Module 5.1

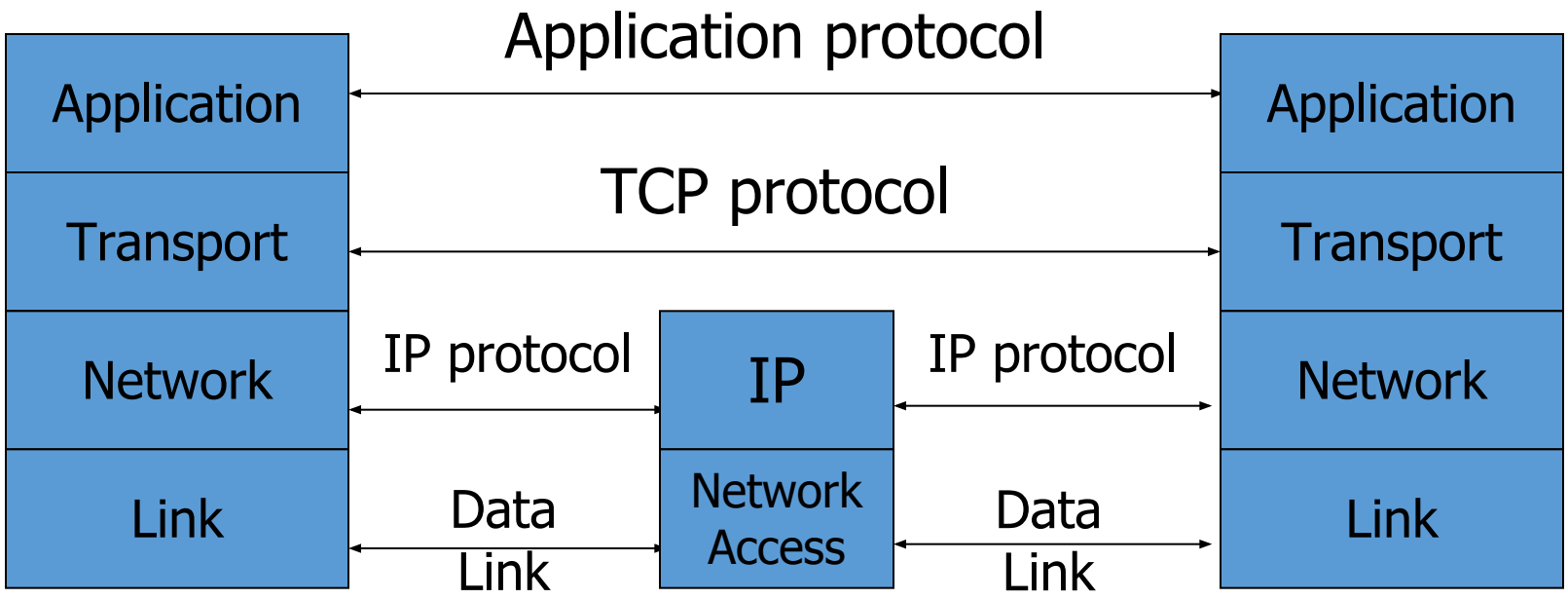
Software Vulnerability

- A vulnerability can be an error in the way that user management occurs in the system, an error in the code or a flaw in how it responds to certain requests.
- Software vulnerabilities involve bugs in software.
- Bugs are coding errors that cause the system to make an unwanted action.
- All software has bugs of one form or another. Some bugs cause the system to crash, some cause connectivity to fail, some do not let a person to log in, and some cause printing not to work properly.
- Some bugs create information leakage or elevate user privileges or grant otherwise unauthorized access. These are **security vulnerabilities**.
- Operating systems are composed of software, as are web browsers, word processing programs, spreadsheets, video players, websites, and every other application.
- Even computer hardware includes a form of software called firmware.
- Networking equipment and cell phones also have software, and therefore inevitably security vulnerabilities.

Types of Software Vulnerability

- Missing data encryption
- OS command injection
- SQL injection
- Buffer overflow
- Missing authentication for critical function
- Missing authorization
- Unrestricted upload of dangerous file types
- Reliance on untrusted inputs in a security decision
- Cross-site scripting and forgery
- Download of codes without integrity checks
- Use of broken algorithms
- URL redirection to untrusted sites
- Path traversal
- Bugs
- Weak passwords
- Software that is already infected with virus

Network Protocols Stack



Types of Addresses in Internet

- Media Access Control (MAC) addresses in the network access layer
  - Associated w/ network interface card (NIC)
  - 48 bits or 64 bits
- IP addresses for the network layer
  - 32 bits for IPv4, and 128 bits for IPv6
  - E.g., 128.3.23.3
- IP addresses + ports for the transport layer
  - E.g., 128.3.23.3:80
- Domain names for the application/human layer
  - E.g., www.purdue.edu

Routing and Translation of Addresses

- Translation between IP addresses and MAC addresses
  - Address Resolution Protocol (ARP) for IPv4
  - Neighbor Discovery Protocol (NDP) for IPv6
- Routing with IP addresses
  - TCP, UDP, IP for routing packets, connections
  - Border Gateway Protocol for routing table updates
- Translation between IP addresses and domain names
  - Domain Name System (DNS)

Threats in Networking

- Confidentiality
  - e.g. Packet sniffing
- Integrity
  - e.g. Session hijacking
- Availability
  - e.g. Denial of service attacks
- Common
  - e.g. Address translation poisoning attacks
  - e.g. Routing attacks

