Security in Digital Age

孫雅麗

國立臺灣大學

December 2016

All Roads to the Digital Future Lead Through Security

Gartner Symposium ITxpo 2014

Security Myth:
Perimeter Defense Will Protect My Applications

Host/App Attacks

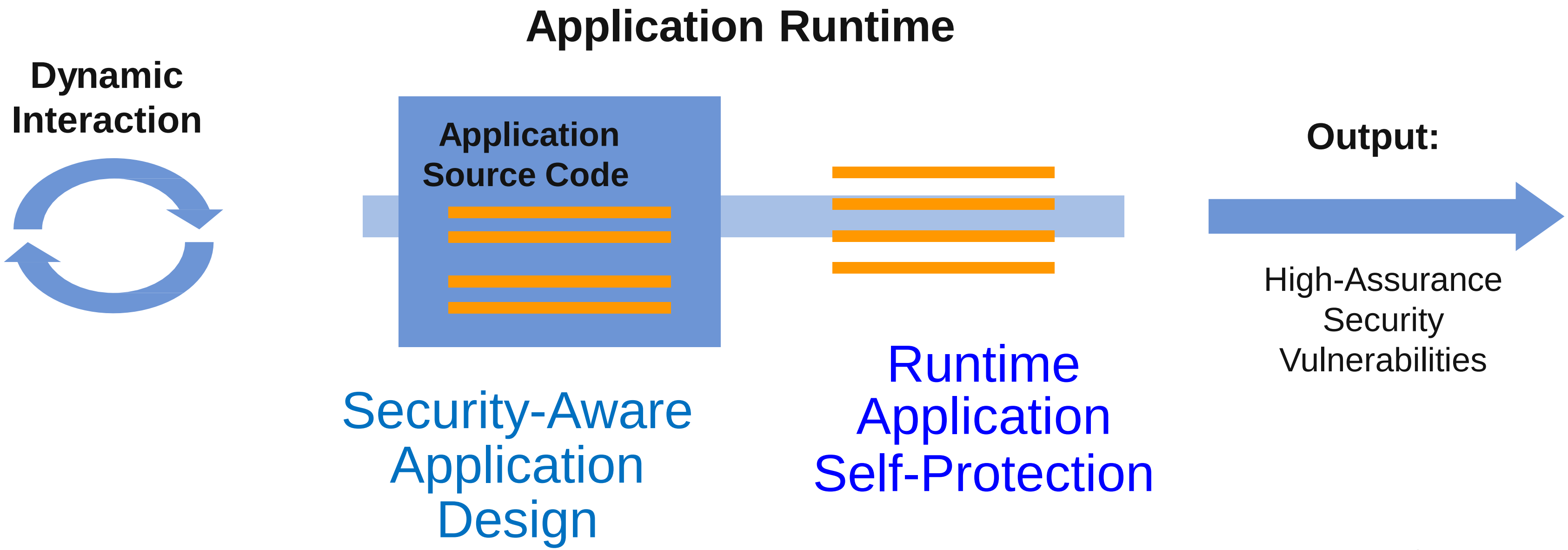
Mobile

Internet of Things

Insider Attacks

External Attacks

Enable Applications to Protect Themselves



Application Runtime

Dynamic Interaction

Application Source Code

Output:

High-Assurance Security Vulnerabilities

Security-Aware Application Design

Runtime Application Self-Protection

Ref: Gartner 2015

Advanced Persistent Threats (APT)

- Top Security Threats to the Organization in 2013-present

Advanced Persistent Threats （APT）

- Attacks on a *specific* organization’s **people**, **systems**, **vulnerabilities** and **data** -> targeted attacks.
- APTs have been growing rapidly.

「iCloud」遭駭客攻擊，大量名人私照洩露

美國多家媒體于當地時間2014年9月1日報導稱，有人非法入侵了多個iCloud帳號，將好萊塢女明星等名人的照片及視訊上傳到匿名方式的圖片論壇“4Chan”上。(4Chan上也有從iCloud之外收集圖片等內容) 隨後，這些內容經由“Twitter”及“Reddit”等其他SNS迅速擴散。

- Apple said that the company‘s core computer systems, which house all its users’ data, were **not** hacked！

How did the hacking happen? (1/2)

- Find My iPhone: the purpose is to protect user's data for a lost iOS device.
- "**Find My iPhone" app and iCloud** does *not* lock access after several unsuccessful attempts to log in.
- Target on certain celebrities and their iOS accounts
  - They lead public lives, hence answers to questions about their **past** are easily found on Wikipedia, Internet and elsewhere.

➢ *Their accounts were compromised.*

Find My iPhone: protect data for a lost iOS device

- If one misplaces his/her iPhone, the Find My iPhone app will let one use any iOS device to locate the missing device on a map, remotely lock it, play a sound, display a message, or erase all the data on it.
- Lost Mode - locks device with a passcode and displays a custom message and contact phone number on the Lock Screen.
- While in Lost Mode, the device can keep track of where it has been and report back.

How did the hacking happen? (2/2)

- Hackers forced their way into **celebrities' accounts** by **repeatedly guessing passwords** -- or **answers to their security questions**.
- It was a combination of *weak* passwords, *easy-to-guess security questions* and **a bug in Apple's photo backup service.**

Lessons

- It is a Targeted Attack
- This stresses the importance of secure passwords.
- Strong, hard-to-guess passwords are a must.
- Multi-factor authentication!

Multi-Factor Authentication (MFA)

- A user is only granted access after successfully presenting *multiple* separate pieces of evidence to authenticate himself/herself.
- Typically the following categories:
  - knowledge (**something only they know (secrete)**) (e.g., password, PIN, personal questions, etc.)
  - possession **(something they have)** (e.g., cellphone, computer, a USB stick with a secret token, a bank card, a key, etc.)
  - physical characteristic (**inherence**), (something they are, biometrics) (e.g., fingerprint, eye iris, voice, typing speed, pattern in key press intervals, etc.)

Two-factor Authentication: Mobile Phone

- Using mobile phones and smartphones to serve as "something that the user possesses".
- User authenticates himself/herself with a personal access code to the phone (i.e. something that only the individual user knows) plus a **one-time-valid, dynamic passcode** consisting of digits.
- If the new code is not entered within a specified time limit, the system automatically replaces it.
  - This ensures that no old, already used codes are left on mobile devices.
  - For added security, it is possible to specify how many incorrect entries are permitted before the system blocks access.
  - Safer to use than fixed (static) log-in information

Security Management

- “What ***resources*** are we trying to protect?”
  - data, files, storage device, computers, network, etc.
  - AAA (authentication, authorization, and accounting), identity management, access control, etc.
- “*Against* ***who***, must the computer systems be defended?
  - Attacker/hacker, (automatic) hacking software, insider, outsider, etc.

Security Management

Network Security

Host Security

Information Security

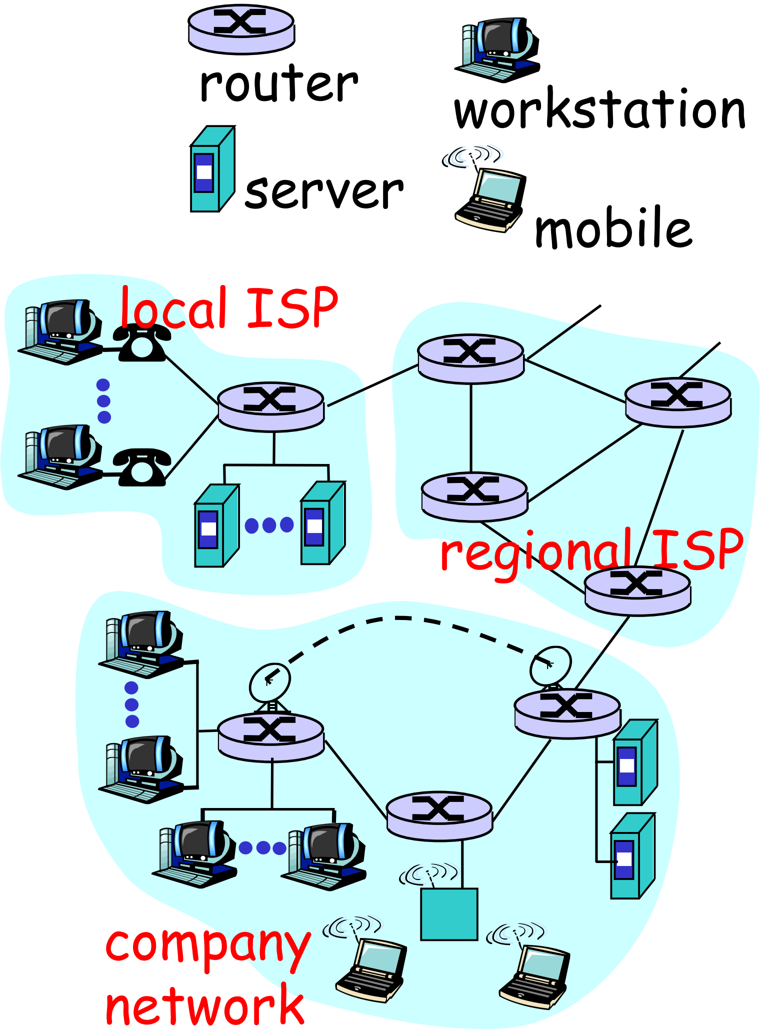
Topics to cover

- Firewall
- Network Intrusion Detection/Prevention System (IDS/IPS)
- IPsec, IP Traceback
- Host Intrusion Detection/Prevention System (HIDS)
- Web Security

Firewall

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

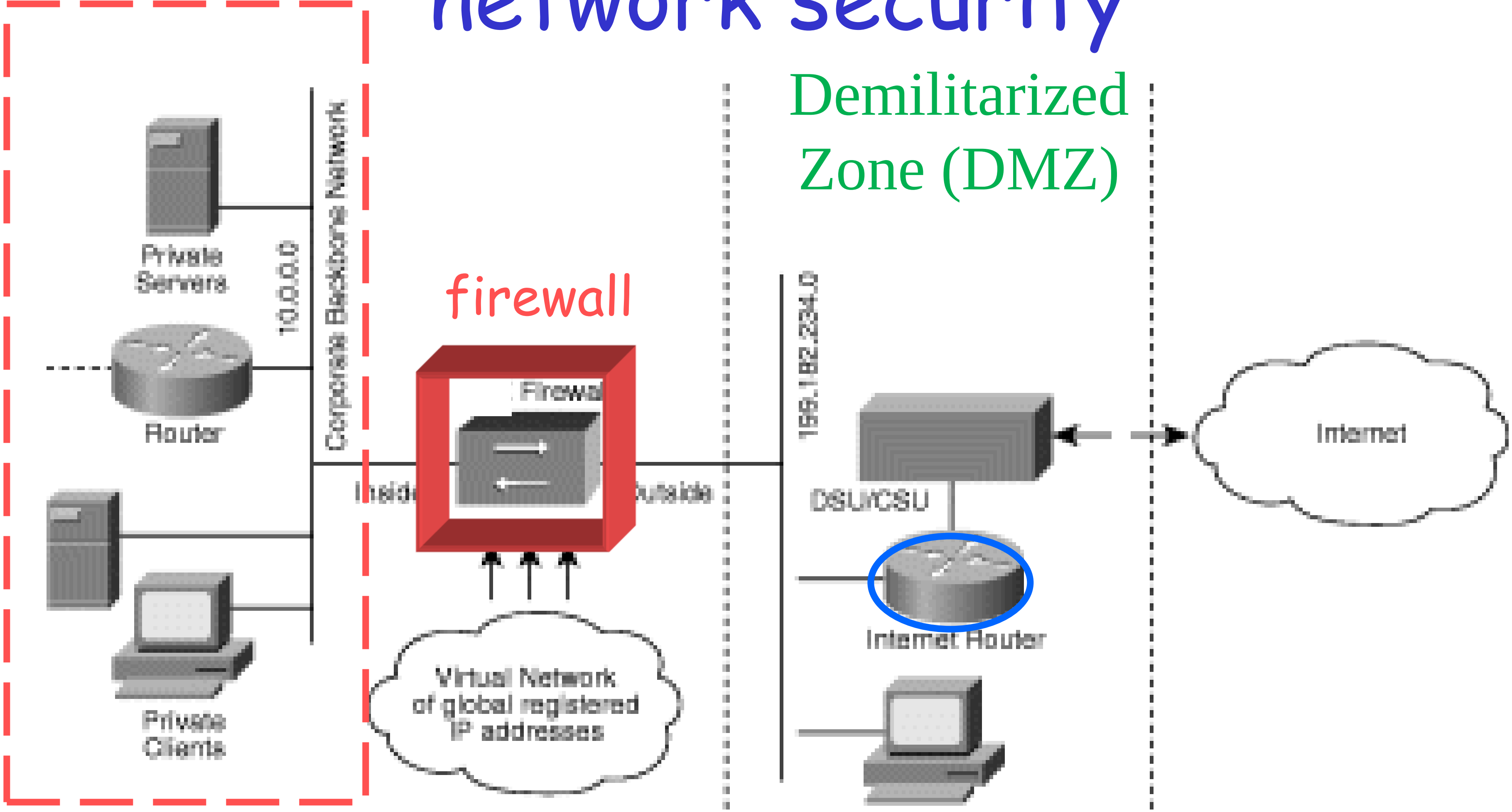
Security Risk Analysis

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

Typical Corporate Network 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.
    - Access control, DoS, smuggling, etc.
- 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.

