Project 1: ?? days left

Exploitation of Network Vulnerabilities

CS 459/559: Science of Cyber Security
4th Lecture

Instructor:

Guanhua Yan

Project 1

- Demonstrate a cyber attack
- In your first project report, please:
  - Explain what are the vulnerability, exploit, attack surface, and attack vector in your project
  - Explain how the attack (exploitation) occurs in your project
  - Explain why the attack (exploitation) works
- Each of your project reports should be **at least five pages, excluding bibliography**
- **Due time: TBD**
  - **Four days of grace period, with 2.5% penalty each half day late**
- **Grading criteria:**
  - **Results, novelty, difficulty, presentation**

How to choose your project?

- You should be familiar with the concepts behind the attack (software, network, system, or human)
- You should be comfortable with developing defensive mechanisms on the target being attacked
- It's OK to work on known vulnerabilities using known exploit code (plenty of online resources)
  - Known vulnerabilities: National Vulnerability Database (https://nvd.nist.gov)
  - Known exploits: Exploit database (https://exploit-db.com)
  - The metasploit framework (https://www.metasploit.com) makes exploitation easy!
- Start early and work on your project hard!
  - **You will continue working on the same project with defensive techniques introduced in this course**

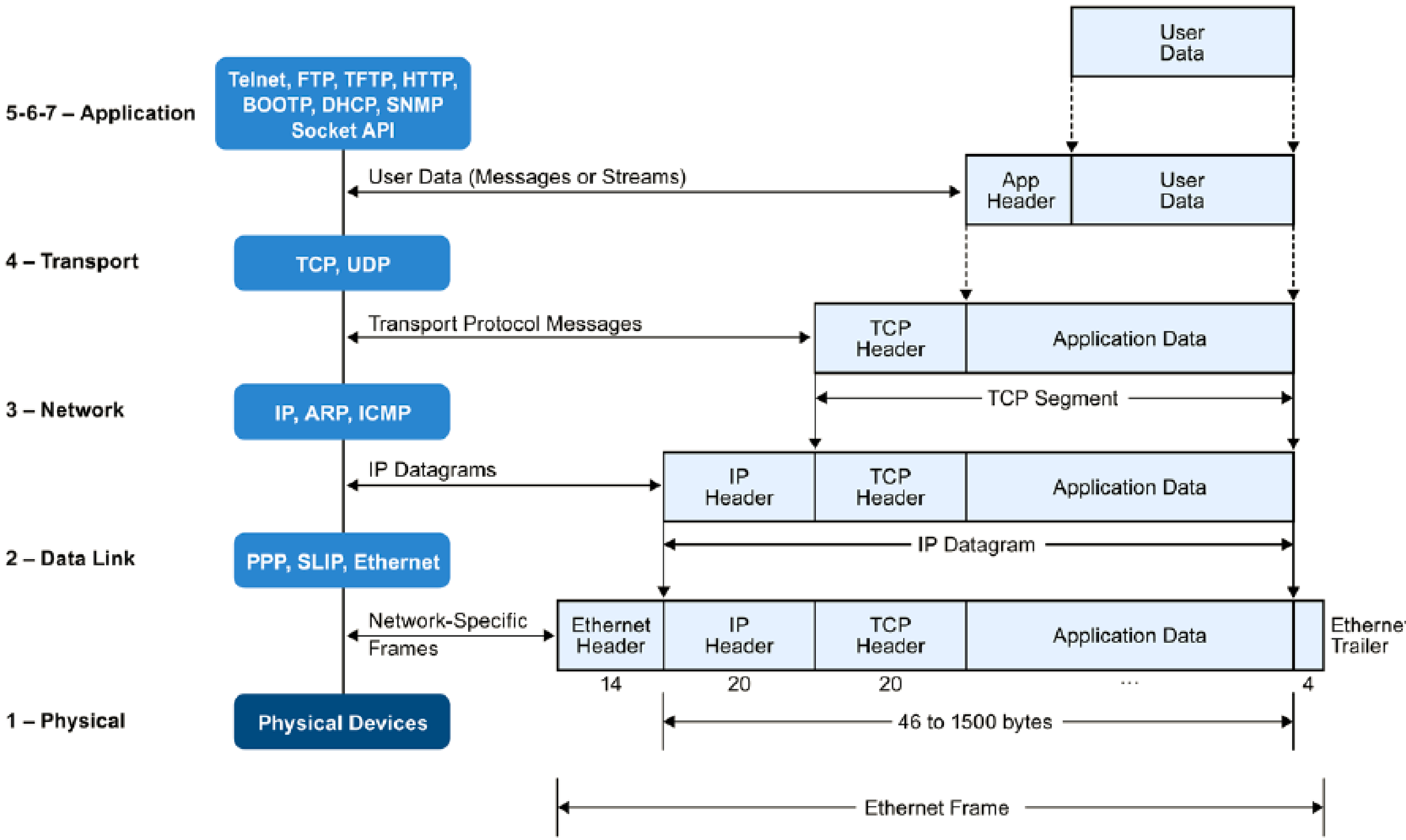
What we have learned from last lecture

- **Reconnaissance tools**
  - Public/online reconnaissance tools
  - Port scanners
  - Network mappers
  - Operating system detection tools
  - Firewall analysis tools
  - Vulnerability scanners
  - Packet sniffers
  - Wireless sniffers

Exploitation of network vulnerabilities

Reconnaissance	• Harvesting email addresses, conference information, etc.
Weaponization	• Coupling exploit with backdoor into deliverable payload
Delivery	• Delivering weaponized bundle to the victim via email, web, USB, etc.
Exploitation	• Exploiting a vulnerability to execute code on victim's system
Installation	• Installing malware on the asset
Command & Control (C2)	• Command channel for remote manipulation of victim
Actions on Objectives	• With "Hands on Keyboard" access, intruders accomplish their original goal

Network protocol stack



Types of network vulnerability exploitation

- Cache poisoning attacks
- Sniffing
- Denial of service attacks

Cache poisoning attacks

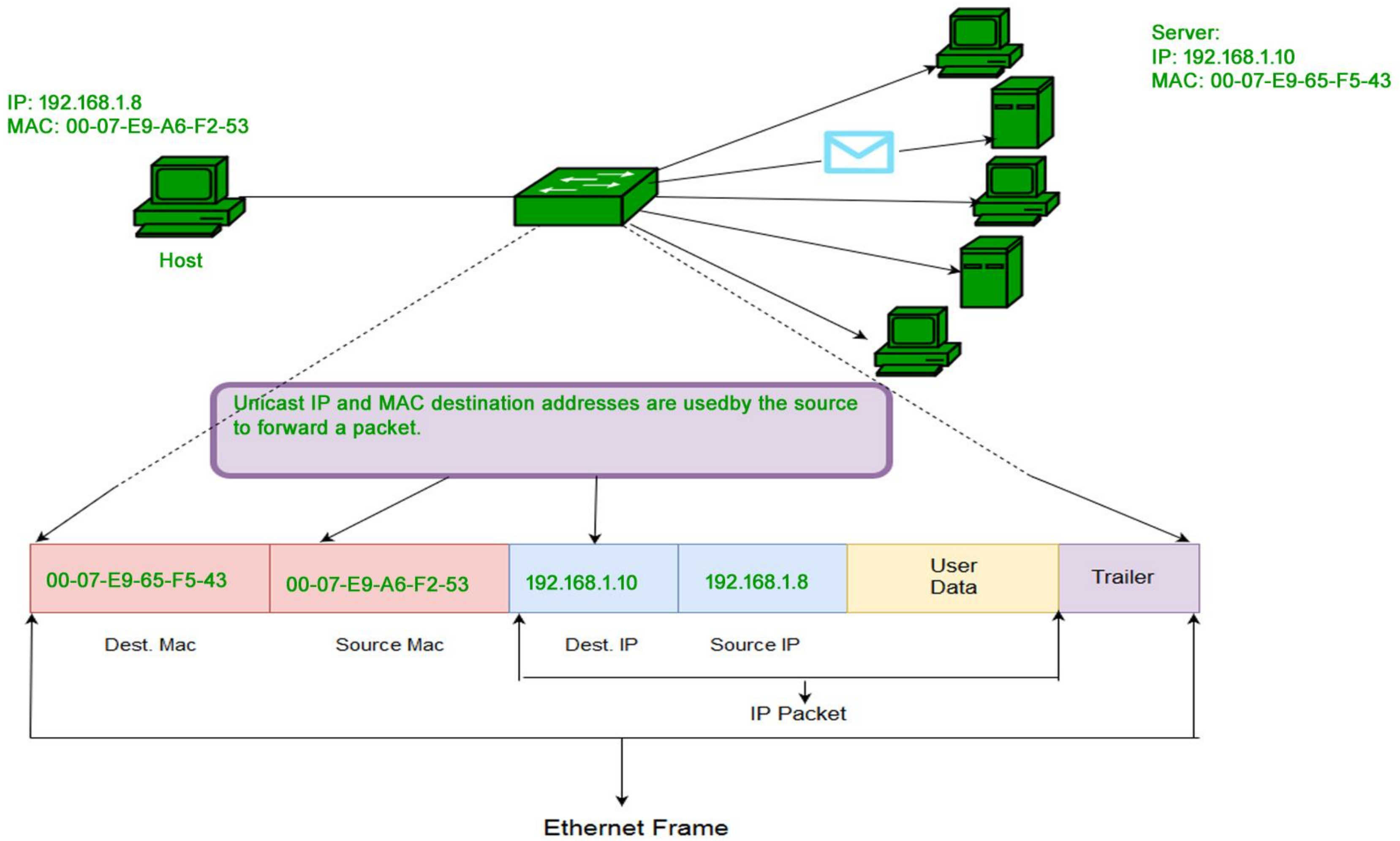
Caching in distributed systems

- Hide latency to improve performance for repeated accesses
- Places to use caching:
  - Server’s disk
  - Server’s buffer cache (memory)
  - Client’s buffer cache (memory)
  - Client’s disk
- **Cache consistency problems** in distributed systems
- Two well-known cache poisoning attacks
  - ARP poisoning
  - DNS poisoning

