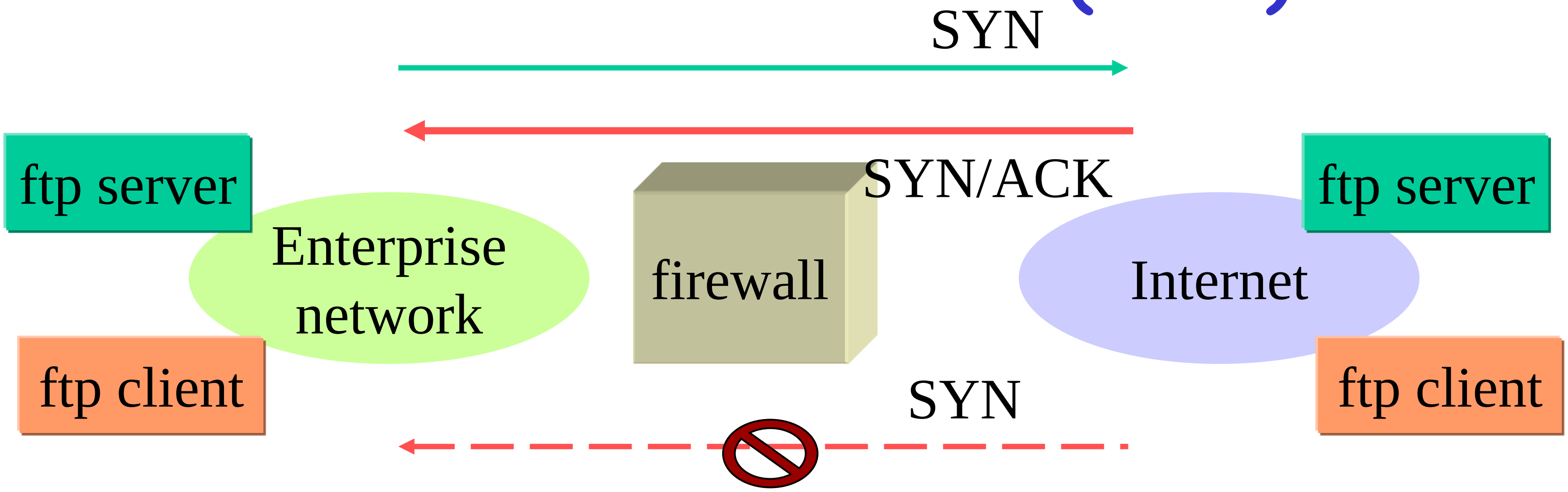
Send SYN seq=y, ACK ackSeq=x+1

Send ACK ackSeq=y+1

(3) ACK

Receive ACK segment

Example: Security control of TCP connections (3/3)



- TCP Connection Blocking - *A rule to block TCP connections initiated from the outside* while allowing responses to internally initiated connections
- "passive open" in FTP - allows only inbound ftp data for sessions that were initiated from inside the private network.

Firewall Service #3: User Control

- Control users’ access to a service.
  - Local users
  - Outside users – authentication is needed.
  - Virtual Private Network (VPN)

Firewall Service #4: Behavior Control

- Control *how* particular services are used, e.g.,
  - Authorization of resource access
  - Only limited access to portions of information on a web server.
  - Filter email to eliminate spam

Limitations

- The firewall cannot protect against attacks that bypass the firewall.
  - e.g., dial-out capability, dial-in modem pool
- Does not protect against internal threats
  - e.g., local users cooperate with external attacker.

現有防火牆的功能區分為三大類

第一類是：低階防火牆 NT$ 43,000

- 採硬體式架構(無硬碟)，具2埠 (含)以上10/100Base-T 介面
- Concurrent sessions達1000個(含)以上及整體處理效能Throughput達20Mbps(含)以上
- 具網路位址轉譯(NAT)及埠位址轉譯(PAT)功能
- 支援IPSec，VPN 功能
- 具備URL Block 內容過濾(Content Filtering)的功能
- 具記錄管理(Syslog/Event logs)和警訊(alarm)及 E-mail notify 功能

第二類是：中階防火牆 NT$ 108,000

- 採硬體式架構(無硬碟)，具3埠(含)以上10/100Base-T 介面
- Concurrent sessions達25000個(含)以上及整體處理效能Throughput達100Mbps(含)以上
- 具網路位址轉譯(NAT)及埠位址轉譯(PAT)功能
- 支援IPSec，VPN 功能
- 具備URL Block 及Java Applet、ActiveX 過濾的功能
- 具記錄管理(Syslog/Event logs)和警訊(alarm)及 E-mail notify 功能
- 具備IDS 入侵攻擊偵測，可紀錄入侵時間及入侵方式，IP 來源

現有防火牆的功能區分為三大類 (cont'd)

第三類是：中高階防火牆 NT$ 350,000

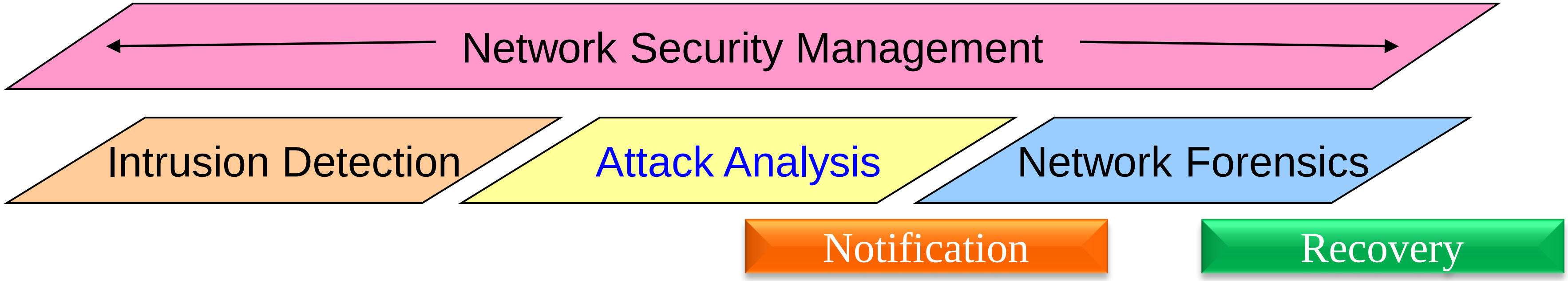
- 採硬體式架構(無硬碟)，具4 埠 (含)以上10/100Base-T 介面
- Concurrent sessions達128000個(含)以上及整體處理效能Throughput達300Mbps(含)以上
- 具網路位址轉譯(NAT)及埠位址轉譯(PAT)功能
- 支援IPSec，VPN 功能
- 具備URL Block 及Java Applet、ActiveX 過濾的功能
- 具記錄管理(Syslog/Event logs)和警訊(alarm)及 E-mail notify 功能
- 具備IDS 入侵攻擊偵測，可紀錄入侵時間及入侵方式，IP 來源

Example of Commercial IDS/IPS Products

- All models may have *similar* functionalities and features.
- But models are configured for a wide range of *performance* and *price*.
  - e.g., entry level price (e.g., 1.5Mbps), price for enterprise models (100Mbps) and price for multi-gigabit for carriers.

Intrusion Detection Systems (IDS)

- A great number of intrusion detection systems (IDS) are software applications running on standard Microsoft windows or Linux platforms.
- For 10 Mbit/s Ethernet links, these platforms provide sufficient power to capture and process the data packets.
- However, for higher-speed links (gigabit and higher) hardware accelerators must be integrated into IDS systems, to process packets in real-time (or near real-time).



Network Security Management

Intrusion Detection

Attack Analysis

Network Forensics

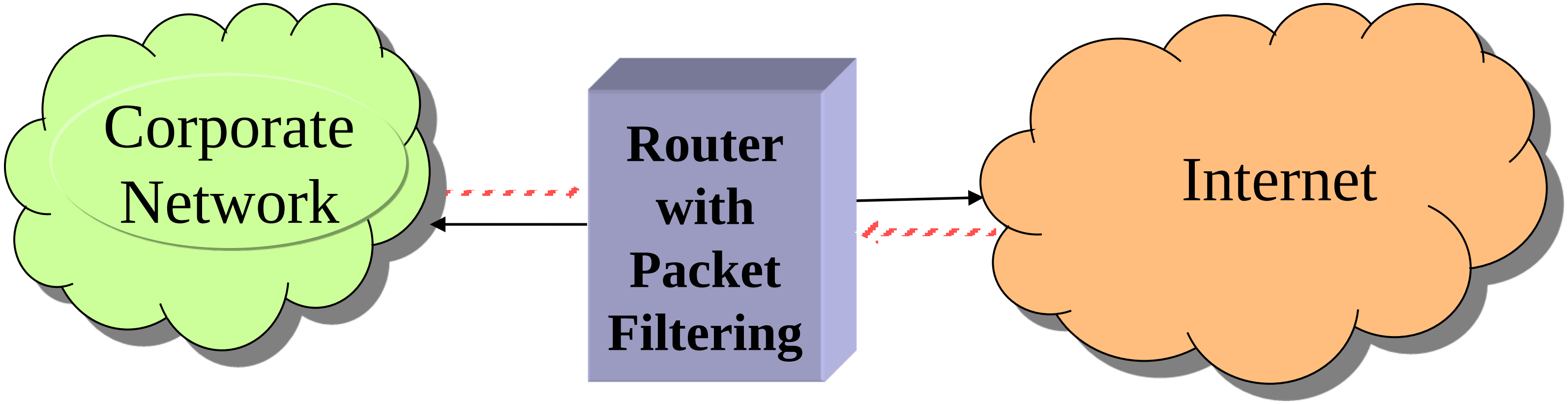
Notification

Recovery

Types of Firewalls

- Packet-filtering
- Stateful inspection firewalls
- Application-level gateway

Packet Filtering Router



- To block transmission of certain classes of traffic
  - Inbound/Outbound filters
  - Access Control List (ACL) – a set of rules
  - Per-packet inspection
- It typically does ***not*** have the ability to maintain session state

Packet-Filtering Gateway-Example

Action	src	port	dest	port	comment
block	SPIGOT	*	*	*	// inbound: don’t trust this host
allow	*	*	our-gw	25	// inbound: connect to our SMTP port
allow	our-gw	25	*	*	//outbound: our mail server connect to other SMTP port
allow	*	*	*	25	// outbound: any internal hosts connect to outside SMTP well-known port ; this however could be a security hole
block	*	*	*	*	default

Packet-Filtering Gateway-Example (cont'd)

Action	src	port	dest	port	flags	comment
■ allow	*our hosts*	*	*	*	*	// **outbound**: *only originating internally*
■ *allow*	*	*	*	*	*ACK*	// **inbound:** *replies to our connections*
■ *allow*	*	*	*	*>1024*		// *traffic to high-numbered ports*
■ *block*	*	*	*	*		*default*

Packet Filtering: filter database (1/3)

- Contains a set of *filters* *(rule)*.
- Each filter is a combination of *K* values, one for each *header field*.
- Packet filtering (dropping) based on ***source address***, ***destination address***, ***source port***, ***destination port***, ***protocol type***, or ***TCP flags***
  - e.g., SYN and ~ACK - connection initiation; others do have ACK bit set
- "Content-based" Inspection and Filtering
  - e.g., more than black mail list (mail spams, bad mail relay hosts), porno sites, etc.

Packet Filtering: filter matching - search (2/3)

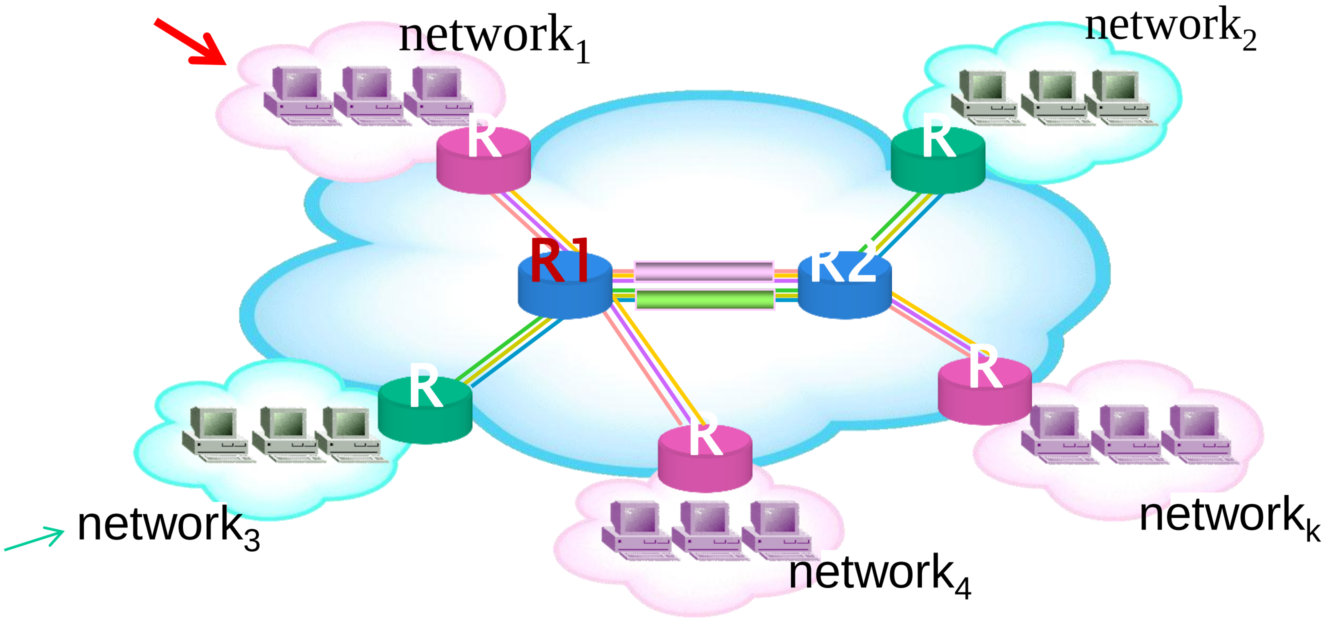
- Three kinds of matches
  - *exact match*, *prefix match*, *range match*
- Exact match
  - useful for protocol and flag fields
- Prefix match
  - The filter field should be a prefix of the header field.
  - useful for blocking access from a certain subnetwork
- Range match
  - The header values should lie in the range specified by the filter.
  - useful for specifying port number ranges, address ranges.
- Each filter has an associated directive
  - *allow* or *block*

Packet Filtering: filter matching - search (3/3)

- Several existing firewall implementations do a linear search.
  - *poor* performance for large filter databases
- Some use caching to improve performance
  - Cache full packet headers to speed up the processing of future lookups
  - The hit rate of caching full IP addresses is at most 80-90%.