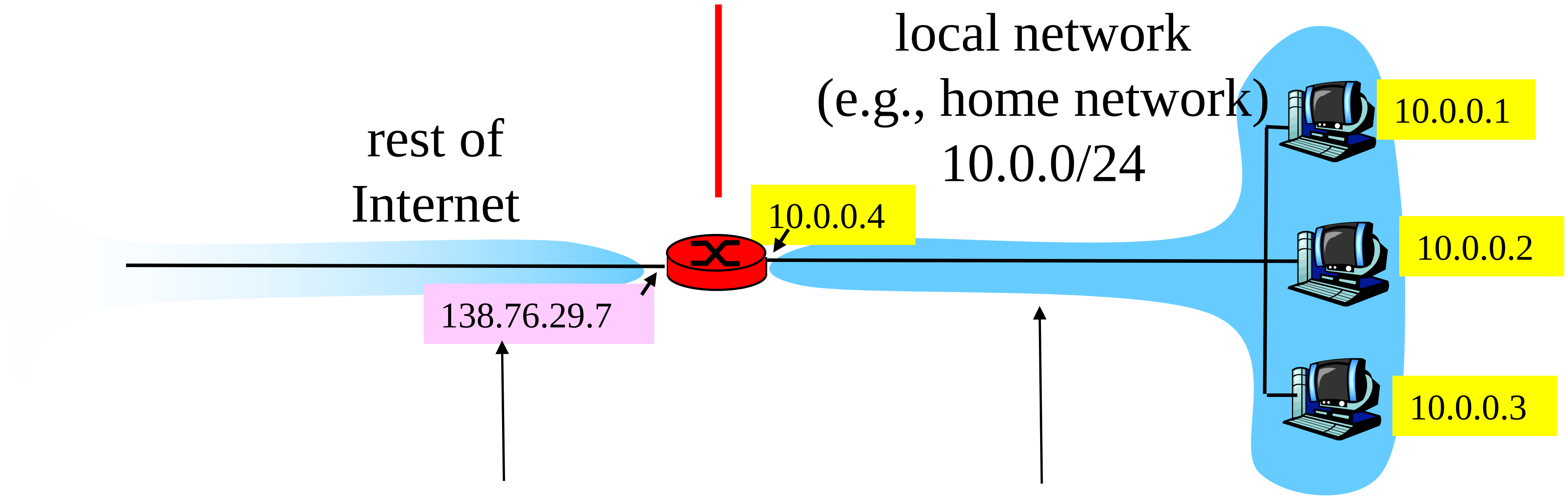
Firewall: other popular services

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

Security Information and Event Management (SIEM)

- Covers security information management (SIM) and security event management (SEM)
- SIEM system a) log security data and retention; b) perform log analysis; c) perform real-time correlation of events generated by network hardware and applications; d) generate security alerts; and e) report generation for compliance purposes.

NAT: Network Address Translation



rest of
Internet

local network
(e.g., home network)
10.0.0/24

10.0.0.4

138.76.29.7

10.0.0.1

10.0.0.2

10.0.0.3

All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

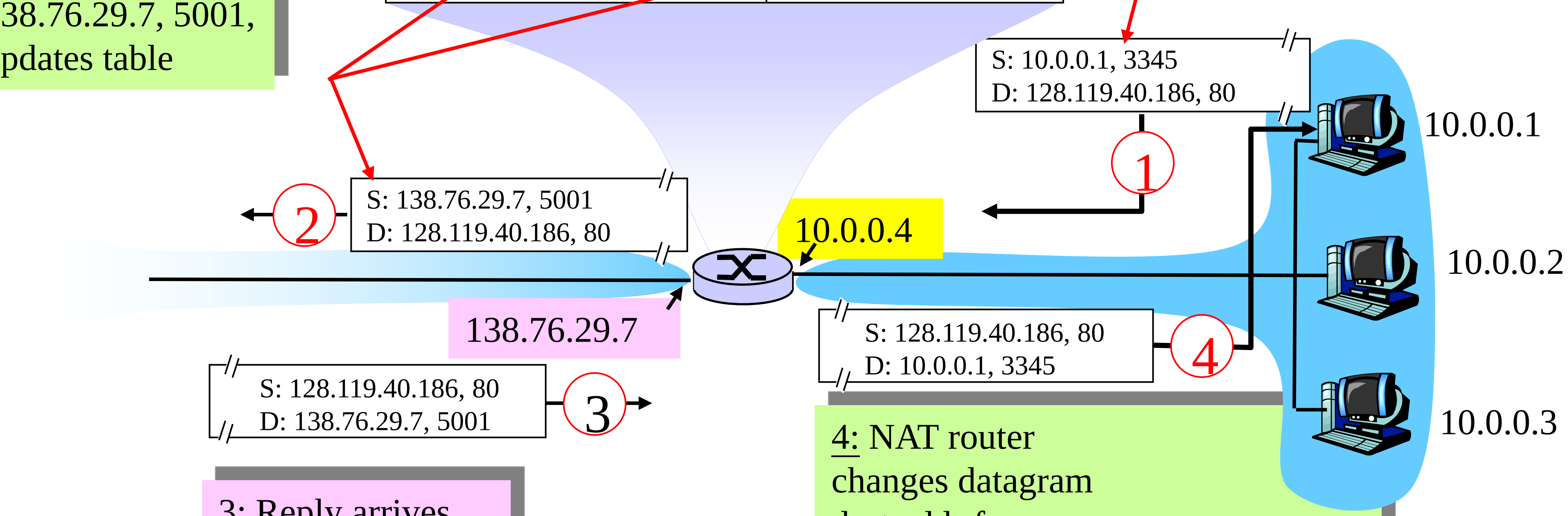
NAT: Network Address Translation

NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40, 80



S: 10.0.0.1, 3345
D: 128.119.40.186, 80

S: 138.76.29.7, 5001
D: 128.119.40.186, 80

S: 128.119.40.186, 80
D: 138.76.29.7, 5001

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

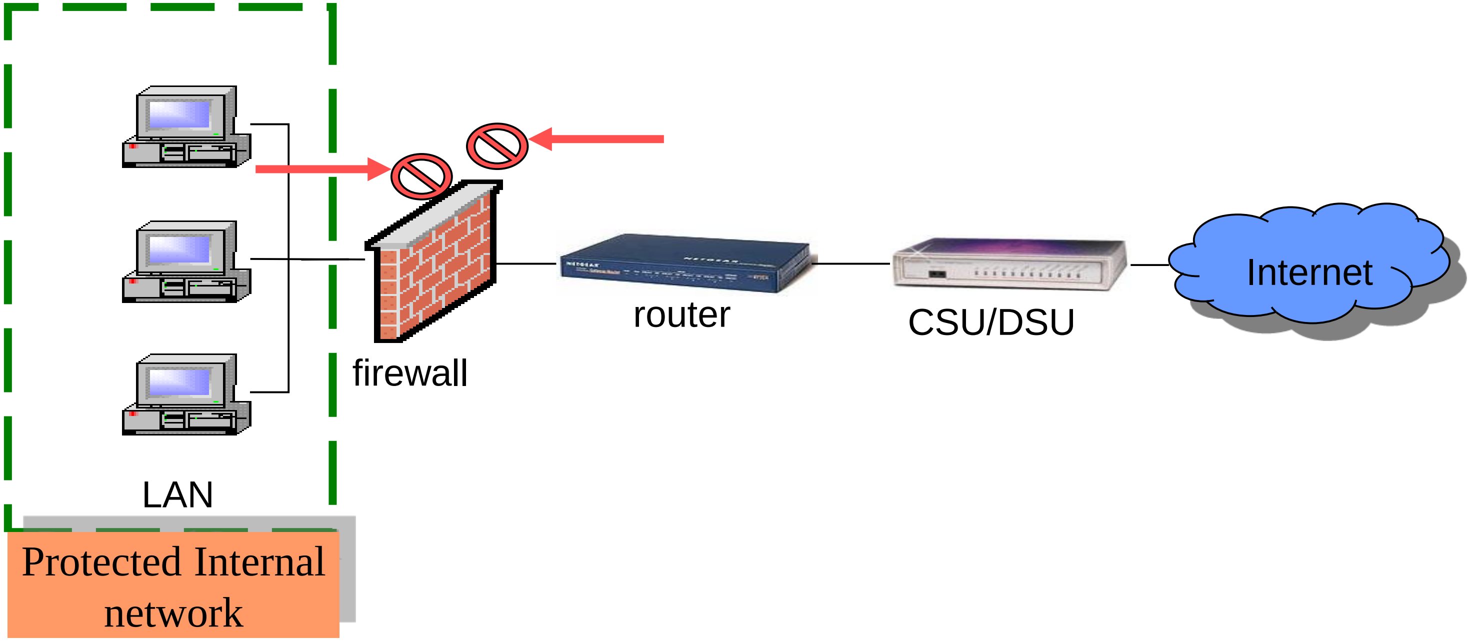
3: Reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

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)

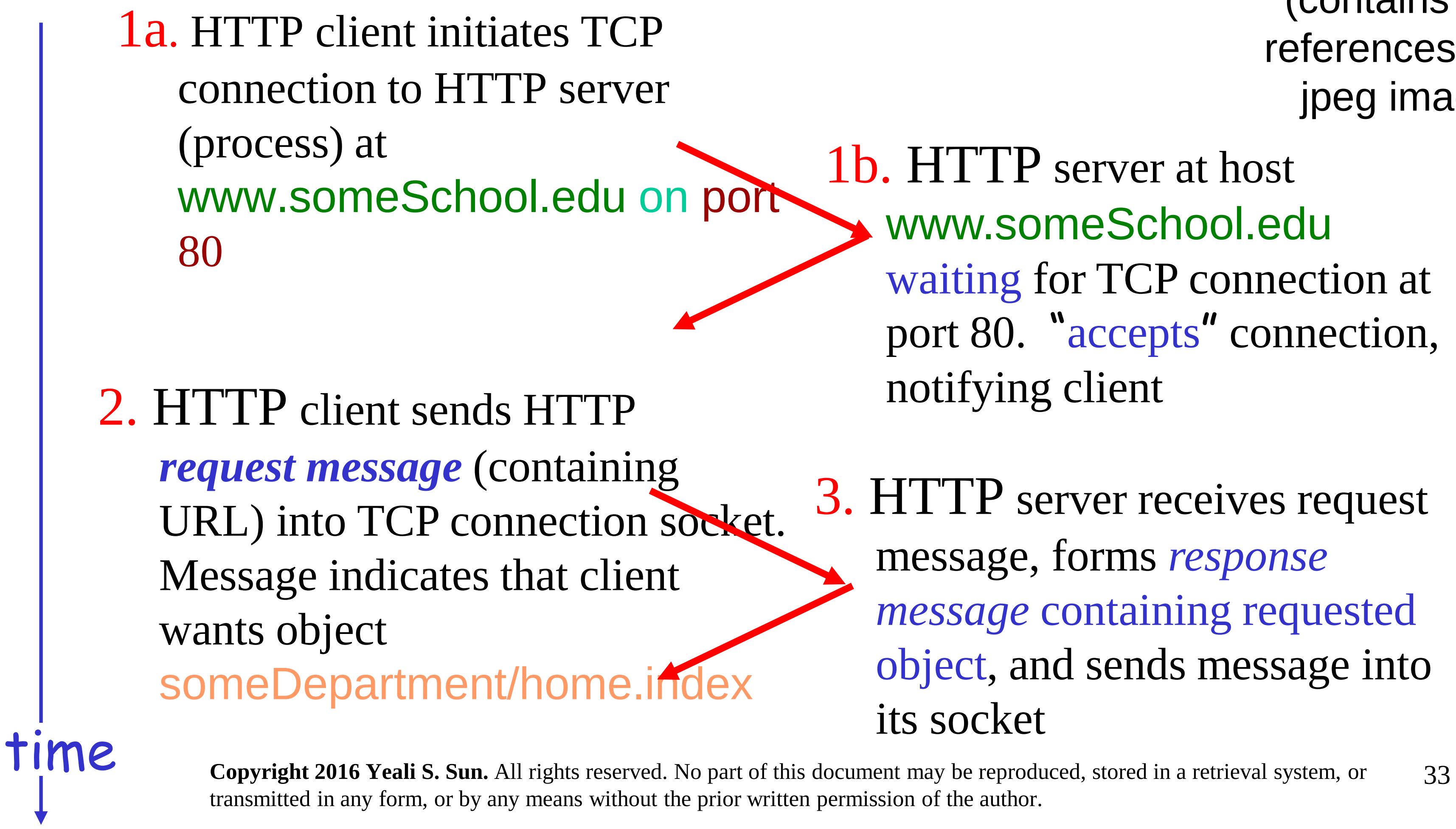
Content Filtering

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.