ARP: Address Resolution Protocol

- ARP turns an IP number into an Ethernet number, very important.
- Instead of asking “Who’s Bob?” you ask “Who’s 172.19.4.15” and if you get a reply, associate the Ethernet address with the IP address in **your ARP table**, and now you can keep sending your data to the intended recipient via the correct Ethernet address.
- **Remember: the only packet you can actually send on Ethernet is an Ethernet packet, everything else has to be stuffed inside it.**

IP & MAC Addresses



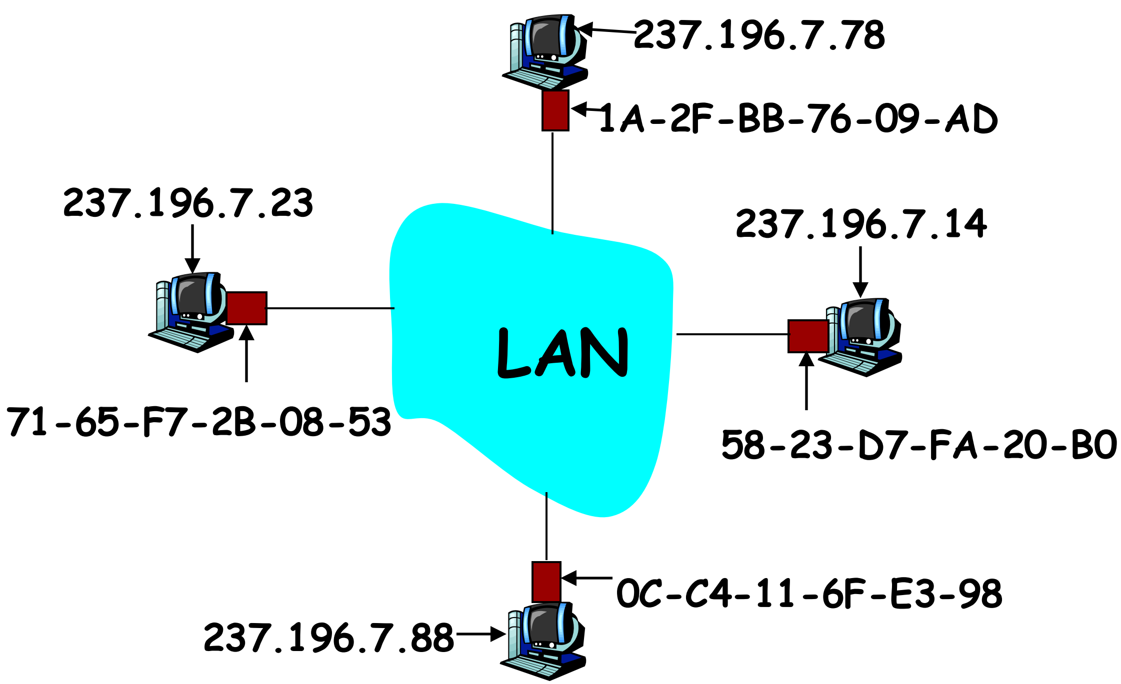
MAC Addresses and ARP

- **32-bit IP address:**
  - *network-layer* address
  - used to get datagram to destination IP subnet

- **MAC (or LAN or physical or Ethernet) address:**
  - Data link layer address
  - used to get datagram from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs) burned in the adapter ROM
  - Some Network interface cards (NICs) can change their MAC

ARP: Address Resolution Protocol

Question: how to determine MAC address of host B when knowing B's IP address?



- Each IP node (Host, Router) on LAN has ARP table
- **ARP Table**: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>

  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP Protocol

- ARP works by broadcasting requests and **caching** responses for future use
- The protocol begins with a computer broadcasting a message of the form

  who has <IP address1> tell <IP address2>

- When the machine with <IP address1> or an ARP server receives this message, its broadcasts the response

  <IP address1> is <MAC address>

- The requestor's IP address <IP address2> is contained in the link header

ARP Table

- The Linux and Windows command arp - a displays the ARP table

Internet Address	Physical Address	Type
128.148.31.1	00-00-0c-07-ac-00	dynamic
128.148.31.15	00-0c-76-b2-d7-1d	dynamic
128.148.31.71	00-0c-76-b2-d0-d2	dynamic
128.148.31.75	00-0c-76-b2-d7-1d	dynamic
128.148.31.102	00-22-0c-a3-e4-00	dynamic
128.148.31.137	00-1d-92-b6-f1-a9	dynamic

ARP Spoofing

- The ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other
- A rogue machine can spoof other machines

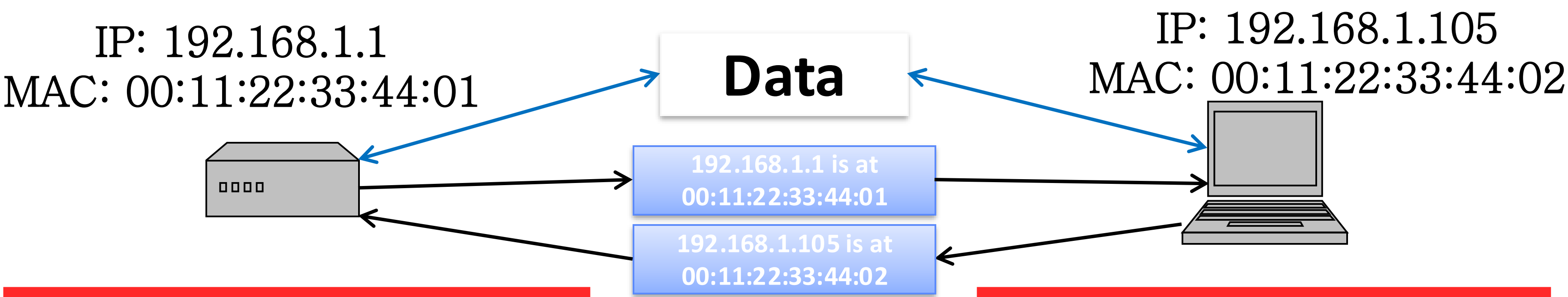
ARP Poisoning (ARP Spoofing)

- According to the standard, almost all ARP implementations are stateless

- **An arp cache updates every time that it receives an arp reply… even if it did not send any arp request!**