Spoofing Attacks

“IP address spoofing” Attacks



- The intruder transmits packets from the outside with internal source address.
  - Solution – discard packets if it is not from the port it is supposed from.
- Spoof trusted IP source address to pass firewall check (need sender authentication)

Source Routing Attacks

- Source routing
  - The source station specifies the route that a packet should take as it crosses the Internet.
- The sender "hopes" to bypass security measures that do NOT analyze the source routing information.
- Solution: discard any packets with source routing.

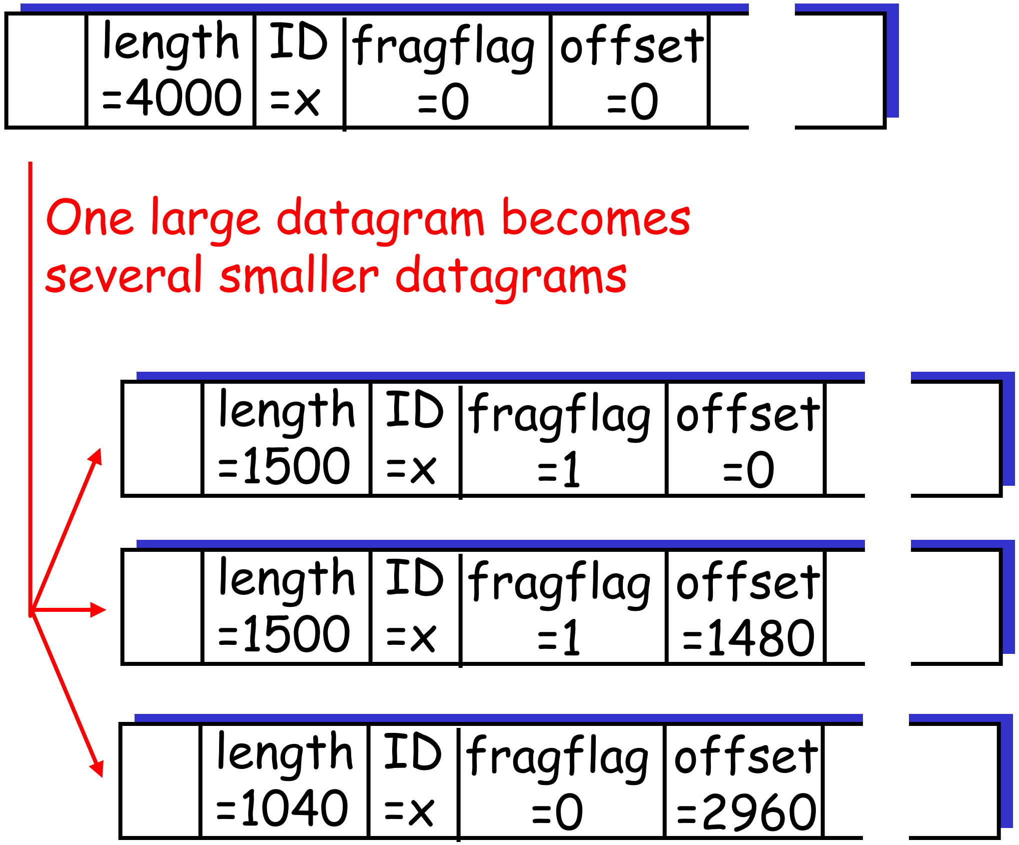
Tiny Fragment Attacks

- The intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment.
- To circumvent filtering rules that depend on TCP header.
- Only the first fragment is examined and the remaining passed through.
- Solution: discard any packets whose protocol number is TCP and IP fragment offset is 1.

IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes



length	ID	fragflag	offset
=4000	=x	=0	=0

One large datagram becomes several smaller datagrams

length	ID	fragflag	offset
=1500	=x	=1	=0

length	ID	fragflag	offset
=1500	=x	=1	=1480

length	ID	fragflag	offset
=1040	=x	=0	=2960

4000=20+3980
=(20+1480)+(20+1480)
+(20+1020)

Tiny Fragment Attacks

- The size of the basic block in IP fragmentation is 8 octets (= 8 bytes = 64 bits)
- Fragment Offset in IP header (in 8 bytes)
- TCP 的 header - 20 octets (not including options)
  - First 8 octets include src port, dest port, seq number
  - second 8 octets include ack number, SYN, ACk, ...
  - The last 4 octets include checksum, urgent data pointer

Tiny Fragment Attacks

- Attacker must put the first 8 octets and the second one in two separate IP datagrams
  - One IP datagram carries the first 8 octets (offset=0)
  - one IP datagram carries the second 8 octets (offset=1)
- Because src port and dest port are in the first 8 octets while SYN and ACK are in the second 8 octets

Types of Firewalls

- Packet-filtering
- Stateful inspection firewalls
- Application-level gateway

Why need stateful inspection

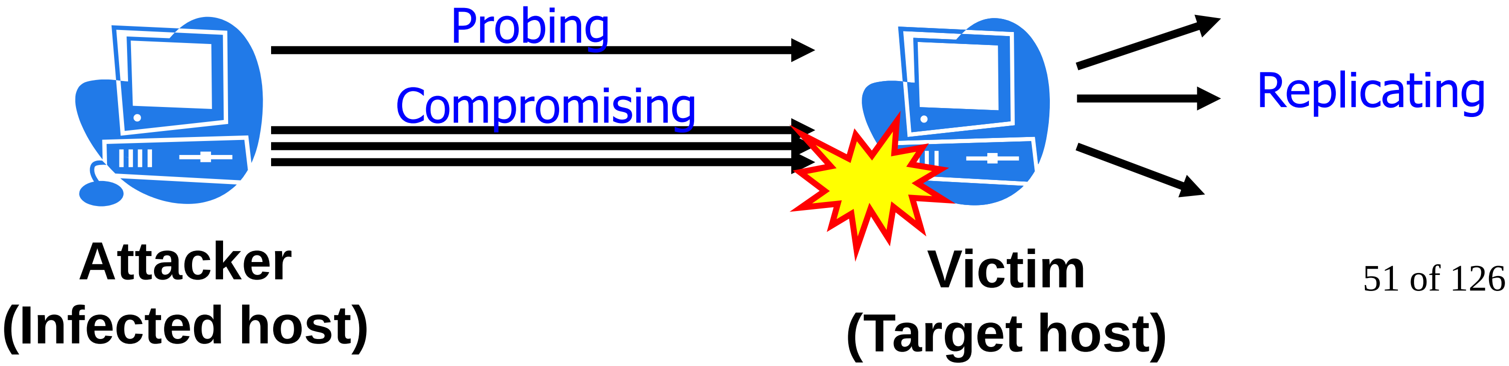
- It is NOT sufficient to examine packets in isolation (i.e. individual packet basis)

Internet Worm

- *Worm* is a self-propagation computer program that automatically exploits the vulnerabilities of the software/computers in the Internet.
- Attack consequences
  - *disrupt* the computer system
  - *consume* network bandwidth
  - *install* any malicious software

Worm Spreading: Stages

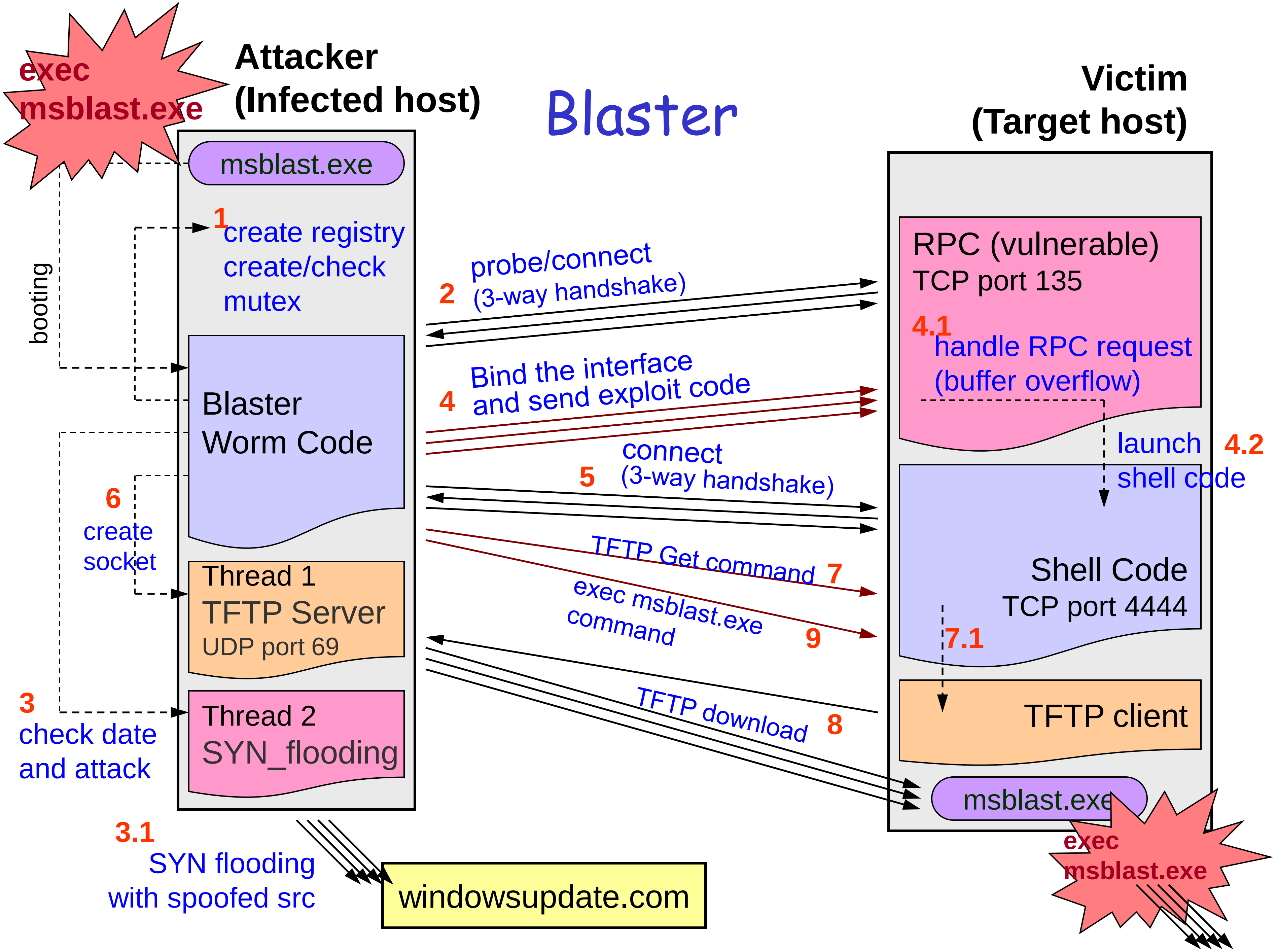
- Probing (optional)
  - Select target hosts (victims) and send probe requests to check the existence of vulnerability
- Compromising
  - Exploit the vulnerability and gain execution privilege
  - Send and execute the worm code
  - Cause certain damages
- Replicating
  - Replicate itself and continue spreading



Worm Attack: Characteristics

- Attack procedures
  - Each worm has its *specific attack procedure* to compromise the network service of the victim.
- Invariant signature
  - The worm payload has *inevitable invariant exploit bytes*.
- Outbreak
  - *High* traffic volume
  - *Address dispersion*
    - Due to the wide spreading, the infected host selects a wide range of IP destination as next targets.
  - *Zero-wait spreading*
    - A victim launches the same attack as soon as it is infected.
  - *Epidemic spreading*
    - Three phases: slow start, fast spread, slow finish

Blaster



exec msblast.exe

Attacker (Infected host)

msblast.exe

booting

1 create registry create/check mutex

Blaster Worm Code

6 create socket

Thread 1 TFTP Server UDP port 69

3 check date and attack

Thread 2 SYN_flooding

3.1 SYN flooding with spoofed src

windowsupdate.com

2 probe/connect (3-way handshake)

4 Bind the interface and send exploit code

5 connect (3-way handshake)

TFTP Get command 7

exec msblast.exe command 9

TFTP download 8

Victim (Target host)

RPC (vulnerable) TCP port 135

4.1 handle RPC request (buffer overflow)

launch shell code 4.2

Shell Code TCP port 4444

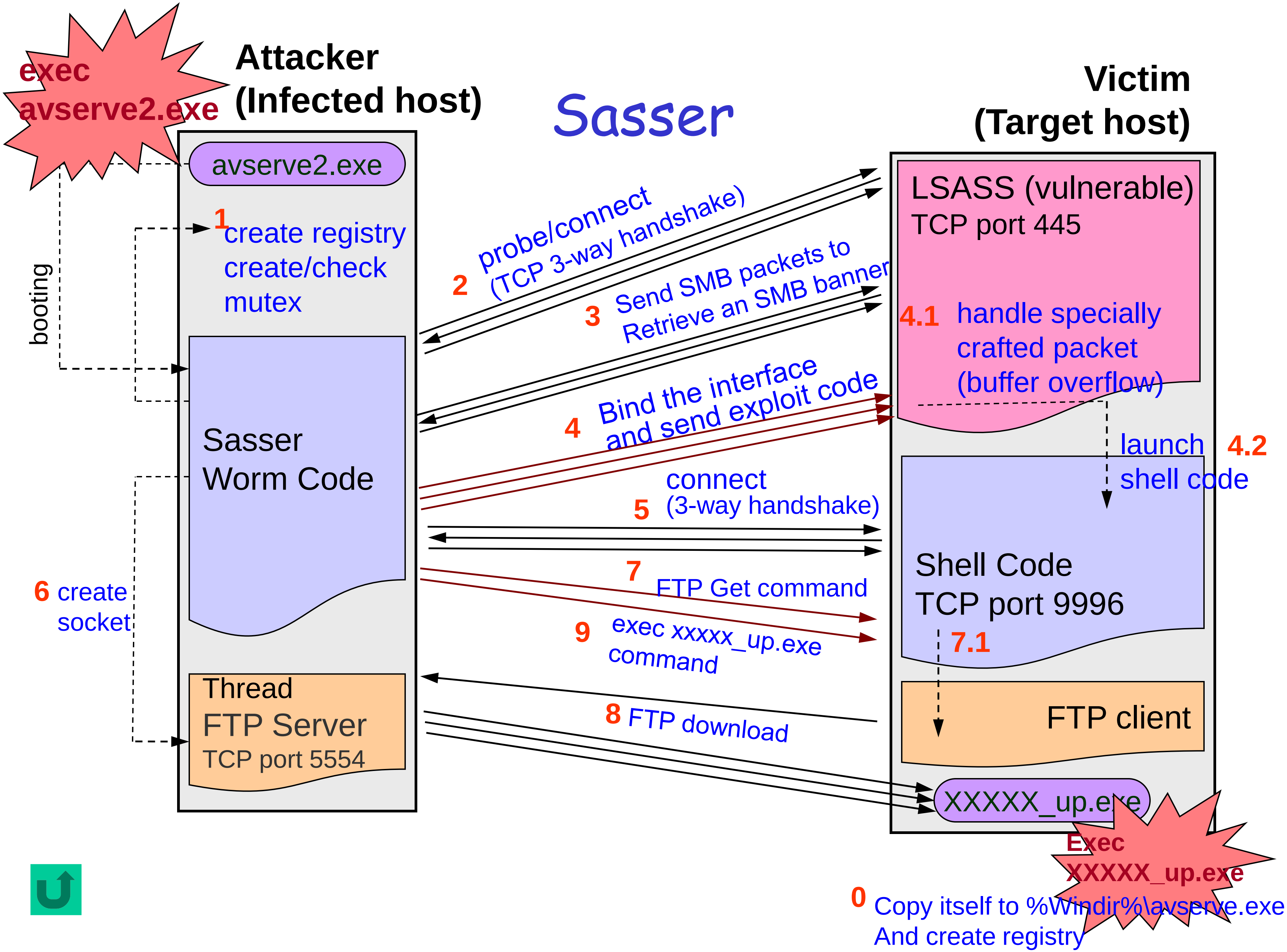
7.1

TFTP client

msblast.exe

exec msblast.exe

Sasser



exec avserve2.exe

Attacker (Infected host)

avserve2.exe

booting

1 create registry create/check mutex

Sasser Worm Code

6 create socket

Thread FTP Server TCP port 5554

Victim (Target host)

LSASS (vulnerable) TCP port 445

2 probe/connect (TCP 3-way handshake)

3 Send SMB packets to Retrieve an SMB banner

4 Bind the interface and send exploit code

4.1 handle specially crafted packet (buffer overflow)

launch shell code 4.2

5 connect (3-way handshake)

Shell Code TCP port 9996

7 FTP Get command

7.1

9 exec xxxxx_up.exe command

FTP client

8 FTP download

XXXXX_up.exe

Exec XXXXX_up.exe

0 Copy itself to %Windir%\avserve.exe
And create registry

Problems

- Internet worms observed in the literature posses *sophisticated* and *complex behaviors*.
  - Target on specific service/application (employing certain communication protocols).
  - The entire course of attack undergoes *a series of actions* for a certain period of time.