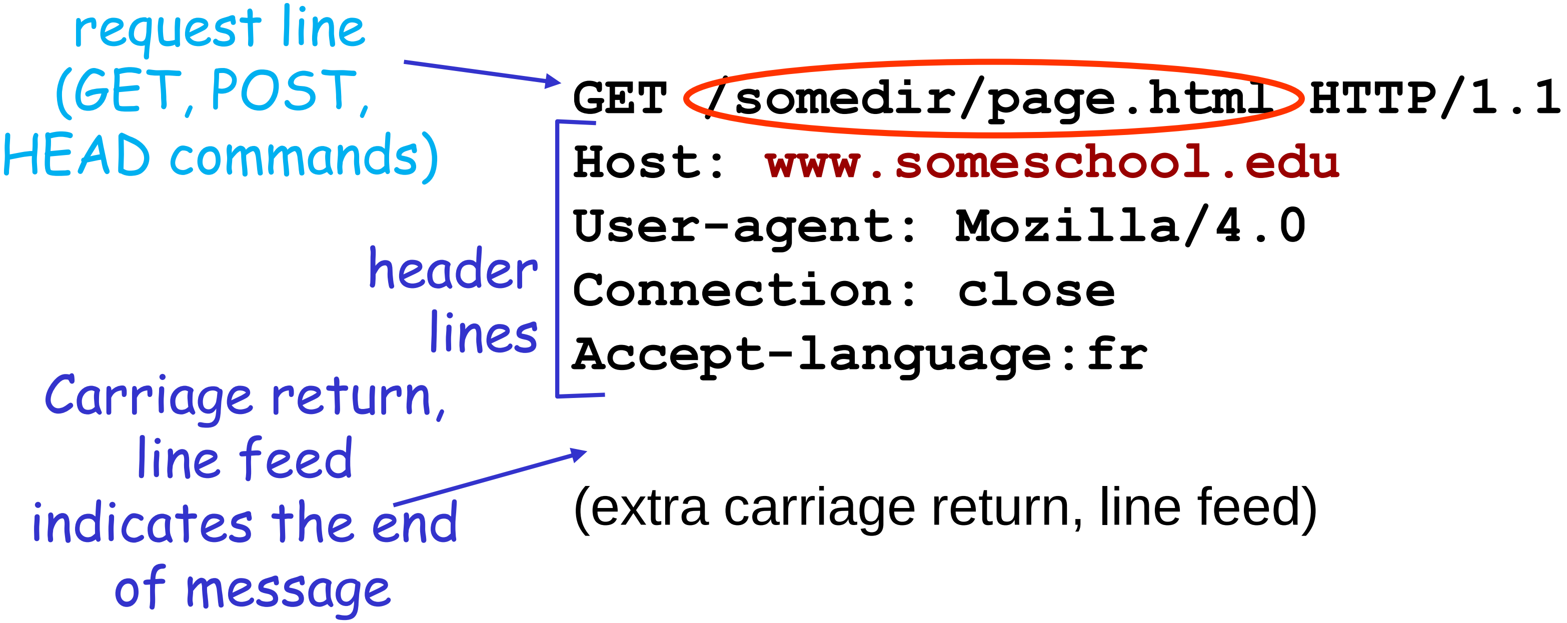
5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

time

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)

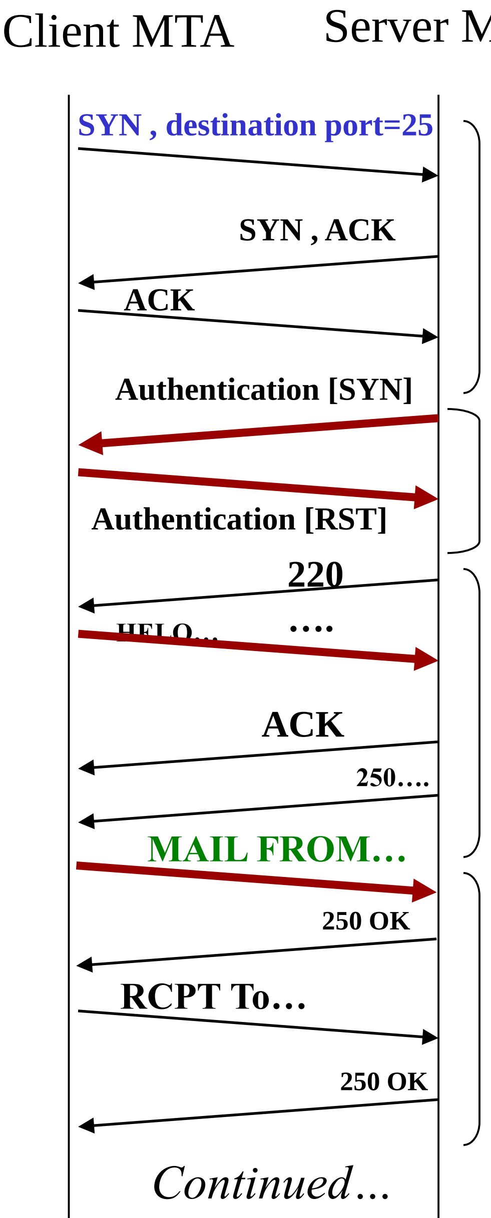
```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language:fr
```

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>

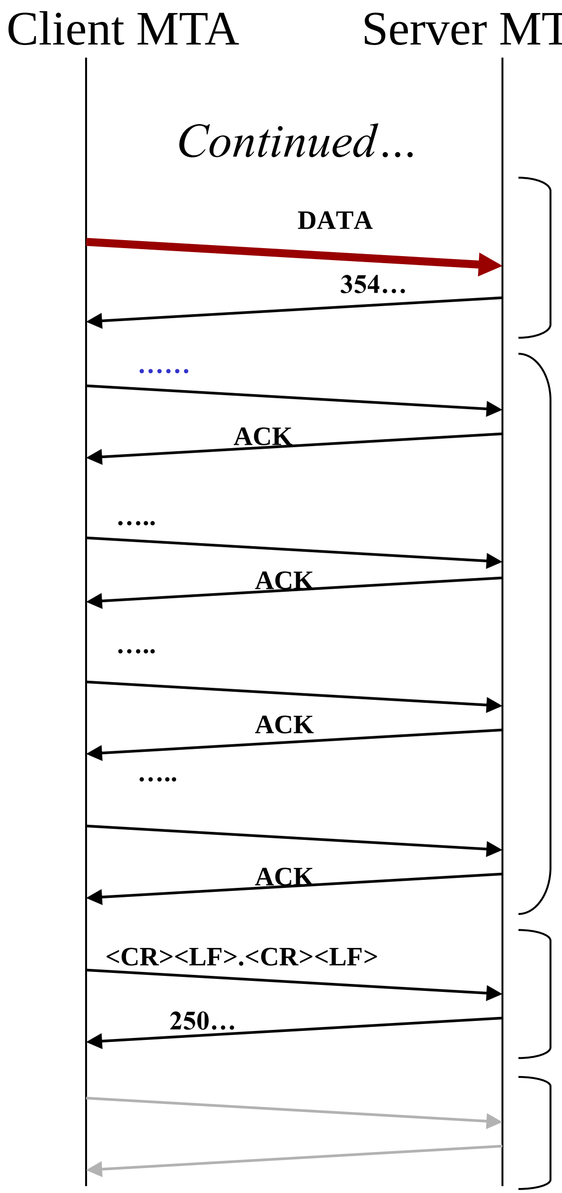
MAIL FROM: reversing path

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

RECP TO: forwarding path

<- receiver's mail account

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

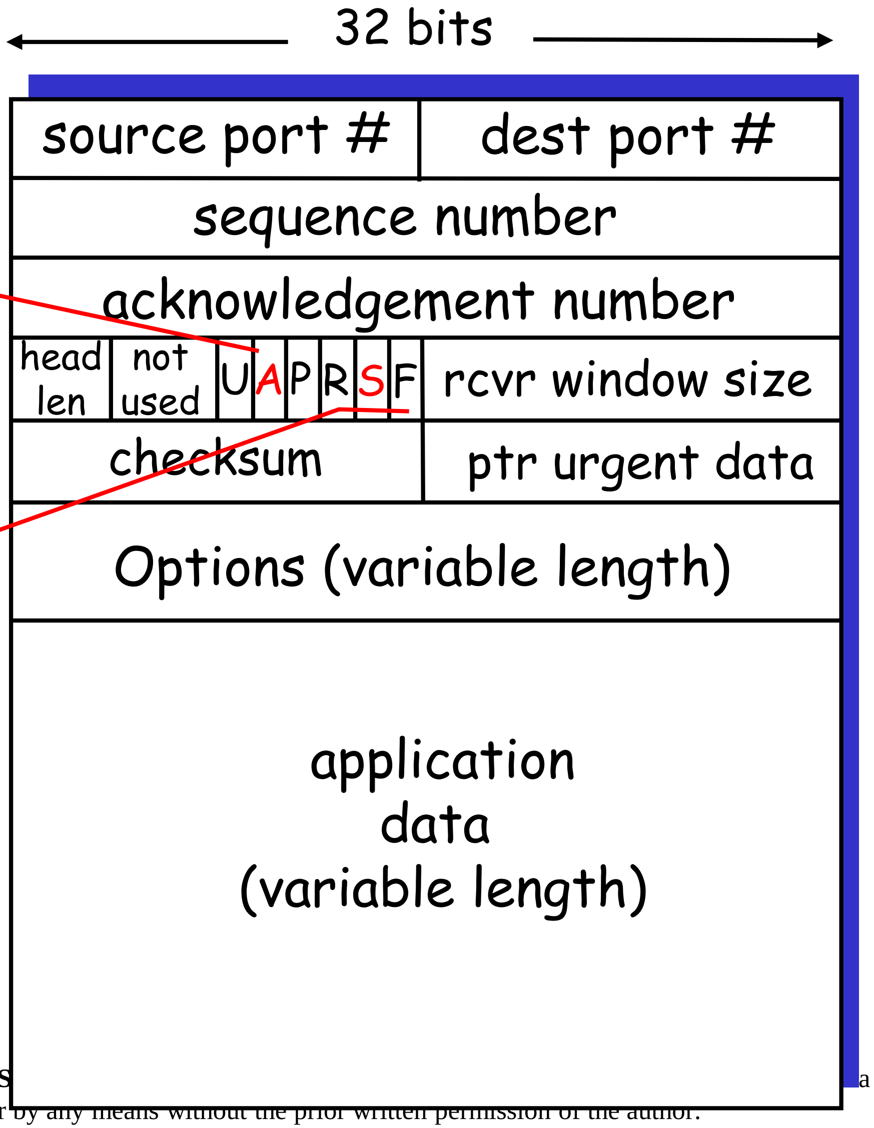
2 cases:

- 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 from outside (the organization).
  - 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)

