Network Protocols and Vulnerabilities

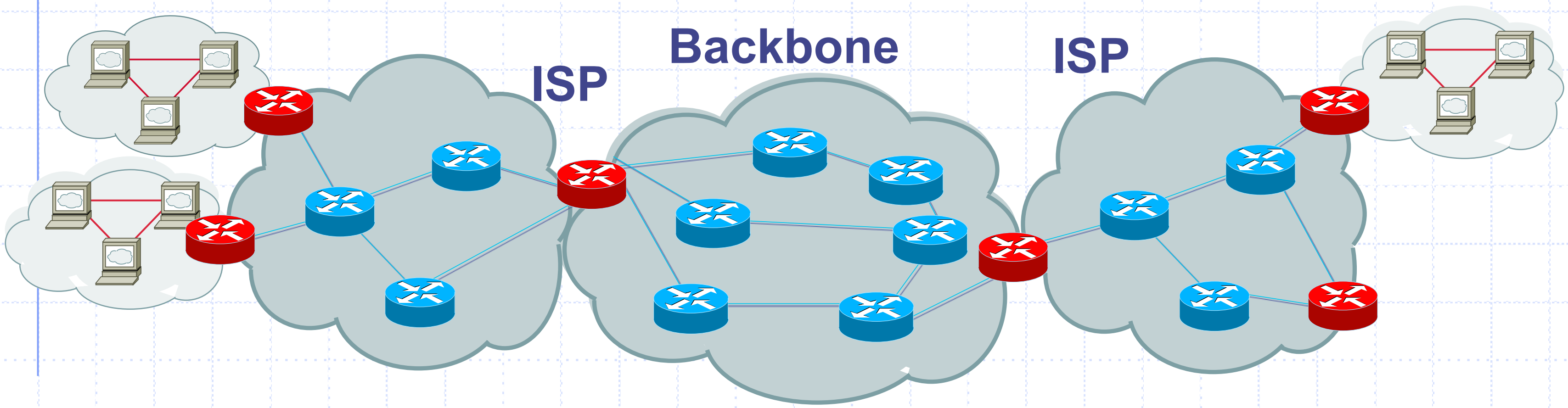
Dan Boneh

Outline

- Basic Networking:
  - How things work now plus some problems

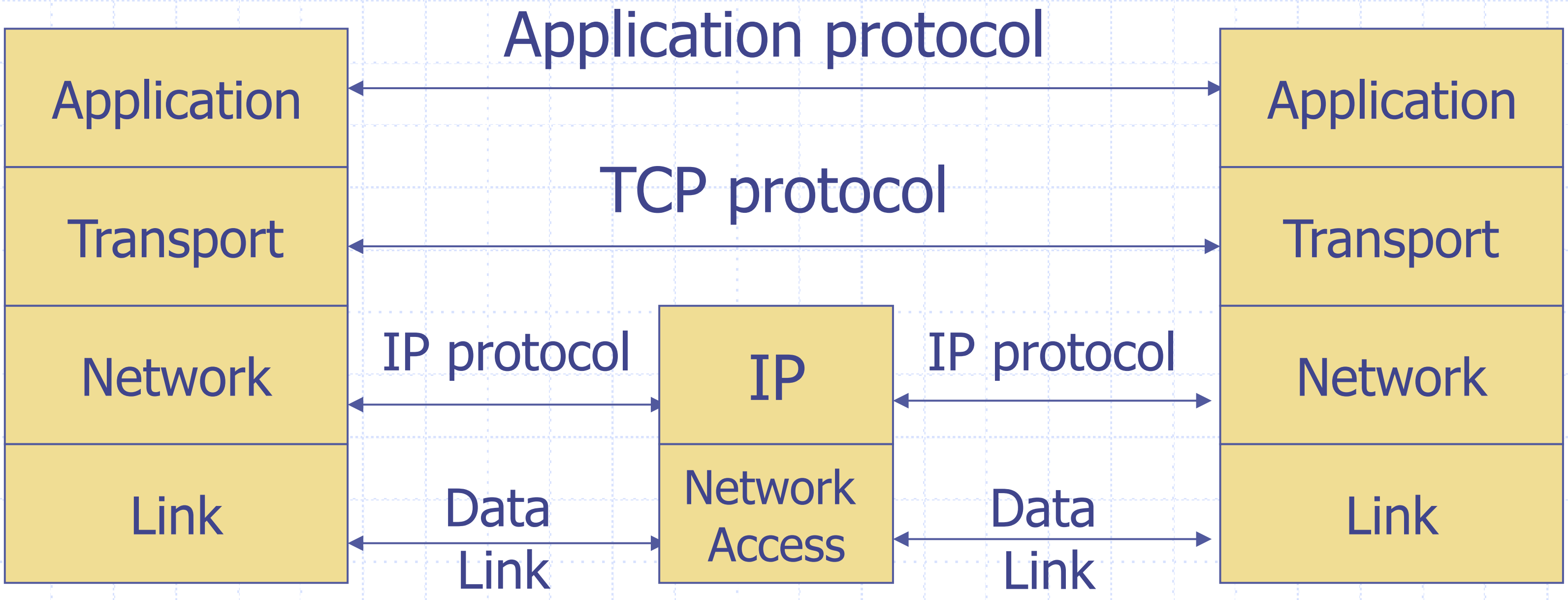
- Some network attacks
  - Attacking host-to-host datagram protocols
    - TCP Spoofing, …
  - Attacking network infrastructure
    - Routing
    - Domain Name System