→ Per-packet or per-connection monitoring is insufficient.

- Procedure or behavior-based monitoring is necessary.

Problems? (cont'd)

- Internet worms *propagate rapidly and cause severe damages*.
- Worst, once compromising target host, they *can secretly transplant any other programs for future attacks*.

➢ An early detection system is necessary and important.

  - avoid severe damages
  - mitigate the threats as early as possible

Information used in stateful inspection technology

- Communication Information
  - Information from all seven layers in the packet
- Communication State
  - derived from *past* communications
  - e.g.,
    - Save the outgoing PORT command of an FTP session;
    - Used to verify an incoming FTP data connection.
- State Information (context basis)
  - derived from past communications and *other* applications.
  - used in making the control decision for new communication attempts.
  - e.g., a *previously authenticated user* would be allowed access through the firewall for *authorized services* only

Stateful Inspection

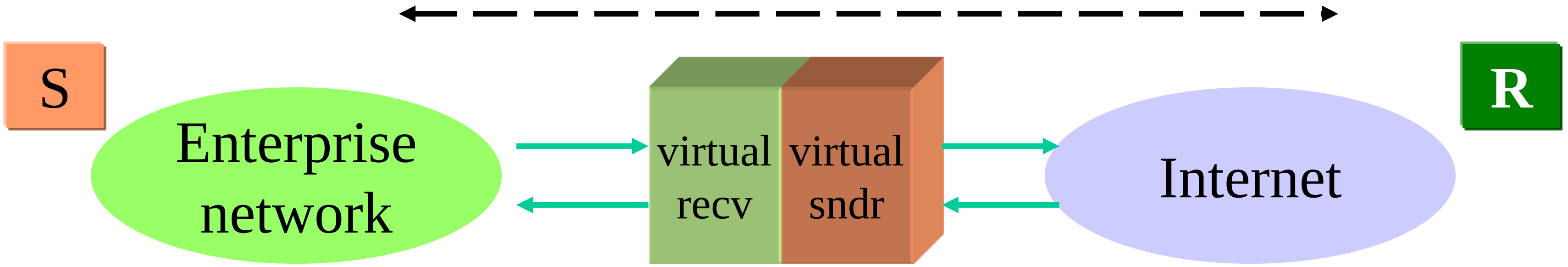
- Packets are intercepted at the network layer.
- Packets are *examined* from all communication layers and relevant data are extracted only.
- These data are analyzed to *derive* communication-derived state and application-derived state and context info
- The system maintains this information in dynamic state tables for evaluating subsequent connection attempts.
- This provides cumulative data against which *subsequent* communication attempts can be evaluated.

Types of Firewalls

- Packet-filtering
- Stateful inspection firewalls
- Application-level gateway

Application-Level Gateway

- Better security than packet filtering
- Service RELAY
  - also known as proxy server
  - e.g., offering controlled TELNET, FTP, and SMTP access.



Application-Level Gateway (cont'd)

- Application gateways *breaks* the client/server model:
  - one from the client to the firewall and
  - one from the firewall to the server.
- To *log* and *control* all incoming and outgoing traffic
  - e.g., restrict outbound FTP traffic to authorized individuals (user authentication)
  - be configured to support only specific features of an application that the administrator considers acceptable.

Application-Level Gateway (cont'd)

- Authentication server for inbound services
  - Users gain access to a network by going through a process that establishes session state, user authentication, and authorization policy.
  - Provides strong security because the session flow is retained at the ***application*** layer.

- Performance is a major issue!
  - Maintaining session states is CPU intensive.
  - Can handle only a limited number of sessions at one time.
  - Must at least compatible with line speed (packet per second (pps)).

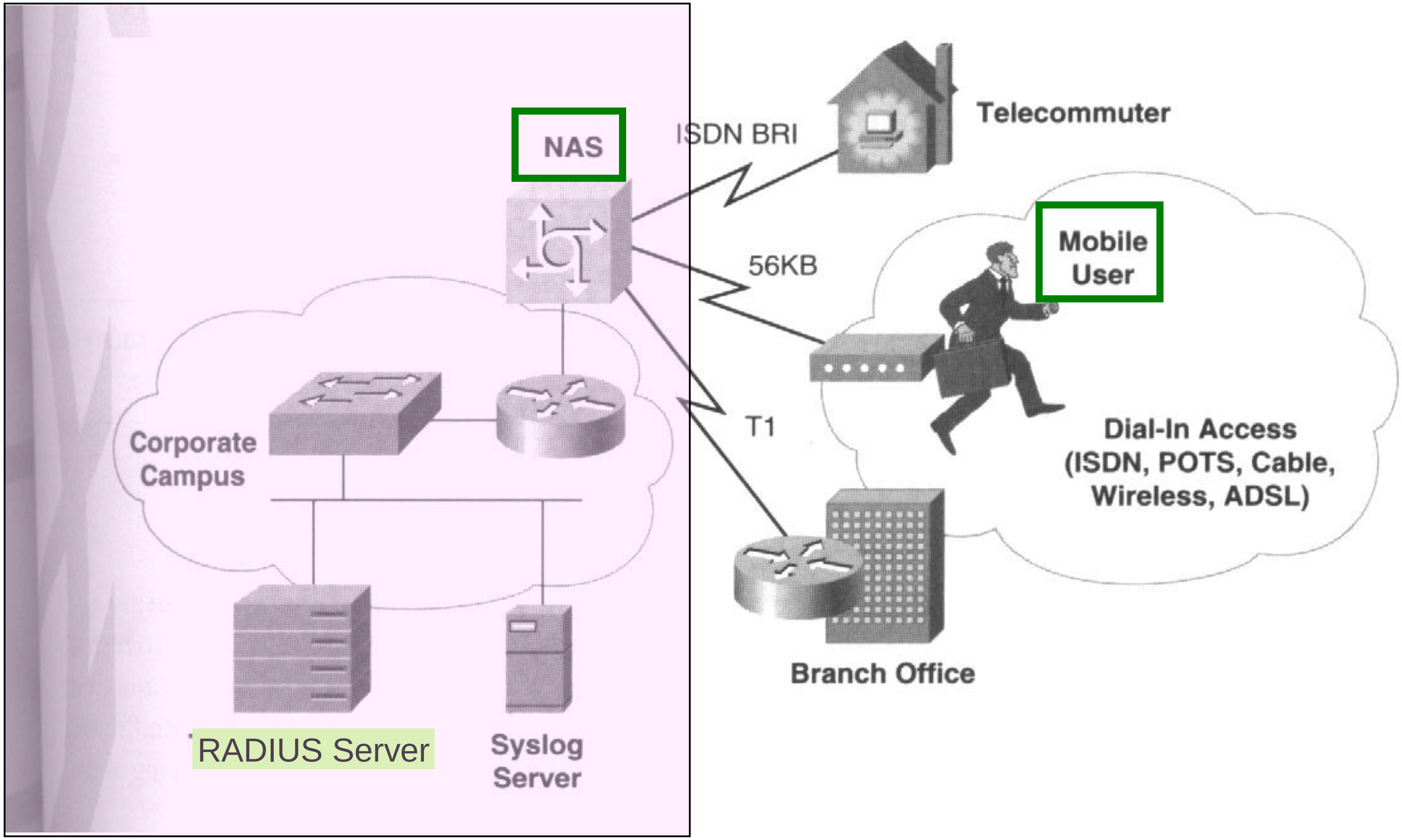
Comparisons

Firewall Capability	Packet filters	Application-level gateway	Stateful inspection
Communication Information	Partial	Partial	Yes
Communication -derived State	No	Partial	Yes
Application-derived State	No	Yes	Yes
Information Manipulation	Partial	Yes	Yes

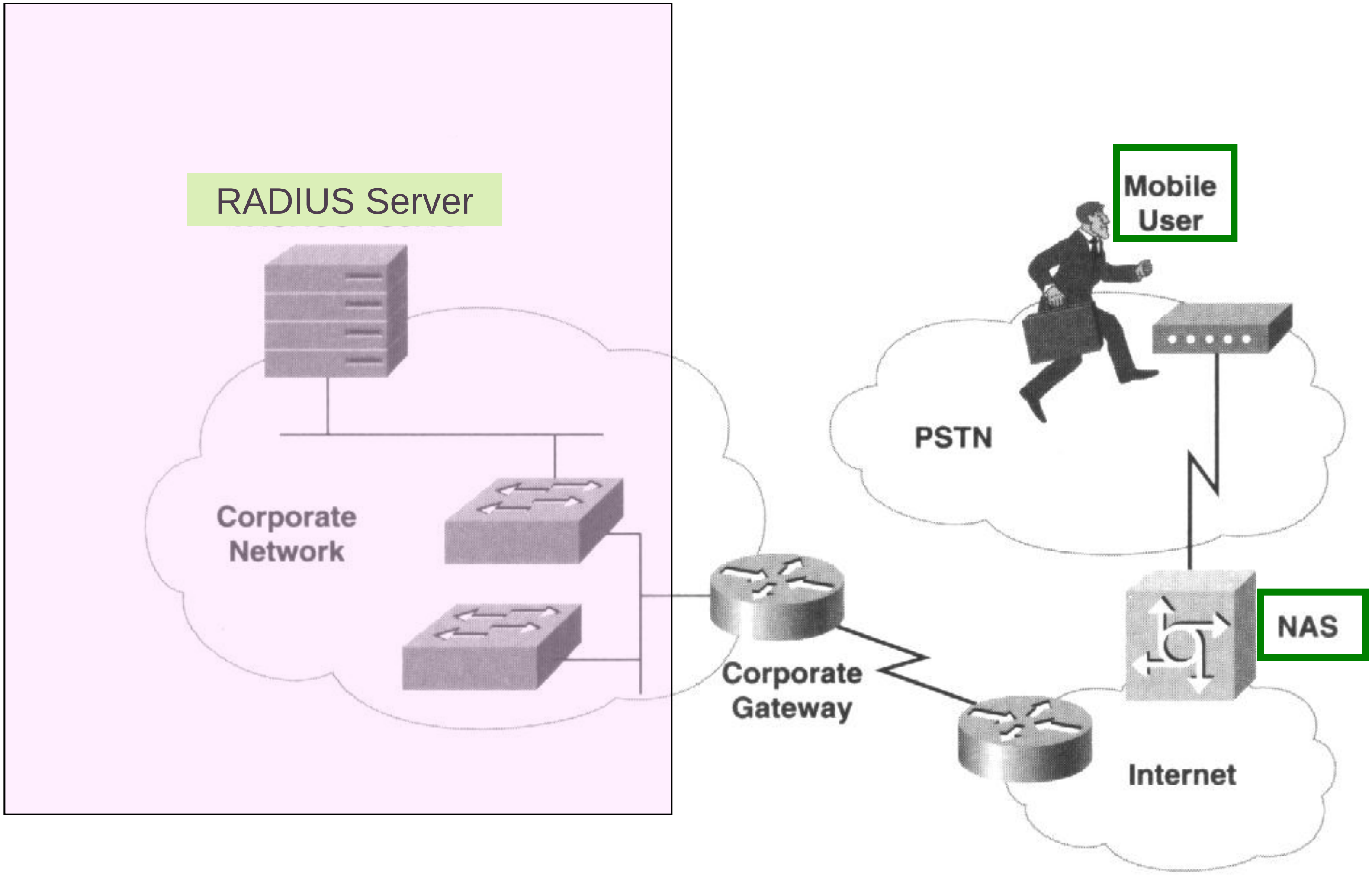
Proxy Server with User Authentication

- Proxy server *challenges* a user initially at the application layer.
- May integrate with an industry-standard user authentication database, e.g.,
  - Terminal Access Controller Access Control System (TACACS)+
  - Remote Authentication Dial-In User Service *(RADIUS)* ▶
- Once passed, the firewall *shifts the session flow*, and all traffic thereafter flows directly between the two parties while maintaining session state, e.g., VPN.