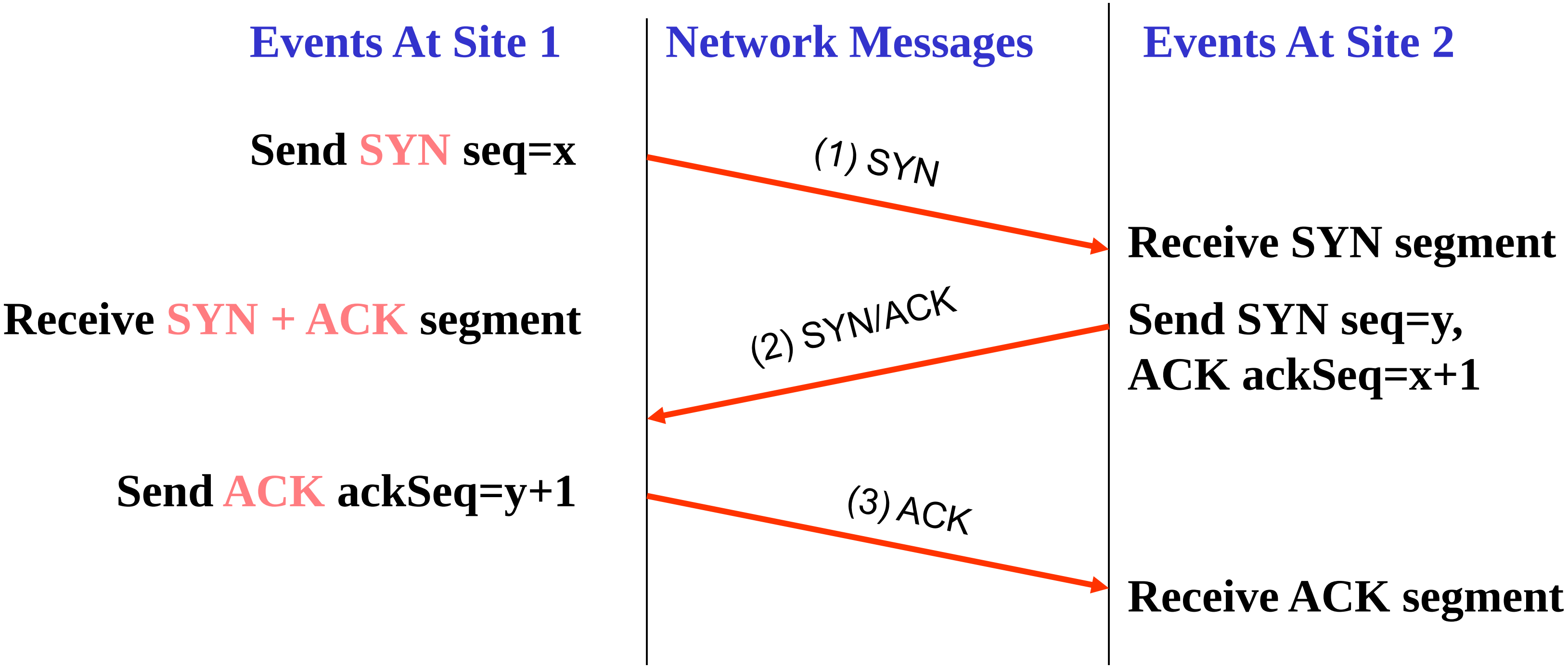
application data (variable length)

ACK

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

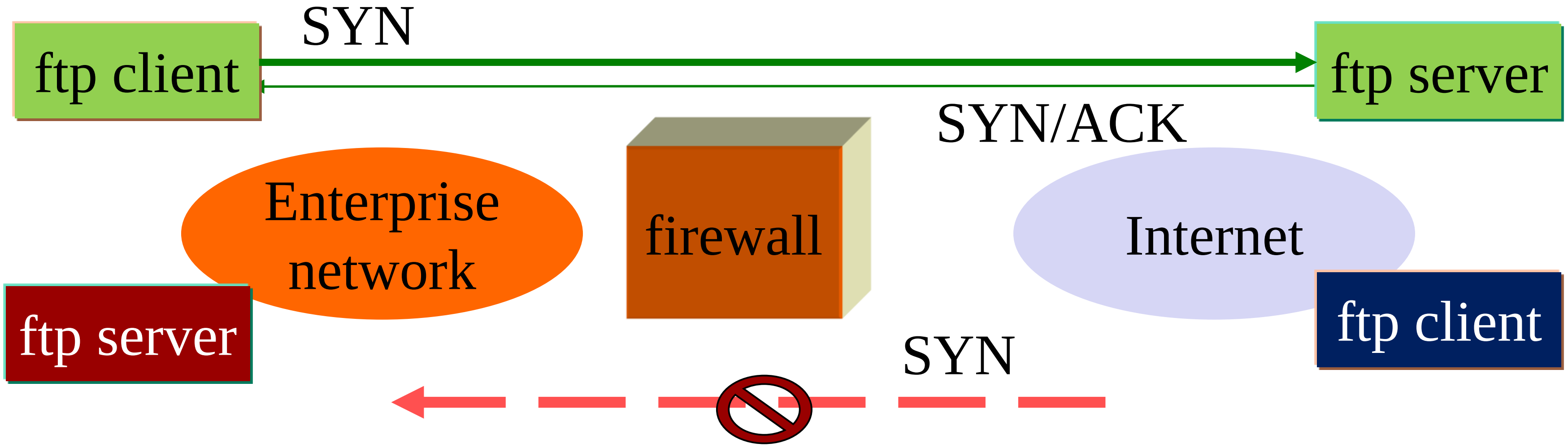
Example: Security control of TCP connections (2/3)

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

Firewall: The first-line defense

Service Characteristics

- Service Control – Deep Packet Inspection (DPI), content inspection
- Direction Control
- User Control – authentication, access control
- Behavior Control – data analysis

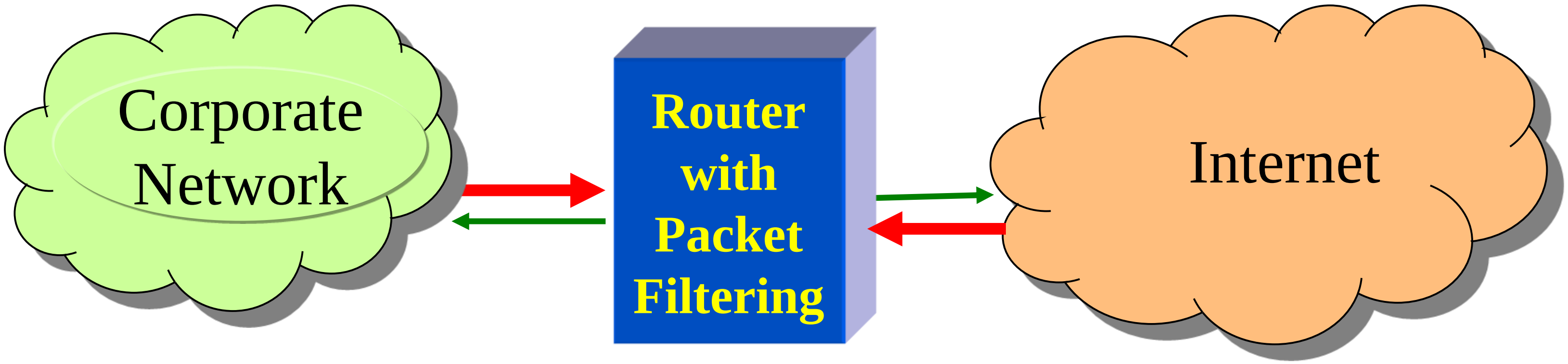
Limitations

- 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.

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; this however could be a security hole**
■ 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) is 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%.

Firewalls: Performance (1/3)

- All models may have *similar* functionalities and features.
- A great number of devices are **software applications** running on standard Microsoft windows or Linux platforms.
- 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.
  - For 100Mbps 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).

Firewalls: Performance (2/3)

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

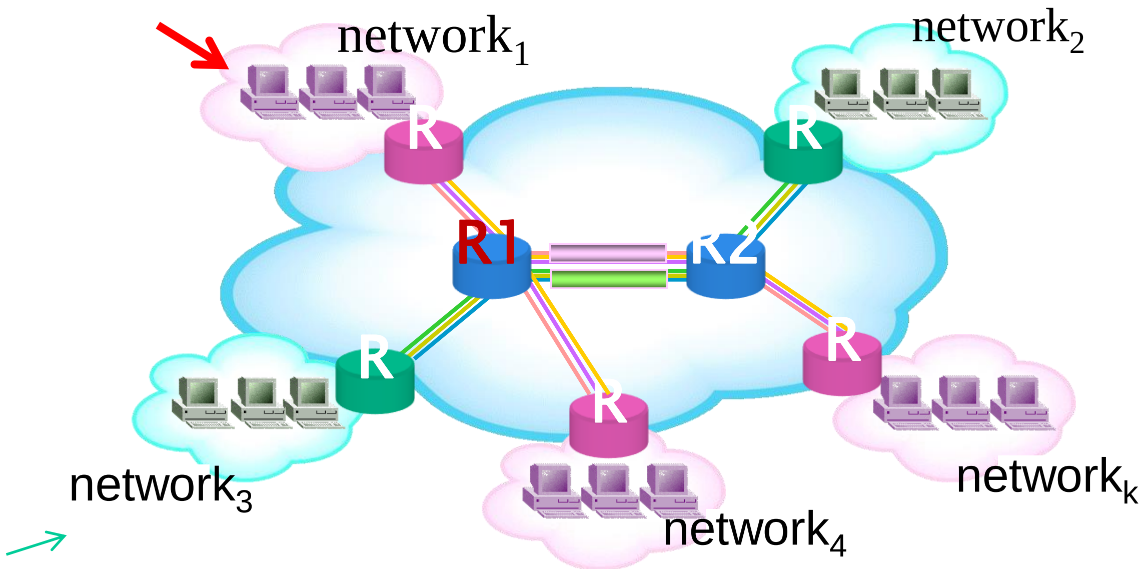
Firewalls: Performance (3/3)

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128,000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

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

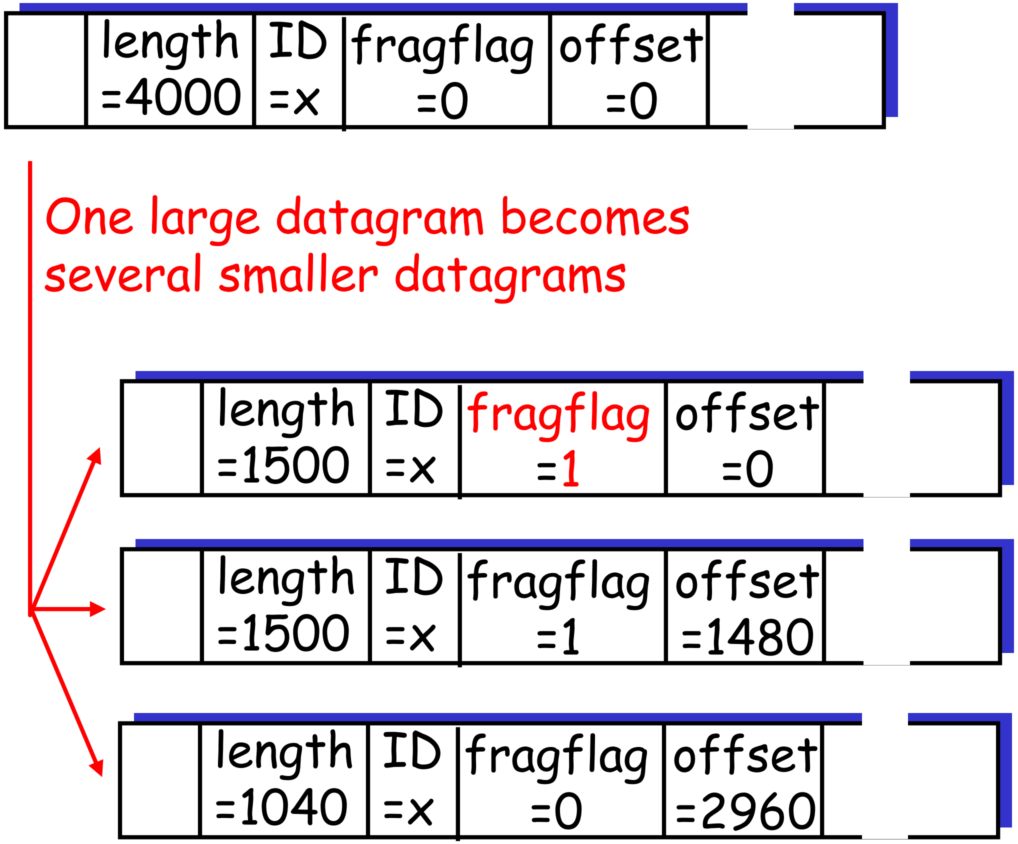
- The intruder uses the IP fragmentation option to create extremely small fragments and **force the TCP header information into a separate packet fragment**.
- To circumvent filtering rules that depend on TCP header.
- Only the first fragment is examined and the remaining passed through.
- Solution: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

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



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

One large datagram becomes
several smaller datagrams

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

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

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

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)
  - The second 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)!