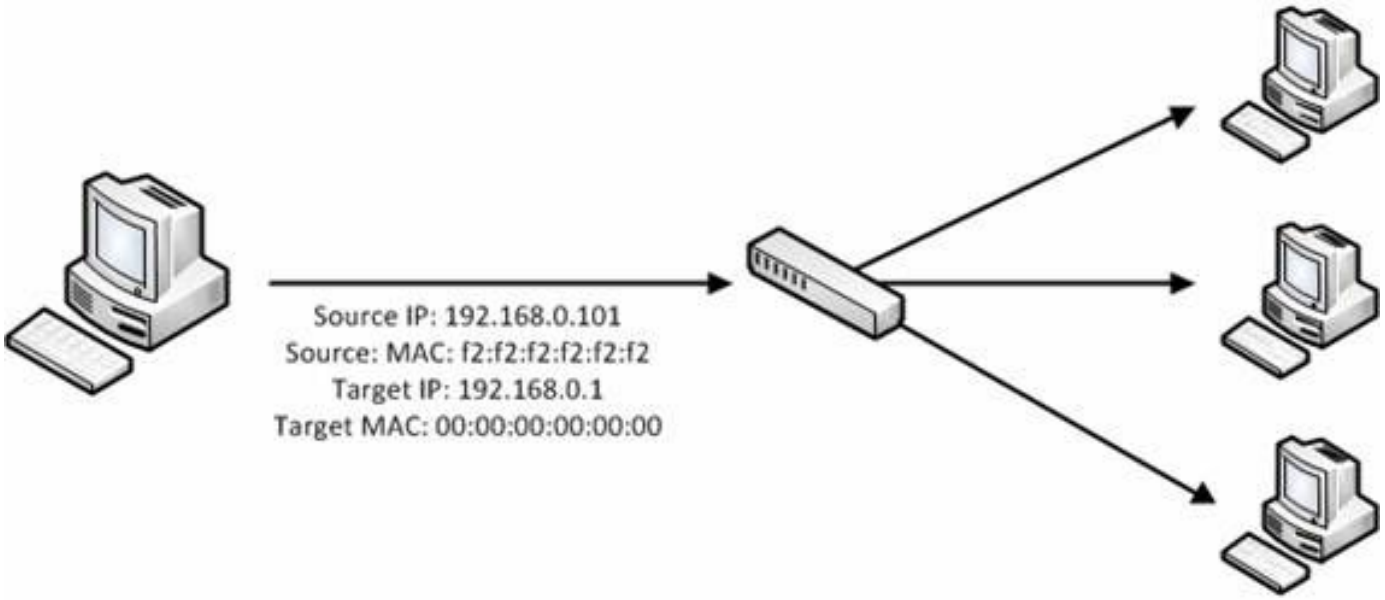
Concrete Security Problems

- ARP is not authenticated
  - APR spoofing (or ARP poisoning)
- Network packets pass by untrusted hosts
  - Packet sniffing
- TCP state can be easy to guess
  - TCP spoofing attack
- Open access
  - Vulnerable to DoS attacks
- DNS is not authenticated
  - DNS poisoning attacks

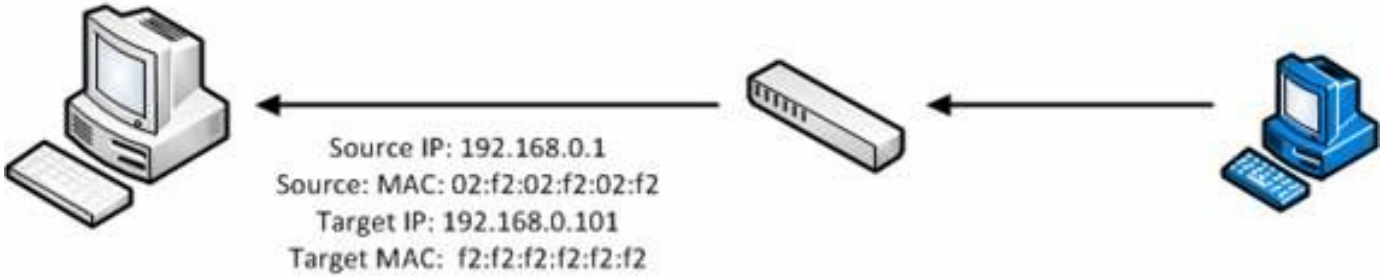
Address Resolution Protocol (ARP)

- Primarily used to translate IP addresses to Ethernet MAC addresses
  - The device drive for Ethernet NIC needs to do this to send a packet
- Also used for IP over other LAN technologies, e.g. IEEE 802.11
- Each host maintains a table of IP to MAC addresses
- Message types:
  - ARP request
  - ARP reply
  - ARP announcement

ARP Request

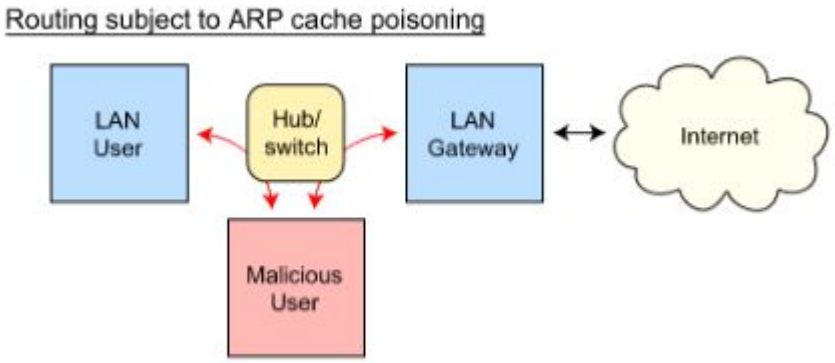


ARP Response

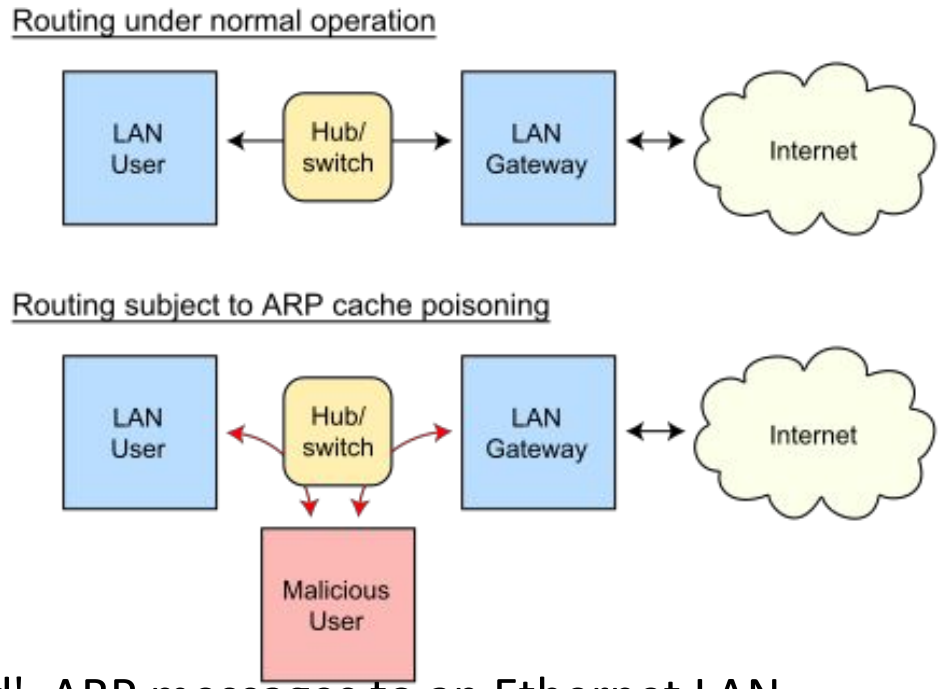


ARP Spoofing

- ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network.
- This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network.
- Once the attacker's MAC address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address.
- ARP spoofing can enable malicious parties to intercept, modify or even stop data in-transit. ARP spoofing attacks can only occur on local area networks that utilize the Address Resolution Protocol.



