

Denial of Service Attacks

Cunsheng Ding
Department of CSE
HKUST, Hong Kong

Acknowledgements: Materials are taken from the Internet

Agenda of this lecture

- Zombie computers, bots, botnets
- Denial of service (DoS) attacks
- Distributed denial of services (DDoS) attacks
- Specific DoS attacks
 - Ping of death, Smurf, Teardrop, DNS amplification
- Defenses against DoS attacks
- Conclusions

Zombie computers, bots, botnets

Zombie computers

- A **zombie computer** is a user's computer controlled and used by a hacker to conduct *illegal* activities.
- The user generally remains unaware that his computer has been taken over -- he can still use it, though it might slow down considerably.
- As his computer begins to either send out massive amounts of spam or attack Web pages, he becomes the focal point for any investigations involving his computer's suspicious activities.

Transforming computers into zombies

- Crackers do it by using small programs that exploit weaknesses in a computer's operating system.
- In order to infect a computer, the cracker must first get the installation program to the victim.
 - Crackers can do this through e-mail, peer-to-peer networks or even on a regular Web site.
 - The program either contains specific instructions to carry out a task at a particular time, or it allows the cracker to directly control the user's Internet activity.
- Most of the time, crackers disguise the malicious program with a name and file extension so that the victim thinks he's getting something entirely different.

Malwares

- Programs designed to harm or compromise a computer are called **malwares** (as in malicious software).
- Malware includes a wide array of nasty batches of code that can wreak *havoc* to your computer, your network and even the Internet itself.

Malwares turning computers into zombies

- Computer viruses
 - programs that disable the victim's computer, either by corrupting necessary files or hogging the computer's resources.
- Worms
 - programs that spread from one machine to another, rapidly infecting hundreds of computers in a short time.
- Trojan horse
 - a program that claims to do one thing, but actually either damages the computer or opens a back door to your system.

Malwares turning computers into zombies

- Rootkits
 - a collection of programs that permits administrator-level control of a computer; not necessarily malware on its own.
 - crackers use rootkits to control computers and evade detection
- Backdoors
 - methods of circumventing the normal operating-system procedures, allowing a cracker to access information on another computer
- Key loggers
 - programs that record keystrokes made by a user, allowing crackers to discover passwords and login codes

Zombie computers for spamming

Using Zombie Computers to send spam



It is hard to trace the hacker!

A zombie by any other name

- A zombie computer can still behave normally, and every action it takes is a result of a cracker's instructions (though these instructions might be automated). Hence, the name “zombie computer” may be misleading.
- Due to this, some people prefer the term "**bot**.”
 - **Bot** comes from the word "**robot**," which in this sense is a device that carries out specific instructions.
 - A collection of networked bots is called a "**botnet**," and a group of zombie computers is called an "**army**."

Denial of Service Attacks

What is a denial-of-service attack?

- A **denial of service** (DoS) is an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as CPUs, memory, bandwidth, and disk space.
- It is a form of attack on the availability of some service.