- It is possible to “poison” an arp cache by sending gratuitous arp replies

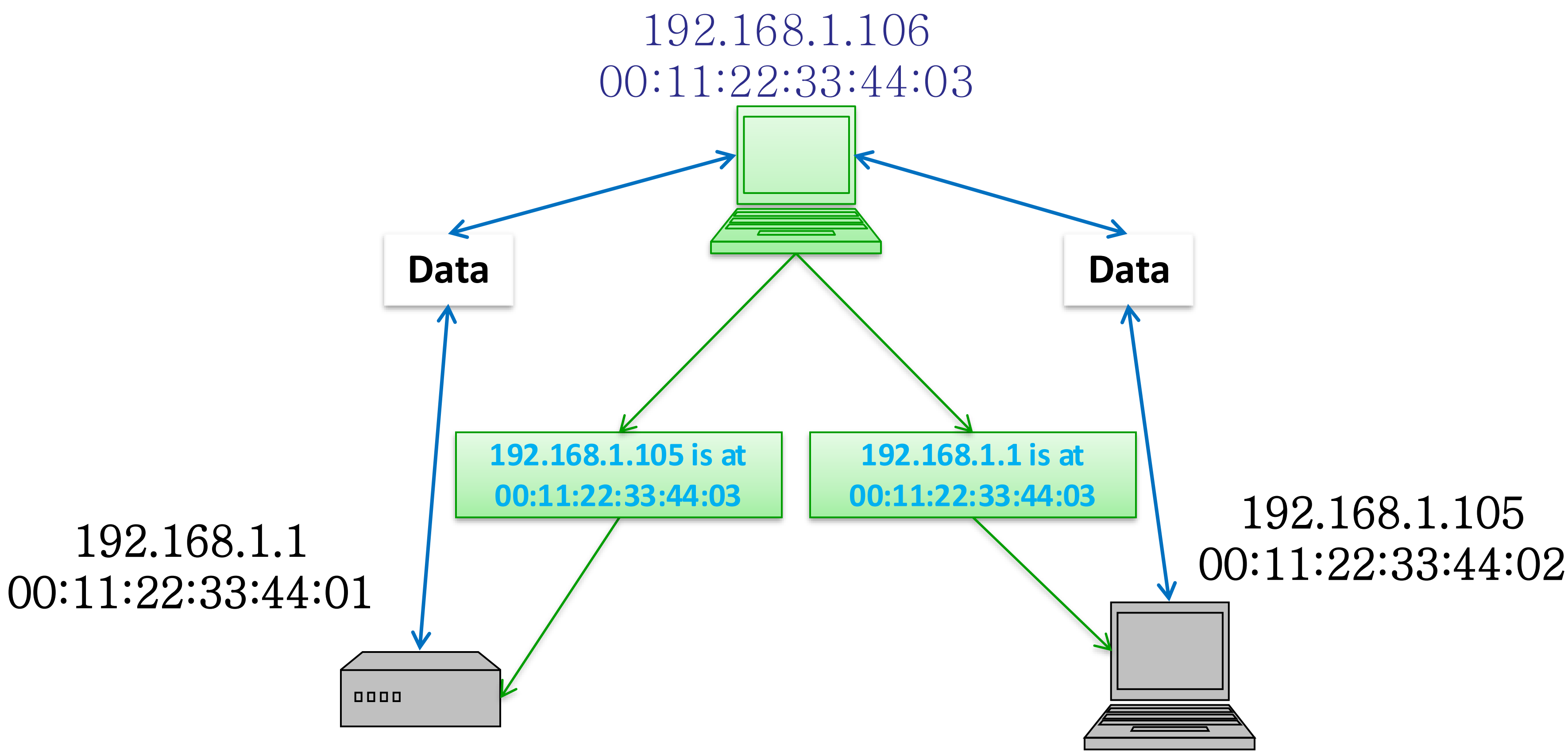
ARP Caches



ARP Cache	
192.168.1.105	00:11:22:33:44:02

ARP Cache	
192.168.1.1	00:11:22:33:44:01

Poisoned ARP Caches (man-in-the-middle attack)



Poisoned ARP Cache	
192.168.1.105	00:11:22:33:44:03

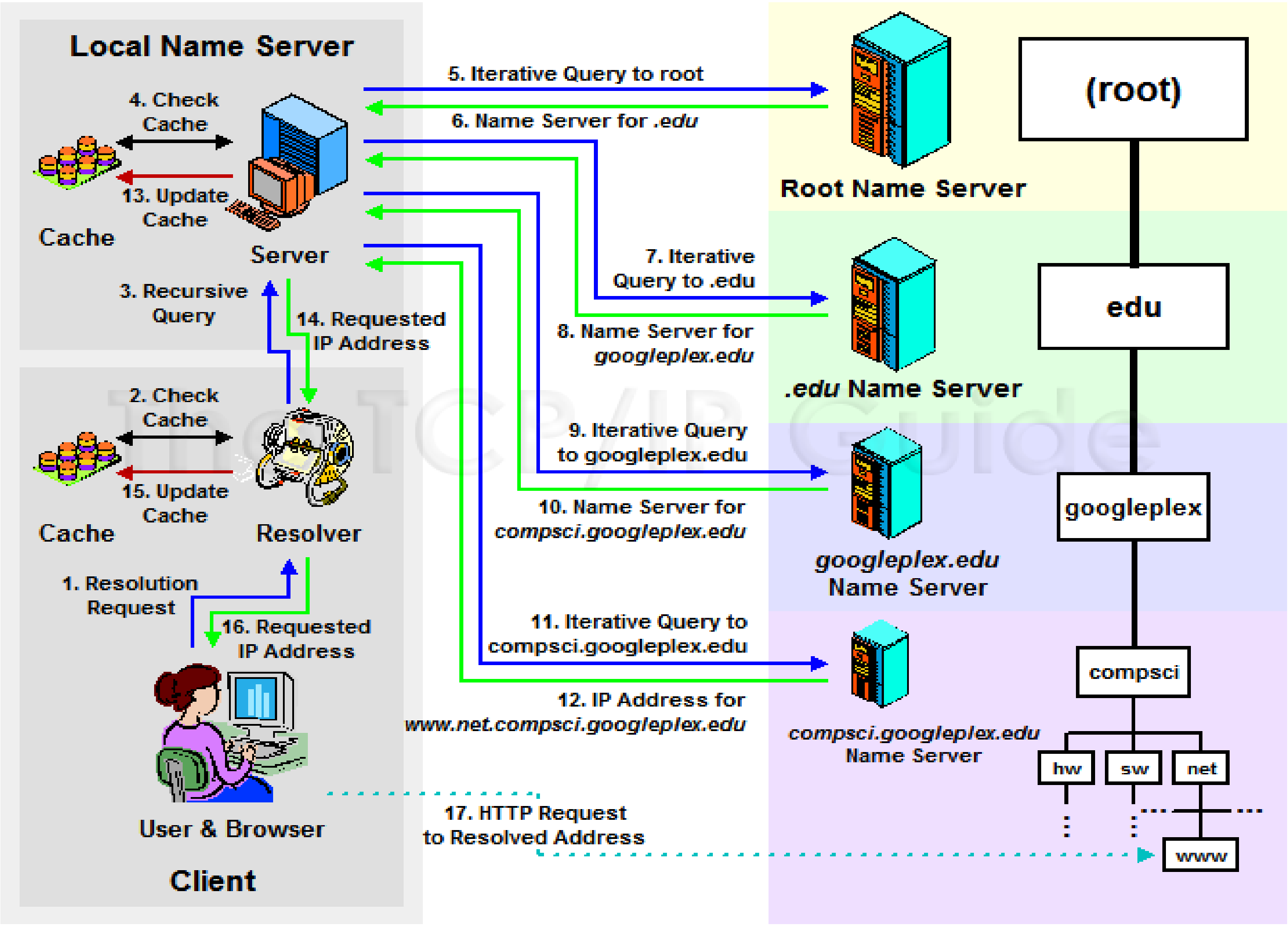
Poisoned ARP Cache	
192.168.1.1	00:11:22:33:44:03

Domain Name Resolution (DNS)

- DNS does **translation between IP addresses and domain names** (which are easy to remember)

- Given domain name bravo.cs.binghamton.edu:
  - the first (or top) level domain (.edu, for educational institutions)
  - the second level domain (binghamton)
  - the third level domain (cs)
  - the actual host's name (bravo)

DNS resolution



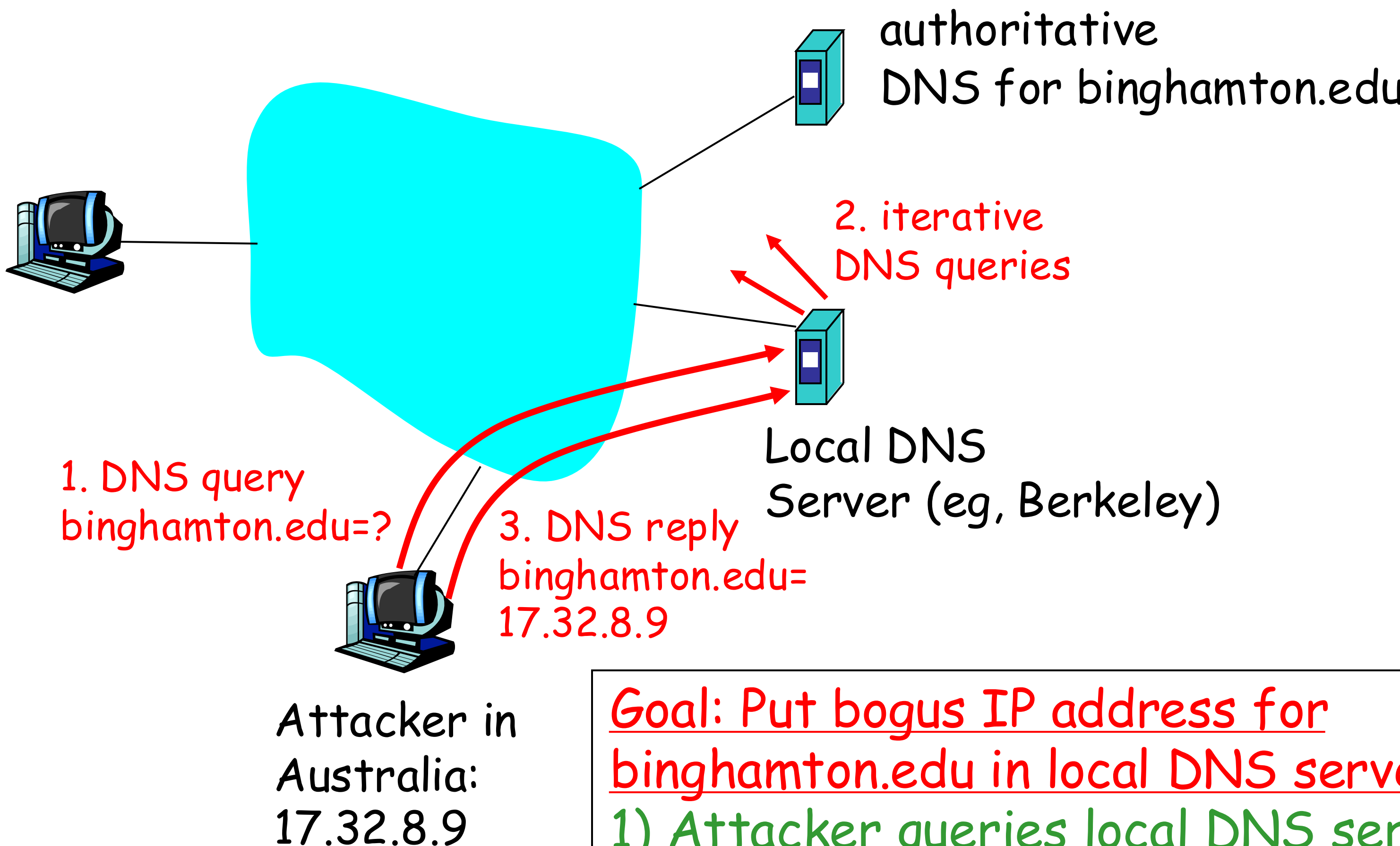
Poisoning DNS Cache (1)