A Dial-In Access Environment



Dial-In Access Using the Internet



Bastion Host

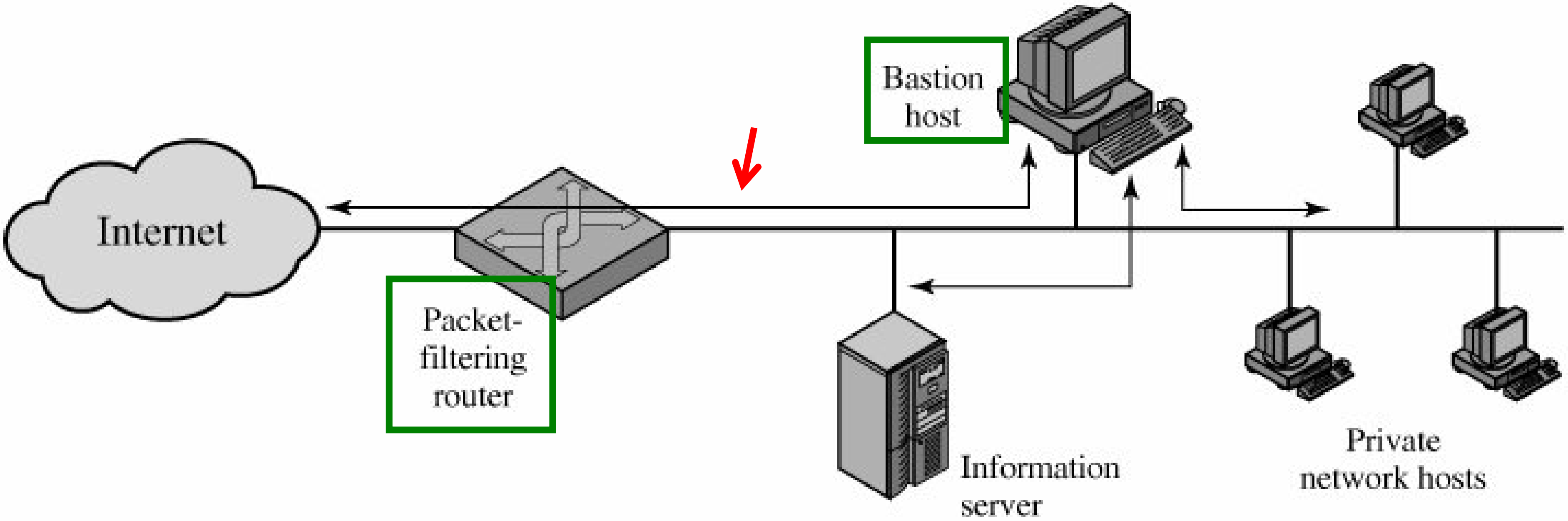
Bastion Host

- It is a system identified by the firewall administrator as a critical strong point in the network's security.
- It is suitable to serve as application level gateways

Bastion Host: Characteristics

- Secure version of operating system
- Only the services that the network administrator considers essential are installed
- May require *user authentication* before a user is allowed to access to the proxy services.
- Maintains detailed *audit* info by logging all traffic, each connection and its duration.
  - Audit log is important to discover and terminate intruder attacks.
- Each proxy module is independent and separately managed.

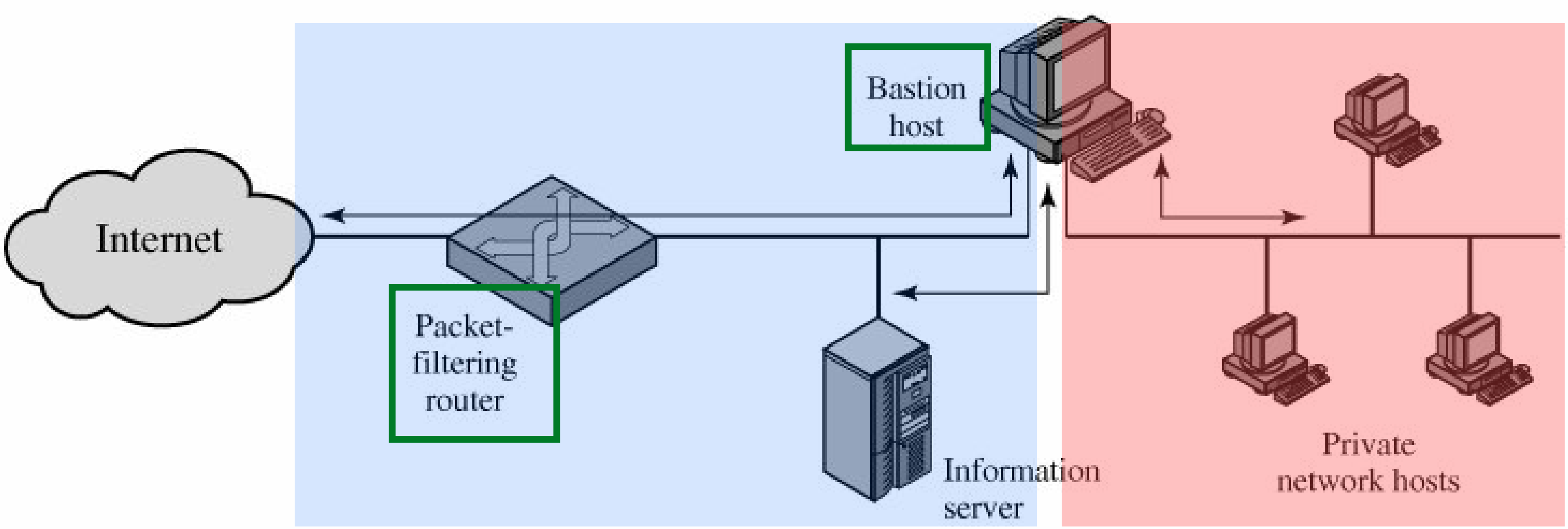
Single-homed bastion host



(a) Screened host firewall system (single-homed bastion host)

- The router
  - for inbound traffic, only allows IP packets destined for the bastion host.
  - for outbound traffic, only allows IP packets sent from the bastion host.
- Bastion host
  - Authentication and proxy functions.
- They are on the *same* network
- If router is comprised, door is opened up.

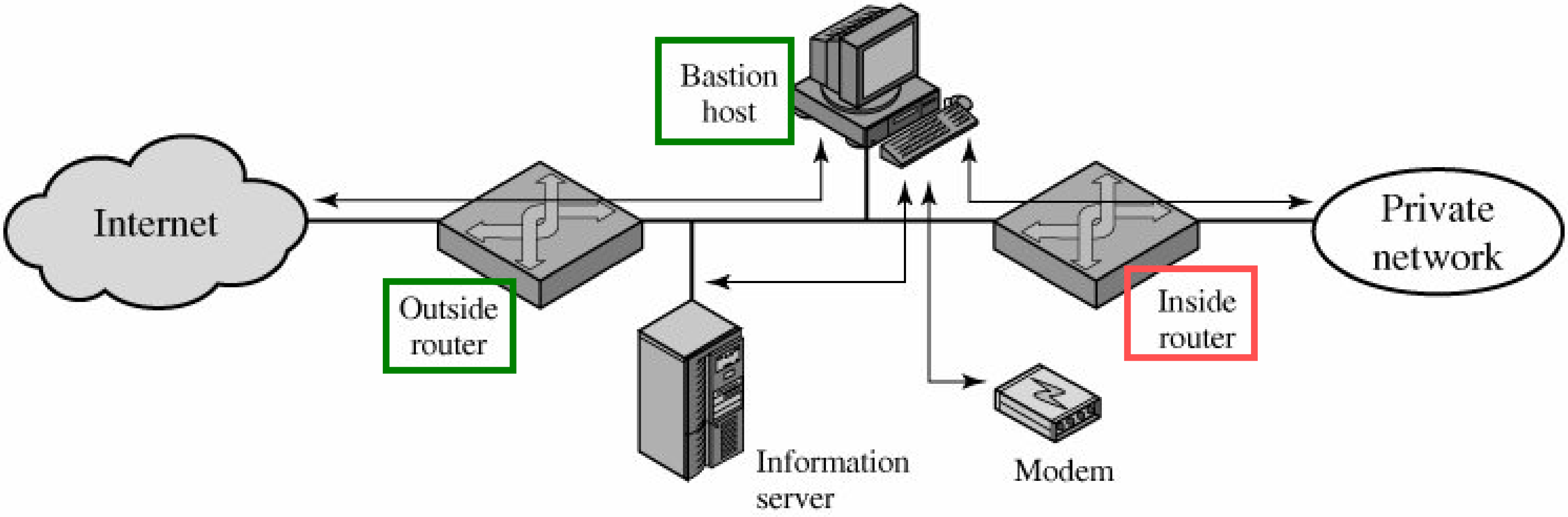
Dual-homed bastion system



(b) Screened host firewall system (dual-homed bastion host)

- *Two* physically separate networks
- Bastion host serves as a gateway

Screened subnet firewall



(c) Screened-subnet firewall system

- Two packet-filtering routers and a bastion host.

網路管理

- **現今企業組織所面臨的問題:**
- 區域網路眾多節點中內，若懷疑某網段有異常情形時，多用「猜測」及「換換看」的除錯方式，完全無法達到網管效率
- 無線網路的架設，因無實體線路存在，若某網段有異常情形時，不知從何查起
- 無法即時針對網路流量進行分析，即時除錯

網路管理 (cont'd)

- 網路分析儀(Protocol Analyzer)
  - 網管模組及網路連接器
  - 當網管人員發現或懷疑某區段網路有異常時，
    - 可將Protocol Analyzer接上網路，
    - 即時的分析該網段的流量及資料封包，
    - 並藉由”專家系統” 快速的找出網路問題所在，並獲得可能解決方法的提示

網路管理 (cont'd)

- RMON2的Agent
  - 利用management station 連接遠端的management agent進行跨網際的網路分析及除錯
  - MIB (Management Information Base)
  - 利用RMON作網路的使用分析，對於網路的長期使用規劃可以提供正確而有效的分析數據

Denial of Service Attacks

Introduction

- Motivated by the widely known February 2000 distributed attacks on Yahoo!, Amazon.com, CNN.com, and other major Web sites.

- A denial of service is characterized by an explicit attempt by an attacker to prevent legitimate users from using resources.

- Even though denial of service attacks have existed for some time, their recent distributed formats have made these attacks *more difficult to prevent*.

Characteristics of DoS Attacks

- Examples of denial of service attacks include:
  - Attempts to "flood" a network, thereby preventing legitimate network traffic.
  - Attempts to disrupt connections between two machines thereby preventing access to a service.
  - Attempts to prevent a particular individual from accessing a service.
  - Attempts to disrupt service to a specific system or person.

Attacks that Exploit Vulnerability of TCP and IP Protocols!

Example Attacks

- IP Spoofing Attacks
- Source Routing Attacks
- Tiny Fragments Attacks
- Stateful inspection
- Sequence number prediction
- TCP SYN flooding
- The Land Attack - IP DOS
- Snipping

Example #1: IP Spoofing Attack

- Attacker 可以自行填寫送出的 IP packet中 source address 的欄位。
- victim 無法知道真正的攻擊來源。
- How to do IP spoofing?
  - Use raw socket
  - 自行算出該 raw packet 的 checksum，以避免被當成有問題的封包，而被 drop 。

Example #1: IP Spoofing Attack - IP Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Datagram Header

Hacker 可自行
算出 checksum

Example #1: IP Spoofing Attack -

TCP Header

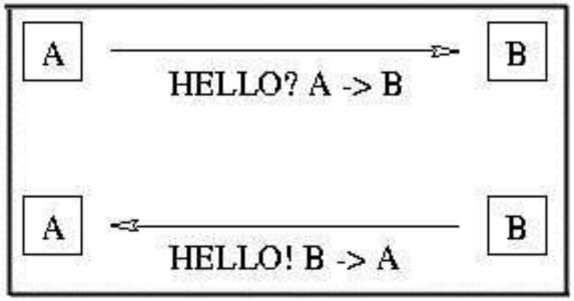
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Hacker 可自行
算出 checksum

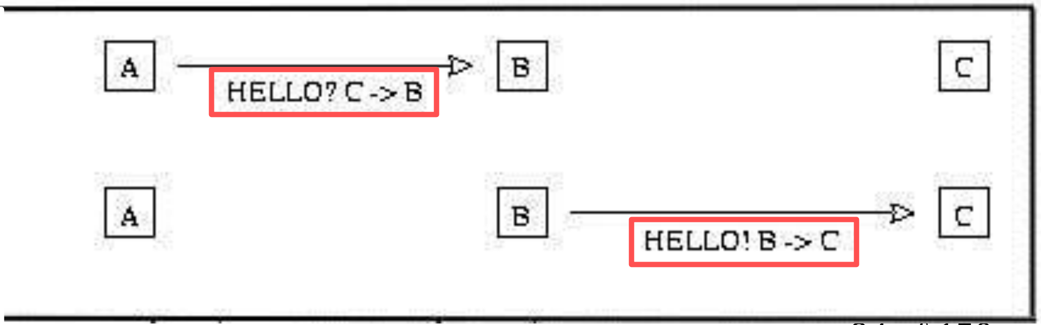
t

Example #1: IP Spoofed Denial-of-Service Attack – "Smurf"

- Use forged ICMP echo request packets sent to IP broadcast addresses.
  - The victim source is flooded with simultaneous replies (1998).
- ICMP Echo/Reply

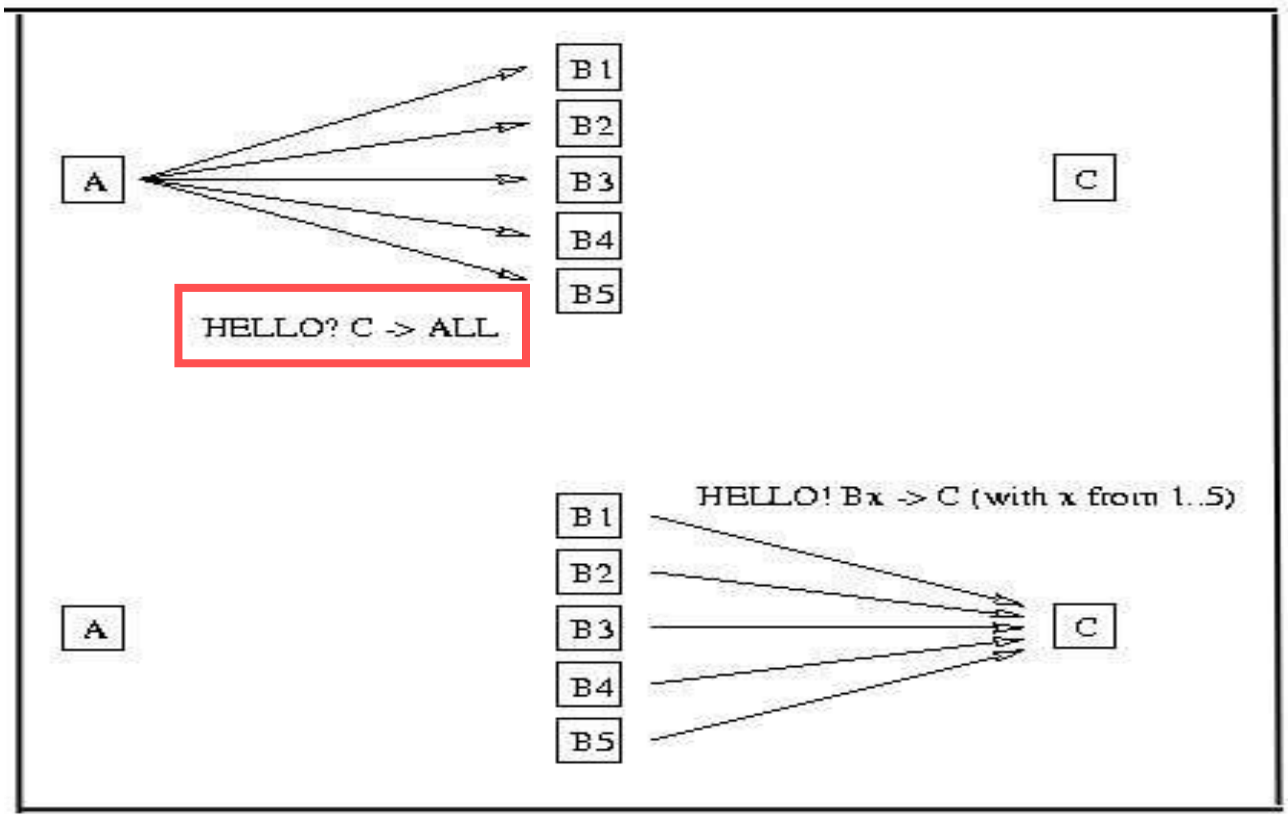


- Spoofing Source address



Example #1: IP Spoofed Denial-of-Service Attack – “Smurf”

- Sending a large amount of ICMP echo traffic to a set of IP broadcast address with a specified spoofing source address.