Targeted resources

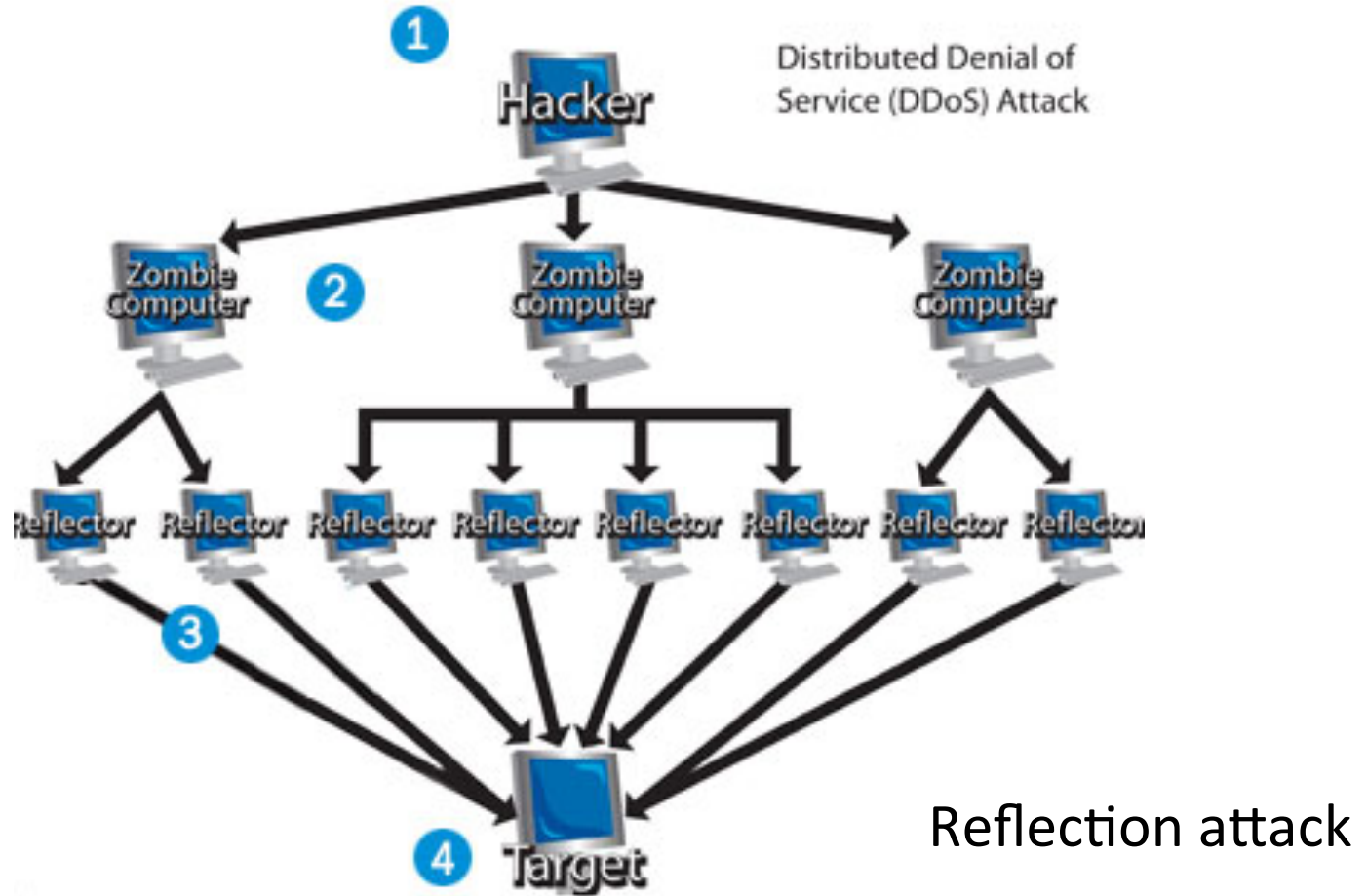
- The following categories of resources could be attached:
 - Network bandwidth
 - E.g., a link between a server and an ISP.
 - System resources
 - E.g., overload or crash a system's network handling software. [e.g., *Ping of Death*, *Teardrop*]
 - Application resources
 - E.g., a Web server, email server

Distributed Denial of Service (DDoS) Attacks

Distributed DoS attacks

- A cracker uses a network of “zombie computers” to sabotage a specific Web site or server. (How?)
- The idea is pretty simple -- a cracker tells all the computers on his *botnet* to contact a specific server or Web site **repeatedly**.
- The sudden increase in traffic can cause the site to load very slowly for legitimate users. Sometimes the traffic is enough to shut the site down completely.

DDoS attacks: pictorial description



How does a DDoS work?

- The cracker sends “the command” to initiate the attack to his zombie army.
- Each computer within the army sends an electronic connection request to an **innocent** computer called a reflector.
- When the reflector receives the request, it looks like it originates not from the zombies, but from the ultimate victim of the attack.
- The reflectors send information to the victim system, and eventually the system's performance suffers or it shuts down completely as it is overwhelmed with multiple unsolicited responses from several computers at once.