- **Poisoning**: Attempt to put bogus records into DNS name server caches
  - Bogus records could point to attacker nodes
  - Attacker nodes could phish

- But unsolicited replies are not accepted at a name server.
  - Name servers use IDs in DNS messages to match replies to queries
  - So can't just insert a record into a name server by sending a DNS reply message.

- But can send a reply to a request.

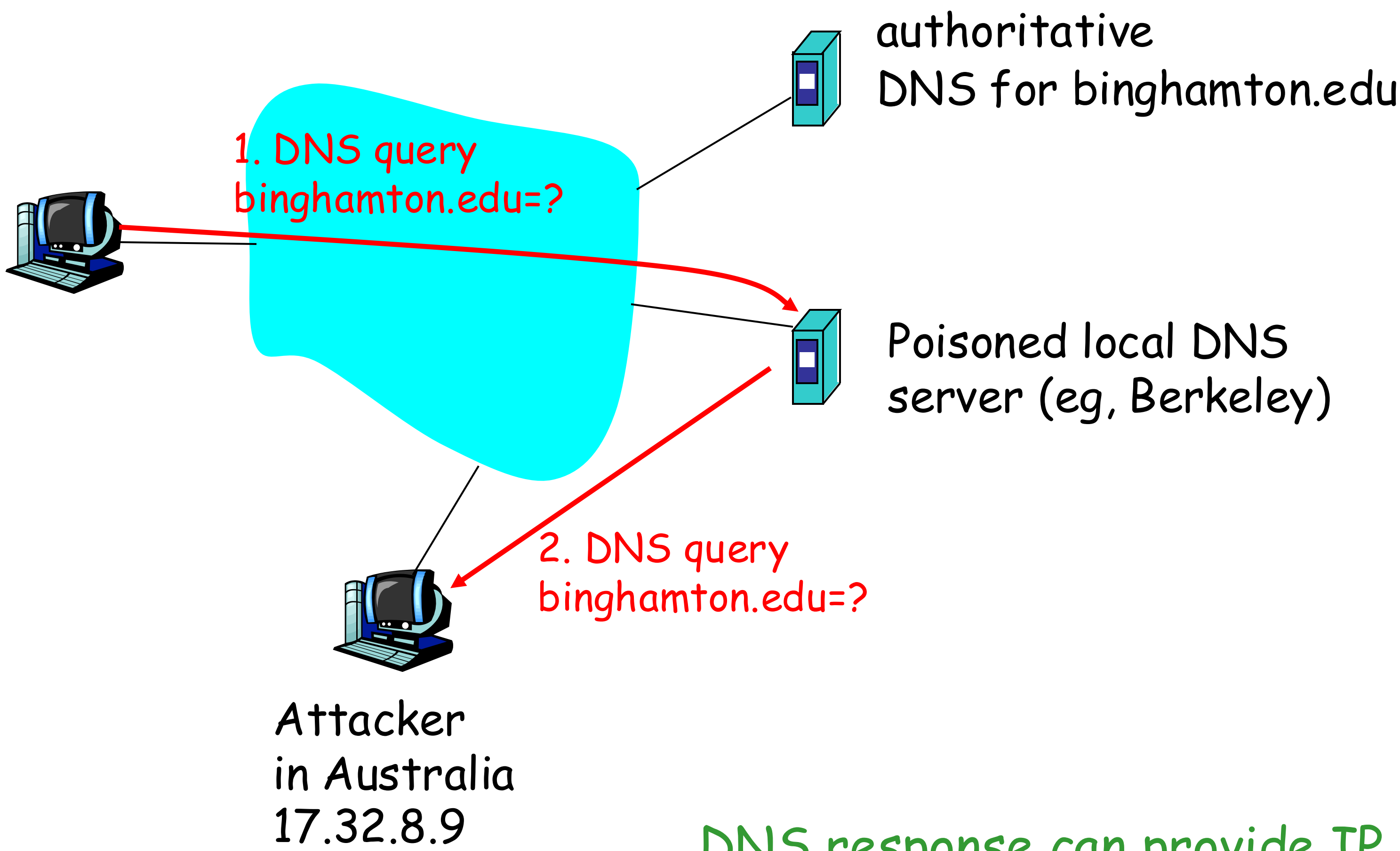
Poisoning local DNS server (2)



Goal: Put bogus IP address for binghamton.edu in local DNS server

1) Attacker queries local DNS server
2) Local DNS makes iterative queries
3) Attacker waits for some time; sends a bogus reply, spoofing authoritative server for binghamton.edu.

Poisoning local DNS server (3)



DNS response can provide IP address of malicious server!

DNS Poisoning (4)

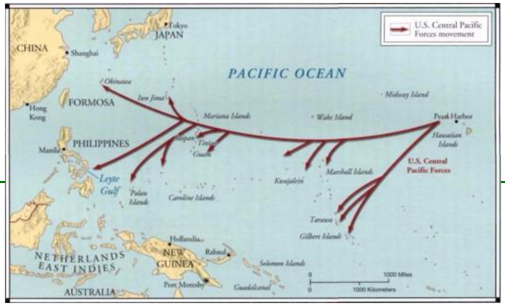
- Issues:
  - **Attacker may need to stop upstream name server from responding**
    - So that server under attack doesn’t get suspicious
    - Denial of service attacks against that server

Sniffing

Sniffing

- Attacker is inside firewall
- Requirements
  - Attacker's host connected to shared medium
  - NIC should be in "promiscuous mode"
    - processes all frames that come to NIC
- Sniffer has two components
  - Capture
  - Packet analysis

- Grab and file away:
  - userids and passwords
  - credit card numbers
  - secret e-mail conversations
- Island hopping attack:
  - Take over single machine (e.g. virus)
  - Install sniffer, observe passwords, take over more machines, install sniffers



Passive sniffing

- Easy to sniff:
  - 802.11 traffic
  - Ethernet traffic passing through a hub
    - Any packets sent to hub is broadcast to all interfaces
    - Not true for a switch
  - Cable modem traffic

- Popular sniffers
  - Wireshark
  - tcpdump (for unix)
  - Snort (sniffing and intrusion detection)

Hubs

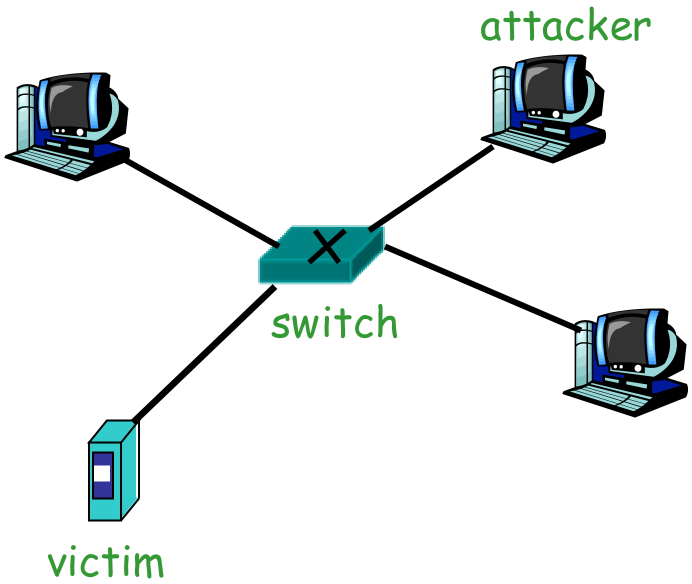
- **Hubs are shared media devices**.
- Everyone sees everyone's packets, you're only supposed to pay attention to those specifically directed to you, or to broadcasts.
- Not too secure, but cheap.
- Most wireless still qualifies as a "hub," while actual wired Ethernet hubs are becoming hard to find.