Example #1: IP Spoofed Denial-of-Service Attack - auto tools

- Automated tools are used to send attacks to multiple *intermediaries* at the same time.
  - Causing all of the intermediaries to direct their responses to the *same victim*.

- Tools looking for potential intermediaries in attacks.
  - *Routers* that do not filter broadcast traffic (broadcast/multicast storming) and *networks* where multiple hosts respond.

Example #1: IP Spoofed Denial-of-Service Attack - performance degradation

- Both the intermediary and victim of the attack may suffer degraded network performance - both on their internal networks or on their connection to the Internet.
- Performance may be degraded to the point that the network cannot be used, i.e. *denial of service*.

Example Attacks

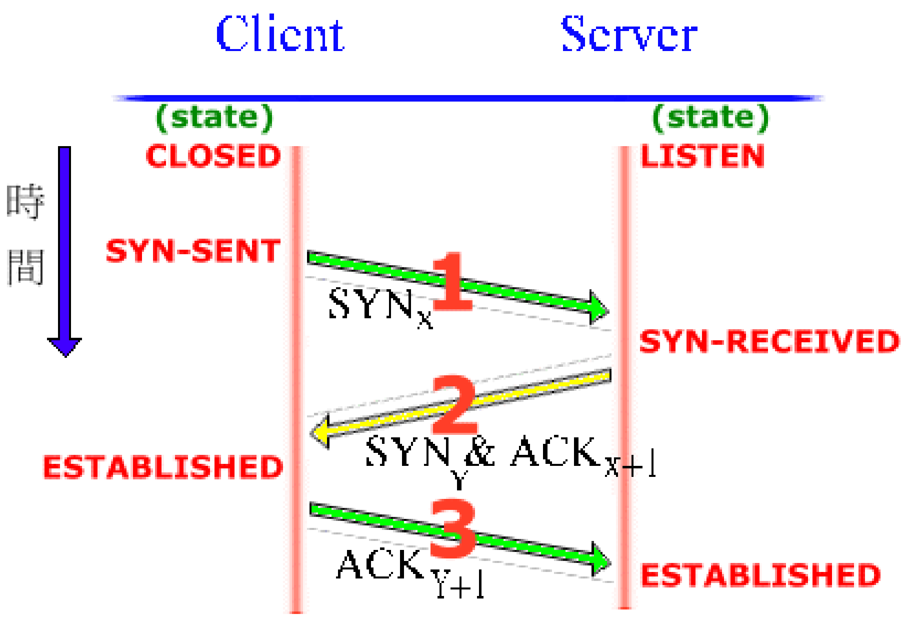
- IP Spoofing Attacks
- Source Routing Attacks
- Tiny Fragments Attacks
- Stateful inspection
- Sequence number prediction
- TCP SYN flooding
- The Land Attack - IP DOS
- Snipping

Example #2: IP-spoofed TCP SYN Flooding Attack

- Any system connected to the Internet and providing TCP-based network services (such as a Web server, FTP server, or mail server) is potentially subject to this attack.
- The attack itself is fundamental to the TCP protocol used by all systems.

Example #2: IP-spoofed TCP SYN Flooding Attack

TCP connection establishment



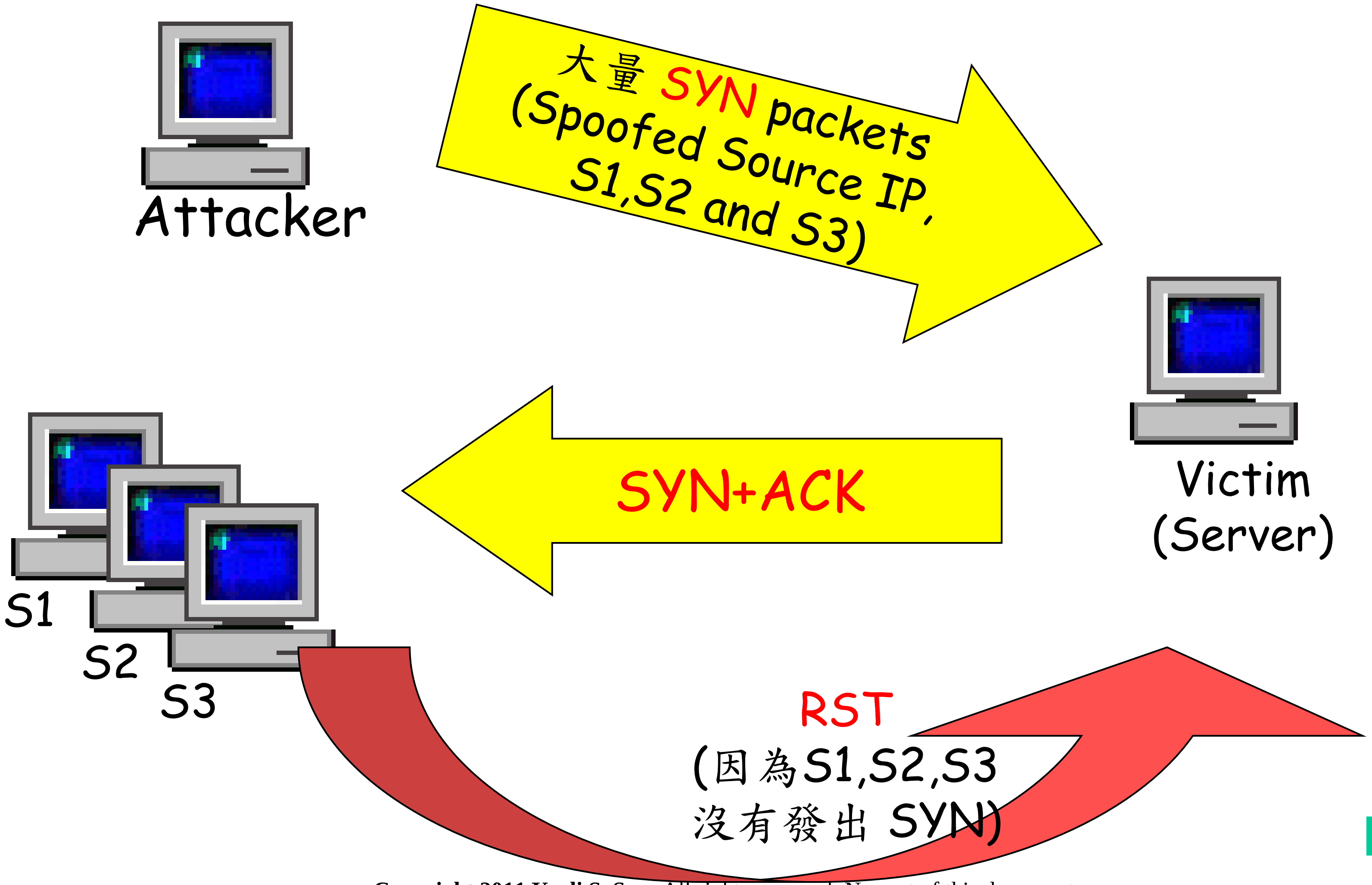
Example #2: IP-spoofed TCP SYN Flooding Attack

- 大量的 SYN packet 送往攻擊主機
- 被攻擊主機通常會保留SYN 5~15個連線
- 連線通常會被保留在系統的queue中，等待連線成功，或者等到timeout時間到達（一般為75秒)
- 假如系統接收 connection 的 buffer 滿了，則新進來的 SYN 會被 drop 掉

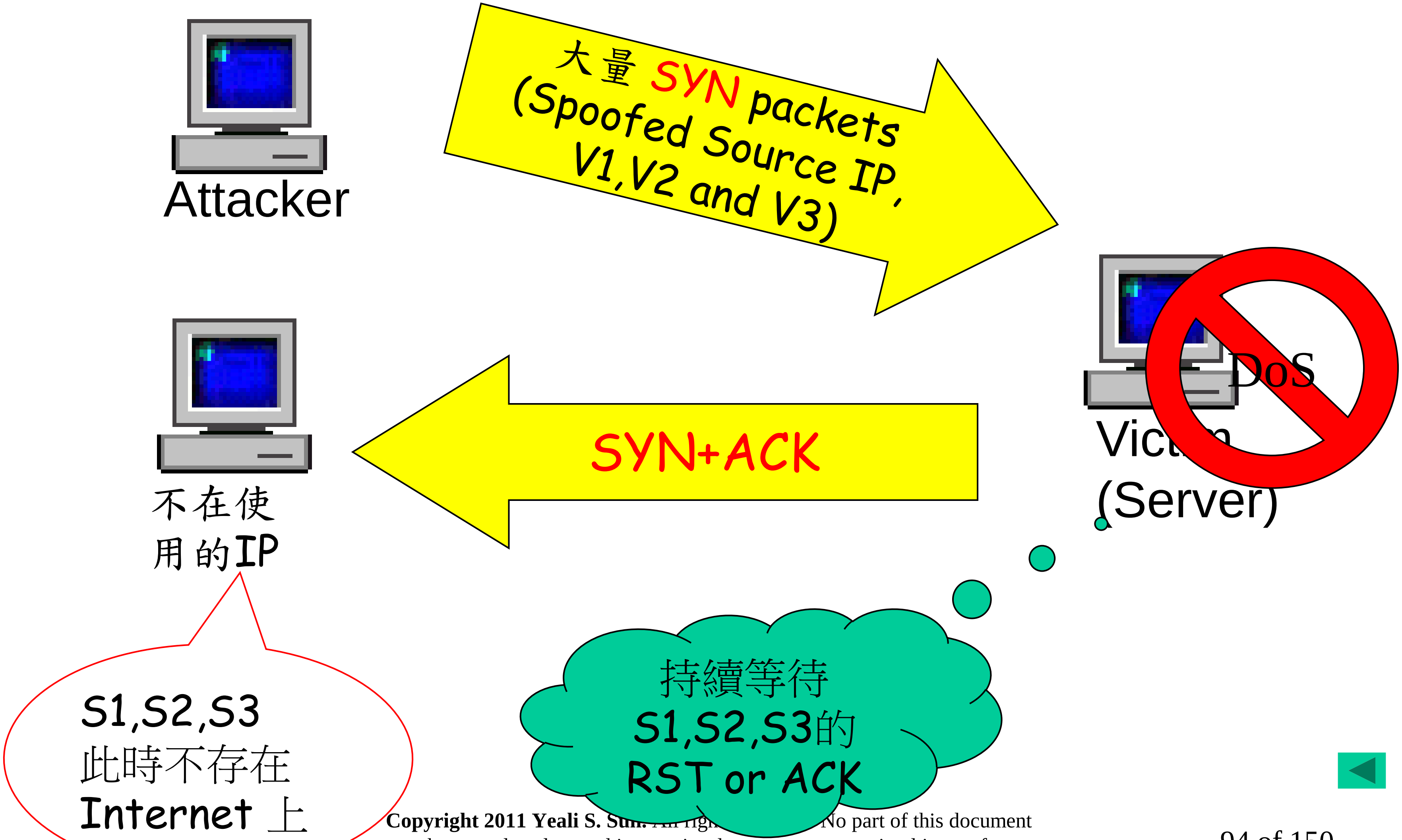
Example #2: IP-spoofed TCP SYN Flooding Attack

- Attacker 要在送出的 SYN packet上 spoof 來源 IP，而且這個 IP 要是當時沒有機器使用的 IP。
  - Victim 會對 spoofed source IP 回 SYN+ACK。
  - 如果 spoofing 的 IP 存在，則會回 RST。Victim 收到 RST 會將 queue 中的SYN清除 (example 1) 。
  - 如果 spoofing 的 IP 不存在，則 victim 則會等候，造成 DoS (example 2) 。
- 因此，選擇當時沒有機器在使用的 IP，可以讓 victim 收不到 RST，而持續等待正常的 ACK ，造成 victim 資源浪費。

Example #2: IP-spoofed TCP SYN Flooding Attack - case 1



Example #2: IP-spoofed TCP SYN Flooding Attack - case 2



Attacker

大量 SYN packets (Spoofed Source IP, V1,V2 and V3)

Victim (Server)

DoS

SYN+ACK

不在使用的IP

S1,S2,S3 此時不存在 Internet 上

持續等待 S1,S2,S3的 RST or ACK

Example Attacks

- IP Spoofing Attacks
- Source Routing Attacks
- Tiny Fragments Attacks
- Stateful inspection
- Sequence number prediction
- SYN flooding
- The Land Attack - IP DOS
- Snipping

Example #3: The Land/ Latierra Attack

- Attacker sends a packet to a victim machine with the source host/port the same as the destination host/port (IP Spoofing).
- Victim 收到此來源與目的位址相同的封包時，會無法正常處理。
- 因為各家作業系統對於這個部分的implementation不同，造成不同結果：
  - Crash/hang
  - Slow down
- 防止方法：安裝各系統提供的修正更新檔案。

Example Attacks

- IP Spoofing Attacks
- Source Routing Attacks
- Tiny Fragments Attacks
- Stateful inspection
- Sequence number prediction
- SYN flooding
- The Land Attack - IP DOS
- Snipping

Example #4: Snipping

- 假設在同一網路上，X 與 Y透過網路在通訊，同時間A正在竊聽之間的訊息
- A可以得到X或Y的 TCP sequence number
- 接著A送出含有正確的TCP sequence number的RST packet到X或Y
  (spoofed IP packet with X or Y)
- 將導致原本X與Y之間的連線中斷 – cut off (snip) an on-going TCP connection

Denial of Service

依照攻擊模式分類：

- 頻寬阻絕
  - UDP Flood Attack
  - ICMP Flood Attack
  - ICMP Smurf Flood Attack
- 伺服器癱瘓
  - TCP SYN Attack
  - File/Socket Descriptor, CPU, Disk, Process 使用過量
- 而這些攻擊，通常都會將自己的來源 IP 隱藏 (IP Spoofing)。