ARP Spoofing (ARP Poisoning)



- Send fake or 'spoofed', ARP messages to an Ethernet LAN.
  - To have other machines associate IP addresses with the attacker's MAC
- Legitimate use
  - redirect a user to a registration page before allow usage of the network.
  - Implementing redundancy and fault tolerance

ARP Spoofing attack

ARP Spoofing attack:

- The effects of ARP spoofing attacks can have serious implications for enterprises.
- In their most basic application, ARP spoofing attacks are used to steal sensitive information. Beyond this, ARP spoofing attacks are often used to facilitate other attacks such as:
- **Denial-of-serviceattacks:**
  DoS attacks often leverage ARP spoofing to link multiple IP addresses with a single target's MAC address. As a result, traffic that is intended for many different IP addresses will be redirected to the target's MAC address, overloading the target with traffic.
- **Session hijacking:**
  Session hijacking attacks can use ARP spoofing to steal session IDs, granting attacker's access to private systems and data.
- **Man-in-the-middle attacks:**
  MITM attacks can rely onARP spoofing to intercept and modify traffic between victims.

ARP Defenses

- Defenses
  - static ARP table
  - DHCP Certification (use access control to ensure that hosts only use the IP addresses assigned to them, and that only authorized DHCP servers are accessible).
  - detection: Arpwatch (sending email when updates occur),

Port Scanning

Port scanning

- Port Scanning is one of the most popular techniques attackers use to discover services that they can exploit to break into systems.
- All systems that are connected to a LAN or the Internet via a modem run services that listen to well-known and not so well-known ports.
- By port scanning, the attacker can find the following information about the targeted systems: what services are running, what users own those services, whether anonymous logins are supported, and whether certain network services require authentication.
- Port scanning is accomplished by sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can be probed for further weaknesses.
- Port scanners are important to network security technicians because they can reveal possible security vulnerabilities on the targeted system.

Port Scan techniques

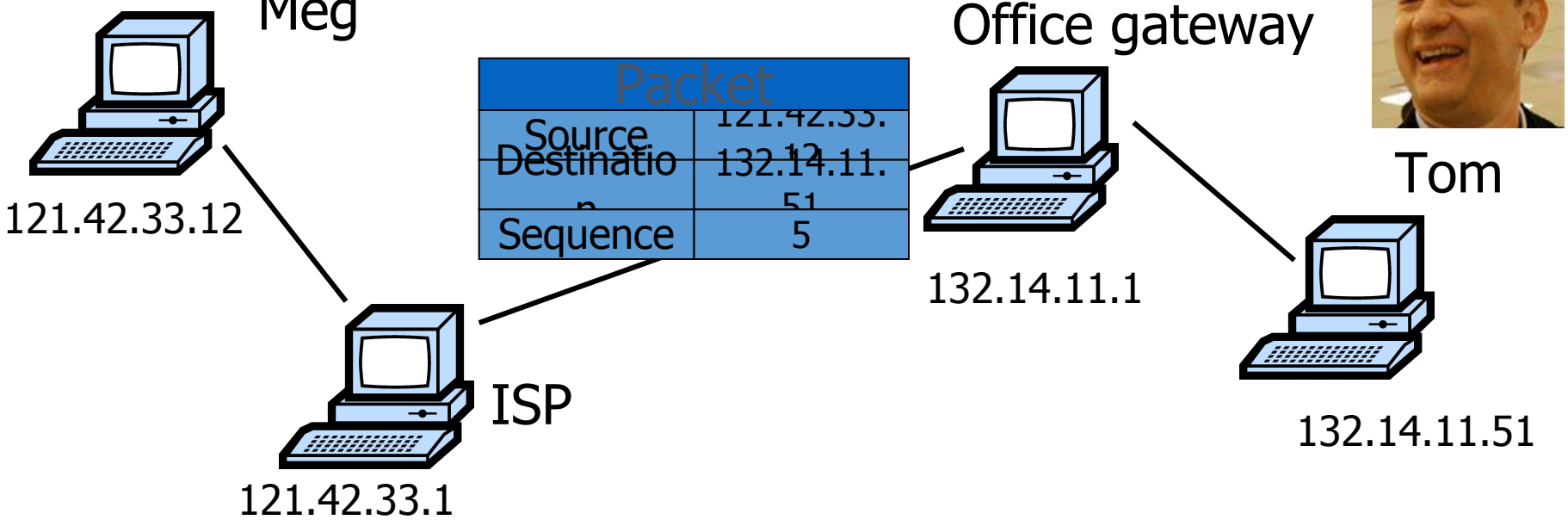
Port Scan Techniques

- **Address Resolution Protocol (ARP)**
  - ARP scans discover active devices on the local network segment by sending a series of ARP broadcasts and incrementing the value for the target IP address field in each broadcast packet.
  - This type of scan will have every IP device on the network respond with its own IP address in response. This scan will effectively map out an entire network.
- **TCP connect**
  - The Vanilla TCP connect scan is the most basic scanning technique.
  - The scan uses the connect system call of an operating system on a target system to open a connection to every port that is open.
  - The scan is extremely noisy and easily detectable. The targeted system logs will show connection requests and error messages for the services that accepted the connections.

Port Scan techniques

- **TCP SYN**
  - The TCP SYN (Half Open) scans are called half open because the attacking system doesn't close the open connections.
  - The attacking scanner will send a SYN packet to the target and wait for a response. If the port is open, the target will send a SYN|ACK.
  - If the port is closed, the target will send an RST.
  - This type of scan is difficult to detect. The target system is in charge of closing the open connections and the target, most likely, will not have the proper logging set up to detect this type of scan.
- **TCP FIN**
  - The TCP FIN scan has the ability to pass undetected through most firewalls, packet filters, and scan detection programs.
  - The attacking system sends FIN packets to the targeted system. The closed ports will respond with an RST. The open ports will ignore the packets. The attacking system will take note of which ports it received an RST on and report on the ports that did not respond with an RST