Worm

Case: Slammer/Sapphire (1/2)

- On January 24, 2003, the W32.SQLExp.Worm (later named Slammer/Sapphire) was released into the wild.
- This worm exploited a stack-based **buffer overflow** vulnerability in Microsoft's SQL Server 2000 software (including MSDE 2000).
- **The speed** at which this worm **propagated** was novel and **scary**.
- The worm was released and within ten minutes it had compromised 90% of all vulnerable systems worldwide.
- Before this incident, worms of this type were merely theoretical, given serious consideration primarily in the academia.

Case: Slammer/Sapphire (2/2)

- It takes even the fastest vendors ***hours or days*** to produce a **signature** for systems.
- A vulnerable network was compromised in seconds, much too quickly for even the most diligently updated signature based or rule-based intrusion detection system.
- Known attacks vs. unknown (anomaly detection, baseline of what is normal.)

Rule-based Intrusion Detection

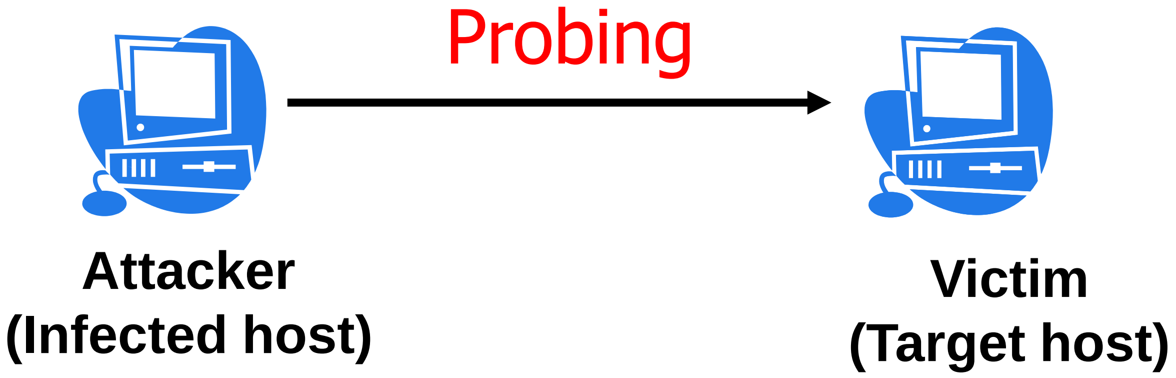
- Fact base + Rule base = Knowledge base
- Predicates (IF-THEN clauses)
- Forward chaining
- Experts (domain experts, subject matter experts)

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 (1/3)

- Probing (optional)
  - Select target hosts (victims) and send probe requests to check the existence of vulnerability



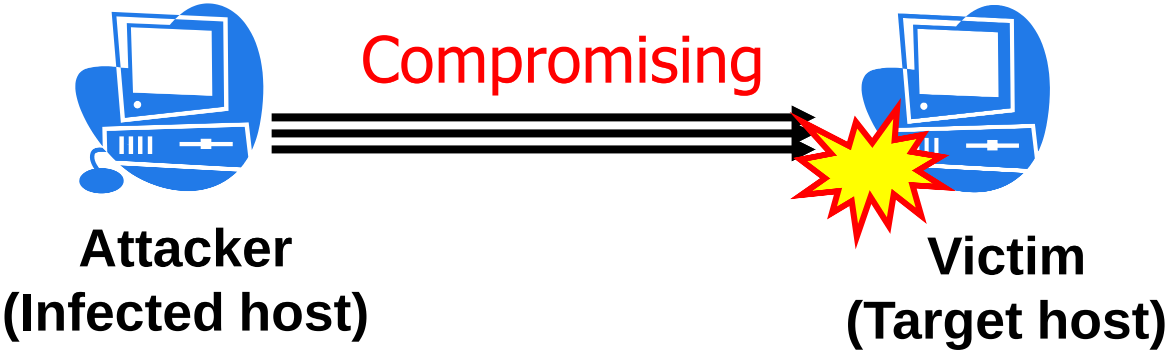
Probing

Attacker
(Infected host)

Victim
(Target host)

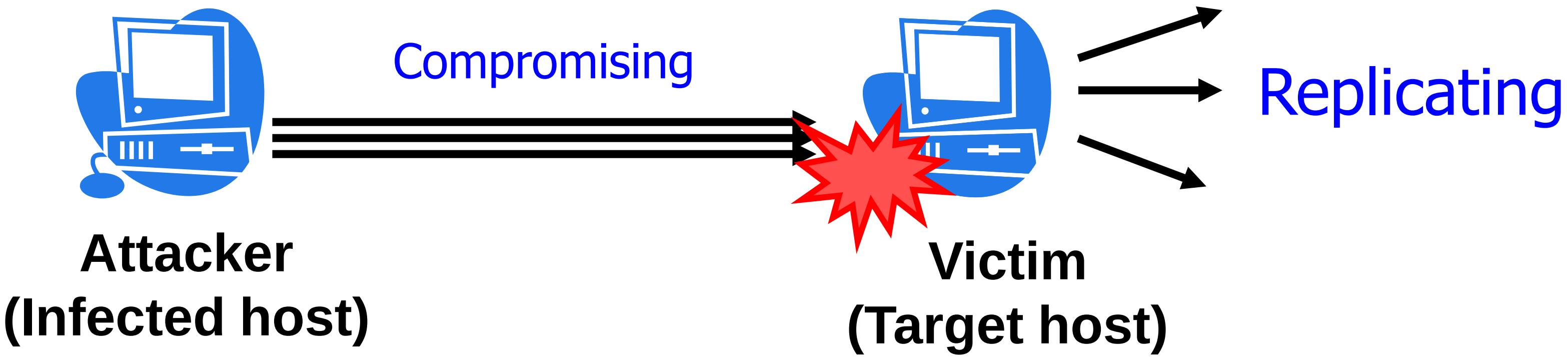
Worm Spreading: Stages (2/3)

- Compromising
  - Exploit the vulnerability and **gain execution privilege**
  - **Send** and execute the **worm code**
  - Cause certain damages



Worm Spreading: Stages (3/3)

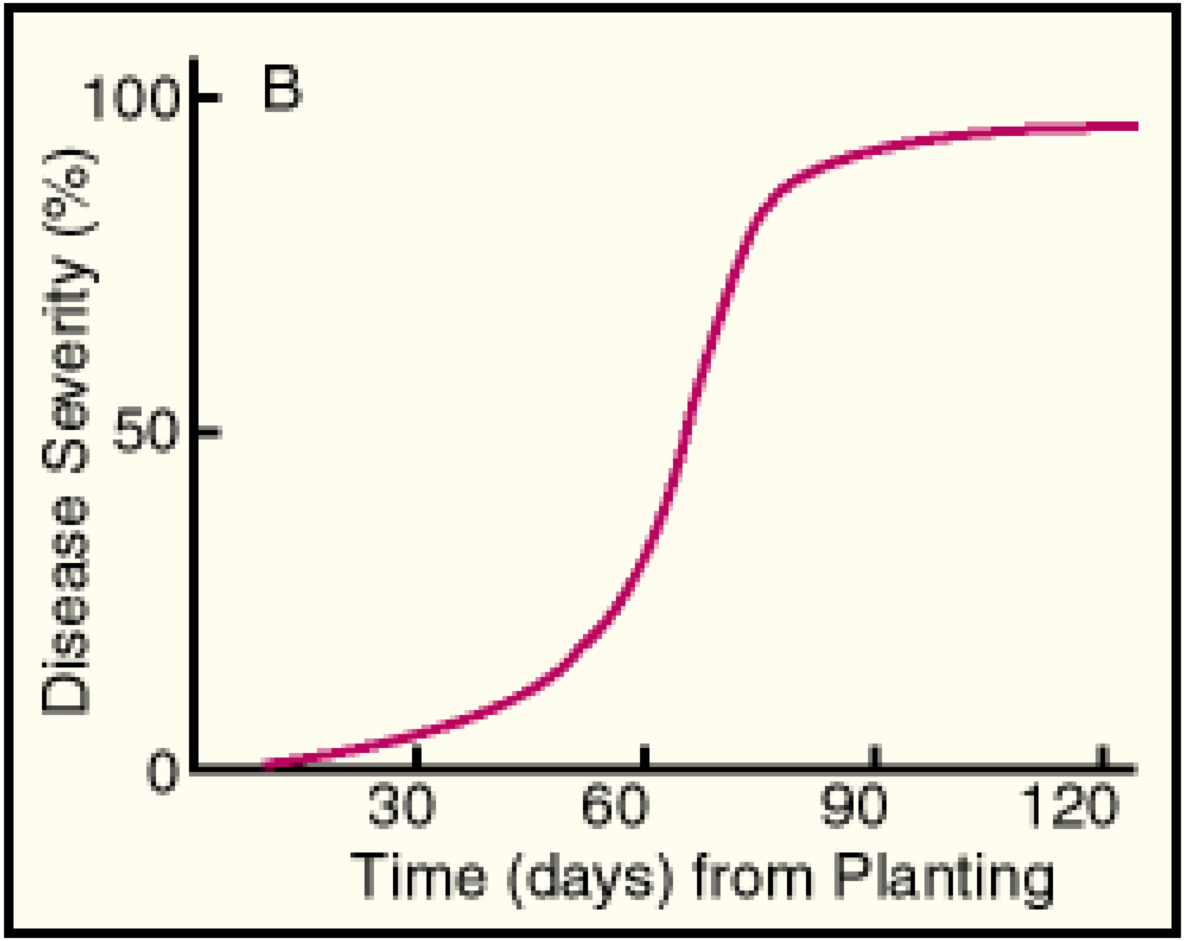
- Replicating
  - Replicate itself and continue spreading



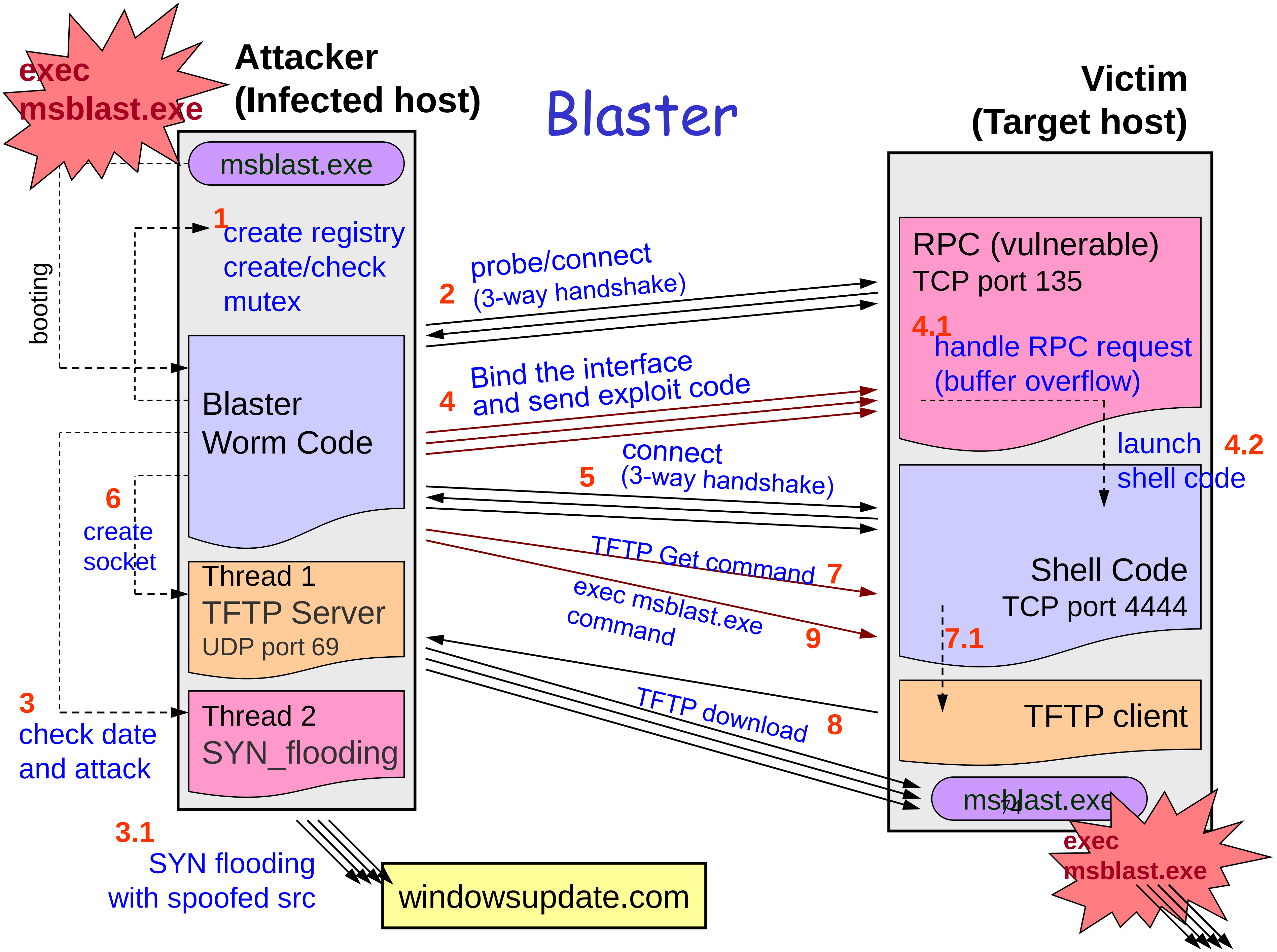
Worm Attack: Characteristics

- 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

Epidemic Disease Spreading



Blaster



Attacker
(Infected host)