Features of DDoS attacks

- From the perspective of the victim, it looks like the reflectors attacked the system.
- From the perspective of the reflectors, it seems like the victimized system requested the packets.
- The zombie computers remain hidden, and even more out of sight is the cracker himself.

Some DDoS attacks

- Ping of Death:
 - bots create huge electronic packets and sends them on to victims.
- Mailbomb:
 - bots send a massive amount of e-mail, crashing e-mail servers.
- Smurf Attack:
 - bots send Internet Control Message Protocol (ICMP) messages to reflectors.
- Teardrop:
 - bots send pieces of an illegitimate packet; the victim system tries to recombine the pieces into a packet and crashes as a result

Examples of Victims

- Companies
 - Microsoft, Amazon, CNN, Yahoo
- Financial institutions
 - eBay,  
 

Script kiddies: so easy to do it

- On May 4th, 2001, a 13-year-old cracker used a denial of service attack to bring down GRC.com, the Web site for Gibson Research Corporation. Ironically, GRC.com focuses on Internet security.
- In 2006, police in Hanoi, Vietnam arrested a high school sophomore for orchestrating a DDoS attack on a Web site for the Nhan Hoa Software Company. He said the reason he did it was because he didn't like the Web site.

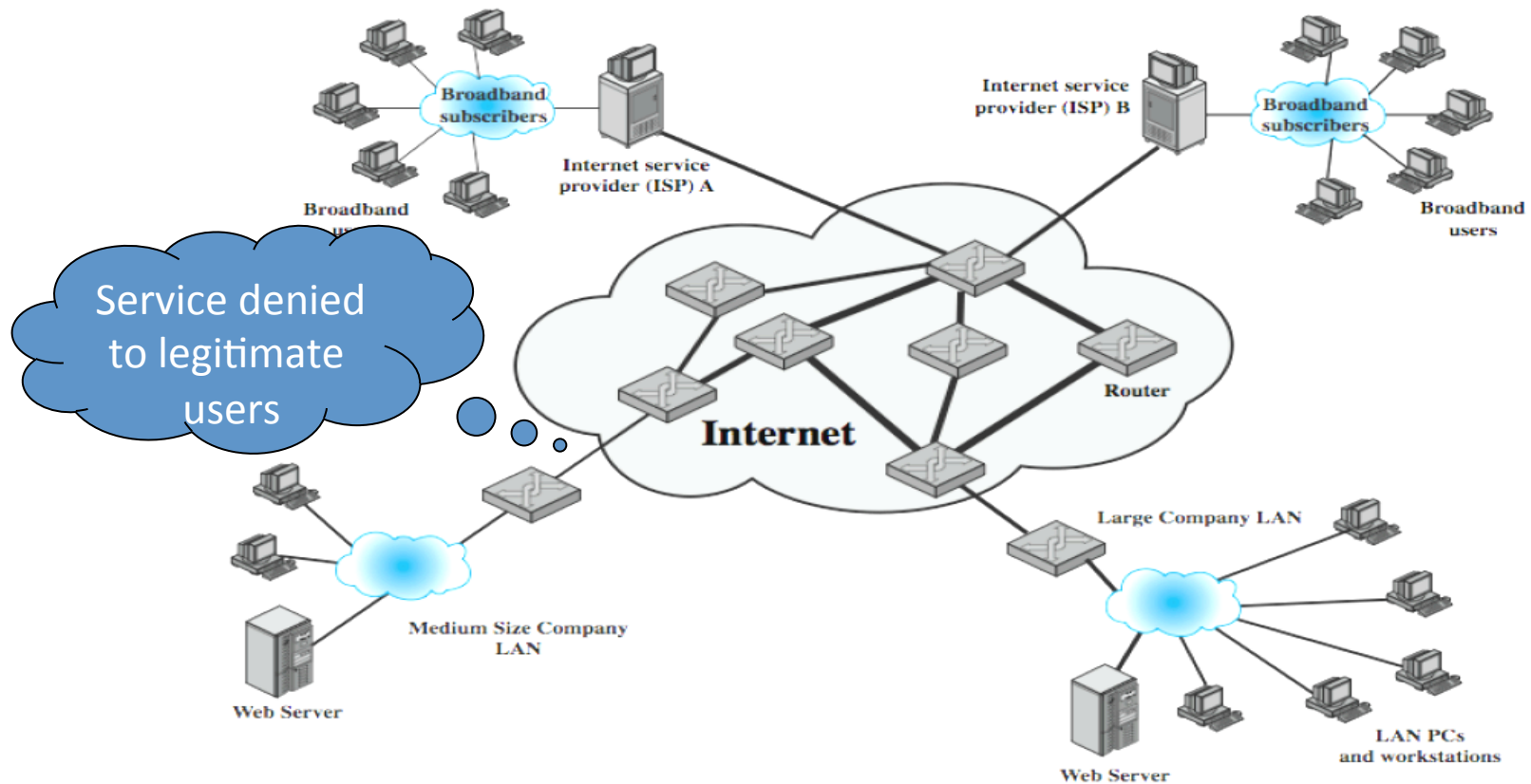
Classification of DoS attacks

- Direct attacks
 - Attacker uses his/her computer to attack the targeted machine or system directly.
 - E.g., sending a huge number of emails to a mail server in a short time period.
 - It is easy to trace back to the attacker.
- Reflection attacks (indirect attacks)
 - Attacker spoofed source addresses to attack the targeted machine or system directly.
 - It is much harder to find out the attacker.
 - Most DDoS attacks are reflection attacks.

Several Denial of Service (DoS) Attacks

Classical DoS attacks

- Simplest classical DoS attack: **Flooding attack** on an organization: E.g., Ping flood attack



Ping of Death

- It exploits a flaw in many vendors' implementations of ICMP.
 - ping is a TCP/IP command that sends out an IP packet to a specified IP address or host name to see if there is a response from the address or host. It is often used to determine if a host is on the network or alive.
 - The typical ping command syntax would be:
ping 150.24.35.46, or, ping www.acme.net
 - It works for Windows and Unix-like operating systems.
- Normally it requires a flood of pings to crash a system.
- It is an attack on the network bandwidth.
- It could be a **direct** or **reflection** attack