IP Routing



- Internet routing uses numeric IP address
- Typical route uses several hops

Packet Sniffing

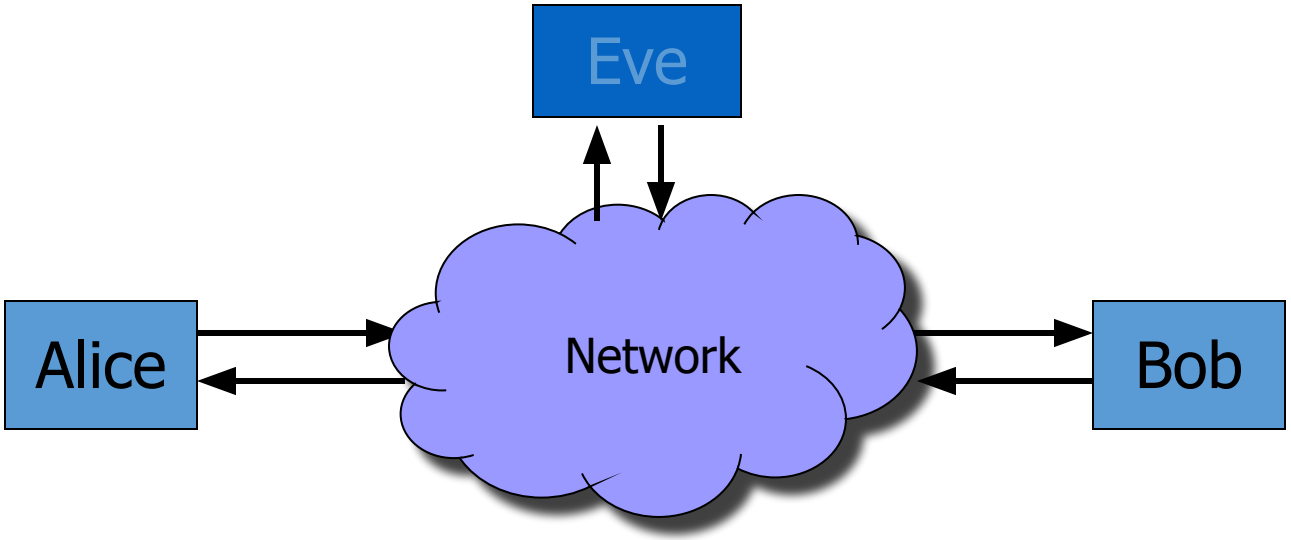
- Monitoring and intercepting data packets passing through a network with the help of specialized tools called packet sniffers is called “sniffing.”
- Data packets carry a wealth of information and facilitate the process of incoming and outgoing traffic.
- A sniffing attack involves the illegal extraction of unencrypted data by capturing network traffic through packet sniffers.

Packet Sniffing-legal/illegal uses

- IT professionals and network administrators use packet sniffers to monitor network traffic, assemble information for security analysis, and identify and troubleshoot network issues from an information security context.
- These are examples of legal usage of packet sniffing to optimize network security.
- Cybercriminals use packet sniffers to steal data and sensitive information from email or web traffic over an unsecured network.
- Sniffing tools are illegally used to steal critical information such as client data, passwords, banking data, or to commit identity theft.
- Hackers can further their nefarious activities by using stolen data in fraudulent transactions with the help of sniffing attack tools such as Wireshark, BetterCAP, WinDump, Ettercap etc.

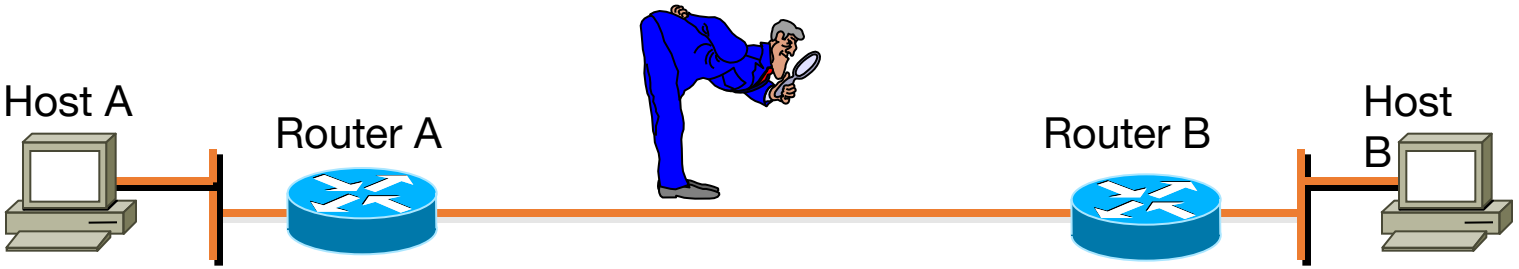
Packet Sniffing

- Promiscuous Network Interface Card reads all packets
  - Read all unencrypted data (e.g., “ngrep”)
  - ftp, telnet send passwords in clear!



Prevention: Encryption (IPSEC, TLS)

Packet Sniffers



- A packet sniffer is a software application that uses a network adapter card in promiscuous mode to capture all network packets. The following are the packet sniffer features:
  - Packet sniffers exploit information passed in clear text. Protocols that pass information in the clear include the following:
    - Telnet
    - FTP
    - SNMP
    - POP
  - Packet sniffers must be on the same collision domain.

Packet Sniffer Mitigation

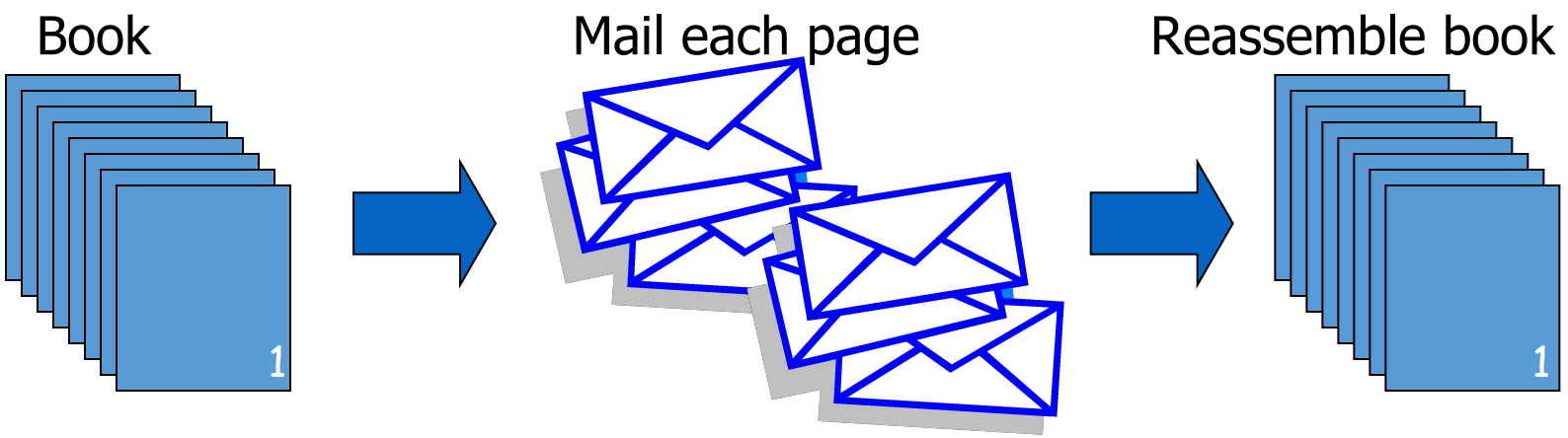
- The following techniques and tools can be used to mitigate sniffers:
  - Authentication—Using strong authentication, such as one-time passwords, is a first option for defense against packet sniffers.
  - Switched infrastructure—Deploy a switched infrastructure to counter the use of packet sniffers in your environment.
  - Antisniffer tools—Use these tools to employ software and hardware designed to detect the use of sniffers on a network.
  - Cryptography—The most effective method for countering packet sniffers does not prevent or detect packet sniffers, but rather renders them irrelevant.