exec
msblast.exe

booting

msblast.exe

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 4.2
shell code

Shell Code
TCP port 4444

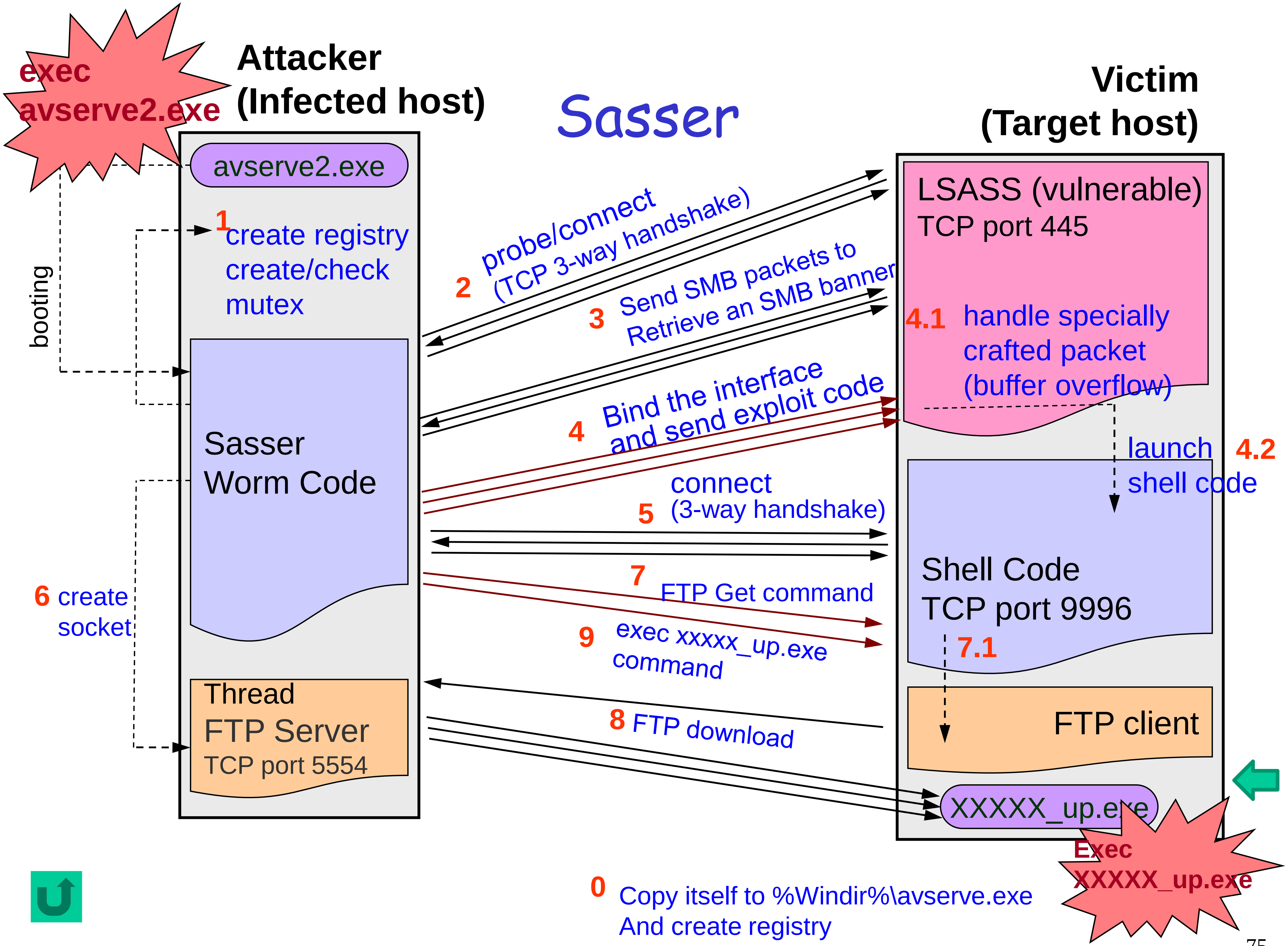
7.1

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

4.1 handle specially crafted packet (buffer overflow)

launch shell code 4.2

Shell Code TCP port 9996

7.1

FTP client

XXXXX_up.exe

Exec XXXXX_up.exe

2 probe/connect (TCP 3-way handshake)

3 Send SMB packets to Retrieve an SMB banner

4 Bind the interface and send exploit code

5 connect (3-way handshake)

7 FTP Get command

9 exec xxxxx_up.exe command

8 FTP download

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

Rapid Epidemic Infection

- So … what is the solution to a worm that doubles its infection rate every 8.5 seconds?
- Behavior-based anomaly detection.
- Benign (normal) vs. anomalous

Network Intrusion Detection/Prevention Systems (IDS/IPS)

Network Intrusion Detection Systems (IDS)

- Network intrusion detection systems (IDS) attempt to detect and report any malicious activity or policy violations, or whether a **network** has been compromised.
- This is done by monitoring and analyzing network traffic or activities.
- Deep packet inspection, stateful inspection, anomaly detection (deviations from normative behavior).
- Use attack "signatures" (rules) to identify or detect attacks in networks, e.g., port number in packet header, specific byte sequence in payload of a series of packets, etc.

Network IDS: attack signature generation

When an attack is detected, typically it takes the following steps to come up with a signature.

- Phase 1: record and analyze the attack packets
- Phase 2: generate the signature
- Phase 3: distribute the new signature
- Phase 4: Network operators implement the new rule for the network IDS system

✓ Zero-day attacks

Stateful Inspection (1/3)

1. Intercept packets at the **network layer**.
2. Examine individual packets from **all** communication **layers** and extract relevant data.
3. Analyze data to *derive* **communication** state and **application-derived** state and **context info.**

Stateful Inspection (2/3)

- Communication information from all seven layers in the packet
- Communication state information (context)
  - derived from *past* communications **and applications**.
  - e.g., save the outgoing PORT command of an FTP session; used to verify an incoming FTP data connection.
  - 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 (3/3)

- The system maintains state information in dynamic state tables for evaluating subsequent connection attempts.
- This provides cumulative data against which ***subsequent* communication attempts** can be evaluated.

Examples (1/2)

- Connection attempt from a reserved IP address.
  - Check the source address field in an IP header.
- Packet with an illegal TCP flag combination.
  - Compare the flags set in a TCP header against known good or bad flag combinations.
- DNS buffer overflow attempt contained in the payload of a query.
  - Parse DNS fields and check the length of each of them
  - Look for exploit shellcode sequences in the payload

Examples (2/2)

- Denial of service attack on a POP3 server caused by issuing the same command thousands of times.
  - Keep track of the number of times the command is issued if it exceeds a certain threshold.
- File access attack on an FTP server by issuing file and directory commands to it without first logging in.
  - Use a state-tracking signature to monitor FTP traffic for a successful login and would alert if certain commands were issued before the user had authenticated properly.

Behavior-based Network IDS: "normal" vs. anomaly (1/3)

- Determines "normal" network activity and then all traffic that falls outside the scope of normal is flagged as anomalous (not normal)!

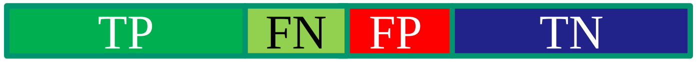
1. Learn network traffic patterns
   - assuming network traffic patterns remain constant,
   - the longer the system remains constant the more accurate!
2. Employee complex statistical or machine learning algorithms to derive the "normal" behavior model

Behavior-based NIDS: “normal” vs. anomaly (2/3)

Learn and distinguish normal from anomalous network activity

3. Detection
   - Look for anomalies in the established normal network traffic patterns.
   - All packets are given an anomaly score (indicating the degree of irregularity for the specific event)
   - If the anomaly score is higher than a certain threshold, generate an alert

Evaluation Metrics



- False positive – 誤判 （indicating a given condition has been fulfilled, when it actually has not been fulfilled)
- False negative －漏判 (indicating that a condition failed, while it actually was successful)
- True positive rate measures the *proportion* of actual positives which are correctly identified as such.

  TPR = TP/P = TP/(TP+FN)

- True negative rate measures the proportion of negatives which are correctly identified as such.

  SPC = TN/N = TN/(FP+TN)

Behavior-based NIDS: summary (3/3)

- Select a target network
- Profiling traffic (“normative” behavior)
- Measure(s) (a vector of features, e.g., statistics)
- Deductive process (rules)
- False positive and false negative

- Good for unknown attacks!

Behavior-based Detection: advantages (1/2)

- ✓ Can detect a previously unseen worm, virus, or Denial of Service (DOS) attack.
- ✓ Can alert based on the presence of the unusual activity
- ✓ Can detect "low and slow" attacks, characterized by their lengthy duration (possibly months at a time), precision, and methodical execution.
  - ■ Usually these attacks are intended to enumerate the network or gather information about a specific system.
  - ■ The detection system will note that this is anomalous traffic and alert on the event.

Behavior-based Detection: limitation (2/2)

- “The only thing "normal" about a network is the fact that it is constantly changing.”
    - Most networks are extremely diverse in terms of protocols, services, and usage times.
- Suffer from the ability to be "taught" by intruders.
    - e.g., an attacker could use a program like Nmap and send numerous SYN-scans at the network.
- Demand highly skilled staff in the art of packet analysis (expert systems, automation)

Two approaches to defeating intrusion detection

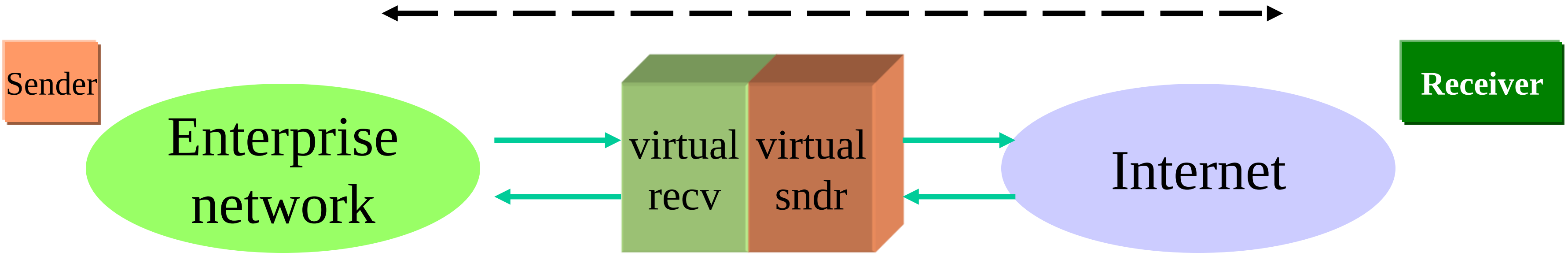
- Intrusion detection systems are defeated either through **attack** or **evasion**.
- Attacking a Network IDS involves tampering with the Network IDS or components it trusts to prevent it from detecting or reporting malicious activity.
- Evading a network IDS is achieved by disguising malicious activity so that the IDS fails to recognize it.
  - APT (get around)

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.