Internet Infrastructure

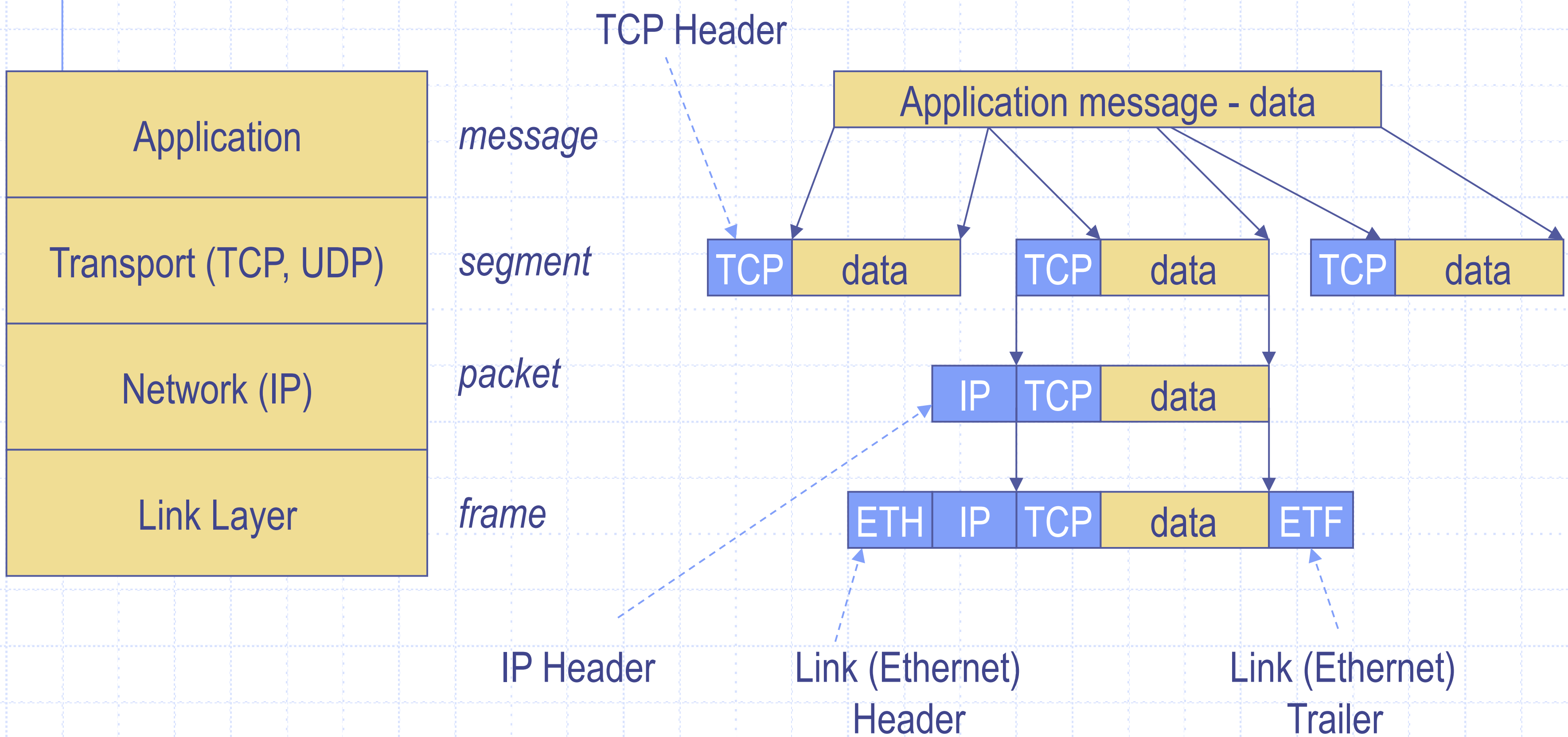


- Local and interdomain routing
  - TCP/IP for routing, connections
  - BGP for routing announcements
- Domain Name System
  - Find IP address from symbolic name (www.cs.stanford.edu)