Ping flood attack

- Use of ping command options -n -l

Ping of Death

```
Microsoft Windows [Version 6.0.6000]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Z>ping 127.0.0.1 -n 5 -l 65500

Pinging 127.0.0.1 with 65500 bytes of data:

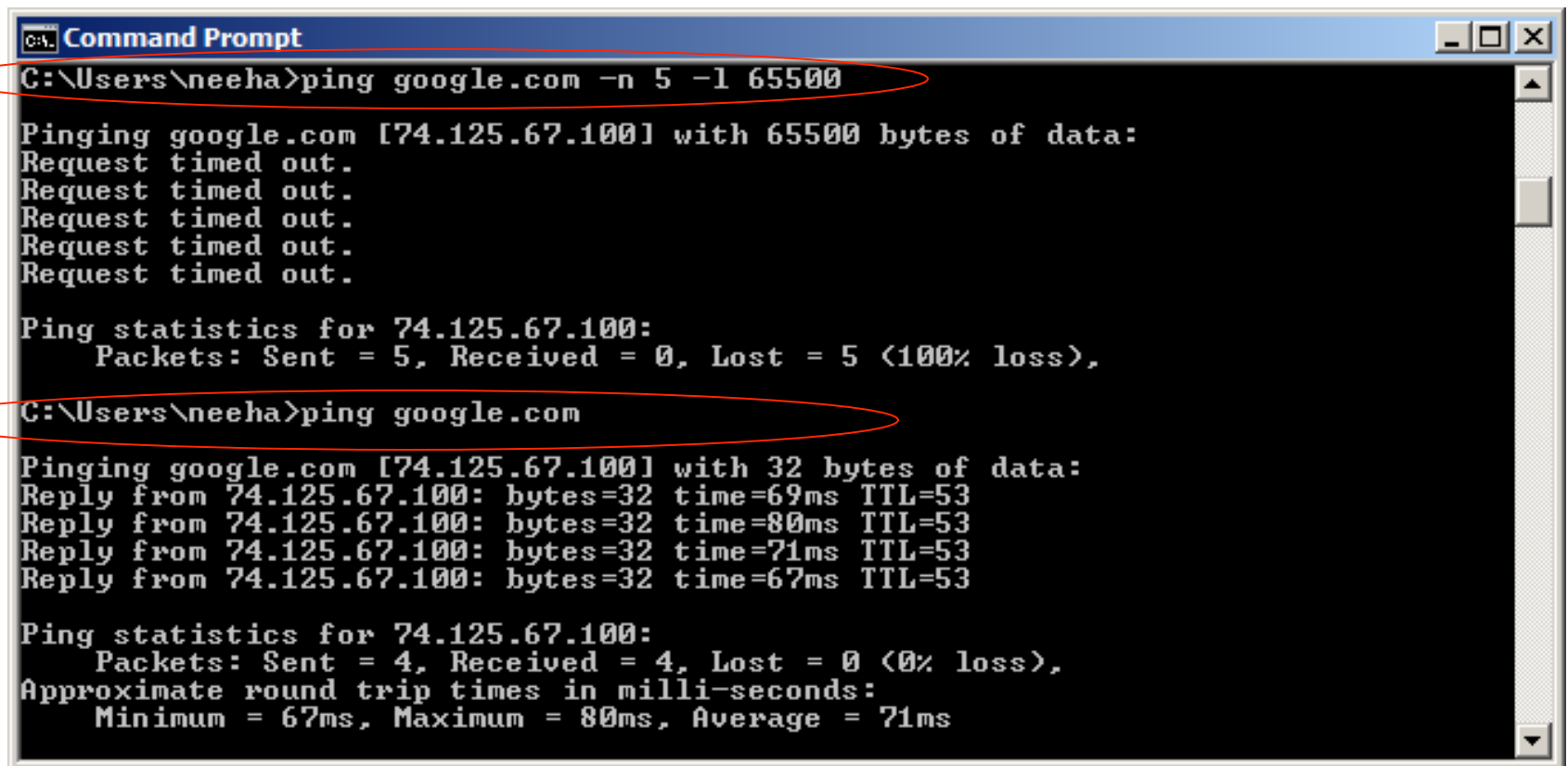
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128
Reply from 127.0.0.1: bytes=65500 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Z>_
```

Ping flood attack cont'd

- Generally useless on larger networks or websites



```
Command Prompt
C:\Users\neeha>ping google.com -n 5 -l 65500

Pinging google.com [74.125.67.100] with 65500 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 74.125.67.100:
    Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),

C:\Users\neeha>ping google.com

Pinging google.com [74.125.67.100] with 32 bytes of data:
Reply from 74.125.67.100: bytes=32 time=69ms TTL=53
Reply from 74.125.67.100: bytes=32 time=80ms TTL=53
Reply from 74.125.67.100: bytes=32 time=71ms TTL=53
Reply from 74.125.67.100: bytes=32 time=67ms TTL=53

Ping statistics for 74.125.67.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 67ms, Maximum = 80ms, Average = 71ms
```

Source address spoofing

- It is one of the most frequently used spoofing attack methods, and can be employed in both direct and reflection attacks.
- In an IP address spoofing attack, an attacker sends IP packets from a false (or “spoofed”) source address in order to disguise itself, and distribute the working load of the attack.
- Denial-of-service attacks often use IP spoofing to overload networks and devices with packets that appear to be from legitimate source IP addresses.
- IP spoofing attacks can also be used to bypass IP address-based authentication.