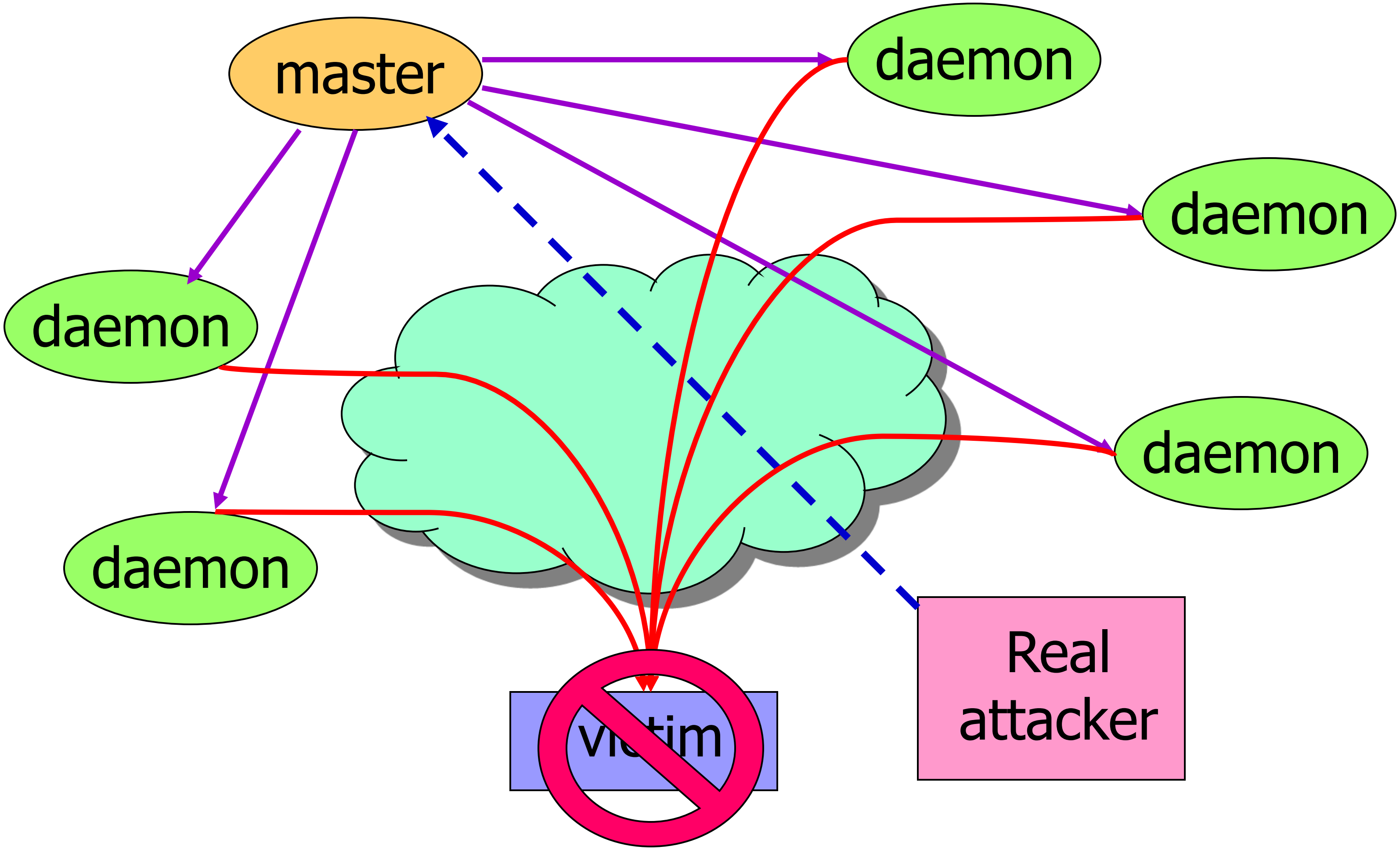
Distributed Denial of Service Attacks

DoS and DDoS

- Denial of Service (DoS)：
  - 這種情形就如同某公司電話總機同一時間被一個人，不停的撥進電話，佔據有限的線路，導致其他正常使用者沒辦法接通。
- Distributed DoS (分散式阻絕服務)：
  - 這種情形就如同某公司電話總機同一時間被一群人，從各地不停的撥進電話，佔據有限的線路，導致其他正常使用者沒辦法。

Distributed Denial of Service (DDoS)

- Tools utilize distributed technology to create *large networks of hosts* capable of launching large coordinated packet flooding denial of service attacks.
- Examples
  - Trinoo
  - Tribe flood network (TFN)
  - Stacheldraht
  - Shaft
  - TFN2K
  - etc.

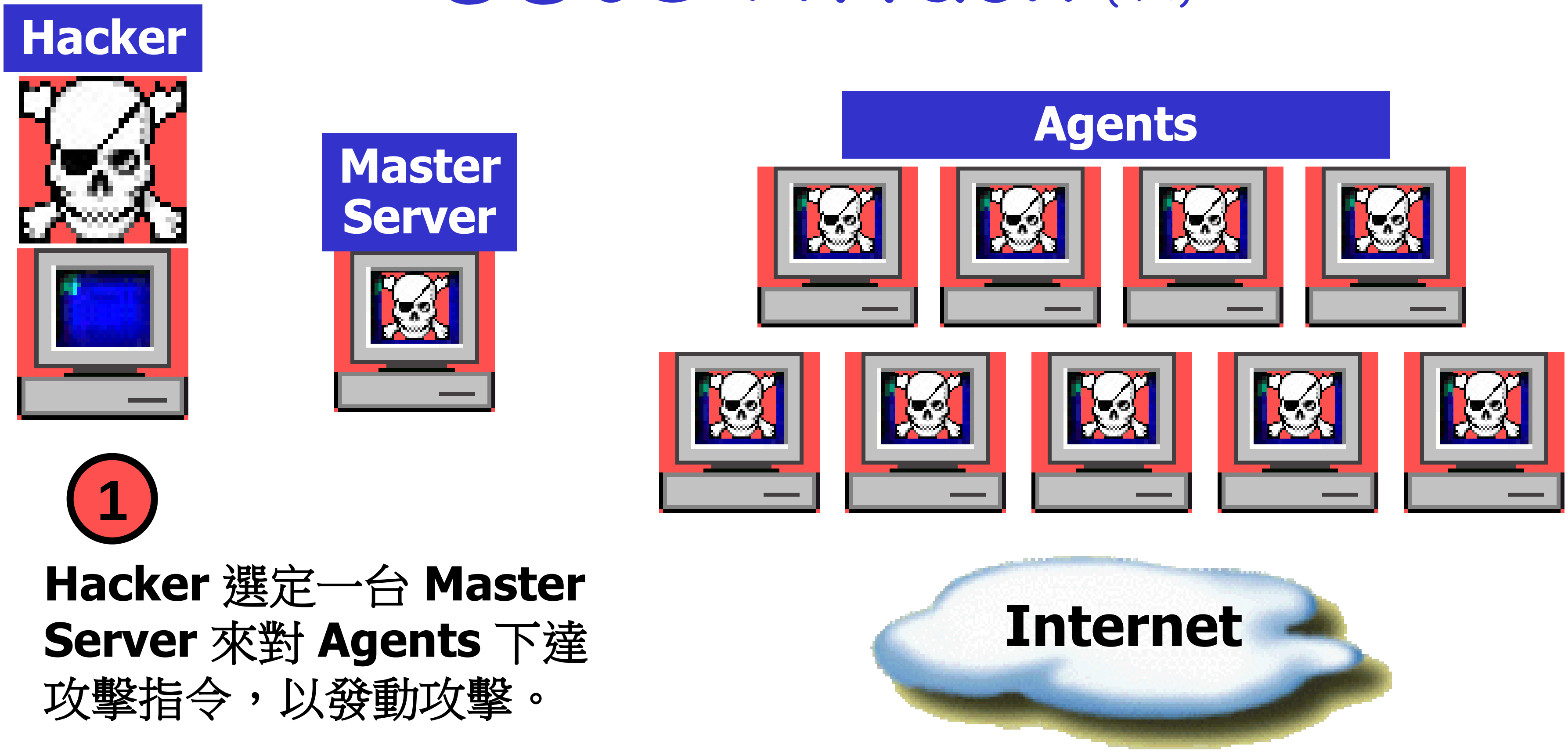


The *four* components of a distributed denial of service attack: a real attacker, a control master program, attack daemon and the victim

A DDoS attack is composed of four elements …

- **Victim**
  - The **target host** that has been chosen to receive the brunt of the attack.
- **Attack daemon agents**
  - These are agent programs that **actually conduct the attack** on the target victim.
  - Deploying these attack daemons requires the attacker to *gain access* and infiltrate (break through secretly) the host computer.
- **Control master program**
  - Its task is to **coordinate the attack**.
- **Real attacker**
  - By using a control master program, the real attacker can stay behind the scenes of the attack.

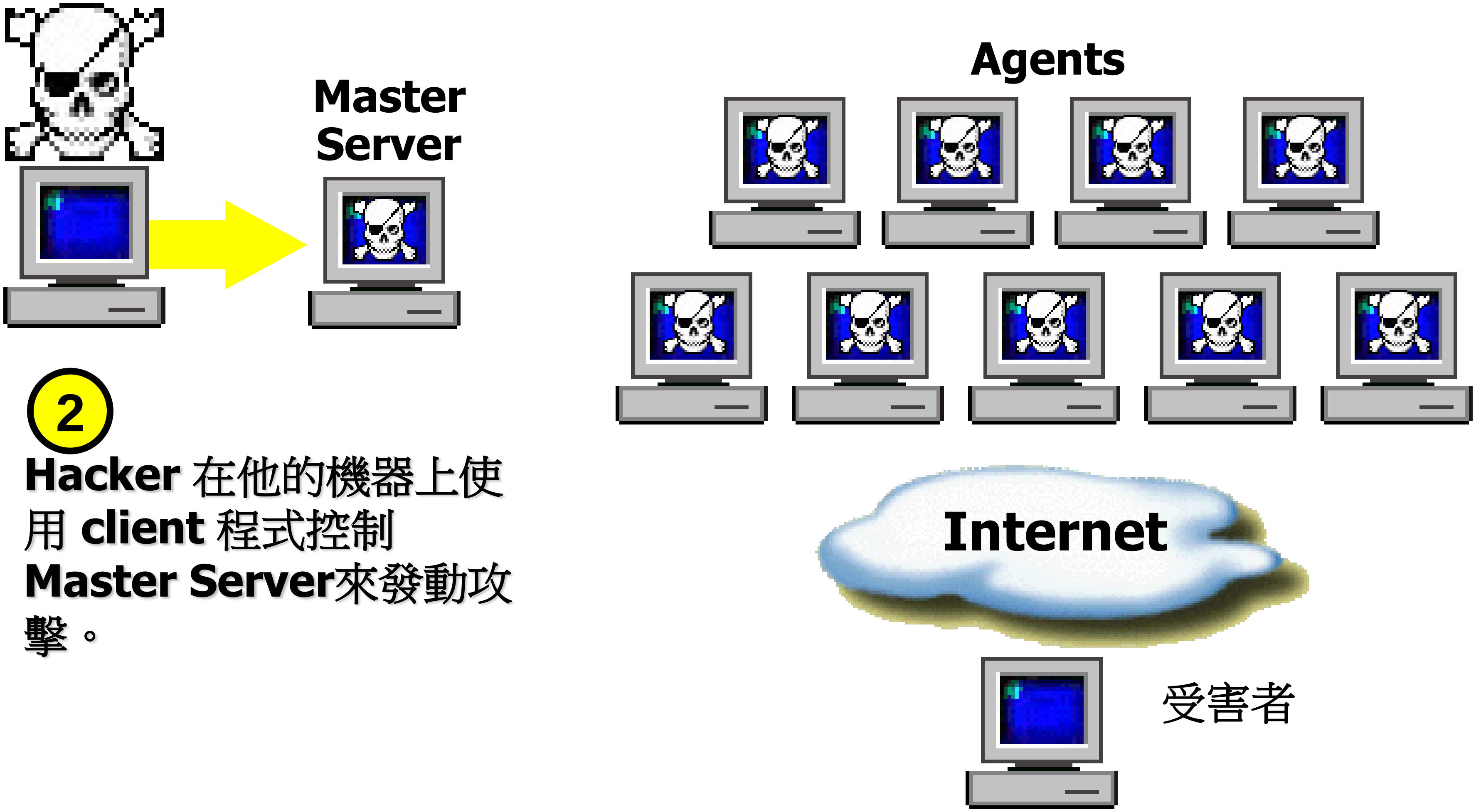
DDoS Attack (1/4)



Hacker 選定一台 **Master Server** 來對 **Agents** 下達攻擊指令，以發動攻擊。

其中 Agents 與 Master Server 是預先透過入侵主機方式取得非法使用權限。並且置入相關攻擊程式。用來接受攻擊指令。

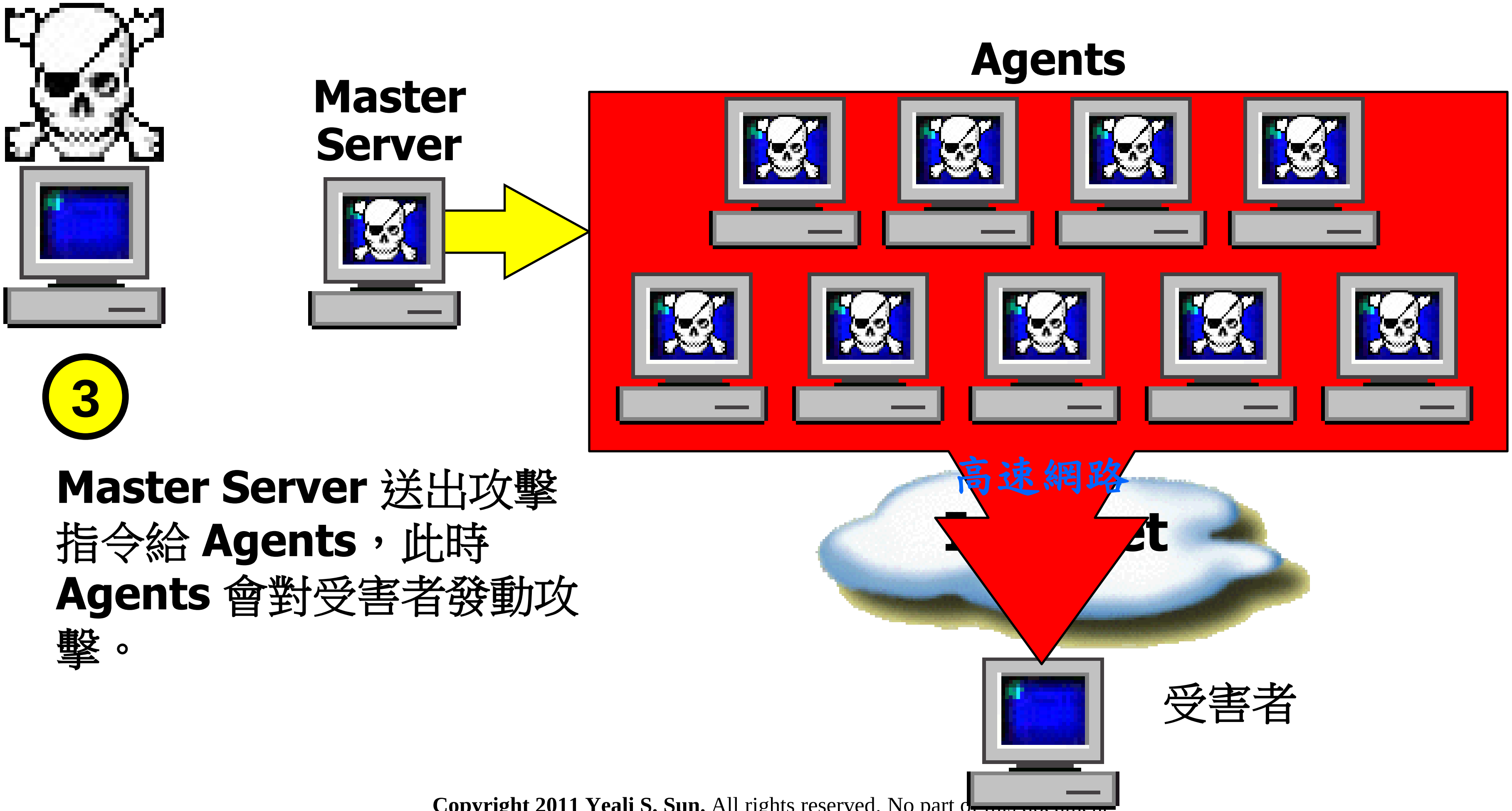
DDoS Attack (2/4)