Switches

- **Switches aren't shared, most of the time.**
- The switch pays attention to the packets and makes a list of the **"sender" Ethernet addresses and makes a table** (it removes old data after a while).
- When a packet comes along whose destination address is in the table (because that host has recently "talked" and identified itself) the packet only goes to that port.
- **Unknown packets and broadcasts still go to all ports**, but overall, there are nearly no collisions and is generally more secure.
- Switches are now much more common than hubs.

Active Sniffing through a switch

How does attacker sniff packets sent to/from the victim?



Have to get victim's packets to attacker!

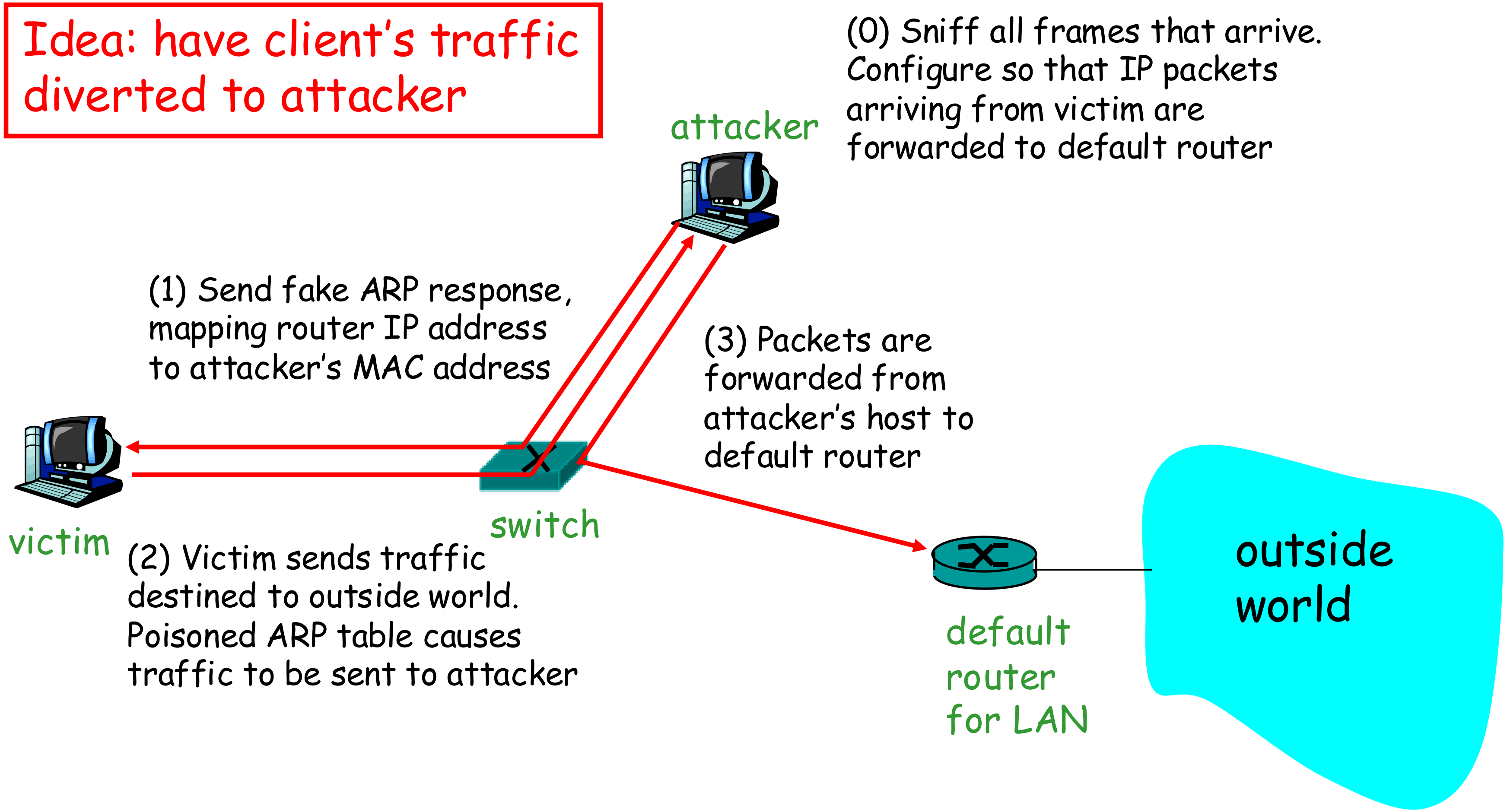
Sniffing through switch: flooding memory

- Host sends **flood of Ethernet frames with random source MAC addresses**
  - Fill switch's forwarding table with bogus MAC addresses
  - When "good packet arrives," destination MAC address not in switch memory
  - Switch broadcasts real packets to all links
- Sniff all the broadcast packets

MAC Address	Port
MAC 1	Port 1
MAC 2	Port 2
MAC 3	Port 3
…	…

Sniffing through LAN: poison victim's ARP table approach

Idea: have client's traffic diverted to attacker



Sniffing tool: dSniff

(https://www.monkey.org/~dugsong/dsniff/)

monkey.org

dsniff

latest release: dsniff-2.3.tar.gz (CHANGES)
beta snapshots

Abstract

dsniff is a collection of tools for network auditing and penetration testing. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.). arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI.

Sniffing tool: Ettercap

(https://www.ettercap-project.org)

Denial of service attacks

Denial-of-Service

Prevent access by legitimate users or stop critical system processes

- **Connection flooding attack**
  - Overwhelming connection queue with TCP SYN flood
- **Bandwidth flooding attack**
  - Overwhelming communications link with packets
  - Strength in flooding attack lies in volume rather than content
- **Implementation vulnerability attack**
  - Send a few crafted messages to target app that has vulnerability
  - Malicious messages called the “exploit”
  - Remotely stopping or crashing services

DoS and DDoS

- DoS:
  - source of attack small # of nodes
  - source IP typically spoofed
- DDoS
  - From thousands of nodes
  - IP addresses often not spoofed

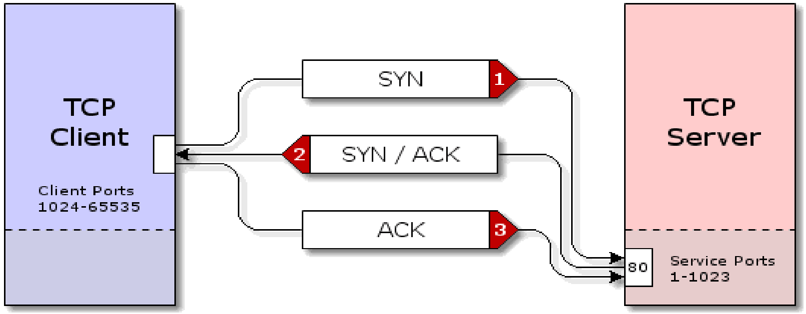
Denial-of-Service

Prevent access by legitimate users or stop critical system processes

- **Connection flooding attack**
  - Overwhelming connection queue with SYN flood
- **Bandwidth flooding attack**
  - Overwhelming communications link with packets
  - Strength in flooding attack lies in volume rather than content
- **Implementation Vulnerability attack**
  - Send a few crafted messages to target app that has vulnerability
  - Malicious messages called the “exploit”
  - Remotely stopping or crashing services