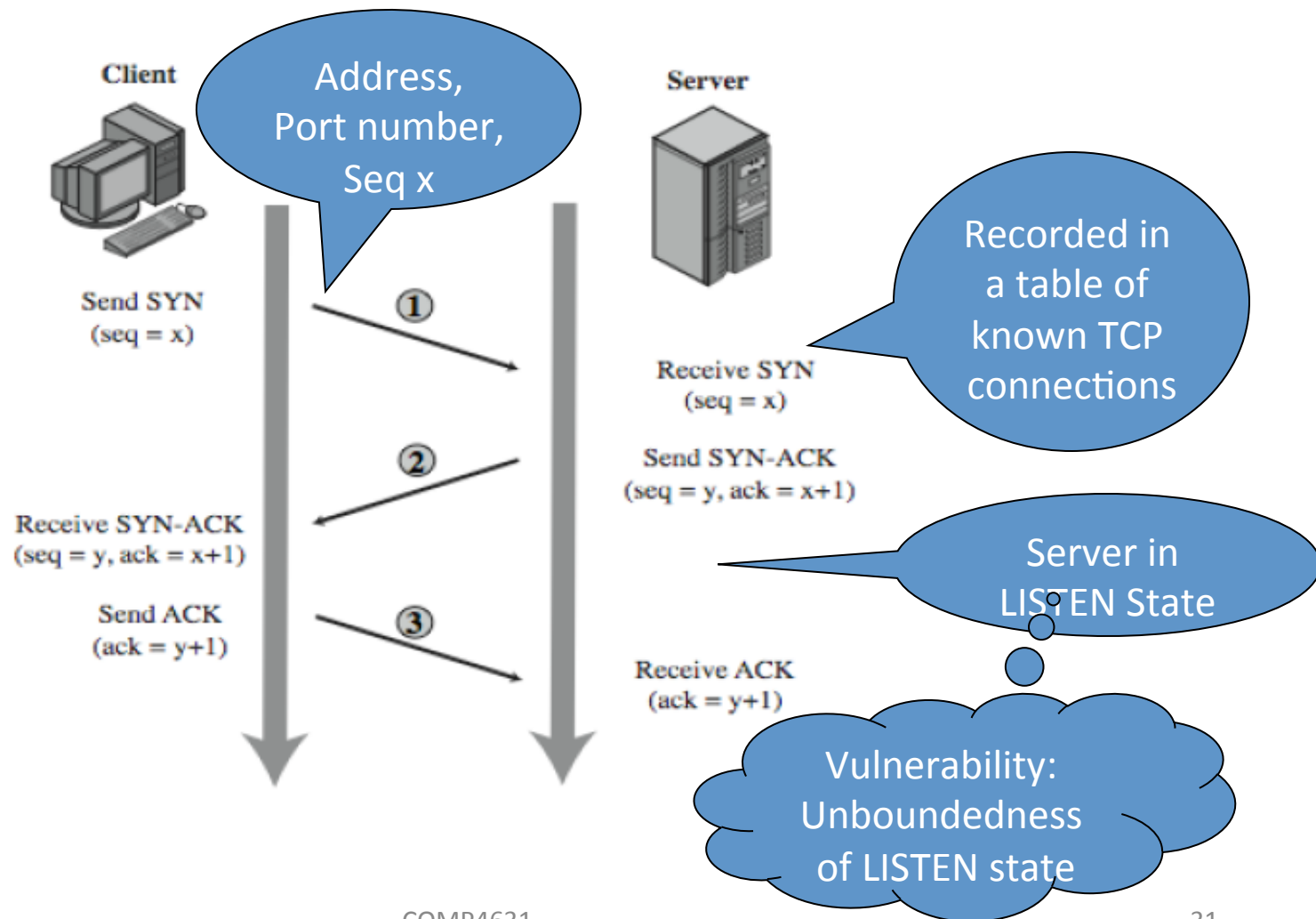
How to spoof source addresses?

- In the case of having privileged access to network handling codes, it can be done via **raw socket interface**
 - Allows direct sending and receiving of information by applications
 - Not needed for normal network operation
- In absence of privilege, install a **custom device driver** on the source system
- How to spoof your IP address using NMAP in Windows
 - <http://gregsumner.blogspot.hk/2013/02/how-to-spoof-your-ip-address-using-nmap.html>
 - <http://seclists.org/nmap-hackers/2004/0008.html>