Points to analyse

- Spoofing/sniffing difference
- How spoofing can be prevented
- Different types of hackers --- black/red/white……

Transmission Control Protocol

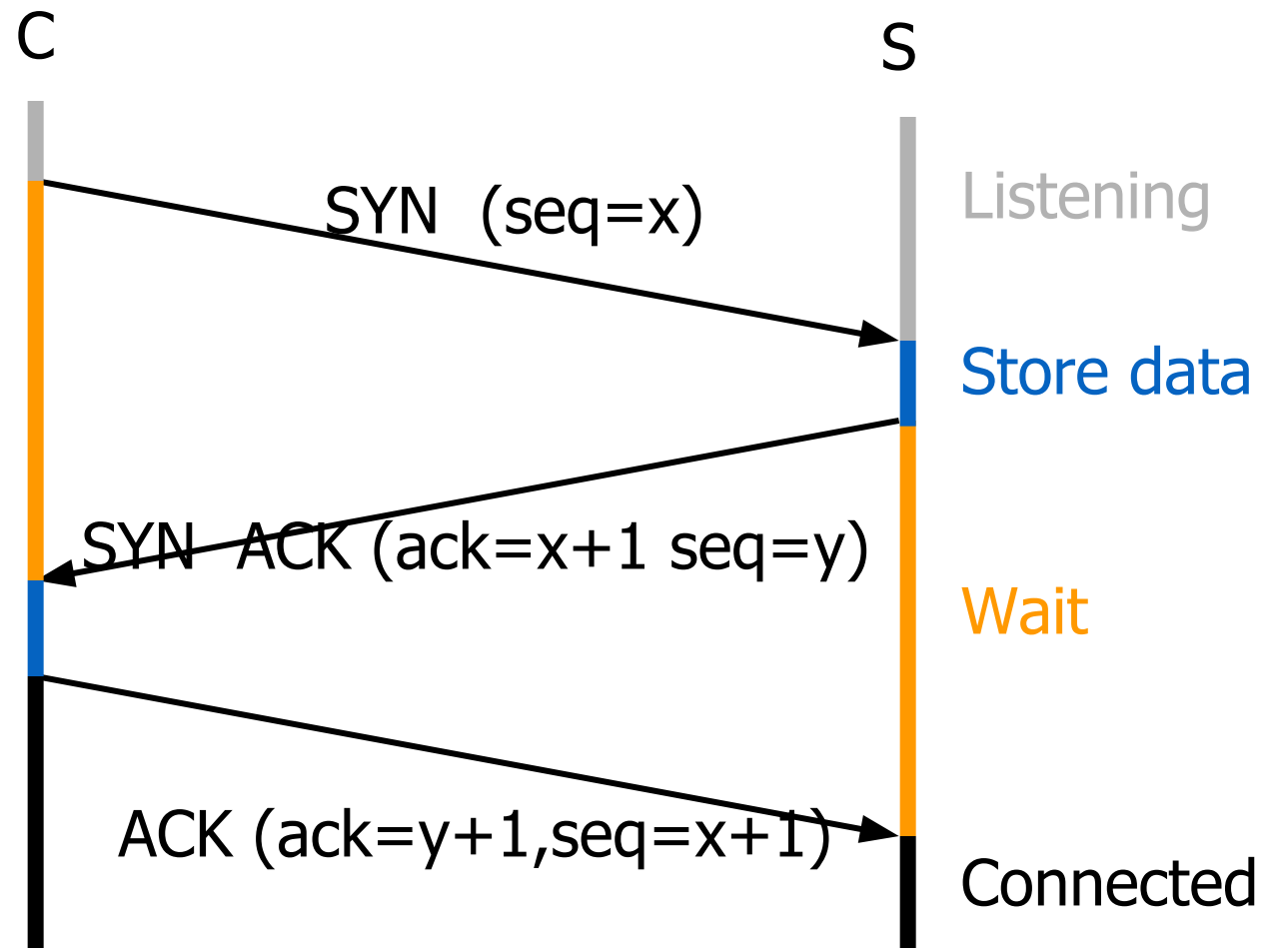
- Connection-oriented, preserves order
  - Sender
    - Break data into packets
    - Attach sequence numbers
  - Receiver
    - Acknowledge receipt; lost packets are resent
    - Reassemble packets in correct order



TCP Sequence Numbers

- Sequence number (32 bits) – has a dual role:
  - If the SYN flag is set, then this is the initial sequence number. The sequence number of the actual first data byte is this sequence number plus 1.
  - If the SYN flag is clear, then this is the accumulated sequence number of the first data byte of this packet for the current session.
- Acknowledgment number (32 bits) –
  - If the ACK flag is set then this the next sequence number that the receiver is expecting.
  - This acknowledges receipt of all prior bytes (if any).