TCP Protocol Stack



Data Formats

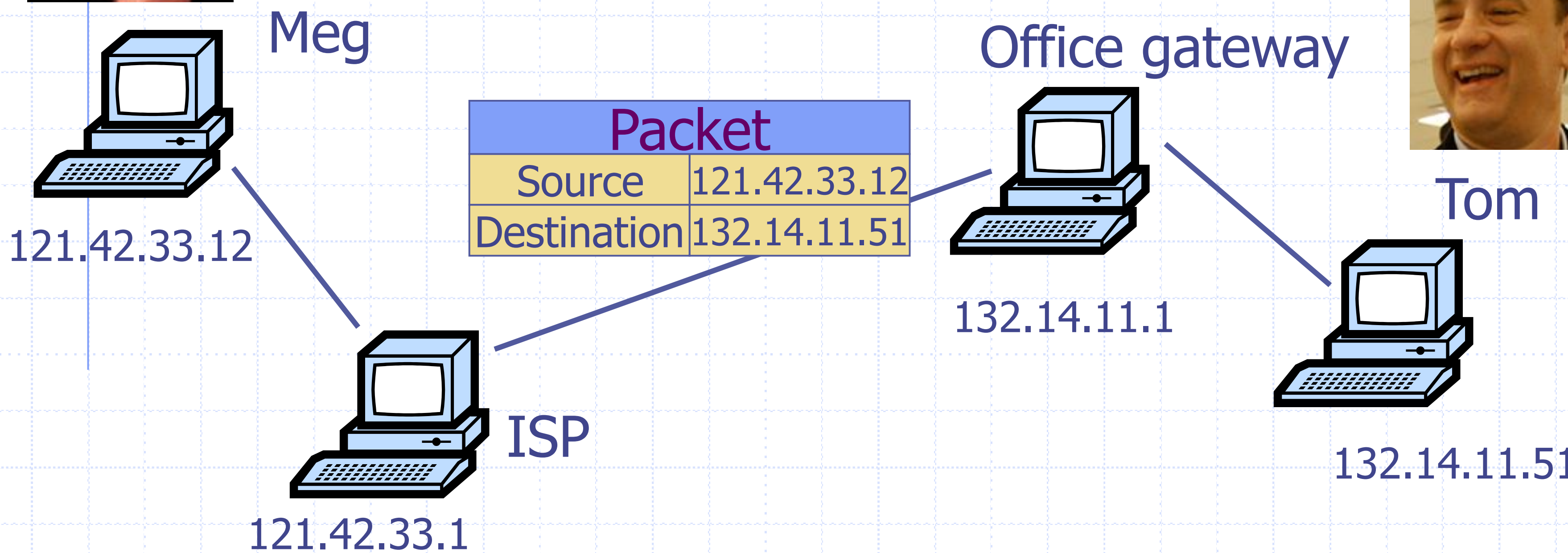


Internet Protocol

- Connectionless
  - Unreliable
  - Best effort
- Notes:
  - src and dest **ports** not parts of IP hdr

Version	Header Length
Type of Service	
Total Length	
Identification	
Flags	Fragment Offset
Time to Live	
Protocol	
Header Checksum	
Source Address of Originating Host	
Destination Address of Target Host	
Options	
Padding	
IP Data	

IP Routing



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

IP Protocol Functions (Summary)

- Routing
  - IP host knows location of router (gateway)
  - IP gateway must know route to other networks
- Fragmentation and reassembly
  - If max-packet-size less than the user-data-size
- Error reporting
  - ICMP packet to source if packet is dropped
- TTL field: decremented after every hop
  - Packet dropped f TTL=0. Prevents infinite loops.

Problem: no src IP authentication

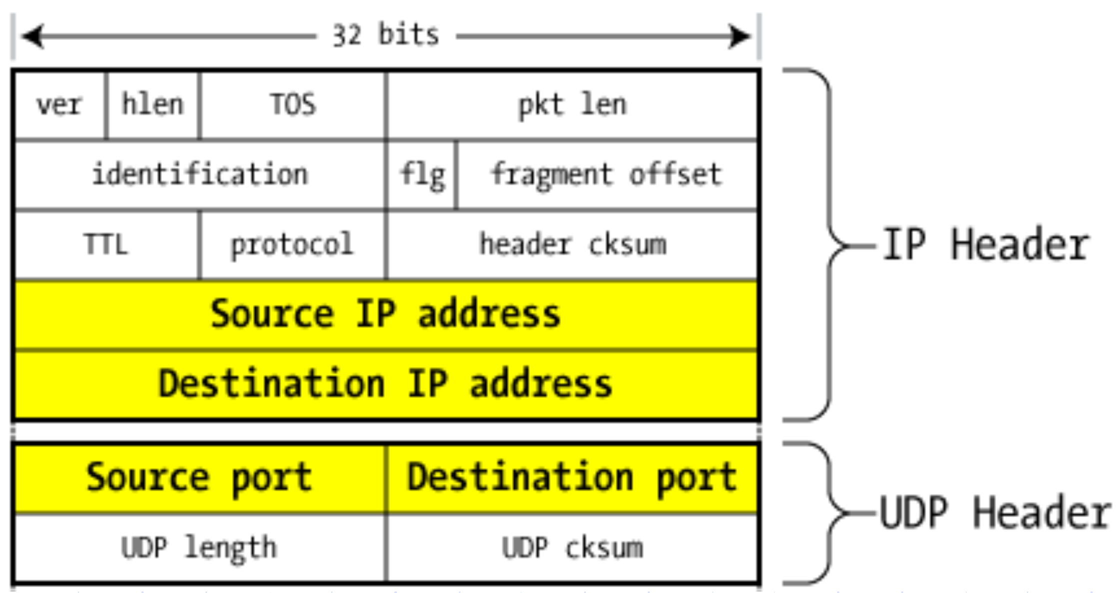
- Client is trusted to embed correct source IP
  - Easy to override using raw sockets
  - **Libnet**: a library for formatting raw packets with arbitrary IP headers