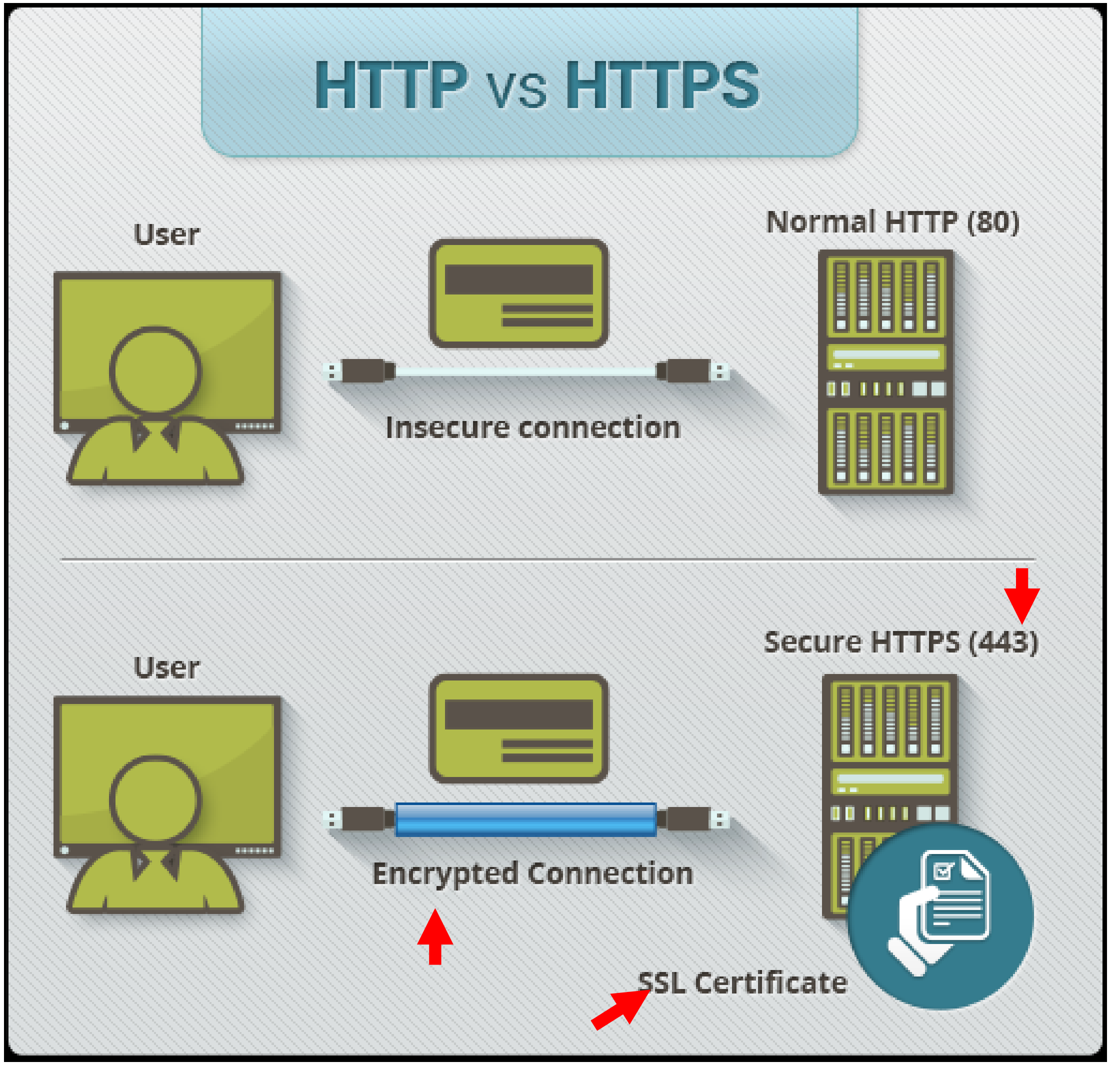


Application-Level Gateway (cont'd)

- 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)
  - 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 an internal 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)).


HTTP vs HTTPS
User
Normal HTTP (80)
Insecure connection
User
Secure HTTPS (443)
Encrypted Connection
SSL Certificate

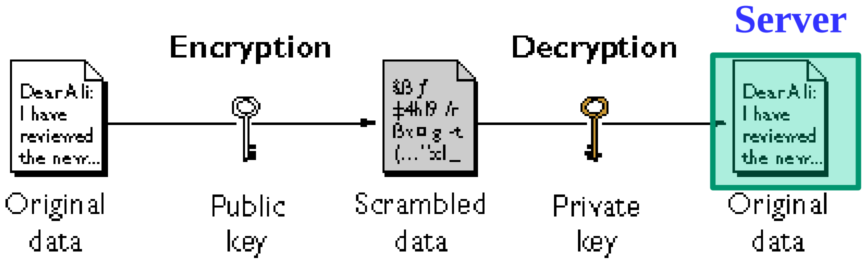
TLS: Design Goals

- Provide authentication, privacy and data integrity between two communicating applications.
- **Mutual** Server and Client authentications
- An encrypted connection
  - *Confidentiality* and *integrity*
- **Interoperability**
- **Extensibility**
  - *New* public key and encryption methods can be incorporated as necessary.

HTTPS: X.509 Certificates (2/4)

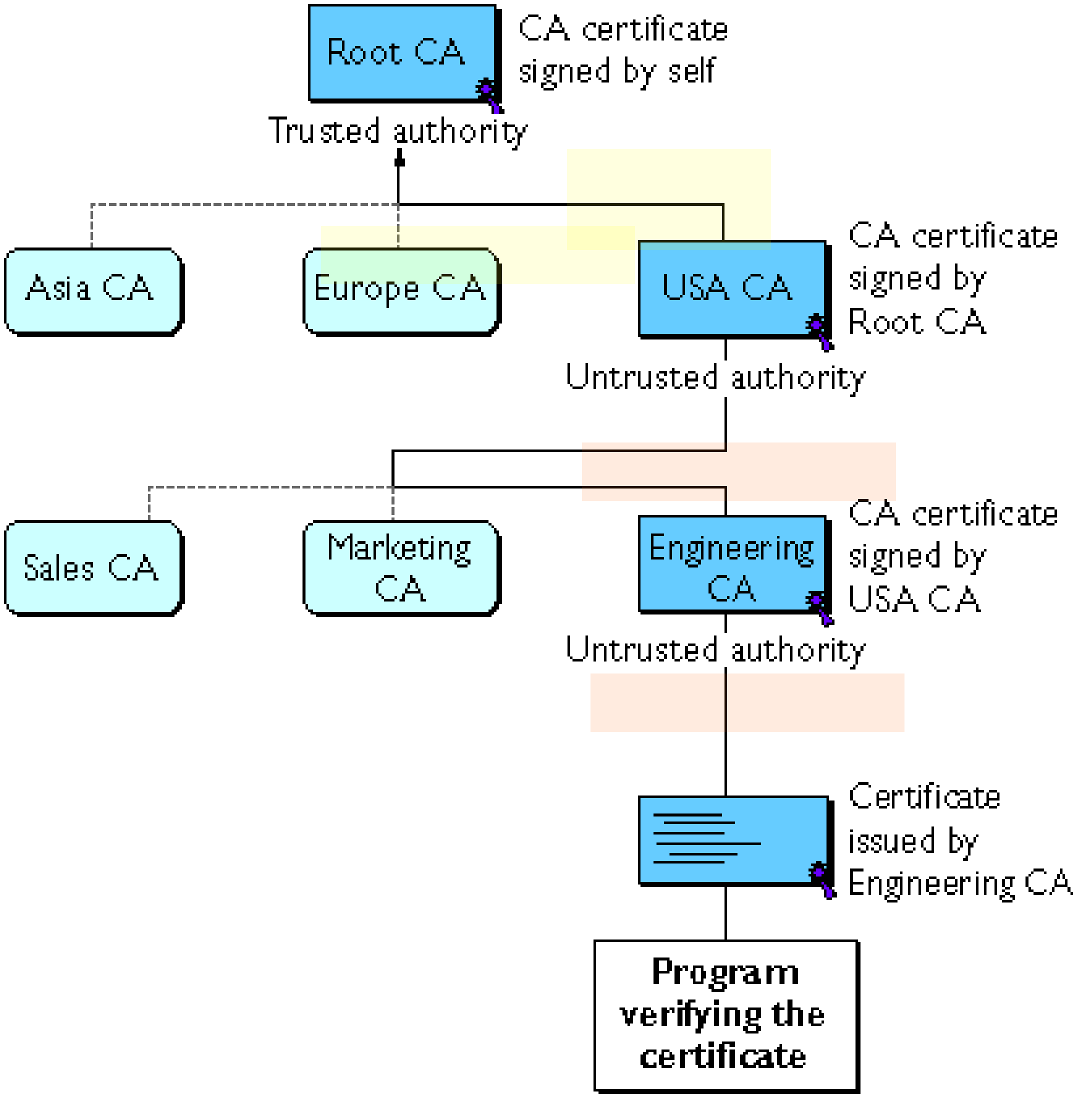
- HTTPS and TLS support the use of ITU-T X.509 **digital certificates from server** ***for user to authenticate the server***, and to negotiate asymmetric session key for the secure session between them.
- Both the TLS and SSL protocols use an 'asymmetric' Public Key Infrastructure (PKI) system.

HTTPS: SSL Certificate (3/4)



- In the case of a website, **server** must first obtain a **SSL Certificate**
  - the **private key** remains *securely* ensconced (or shield) on the web **server**.
  - the **public key** is intended to be *distributed* to anybody and everybody that needs to be able to decrypt information that was encrypted with the private key.

Certificate Chain

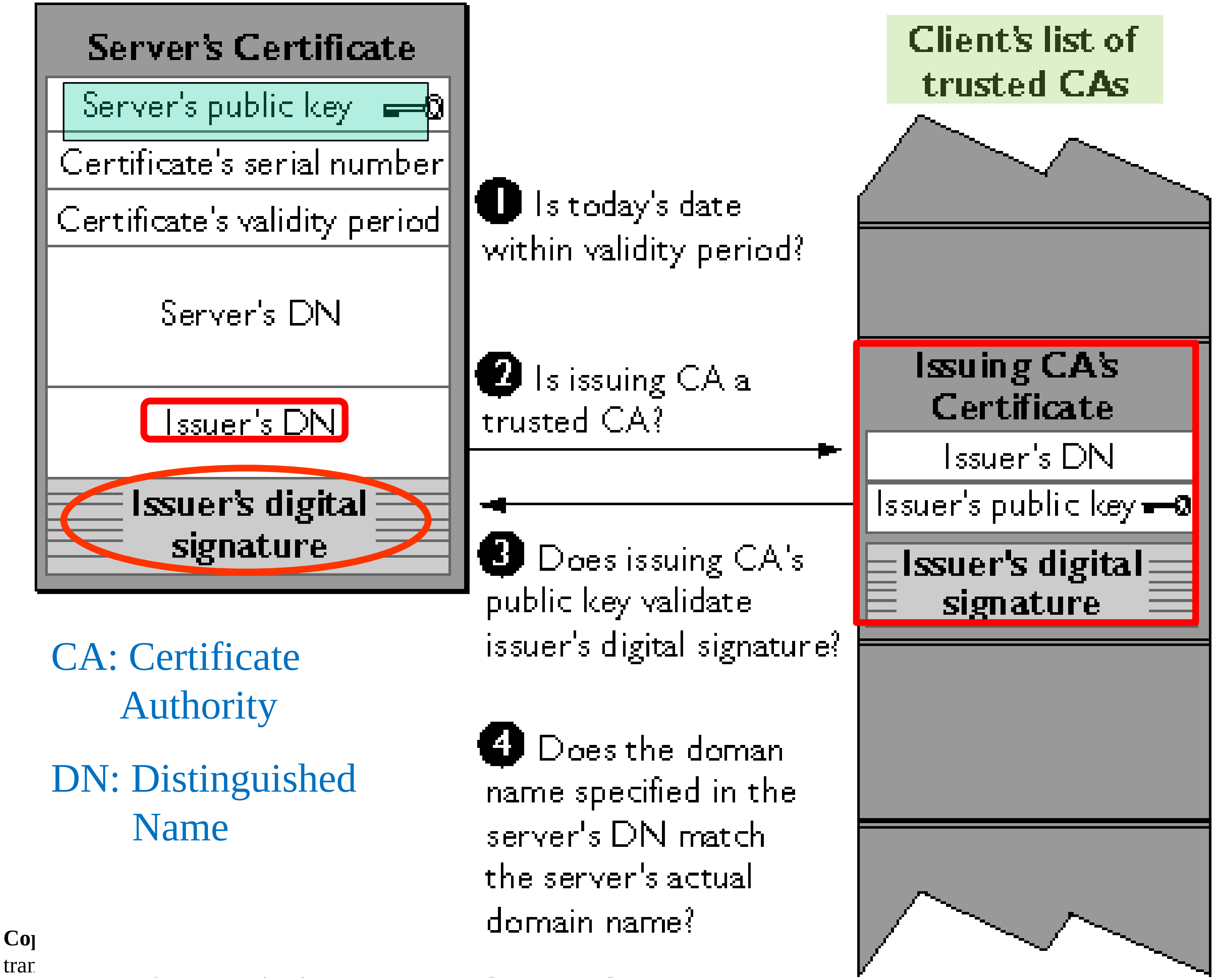

Root CA
CA certificate signed by self
Trusted authority
Asia CA
Europe CA
USA CA
CA certificate signed by Root CA
Untrusted authority
Sales CA
Marketing CA
Engineering CA
CA certificate signed by USA CA
Untrusted authority
Certificate issued by Engineering CA
Program verifying the certificate

HTTPS: Session Key (4/4)

User
Secure HTTPS (443)
Encrypted Connection
SSL Certificate

- The **session key** is used to **encrypt data flowing between the parties**.
- This allows for data/message **confidentiality**, and *message authentication codes* for message **integrity** and as a by-product, **message authentication**.
- The use of HTTPS protects against *eavesdropping* and *man-in-the-middle attacks*.