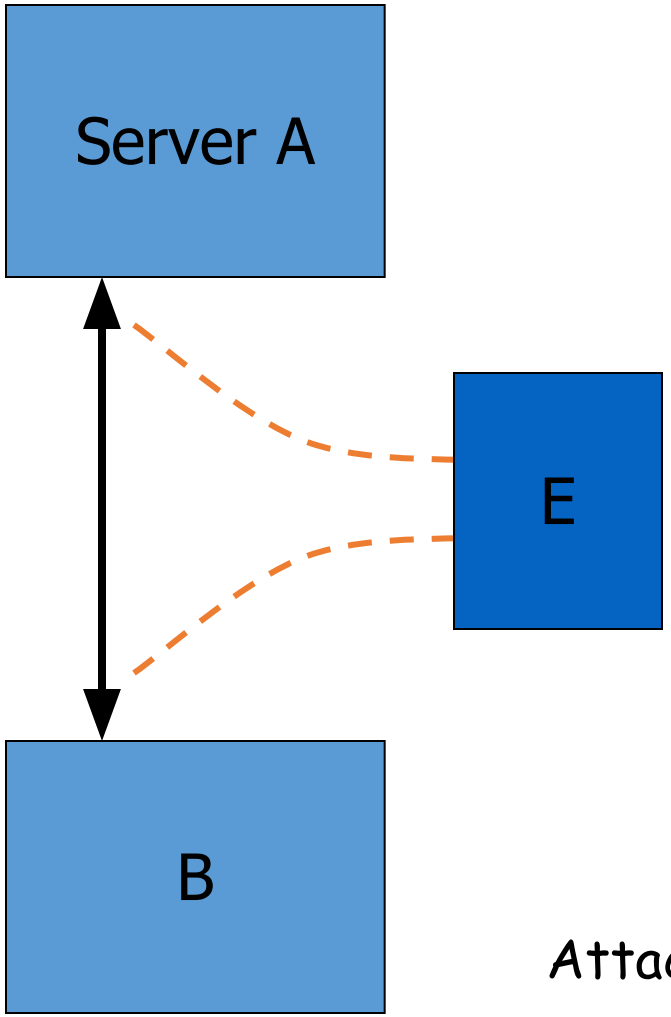
TCP Handshake



TCP sequence prediction attack

- Predict the sequence number used to identify the packets in a TCP connection, and then counterfeit packets.
- Adversary: do not have full control over the network, but can inject packets with fake source IP addresses
  - E.g., control a computer on the local network
- TCP sequence numbers are used for authenticating packets
- Initial seq# needs high degree of unpredictability
  - If attacker knows initial seq # and amount of traffic sent, can estimate likely current values
  - Some implementations are vulnerable

Blind TCP Session Hijacking



- A, B trusted connection
  - Send packets with predictable seq numbers
- E impersonates B to A
  - Opens connection to A to get initial seq number
  - DoS B's queue
  - Sends packets to A that resemble B's transmission
  - E cannot receive, but may execute commands on A

Attack can be blocked if E is outside firewall.

Risks from Session Hijacking

- Inject data into an unencrypted server-to-server traffic, such as an e-mail exchange, DNS zone transfers, etc.
- Inject data into an unencrypted client-to-server traffic, such as ftp file downloads, http responses.
- Spoof IP addresses, which are often used for preliminary checks on firewalls or at the service level.
- Carry out MITM attacks on weak cryptographic protocols.
  - often result in warnings to users that get ignored
- Denial of service attacks, such as resetting the connection.

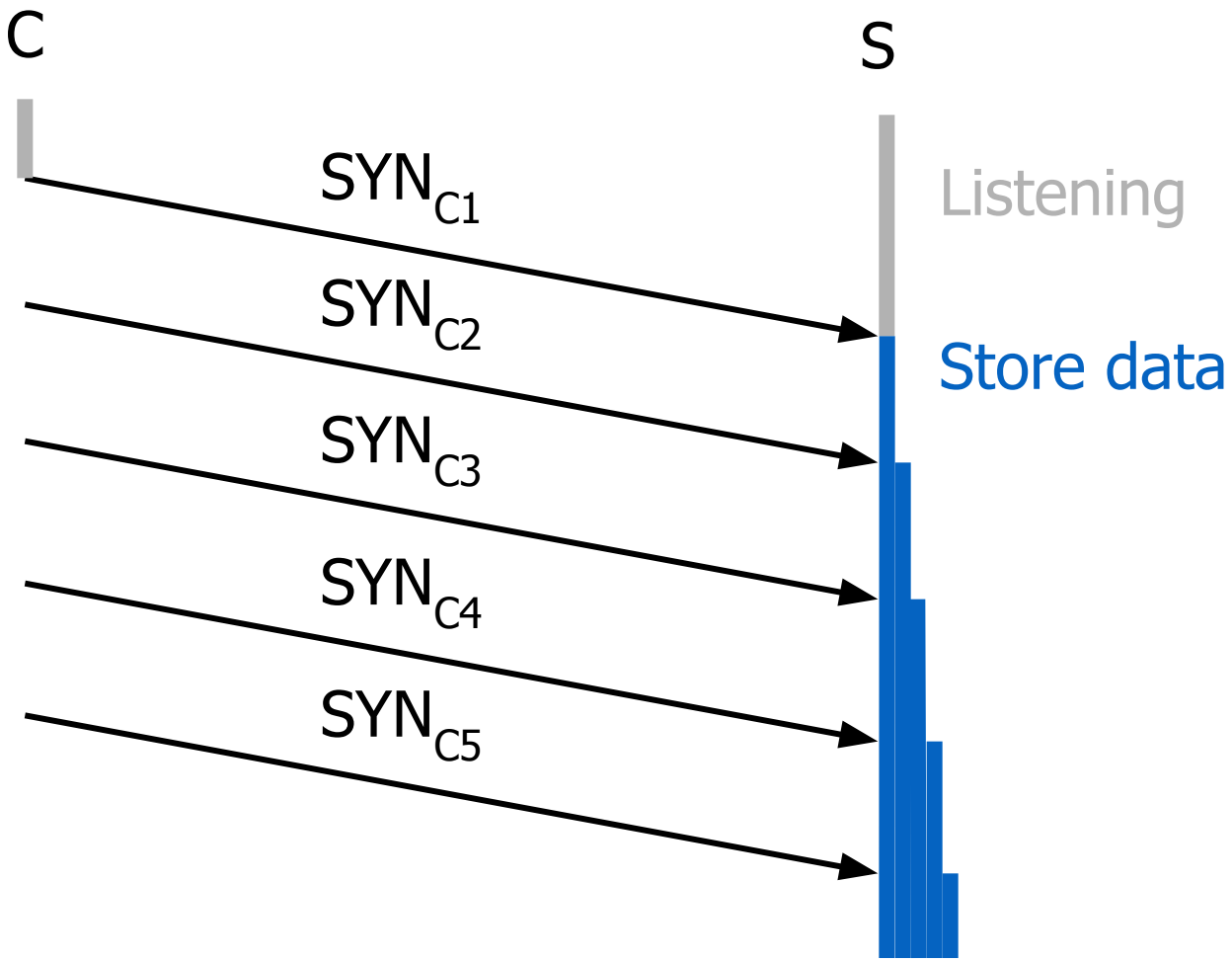
DoS vulnerability caused by session hijacking

- Suppose attacker can guess seq. number for an existing connection:
  - Attacker can send Reset packet to close connection. Results in DoS.
  - Naively, success prob. is $1/2^{32}$ (32-bit seq. #'s).
  - Most systems allow for a large window of acceptable seq. #'s
    - Much higher success probability.
- Attack is most effective against long lived connections, e.g. BGP.