- Anyone who owns their machine can send packets with arbitrary source IP
  - … response will be sent back to forged source IP

- Implications: (solutions in DDoS lecture)
  - Anonymous DoS attacks;
  - Anonymous infection attacks (e.g. slammer worm)

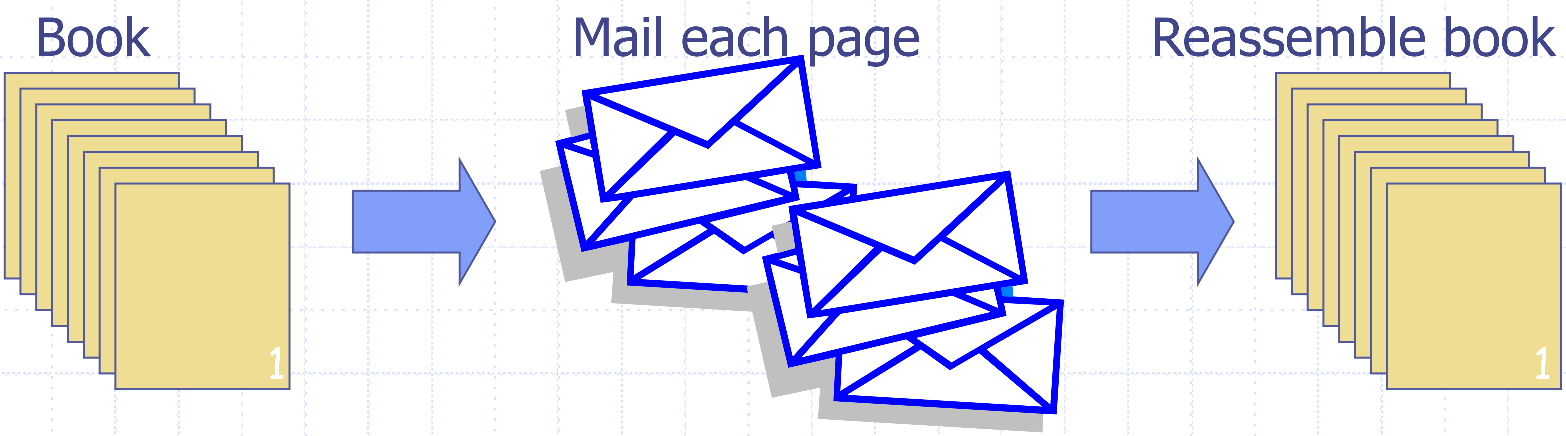
User Datagram Protocol

- Unreliable transport on top of IP:
  - No acknowledgment
  - No congenstion control
  - No message continuation