Connection flooding: Overwhelming connection queue w/ SYN flood

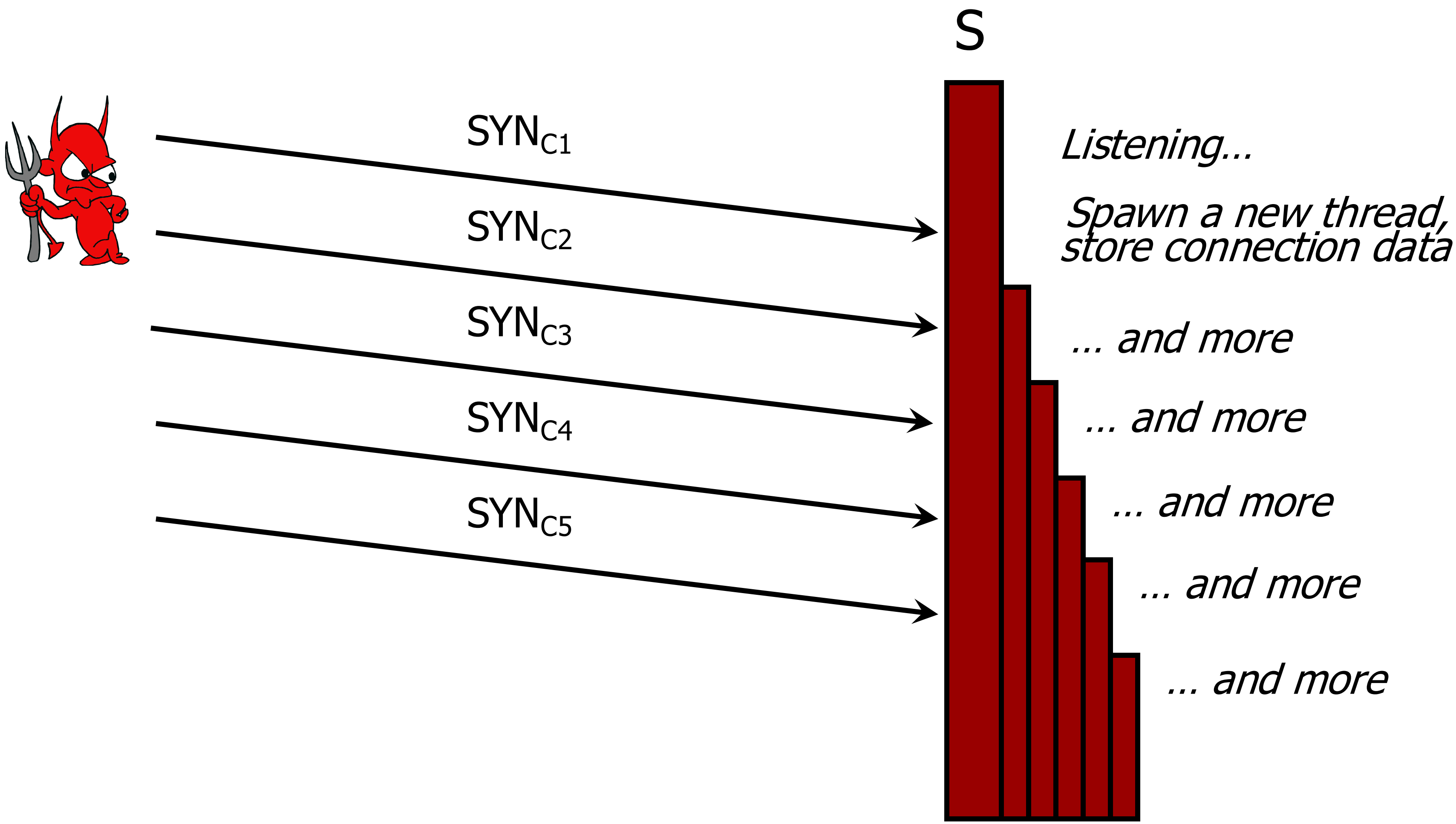
- Recall client sends SYN packet with initial seq. number when initiating a connection.
- TCP on server machine allocates memory on its connection queue, to track the status of the new half-open connection.
- For each half-open connection, server waits for ACK, using a timeout that is often > 1 minute



- **Attack:** Send many SYN packets, filling connection queue with half-open connections.
  - Can spoof source IP address!
- When connection queue is exhausted, no new connections can be initiated by legit users.

Need to know of open port on victim's machine: Port scanning.

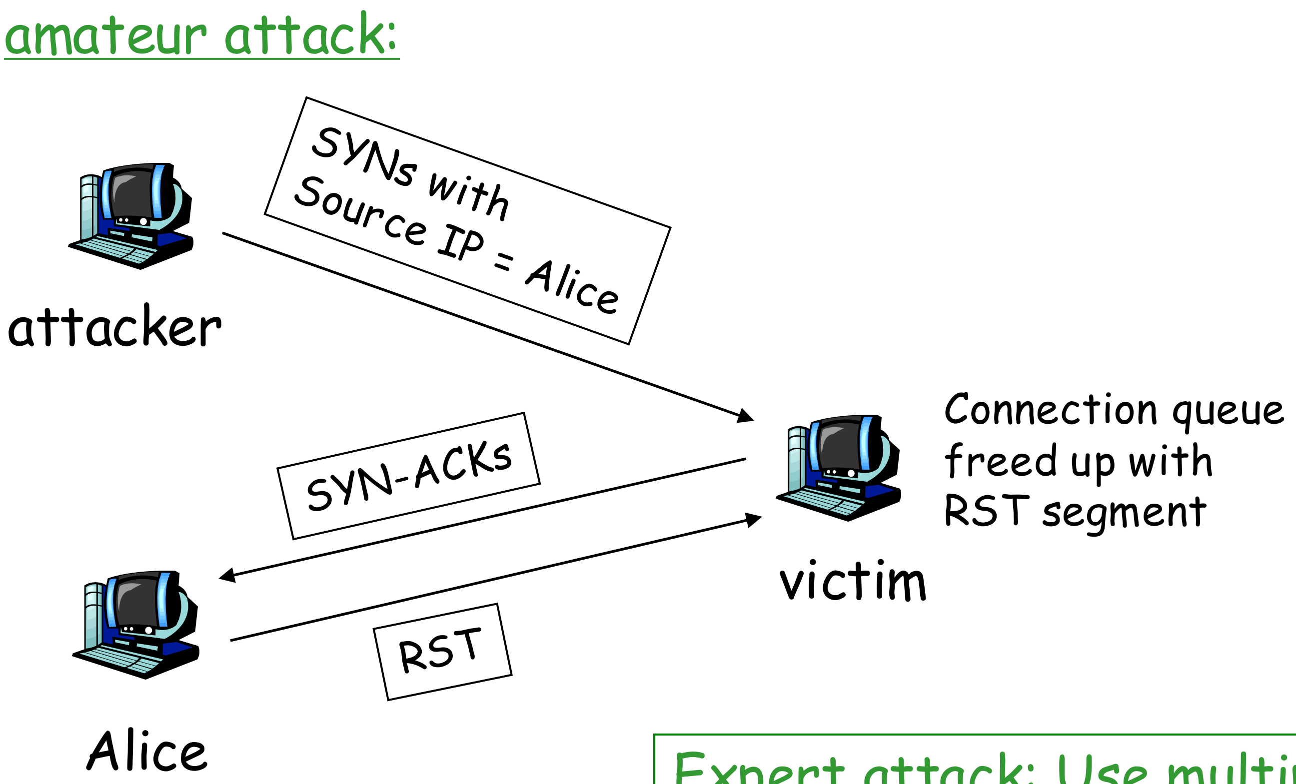
SYN Flooding Attack



SYN Flooding Explained

- Attacker sends many connection requests (SYNs) with spoofed source addresses
- Victim allocates resources for each request
  - New thread, connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a classic denial of service attack
  - Common pattern: it costs nothing to TCP client to send a connection request, but TCP server must spawn a thread for each request - asymmetry!

SYN flood Issue



Expert attack: Use multiple source IP addresses, each from unresponsive addresses.

Denial-of-Service

Prevent access by legitimate users or stop critical system processes

- **Connection flooding attack**
  - Overwhelming connection queue with SYN flood
- **Bandwidth flooding attack**
  - Overwhelming communications link with packets
  - Strength in flooding attack lies in volume rather than content
- **Implementation Vulnerability attack**
  - Send a few crafted messages to target app that has vulnerability
  - Malicious messages called the “exploit”
  - Remotely stopping or crashing services

Overwhelming link bandwidth with packets

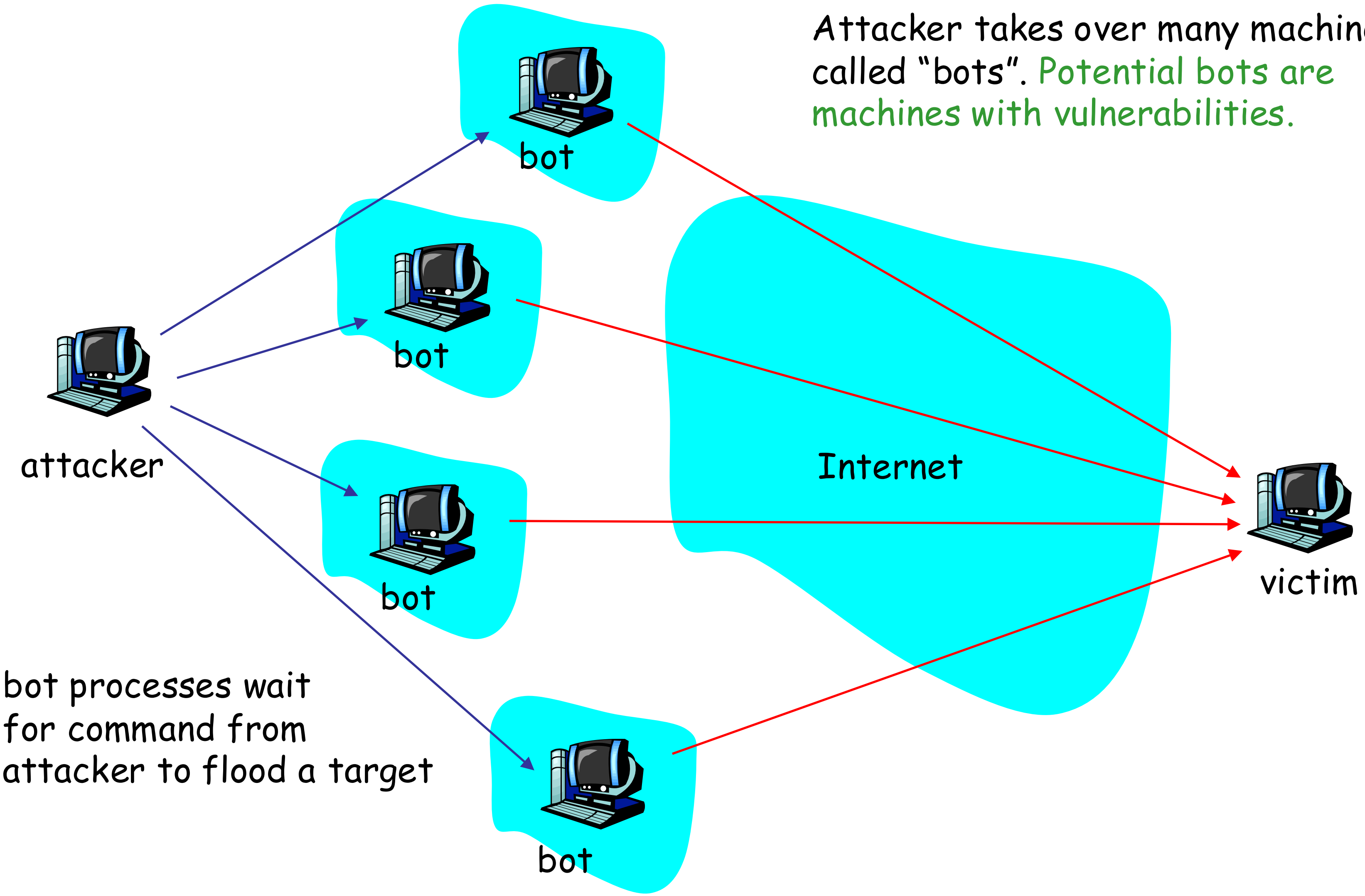
- Attack traffic can be made similar to legitimate traffic, hindering detection.