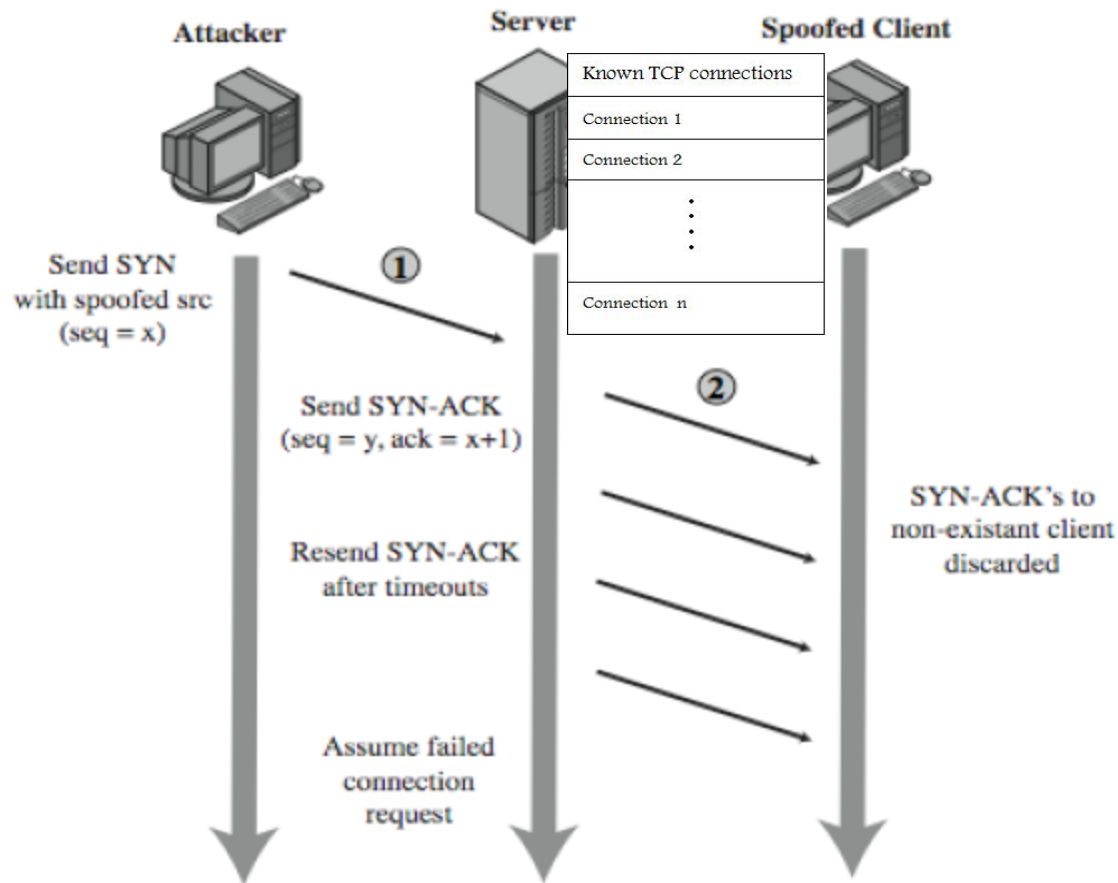
SYN spoofing

- Takes advantage of the ***three-way handshake*** that occurs any time two systems across the network initiate a TCP connection request
- Unlike usual brute-force attack, not done by exhausting network resources but done by ***overflowing the system resources*** (tables used to manage TCP connections)
- **Require fewer packets** to deplete
- Consequence: **Failure of future connection requests**, thereby denying access to the server for legitimate users
- Example: land.c sends TCP SYN packet using target's address as source as well as destination

TCP 3-way connection handshake



SYN spoofing cont'd



Factors considered by attacker for SYN spoofing

- The number of sent forged packets are just **large enough to exhaust the table** but small as compared to a typical flooding attack
- Keep sufficient volume of forged requests **flowing**
 - Keep the table constantly full with no timed-out requests
- Make sure to use addresses that will **not respond to the SYN-ACK with a RST** (reset the connection)
 - Overloading the spoofed client
 - Using a wide range of random addresses
 - A collection of compromised hosts under the attacker's control (i.e., a "botnet") could be used