Server Certification


Server's Certificate
Server's public key
Certificate's serial number
Certificate's validity period
Server's DN
Issuer's DN
Issuer's digital signature
Client's list of trusted CAs
Issuing CA's Certificate
Issuer's DN
Issuer's public key
Issuer's digital signature

❶ Is today's date within validity period?

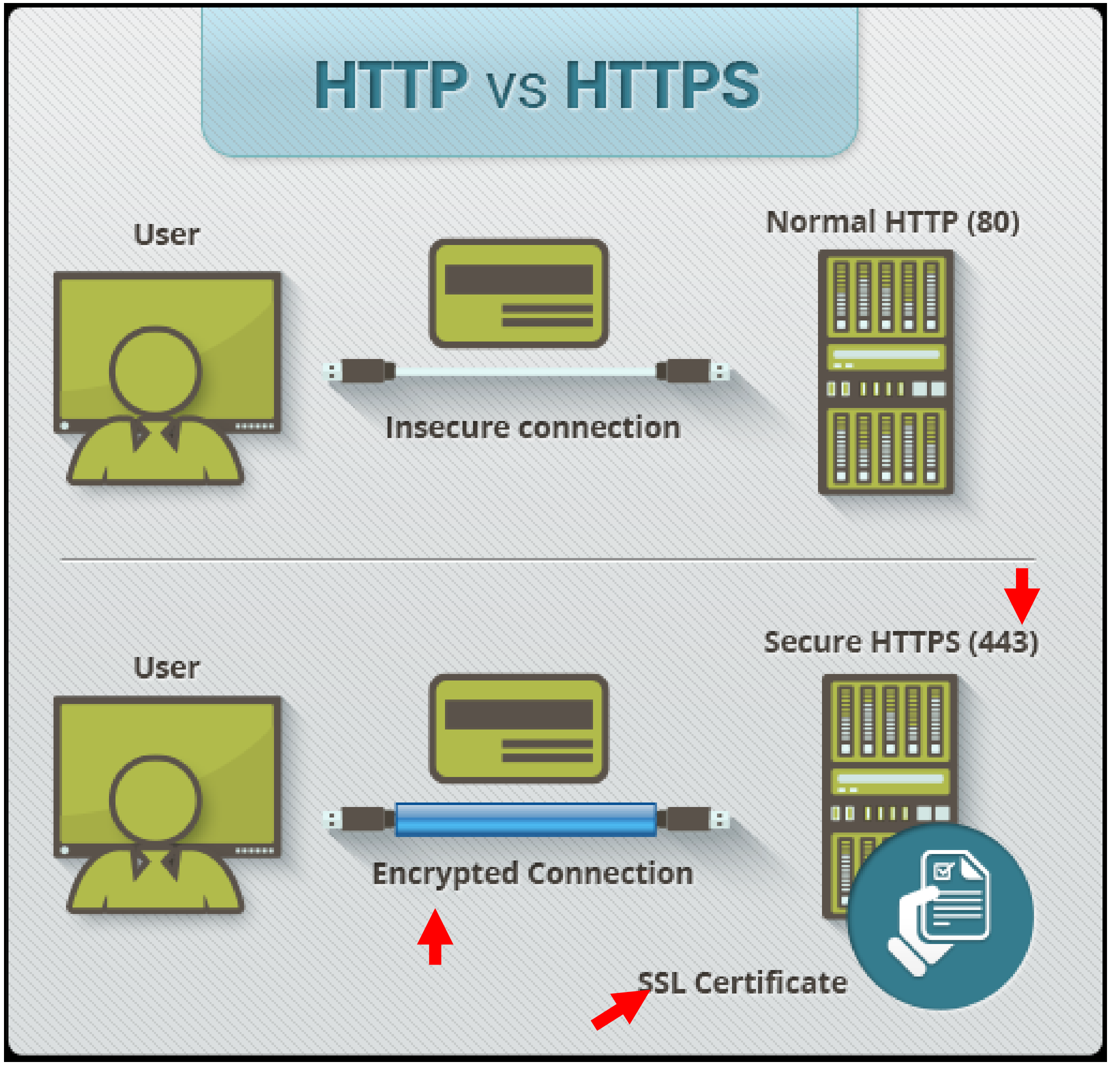
❷ Is issuing CA a trusted CA?

❸ Does issuing CA's public key validate issuer's digital signature?

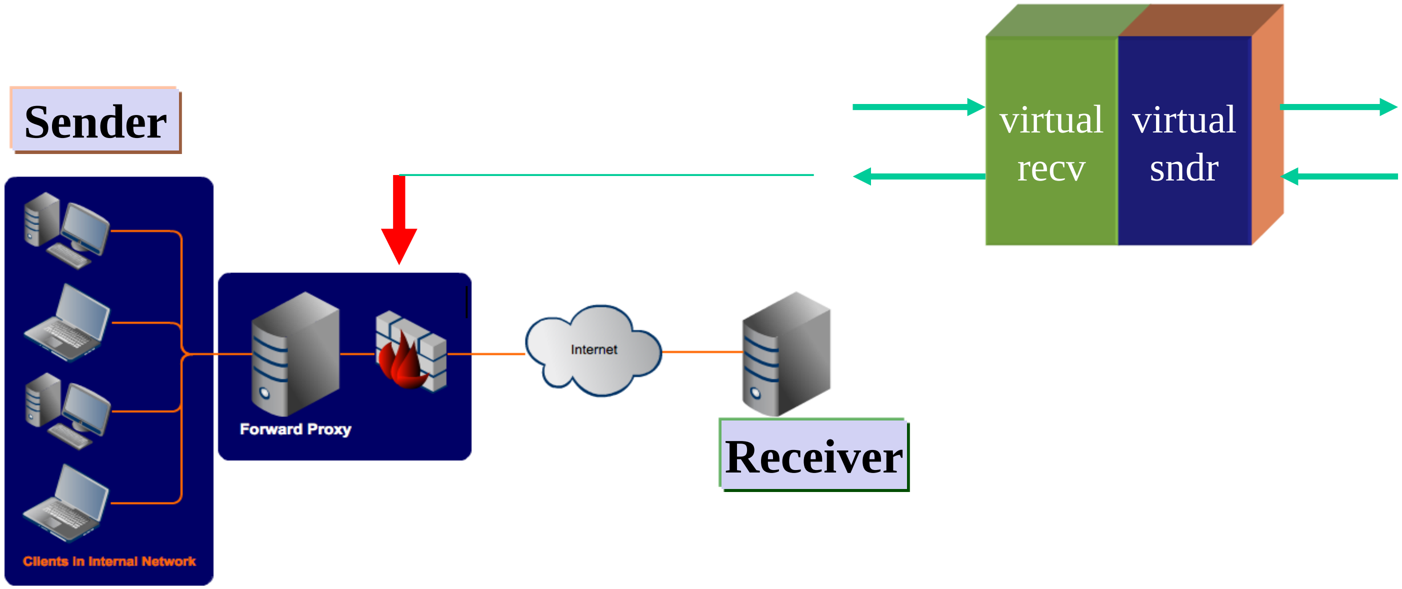
❹ Does the doman name specified in the server's DN match the server's actual domain name?

CA: Certificate Authority

DN: Distinguished Name


HTTP vs HTTPS
User
Normal HTTP (80)
Insecure connection
User
Secure HTTPS (443)
Encrypted Connection
SSL Certificate

SSL Forward Proxy (1/3)



Sender

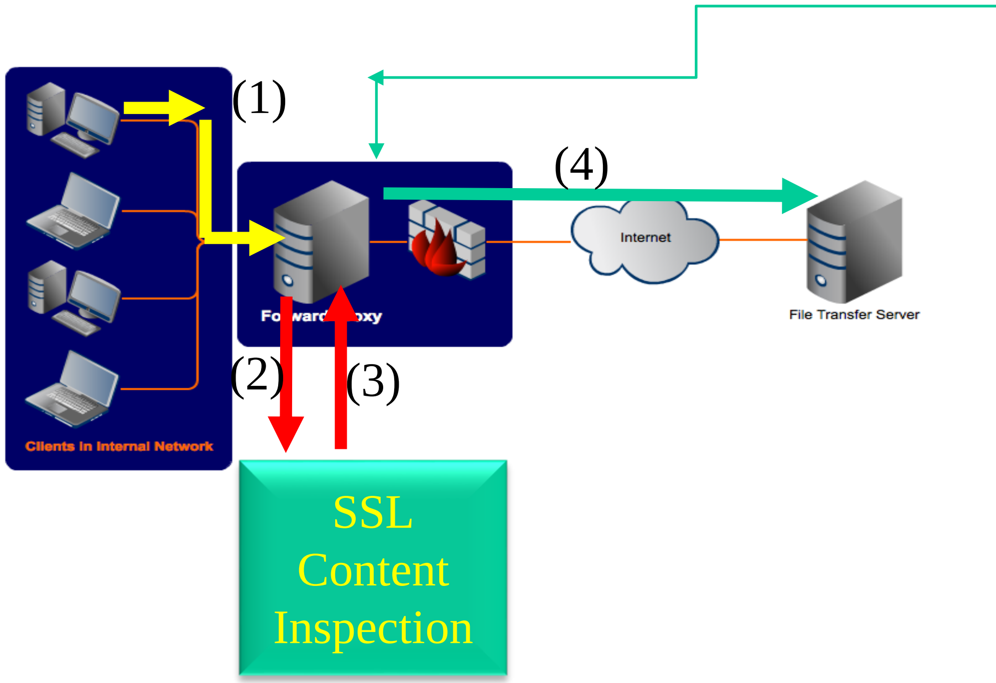
virtual recv

virtual sndr

Receiver

- A forward proxy is typically used **in tandem with a firewall** to enhance an internal network's security
- It **controls traffic originating from clients in the internal network to hosts on the Internet.**

SSL Forward Proxy (2/3)



SSL Content Inspection

- An <u>SSL forward proxy</u> consists of **two** SSL termination devices that have **separate secured sessions between server and client**.
- From the point of view of the web server, **it is the proxy server that issued the request**, not the client.
- Hence, the server **addresses its response to the proxy**.

1) **Decrypt** SSL-encrypted traffic;
2) The traffic is **inspected** and **analyzed**.
3) Apply security policy, an HTTP request can be **allowed** or *denied*.
4) The traffic, possibly scrubbed, is **encrypted** and forwarded to the intended destination.

The SSL Forward Proxy Server (3/3)

- NAT+application-level security control (e.g., A10 Thunder application delivery control SSL Insight)
- It can serve as a **single point of access and control**, making it easier for a corporate to enforce security policies.
- The proxy servers can **keep track of requests, responses**, and their **sources** and their **destinations**.

To be continued. ☺