- Flow of traffic must consume target’s bandwidth resources.
  - Attacker needs to engage more than one machine => DDoS

- May be easier to get target to fill-up its upstream bandwidth: async access

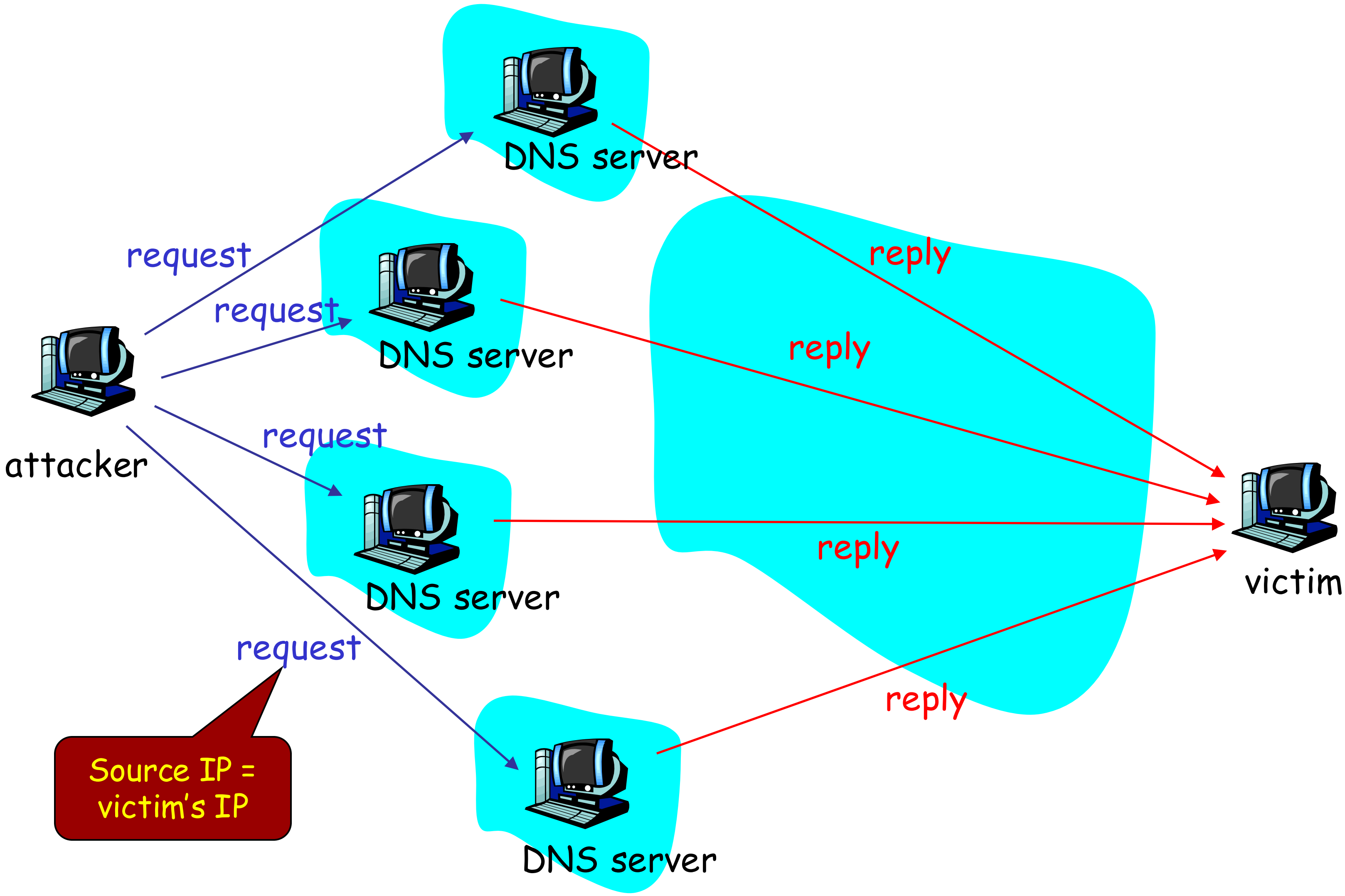
Distributed DoS: DDoS

Attacker takes over many machines, called "bots". Potential bots are machines with vulnerabilities.