2

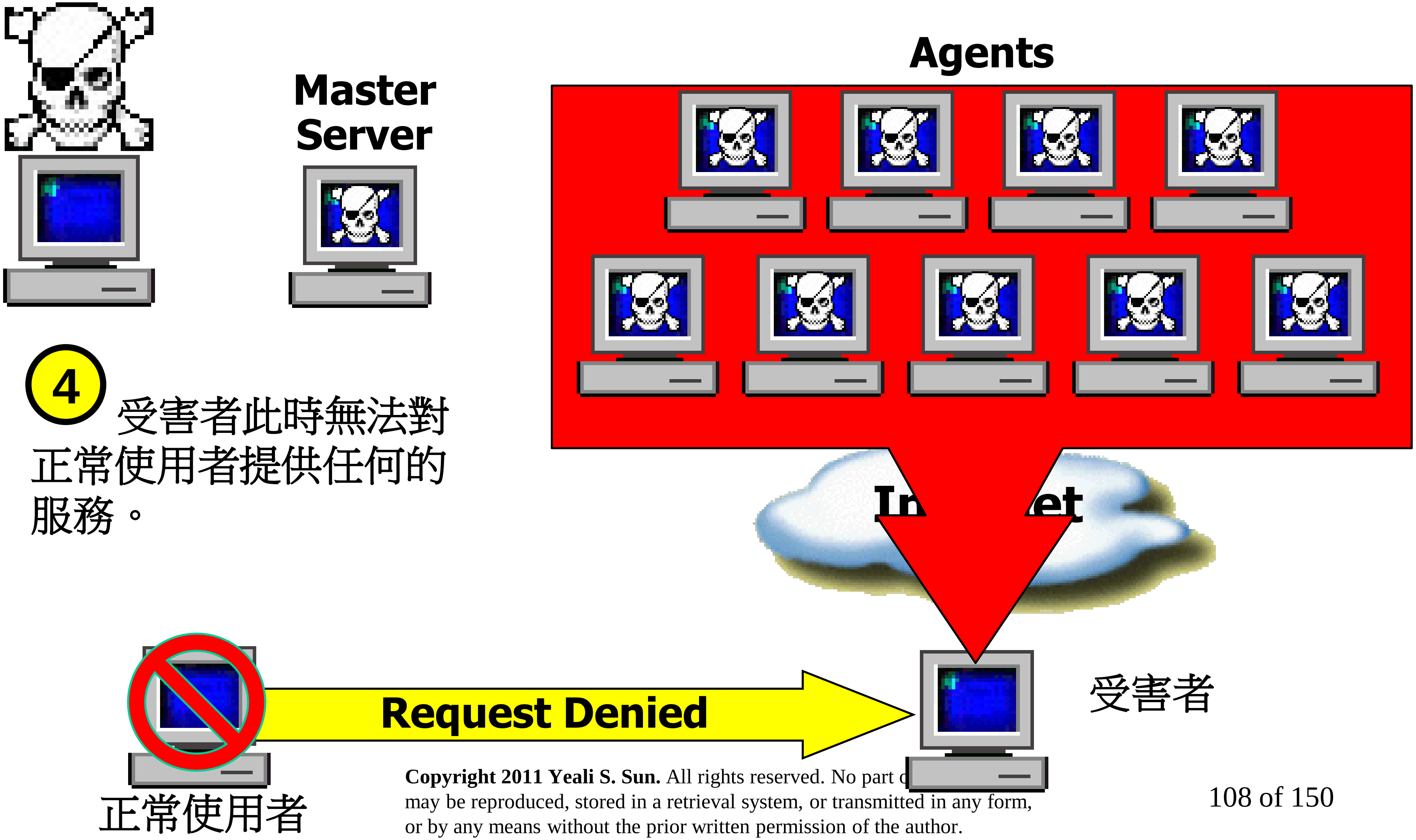
Hacker 在他的機器上使用 **client** 程式控制 **Master Server**來發動攻擊。

DDoS Attack (3/4)



Master Server 送出攻擊指令給 Agents，此時 Agents 會對受害者發動攻擊。

DDoS Attack (4/4)



4 受害者此時無法對正常使用者提供任何的服務。

Attack scenario

- Attacker -> Master : “execute” message
- Master -> Daemons : command
- Daemons -> Victim : attack

- The attacker must study the **target’s network topology** and **search for *bottlenecks* and *vulnerabilities*** that can be exploited during the attack.

Trinoo

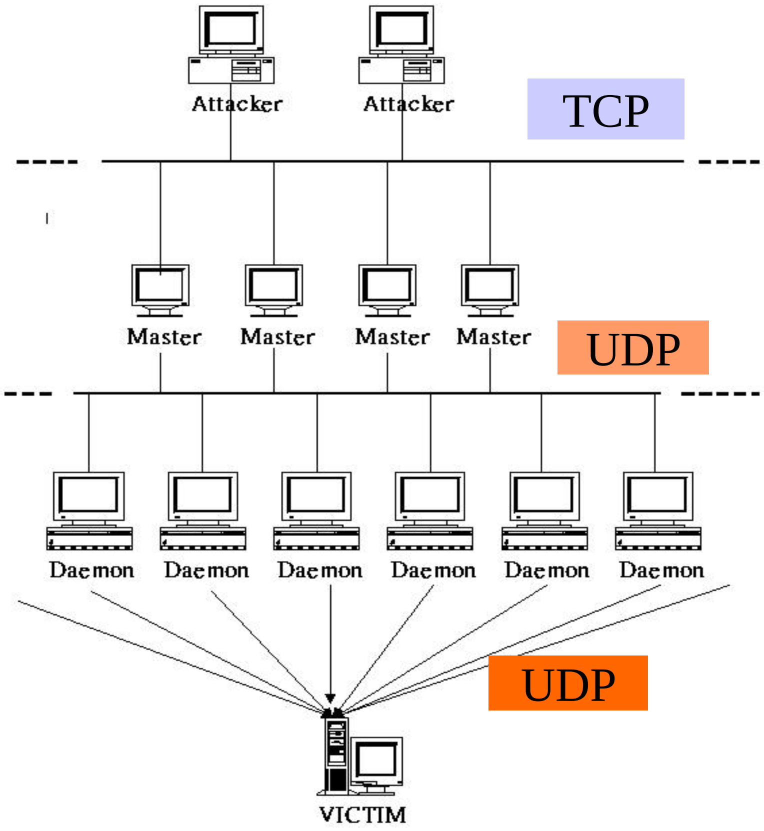
- A distributed tool used to launch coordinated UDP flood denial of service attacks from many sources.
- A trinoo network consists of a small number of servers, or *masters*, and a large number of clients, or *daemons*.

Trinoo (cont'd)

Scenario

- An intruder connects to a trinoo master and instructs that master to launch a denial of service attack against one or more IP addresses.
- The trinoo master communicates with the daemons giving instructions to attack one or more *IP addresses* for a specified *period of time*.

- intruder --> master; destination port 27665/tcp
- master ---> daemons; destination port 27444/udp
- daemons ---> UDP flood to target with randomized destination ports



DDoS Attacks (1/2)

- TFN:

  A $\xrightarrow{\text{TCP、UDP、ICMP}}$ M $\xrightarrow{\text{ICMP echo reply}}$ D =>

  UDP Flood
  SYN Flood
  Smurf
  ICMP Flood

- Stacheldraht:

  A $\xrightarrow[\text{(encrypted)}]{\text{TCP}}$ M $\xrightarrow{\text{TCP、ICMP}}$ D => UDP Flood、 SYN Flood
  ICMP Flood 、Smurf

DDoS Attacks (2/2)

- Shaft:

  $A \xrightarrow{\text{TCP}} M \xrightarrow{\text{UDP}} D$ => UDP Flood、 SYN Flood
  ICMP Flood 、Smurf

  - It has the ability to switch control master servers and ports in real time.

- TFN2K:

  $A \xrightarrow{\text{TCP、UDP、ICMP}} M \xrightarrow{\text{TCP、UDP、ICMP}} D$ =>Smurf

(encrypted by a key-based CAST-256)

UDP Flood
SYN Flood
ICMP Flood

Defenses Against Attacks

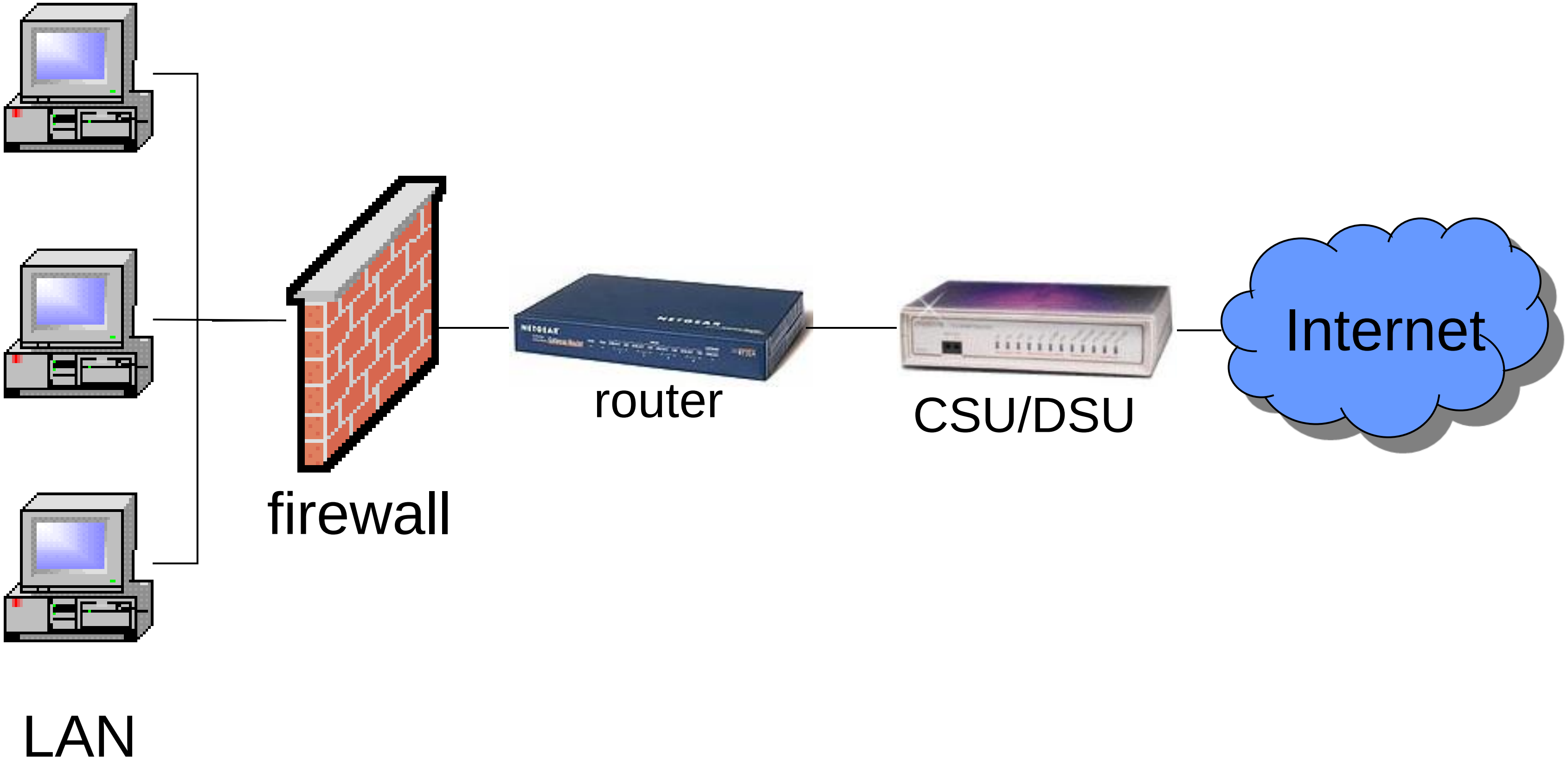
- Many observers have stated that there are currently *no* successful defenses against a fully distributed denial of service attack.

- Nevertheless, there are numerous safety measures to make the network more secure.