Categories of Denial-of-service Attacks

	Stopping services	Exhausting resources
Locally	• Process killing • Process crashing • System reconfiguration	• Spawning processes to fill the process table • Filling up the whole file system • Saturate comm bandwidth
Remotely	• Malformed packets to crash buggy services	• Packet floods (Smurf, SYN flood, DDoS, etc)

SYN Flooding



SYN Flooding

- Attacker sends many connection requests
  - Spoofed source addresses
- Victim allocates resources for each request
  - Connection requests exist until timeout
  - Old implementations have a small and fixed bound on half-open connections
- Resources exhausted ⇒ requests rejected

- No more effective than other channel capacity-based attack today

TCP Syn Flood Attack

- TCP SYN flood is a type of Distributed Denial of Service (DDoS) attack that exploits part of the normal TCP three-way handshake to consume resources on the targeted server and render it unresponsive.
- Essentially, with SYN flood DDoS, the offender sends TCP connection requests faster than the targeted machine can process them, causing network saturation.

Attack description

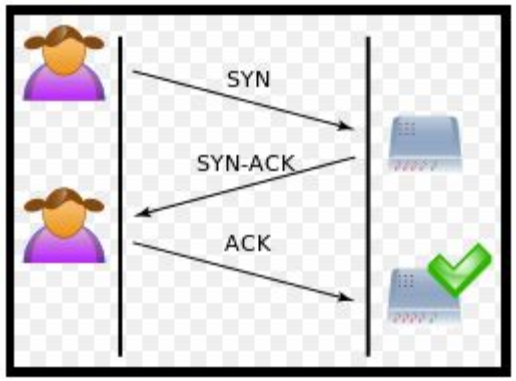
- When a client and server establish a normal TCP “three-way handshake,” the exchange looks like this:

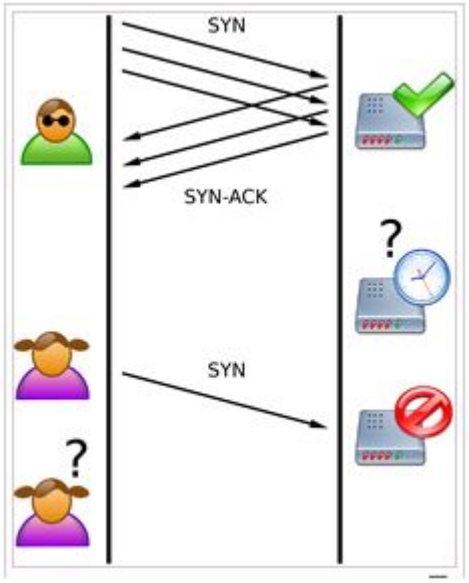
1. Client requests connection by sending SYN (synchronize) message to the server.
2. Server acknowledges by sending SYN-ACK (synchronize-acknowledge) message back to the client.
3. Client responds with an ACK (acknowledge) message, and the connection is established.

TCP Syn Flood Attack

- In a SYN flood attack, the attacker sends repeated SYN packets to every port on the targeted server, often using a fake IP address.
- The server, unaware of the attack, receives multiple, apparently legitimate requests to establish communication. It responds to each attempt with a SYN-ACK packet from each open port.
- The malicious client either does not send the expected ACK, or—if the IP address is spoofed—never receives the SYN-ACK in the first place. Either way, the server under attack will wait for acknowledgement of its SYN-ACK packet for some time.
- During this time, the server cannot close down the connection by sending an RST packet, and the connection stays open.
- Before the connection can time out, another SYN packet will arrive. This leaves an increasingly large number of connections half-open – and indeed SYN Food attacks are also referred to as “half-open” attacks.
- Eventually, as the server’s connection overflow tables fill, service to legitimate clients will be denied, and the server may even malfunction or crash

normal\Syn flood attack





IP Spoofing

- IP address spoofing is one of the most frequently used spoofing attack methods. In an IP address spoofing attack, an attacker sends IP packets from a false (or “spoofed”) source address in order to disguise itself.
- 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 is the action of masking a computer IP address so that it looks like it is authentic.
- During this masking process, the fake IP address sends what appears to be a malevolent message coupled with an IP address that appears to be authentic and trusted.
- In IP spoofing, IP headers are masked through a form of Transmission Control Protocol (TCP) in which spoofers discover and then manipulate vital information contained in the IP header such as IP address and source and destination information.

IP Spoofing

Non Blind Spoofing

Blind Spoofing

Man in the middle attack

- Both types of spoofing are forms of a common security violation known as a man in the middle (MITM) attack. In these attacks, a malicious party intercepts a legitimate communication between two friendly parties.
- The malicious host then controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient.
- In this way, an attacker can fool a victim into disclosing confidential information by “spoofing” the identity of the original sender, who is presumably trusted by the recipient.

DOS Attack

- IP spoofing is almost always used in what is currently one of the most difficult attacks to defend against – denial of service attacks, or DoS.
- Since crackers are concerned only with consuming bandwidth and resources, they need not worry about properly completing handshakes and transactions.
- Rather, they wish to flood the victim with as many packets as possible in a short amount of time. In order to prolong the effectiveness of the attack, they spoof source IP addresses to make tracing and stopping the DoS as difficult as possible.
- When multiple compromised hosts are participating in the attack, all sending spoofed traffic, it is very challenging to quickly block traffic.