Detecting SYN spoof attacks

- After the target system has tried to send a SYN/ACK packet to the client and while it is waiting to receive an ACK packet, the existing connection is said to be half open or host in `SYN_RECEIVED` state
- If your system is in this state, it may be experiencing SYN-spoof attack
- To determine whether connections on your system are half open, type **`netstat -a`** command
- This command gives a set of active connections. Check for those in the state **`SYN_RECEIVED`** which is an indication of the threat of SYN spoof attack

Smurf DoS attack

- Two main components
 - Send **source-forged ICMP echo packet** requests from **remote locations**
 - Packets directed to **IP broadcast addresses**
- If the intermediary does not filter this broadcast traffic, many of the machines on the network would receive and respond to these spoofed packets
 - When entire network responds, successful smurf DoS has been performed on the target network
- Besides victim network, intermediary network may also suffer
 - Smurf DoS attack with single/multiple intermediary(s)
 - Analyze network routers that do not filter broadcast traffic
 - Look for networks where multiple hosts respond

DNS amplification attacks

- DNS servers are the intermediary system
- Exploit DNS behavior to convert a small request to a much larger response
 - 60 byte request to 512 – 4000 byte response
- Sending DNS requests with spoofed source address being the target to the chosen servers
- Attacker sends requests to multiple well connected servers, which flood target
 - Moderate flow of packets from attacker is sufficient
 - Target overwhelmed with amplified responses from server

Teardrop

- This DoS attack affects Windows 3.1, 95 and NT machines and Linux versions previous to 2.0.32 and 2.1.63
- **Teardrop** is a program that sends IP fragments to a machine connected to the Internet or a network
- **Teardrop** exploits an overlapping IP fragment bug
 - The bug causes the TCP/IP fragmentation re-assembly code to improperly handle overlapping IP fragments
 - A 4000 bytes of data is sent as
 - Legitimately (Bytes 1-1500) (Bytes 1501 – 3000) (Bytes 3001-4500)
 - Overlapping (Bytes 1-1500) (Bytes 1501 – 3000) (Bytes 1001-3600)
- This attack has not been shown to cause any significant damage to systems
- The primary problem with this is loss of data

Defense against DoS Attacks

Defenses against DoS attacks

- DoS attacks cannot be prevented entirely
- Impractical to prevent the **flash crowds** without compromising network performance
- Three lines of defense against (D)DoS attacks
 - Attack prevention and pre-emption
 - Attack detection and filtering
 - Attack source trace-back and identification

Attack prevention

- Limit ability of systems to send spoofed packets
 - Filtering done as close to source as possible by routers/gateways
 - Reverse-path filtering ensure that the path back to claimed source is same as the current packet's path
 - On Cisco router “ip verify unicast reverse-path” command
- Rate controls in upstream distribution nets
 - On specific packet types
 - E.g., Some ICMP, some UDP, TCP/SYN
- Use modified TCP connection handling
 - Use SYN-ACK cookies when table full
 - Or selective or random drop when table full

Attack prevention cont'd

- Block IP broadcasts
- Block suspicious services & combinations
- Manage application attacks with “puzzles” to distinguish legitimate human requests
- Good general system security practices
- Use mirrored and replicated servers when high performance and reliability required
- **Different governmental legislation (perhaps the most effective solution)**

Responding to attacks

- Need good incident response plan
 - With contacts for ISP
 - Needed to impose traffic filtering upstream
 - Details of response process
- Have standard antispoofing, rate limiting, directed broadcast limiting filters
- Ideally have network monitors and intrusion detection systems
 - To detect and notify abnormal traffic patterns

Responding to attacks cont' d

- Identify the type of attack
 - Capture and analyze packets
 - Design filters to block attack traffic upstream
 - Identify and correct system application bugs
- Have ISP trace packet flow back to source
 - May be difficult and time consuming
 - Necessary if legal action desired
- Implement contingency plan
- Update incident response plan

Conclusions

Conclusions

- (D)DoS attacks are *genuine threats* to many Internet users
- Level of loss is related to motivation as well shielding attempts from the defender
- Defensive measures might not always work
- Prognosis for DDoS
 - Increase in size
 - Increase in sophistication
 - Increase in semantic DDoS attacks
 - Infrastructure attacks
- DDoS are significant threats to the future growth and stability of Internet