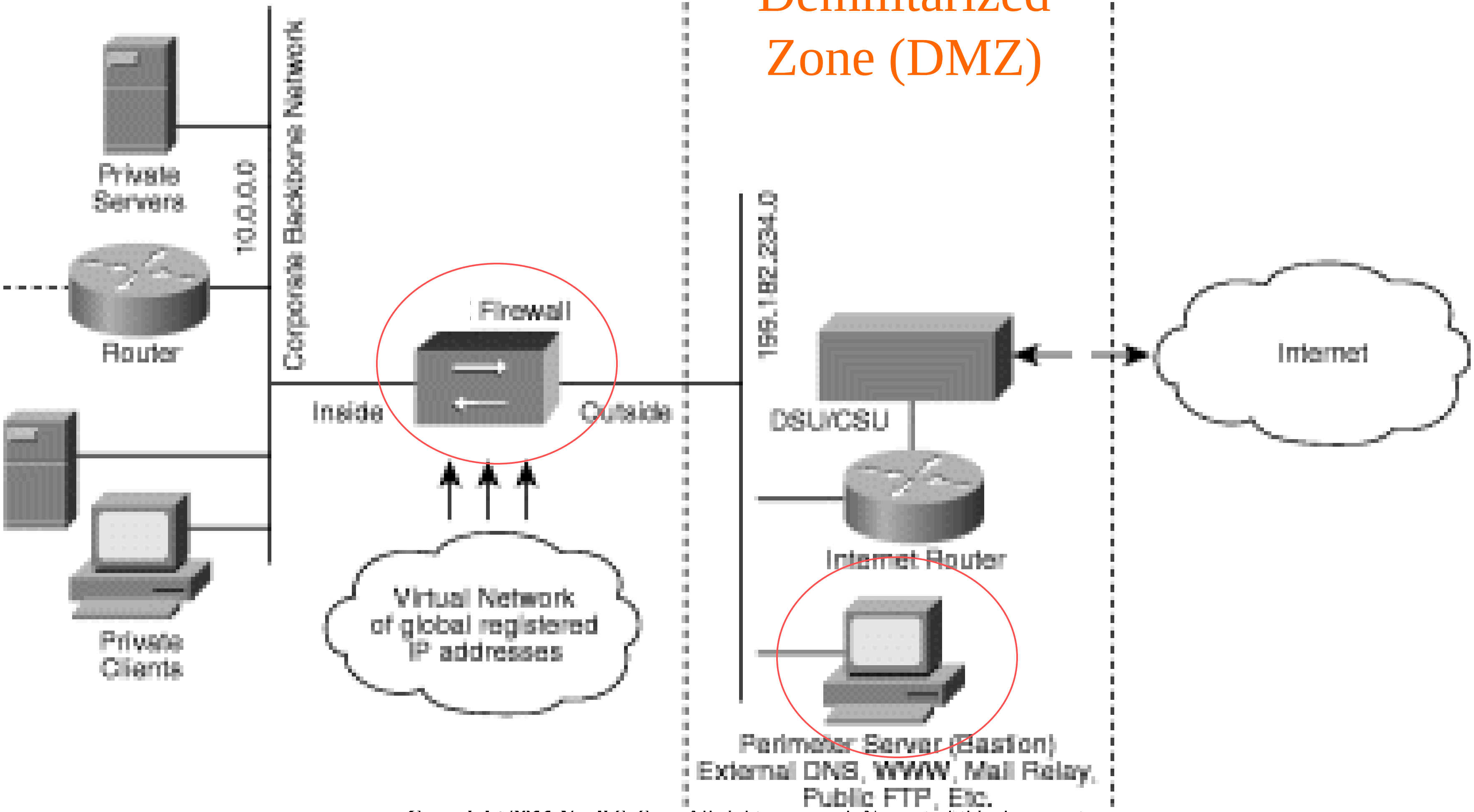
Defenses Against Attacks

- (1) Filtering Routers:
  - Filtering all packets entering and leaving the network protects the network from attacks.
  - The measure requires installing **ingress** and **egress** packet filters on **all** routers.
  - IP Traceback

Packet-Filtering in a Stand-alone Firewall



Two-tiered approach to network security



Demilitarized Zone (DMZ)

Private Servers
10.0.0.0
Corporate Backbone Network
Router
Firewall
Inside
Outside
Private Clients
Virtual Network of global registered IP addresses
199.182.234.0
DSU/CSU
Internet
Internet Router
Perimeter Server (Bastion) External DNS, WWW, Mail Relay, Public FTP, Etc.

Defenses Against Attacks

- (2) Disabling IP Broadcasts:
  - By disabling IP broadcasts, host computers can no longer be used as amplifiers in ICMP Flood and Smurf attack.
  - “Traceroute”
  - “Ping”

Defenses Against Attacks

- (3) Applying Security Patches:
  - Hosts must be updated with the latest security patches and techniques.
  - Example: Program bugs
    - Web program -- escape code.
    - Buffer overflow

Defenses Against Attacks

- (4) Disable Unused Services:
  - If network services are unneeded or unused, the service should be disable to prevent tampering and attacks.

Defenses Against Attacks

- (5) Performing Intrusion detection
  - Network *monitoring* is a very good preventive way of guarding against denial of service attacks.
  - By *monitoring* traffic patterns, a network can determine when it is under attack, and take the required step to defend itself.
  - Passive, off-line

Defenses Against Attacks

- (6) IETF's IP Security Authentication Header/ Encapsulating Security Payload (AH /ESP) protocols/algorithms to authentication and encrypt data packets.
  - Allows companies to create a virtual private network (VPN) across the Internet or any other packet network.

Conclusion

- Hackers/Intruders will keep hacking and intruding.
- Administrators should keep systems/networks working.
- Firewall serves as the first-line protection.
- Firewall products mainly differ in their performances

The end. ☺

Network Address Translation

- NAT, RFC 1631
- To alleviate the problem of IP address depletion
- Address reuse by private networks
  - To map the reusable IP addresses of the leaf domain to the globally unique ones required for communication with hosts on other networks.
- Many proxy server (firewall, router) provides NAT functionality.

Dynamic Address Allocation

- Mapping between local and global addresses is done dynamically.
- An Internet-bound packet sent by a host on the inside network follows default routes to the inside interface of the Firewall.
- Upon receipt of the outbound packet, the source address is extracted and compared to an internal table of existing translations.
- If the inside host's address does not appear in the translation table, a new entry is created for the host.

Dynamic Address Allocation (cont'd)

- A a globally unique IP number from the pool of available addresses is assigned to the host.
- The actual translation is accomplished by changing the source address of the packet to this "legal" address.
- The checksums are updated.
- After a user-configurable timeout period, during which there have been no translated packets for a particular address mapping, the entry is removed and the global address is freed for use by another inside host.

FTP Example

- Packet filters have two choices with regard to outbound FTP connections.
- Either leave the entire upper range (greater than 1023) of ports open which allows the file transfer session to take place over the dynamically allocated port, but exposes the internal network, or they can shut down the entire upper range of ports to secure the internal network which blocks other services.
- This trade-off between application support and security is not acceptable to users today.

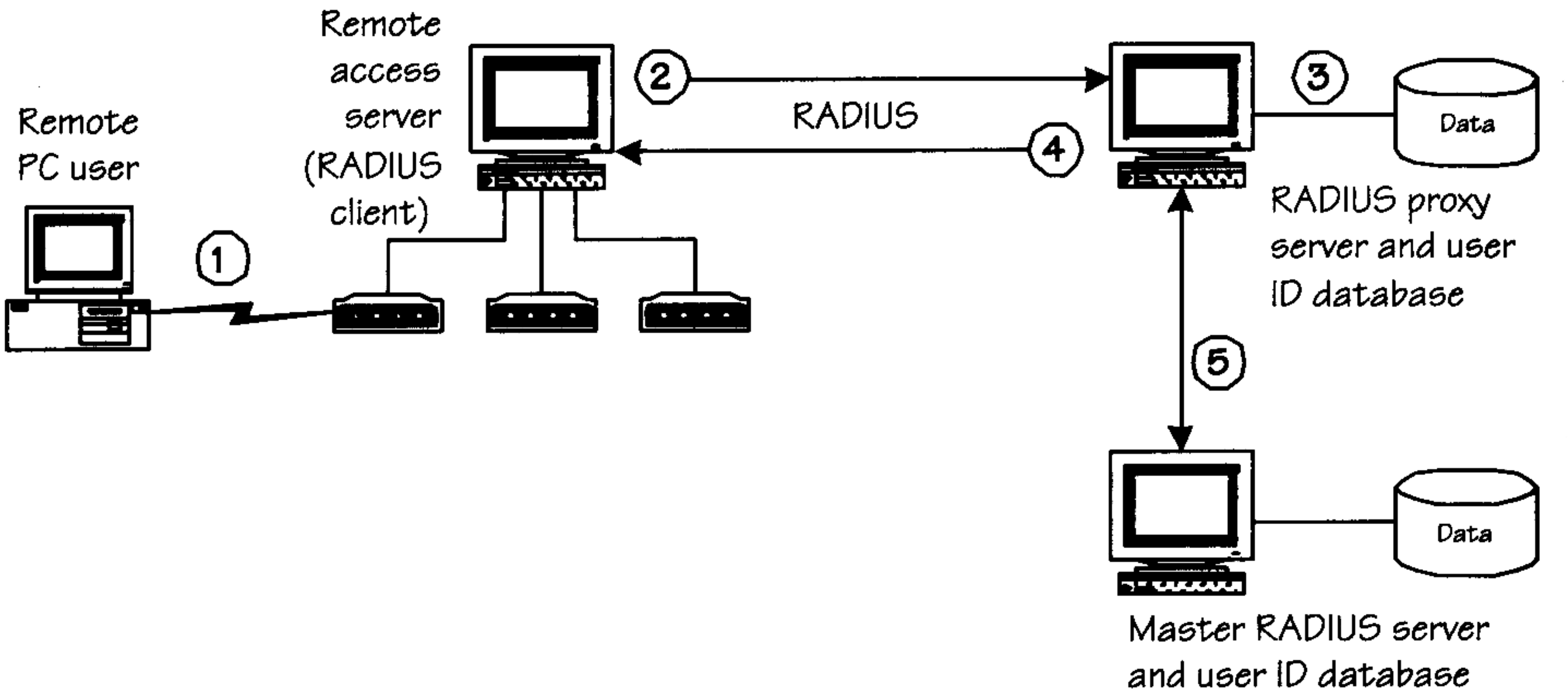
Remote Address Dial-In User Service (RADIUS)

RADIUS

- Remote Address Dial-In User Service (RADIUS)
- An access server authentication and accounting protocol
- RFC2058 (protocol specification) and RFC 2059 (accounting)
- Use UDP protocol
  - RADIUS implementation provides the functions of server availability, retransmission and timeouts

RADIUS (cont'd)

- A client/server protocol
  - client: Network Access Server (NAS)
  - server: a daemon process running on some UNIX or NT machine
- Server can act as a proxy to RADIUS server


Remote access server (RADIUS client)
Remote PC user
1
2
RADIUS
4
3
Data
RADIUS proxy server and user ID database
5
Data
Master RADIUS server and user ID database

1 User dials in to remote access server.

2 Using the RADIUS protocol, the remote access server, a RADIUS client, sends requests for authentication/ authorization to the proxy server.

3 The authentication server checks request against its user ID database.

4 Via RADIUS, the proxy server instructs the remote access server to grant (or deny) the user access.

5 The Master RADIUS server periodically updates the user database in the proxy server as needed.

Back

FIGURE 6.6 Interactions among a RADIUS server, proxy server, and clients.

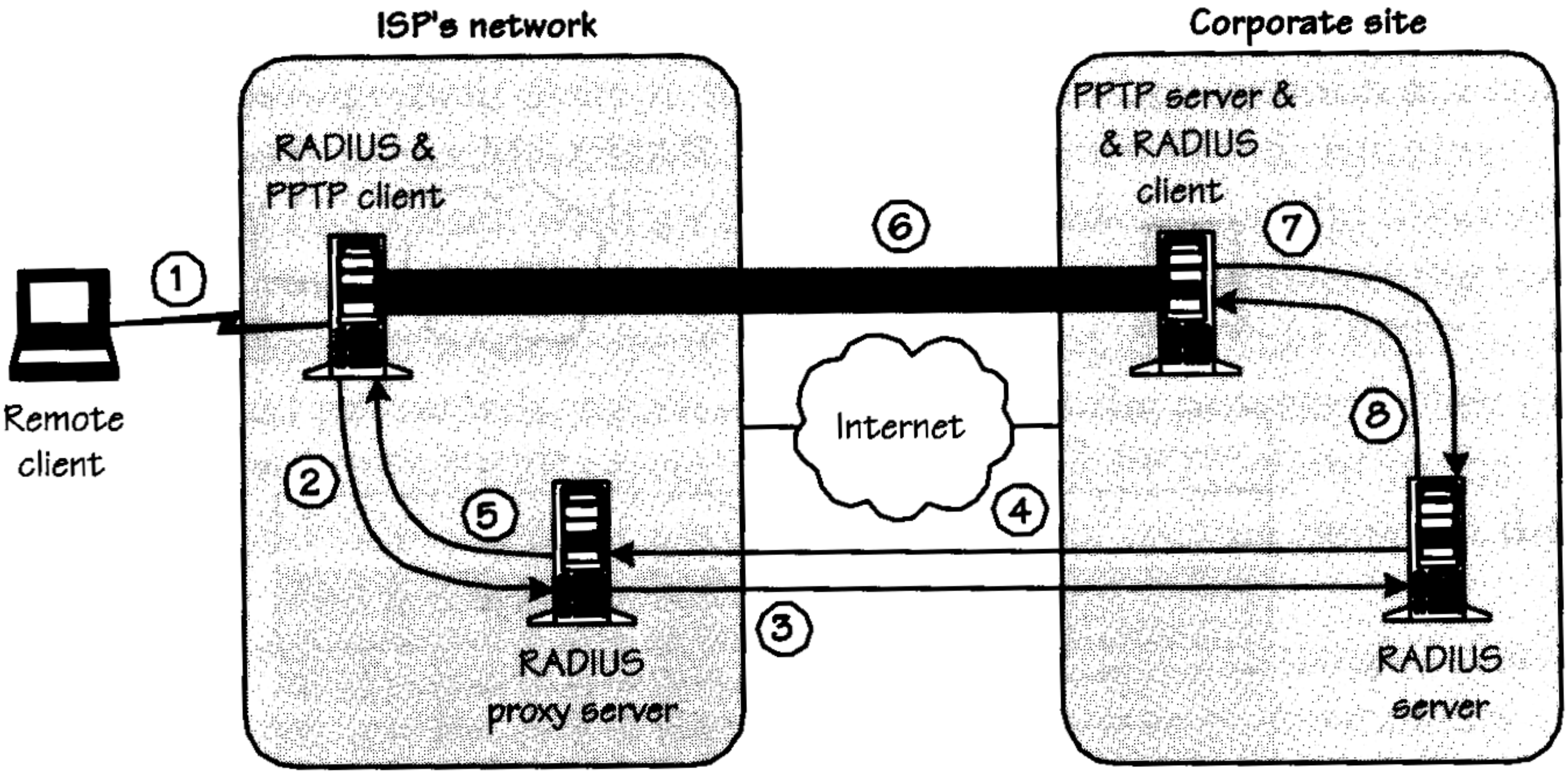


FIGURE 6.7 RADIUS authentication for dynamic tunnels.

Back

RADIUS Authentication

- Can support PPP PAP or CHAP, UNIX login and other mechanisms
- What is supported depends on what a vendor has implemented