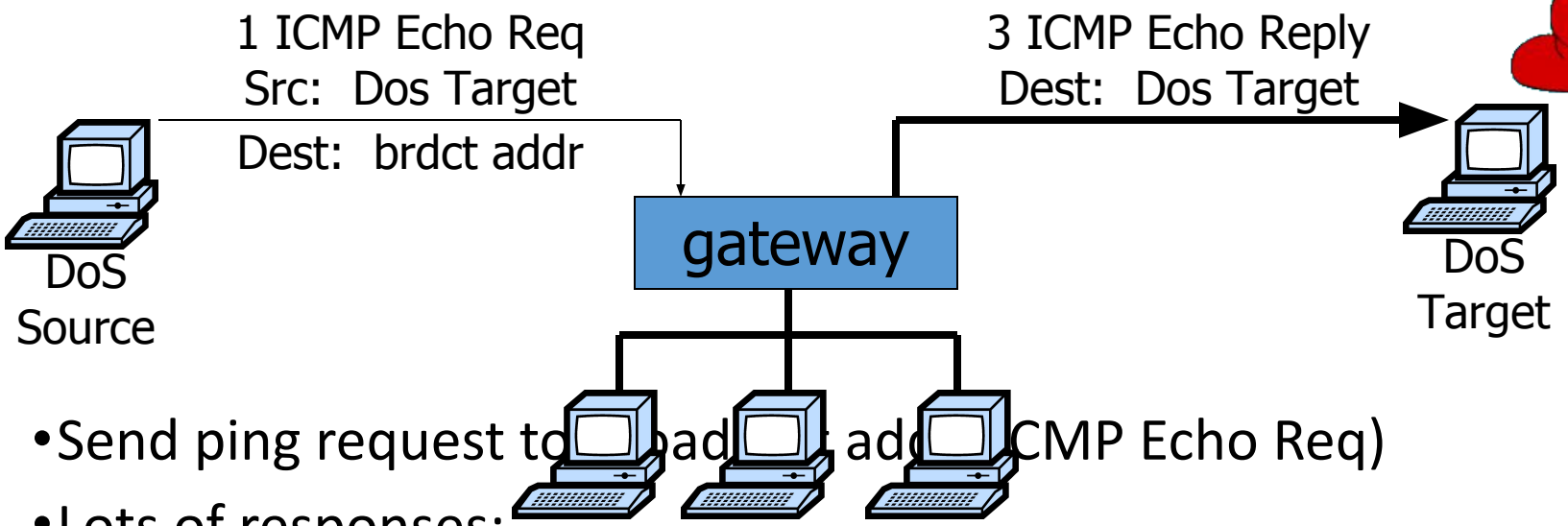
DNS spoofing attack

- The Domain Name System (DNS) is a system that associates domain names with IP addresses.
- Devices that connect to the internet or other private networks rely on the DNS for resolving URLs, email addresses and other human-readable domain names into their corresponding IP addresses.
- In a DNS server spoofing attack, a malicious party modifies the DNS server in order to reroute a specific domain name to a different IP address.
- In many cases, the new IP address will be for a server that is actually controlled by the attacker and contains files infected with malware. DNS server spoofing attacks are often used to spread computer worms and viruses.

User Datagram Protocol

- IP provides routing
  - IP address gets datagram to a specific machine
- UDP separates traffic by port (16-bit number)
  - Destination port number gets UDP datagram to particular application process, e.g., 128.3.23.3:53
  - Source port number provides return address
- Minimal guarantees
  - No acknowledgment
  - No flow control
  - No message continuation

Smurf DoS Attack



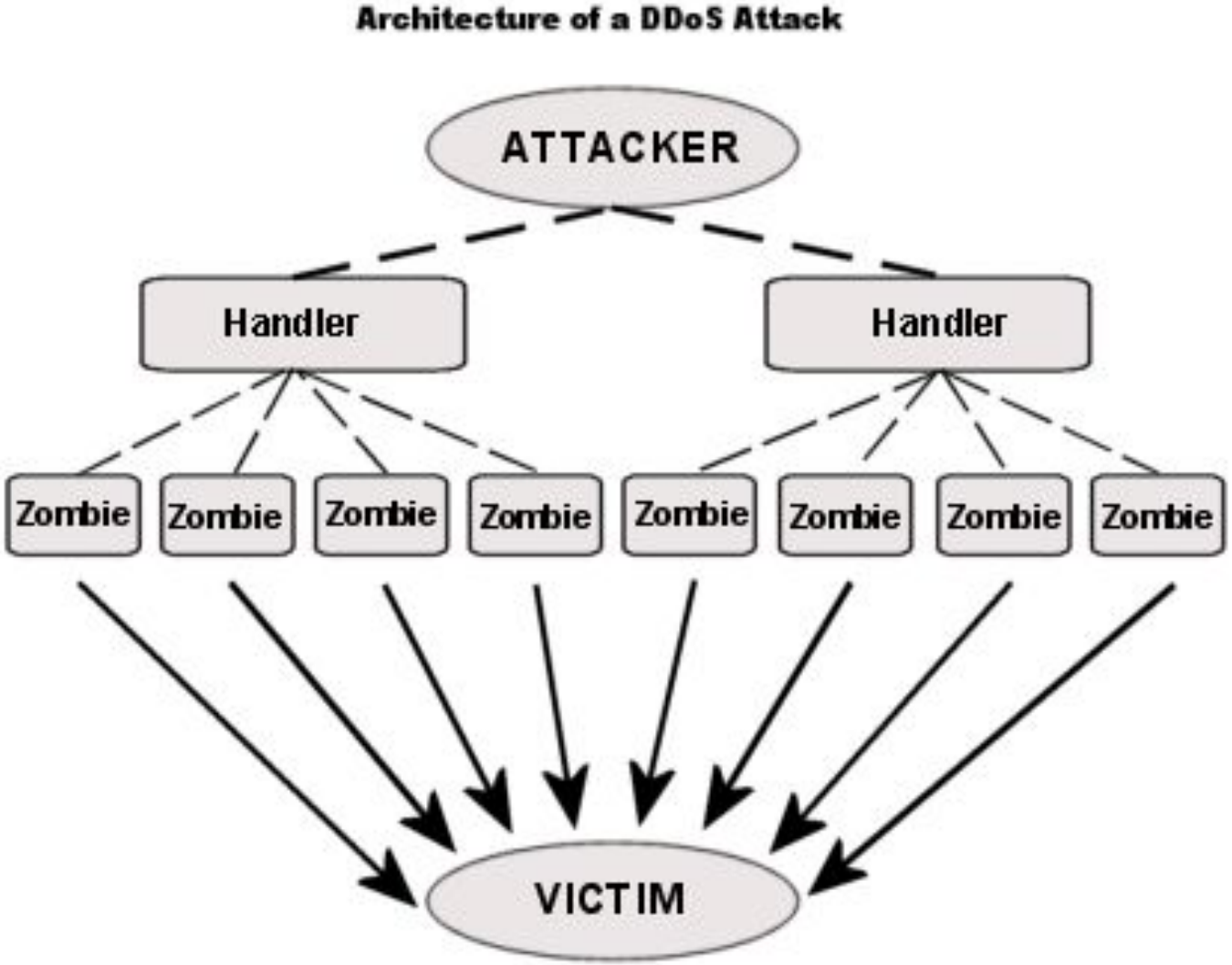
- Send ping request to broadcast address (ICMP Echo Req)
- Lots of responses:
  - Every host on target network generates a ping reply (ICMP Echo Reply) to victim
  - Ping reply stream can overload victim

Prevention: reject external packets to broadcast address

Internet Control Message Protocol

- Provides feedback about network operation
  - Error reporting
  - Reachability testing
  - Congestion Control
- Example message types
  - Destination unreachable
  - Time-to-live exceeded
  - Parameter problem
  - Redirect to better gateway
  - Echo/echo reply - reachability test

Distributed DoS (DDoS)



Hiding DDoS Attacks

- Reflection
  - Find big sites with lots of resources, send packets with spoofed source address, response to victim
    - PING => PING response
    - SYN => SYN-ACK
- Pulsing zombie floods
  - each zombie active briefly, then goes dormant;
  - zombies taking turns attacking
  - making tracing difficult

Cryptographic network protection

- Solutions above the transport layer
  - Examples: SSL and SSH
  - Protect against session hijacking and injected data
  - Do not protect against denial-of-service attacks caused by spoofed packets
- Solutions at network layer
  - Use cryptographically random ISNs [RFC 1948]
  - More generally: IPsec
  - Can protect against
    - session hijacking and injection of data.
    - denial-of-service attacks using session resets.