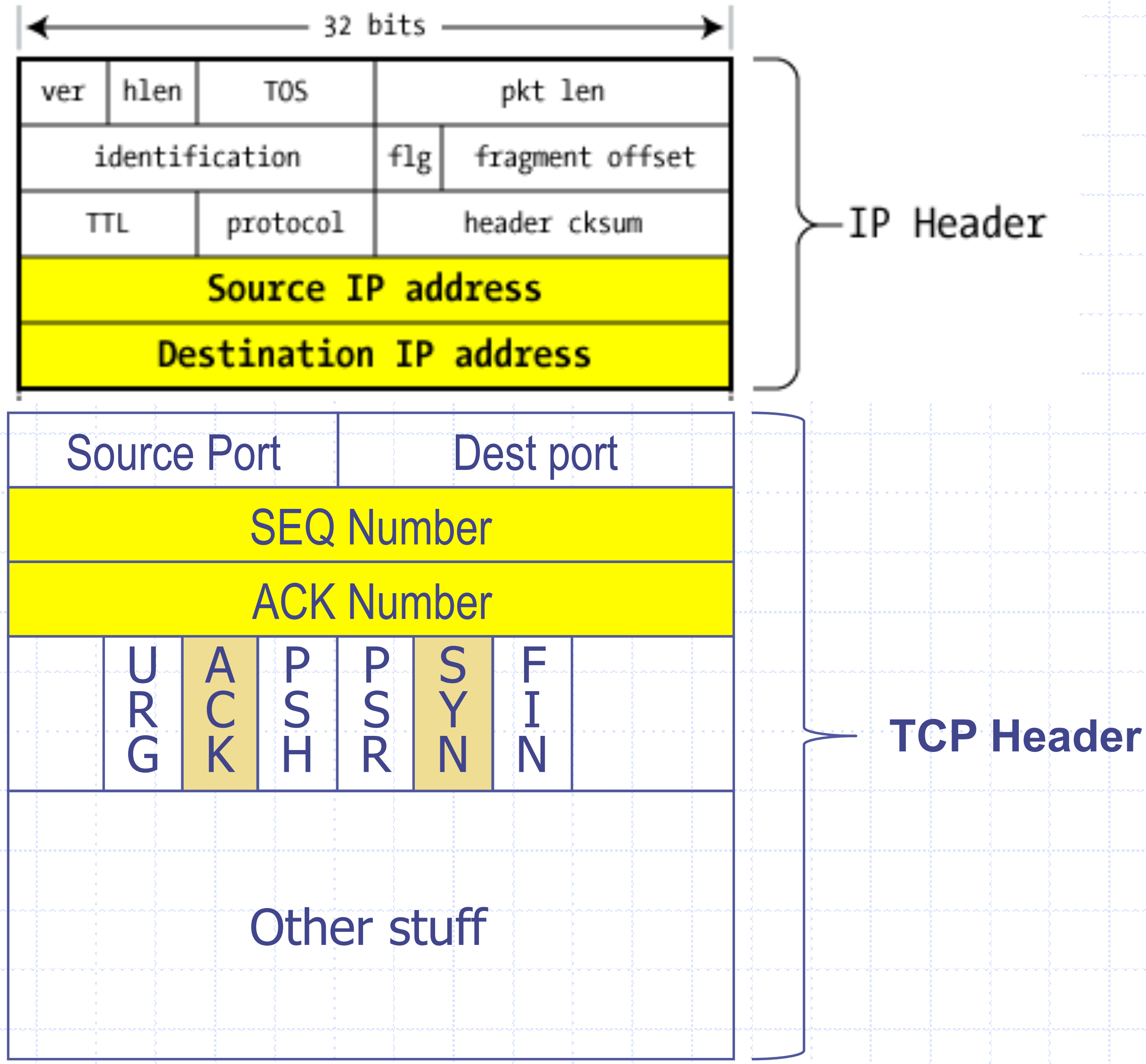


Transmission Control Protocol

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

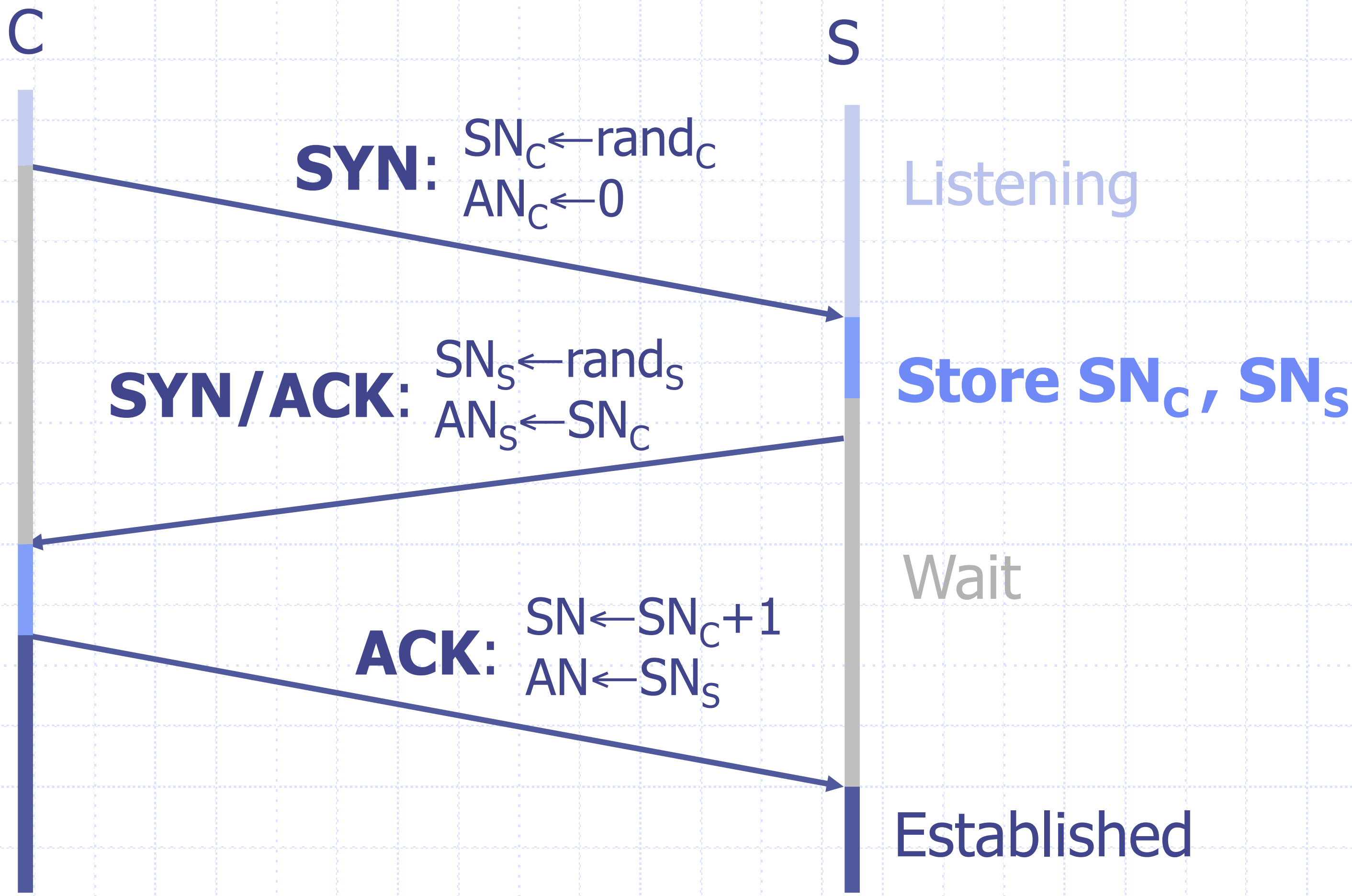


TCP Header



32 bits
ver
hlen
TOS
pkt len
identification
flg
fragment offset
TTL
protocol
header cksum
Source IP address
Destination IP address
IP Header
Source Port
Dest port
SEQ Number
ACK Number
URG
ACK
PSH
PSR
SYN
FIN
Other stuff
TCP Header

Review: TCP Handshake



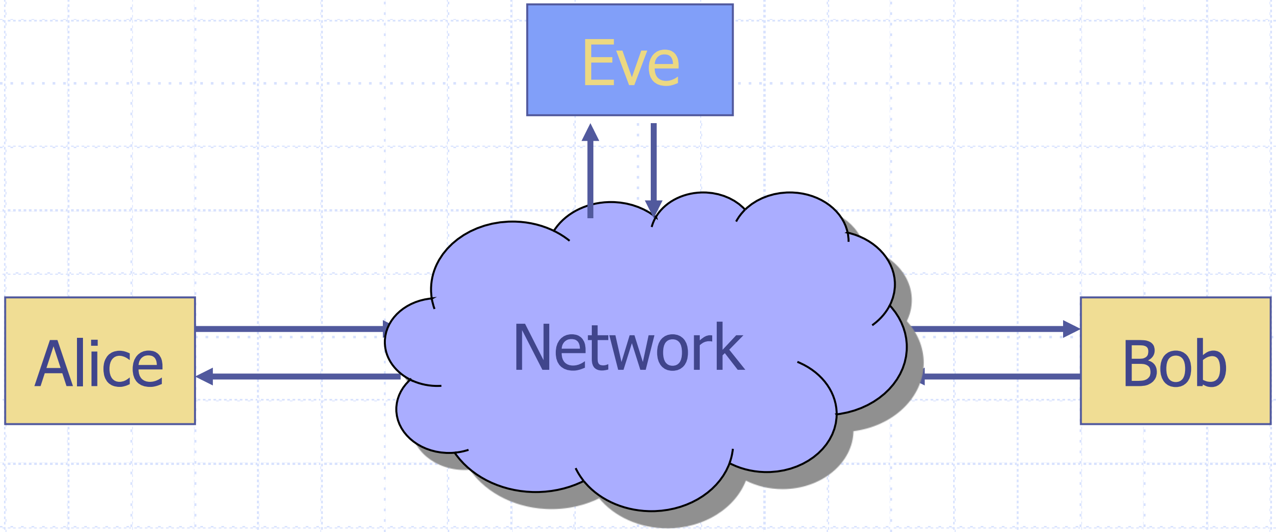
Received packets with SN too far out of window are dropped

Basic Security Problems

1. Network packets pass by untrusted hosts
   - Eavesdropping, packet sniffing
   - Especially easy when attacker controls a machine close to victim
2. TCP state can be easy to guess
   - Enables spoofing and session hijacking
3. Denial of Service (DoS) vulnerabilities
   - DDoS lecture

1. Packet Sniffing

- Promiscuous NIC reads all packets
  - Read all unencrypted data (e.g., "wireshark")
  - ftp, telnet (and POP, IMAP) send passwords in clear!