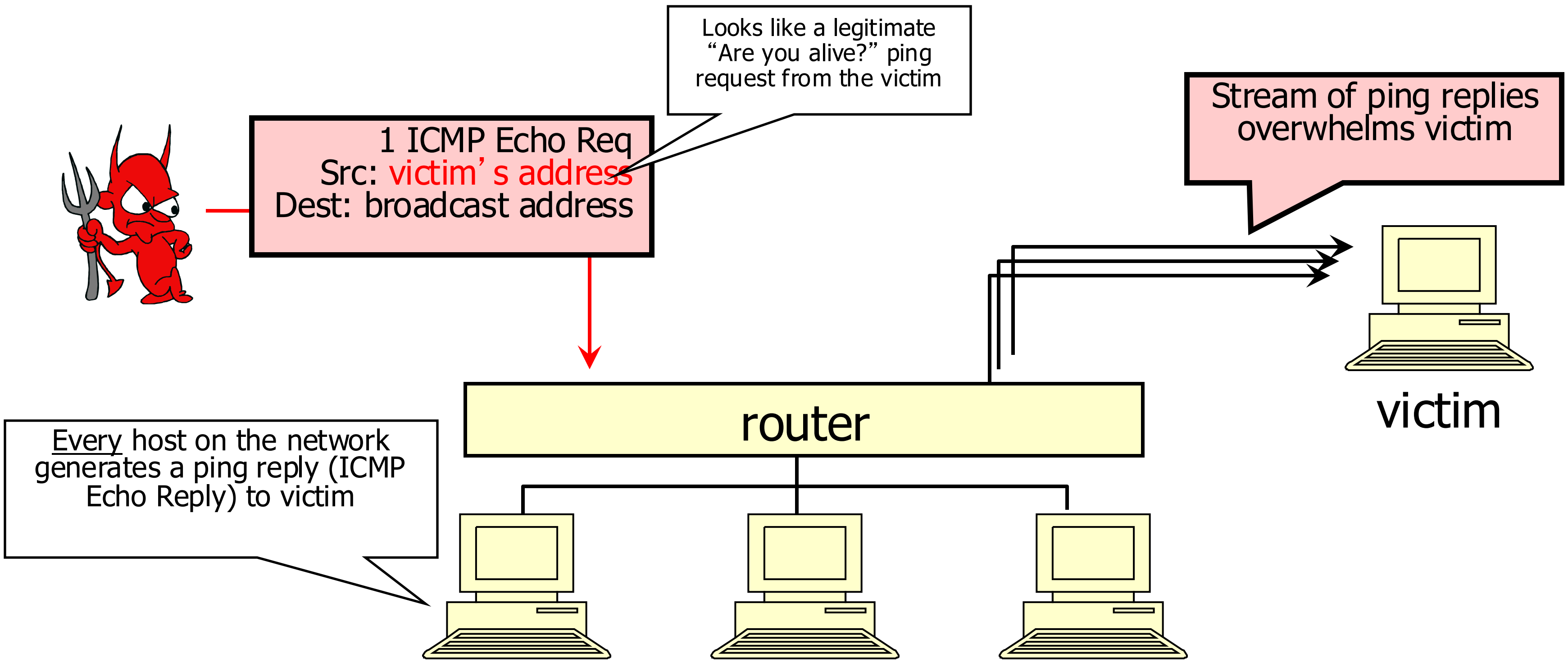


bot processes wait for command from attacker to flood a target

DDoS: Reflection attack



“Smurf” Attack



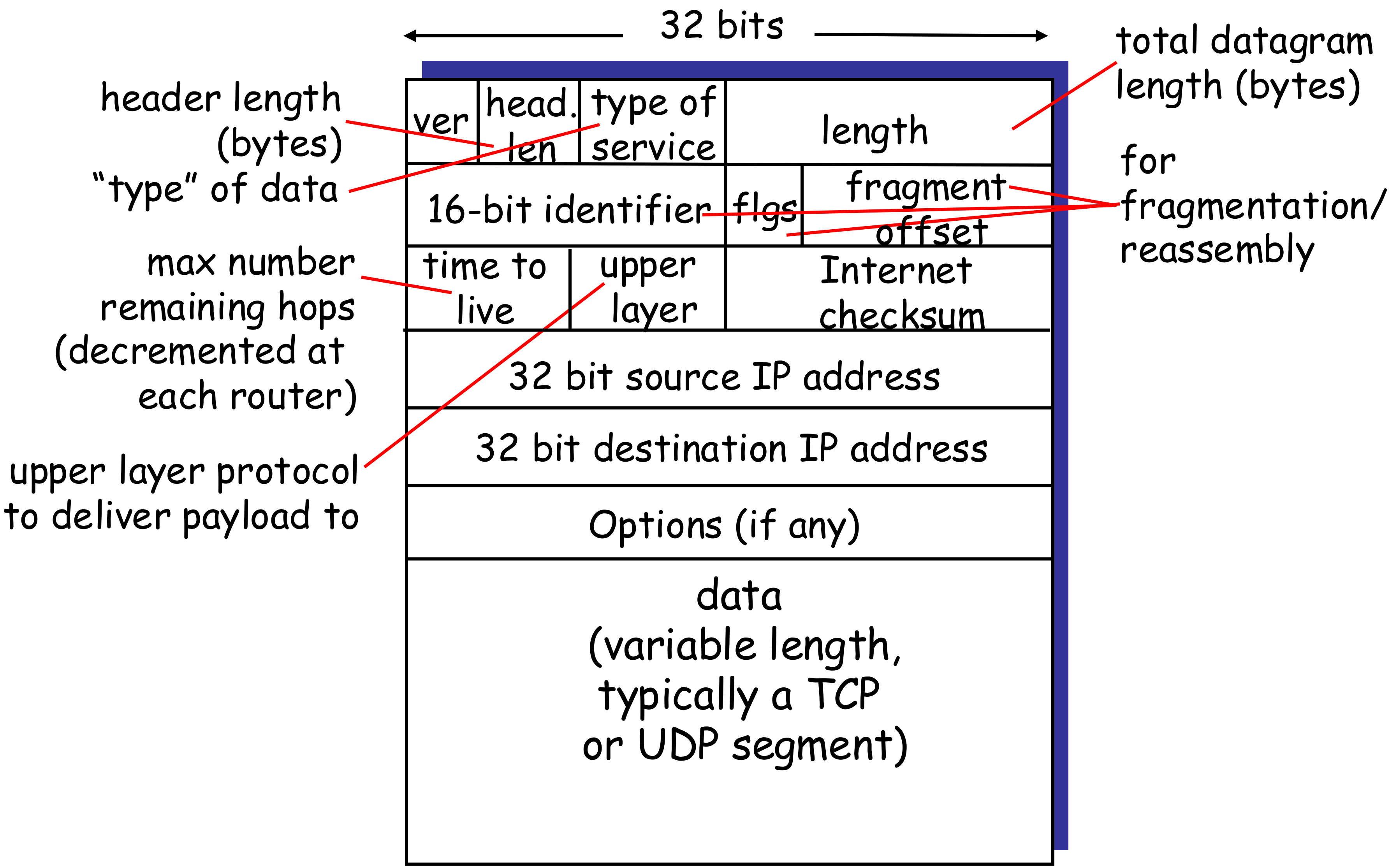
Solution: reject external packets to broadcast addresses

Denial-of-Service

Prevent access by legitimate users or stop critical system processes

- **Connection flooding attack**
  - Overwhelming connection queue with SYN flood
- **Bandwidth flooding attack**
  - Overwhelming communications link with packets
  - Strength in flooding attack lies in volume rather than content
- **Implementation vulnerability attack**
  - Send a few crafted messages to target app that has vulnerability
  - Malicious messages called the “exploit”
  - Remotely stopping or crashing services

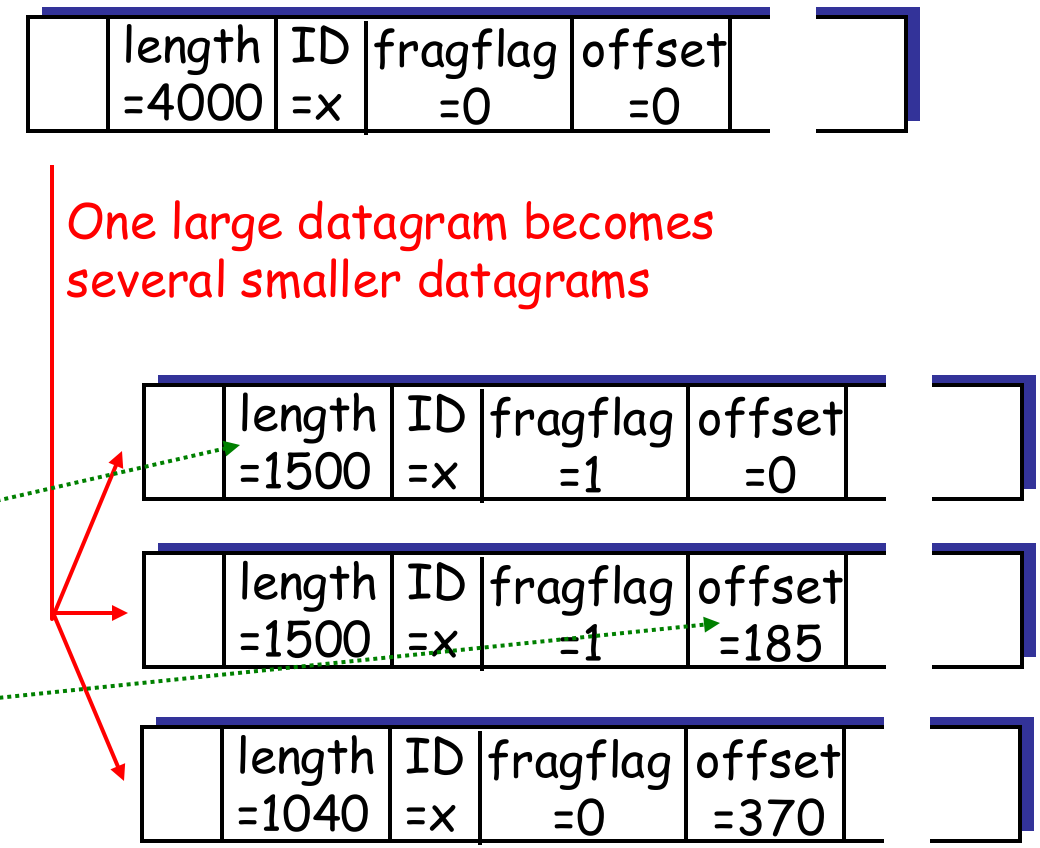
Interlude: IP datagram format



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes



1480 bytes in data field

offset = 1480/8

MTU: Maximum Transmission Unit (the size of the largest network layer protocol data unit that can be communicated in a single network transaction)

DoS: examples of vulnerability attacks

see http://www.cert.org/advisories/CA-1997-28.html

- Land: sends spoofed packet with source and dest address/port the same
- Ping of death: sends oversized ping packet
- Jolt2: sends a stream of fragments, none of which have offset of 0. Rebuilding consumes all processor capacity.
- Teardrop, Newtear, Bonk, Syndrop: tools send overlapping segments, that is, fragment offsets incorrect.

Patches fix the problem, but malformed packet attacks continue to be discovered.

LAND

- LAND: Local Area Network Denial
- Spoofed TCP SYN packet with source and destination both being the victim
- On receipt, victim's machine keep on responding to itself in a loop
  - Causes the victim to crash
- Many OSs were vulnerable, e.g.,
  - Windows 95, NT, XP SP2
  - Mac OS MacTCP

Ping of Death

- ICMP Echo Request (Ping) is 56 bytes
- If a ping message is more than 65536 bytes (max for IP packet), this can cause some machines to crash
- Older windows systems

Solution: patch OS, filter out ICMP packets

“Teardrop”, “Bonk” and kins

- TCP/IP fragments contain Offset field

- Attacker sets Offset field to:
  - **overlapping values**
    - Bad/old implementation of TCP/IP stack crashes when attempting to re-assemble the fragments
  - … **or to very large values**
    - Target system crashes

Solution: use up-to-date TCP/IP implementation

End of Lecture 4