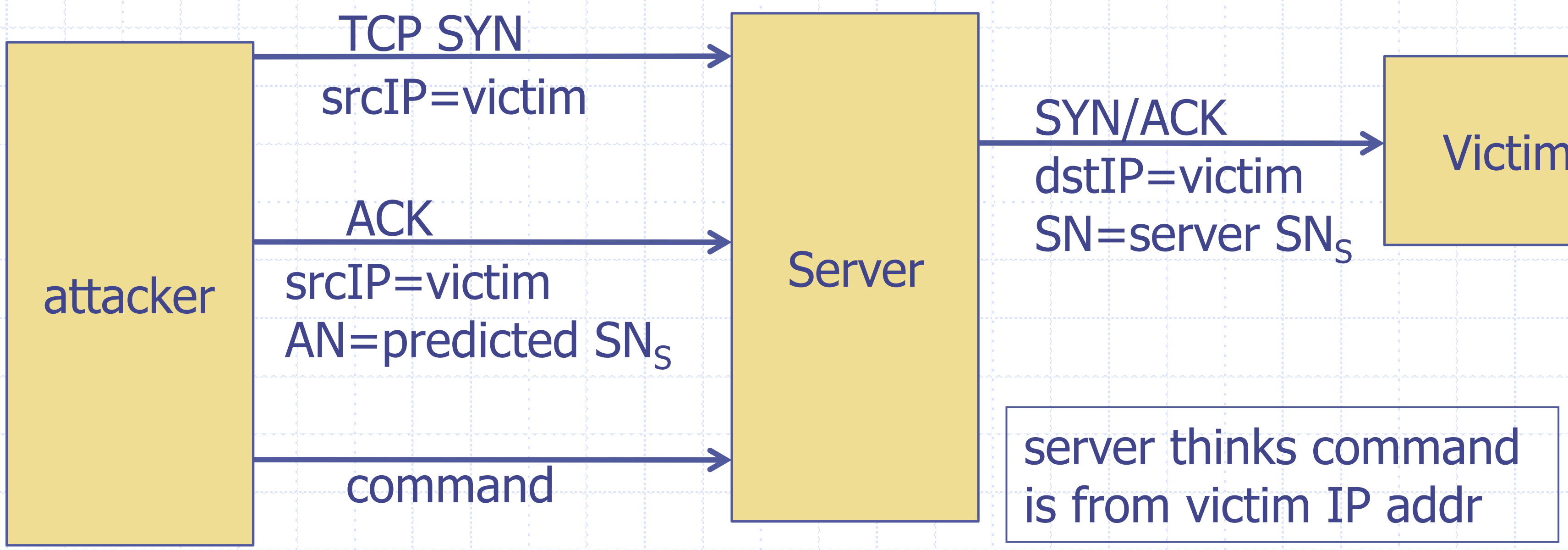


Sweet Hall attack installed sniffer on local machine

Prevention: Encryption (next lecture: IPSEC)

2. TCP Connection Spoofing

- Why random initial sequence numbers? (SN_C , SN_S)
- Suppose init. sequence numbers are predictable
  - Attacker can create TCP session on behalf of forged source IP
    - Breaks IP-based authentication (e.g. SPF, /etc/hosts)



Example DoS vulnerability [Watson'04]

- 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

Random initial TCP SNs

- Unpredictable SNs prevent basic packet injection
  - … but attacker can inject packets after eavesdropping to obtain current SN

- Most TCP stacks now generate random SNs
  - Random generator should be unpredictable
  - GPR'06: Linux RNG for generating SNs is predictable
    - Attacker repeatedly connects to server
    - Obtains sequence of SNs
    - Can predict next SN
    - Attacker can now do TCP spoofing (create TCP session with forged source IP)

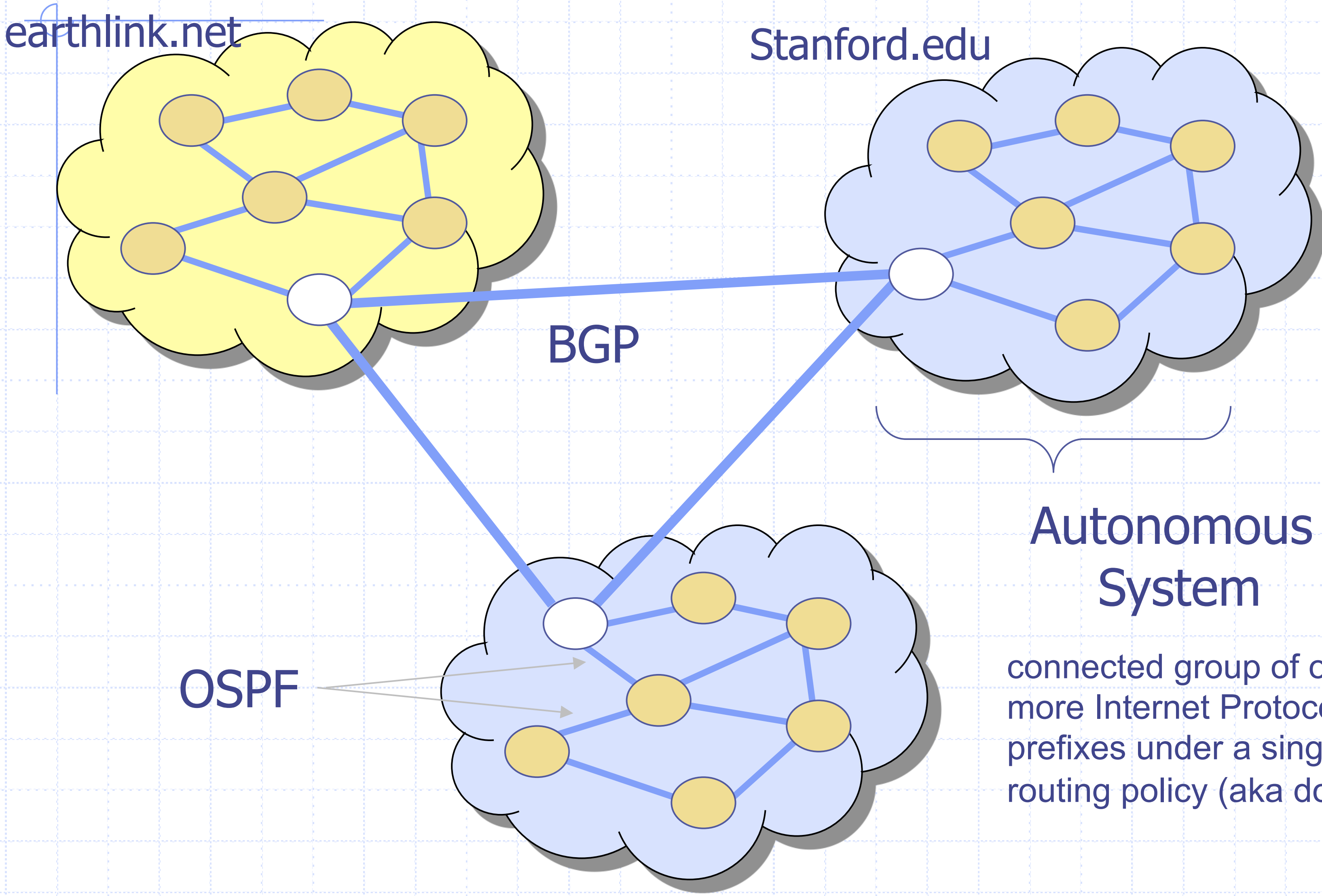
Routing Vulnerabilities

Routing Vulnerabilities

- Common attack: advertise false routes
  - Causes traffic to go though compromised hosts

- ARP (addr resolution protocol): IP addr -> eth addr
  - Node A can confuse gateway into sending it traffic for B
  - By proxying traffic, attacker A can easily inject packets into B’s session (e.g. WiFi networks)
- OSPF: used for routing within an AS
- BGP: routing between ASs
  - Attacker can cause entire Internet to send traffic for a victim IP to attacker’s address.
  - Example: Youtube mishap (see DDoS lecture)

Interdomain Routing



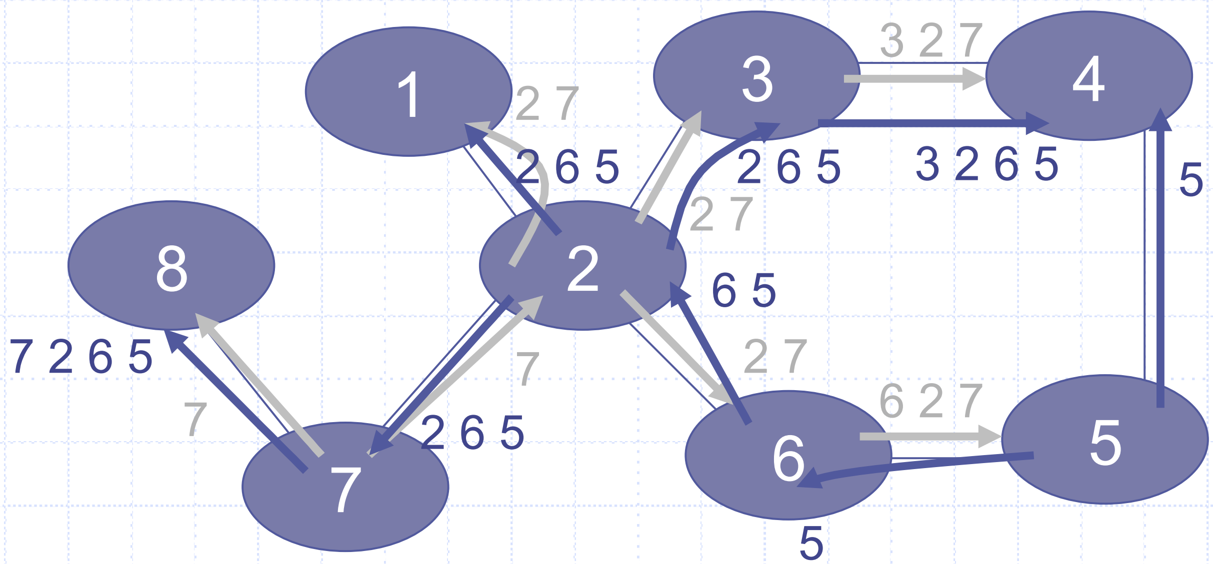
Autonomous System

connected group of one or more Internet Protocol prefixes under a single routing policy (aka domain)

BGP overview

- Iterative path announcement
  - Path announcements grow from destination to source
  - Packets flow in reverse direction
- Protocol specification
  - Announcements *can* be shortest path
  - Not obligated to use announced path

BGP example [D. Wetherall]



- Transit: 2 provides transit for 7
- Algorithm seems to work OK in practice
  - BGP is does not respond well to frequent node outages

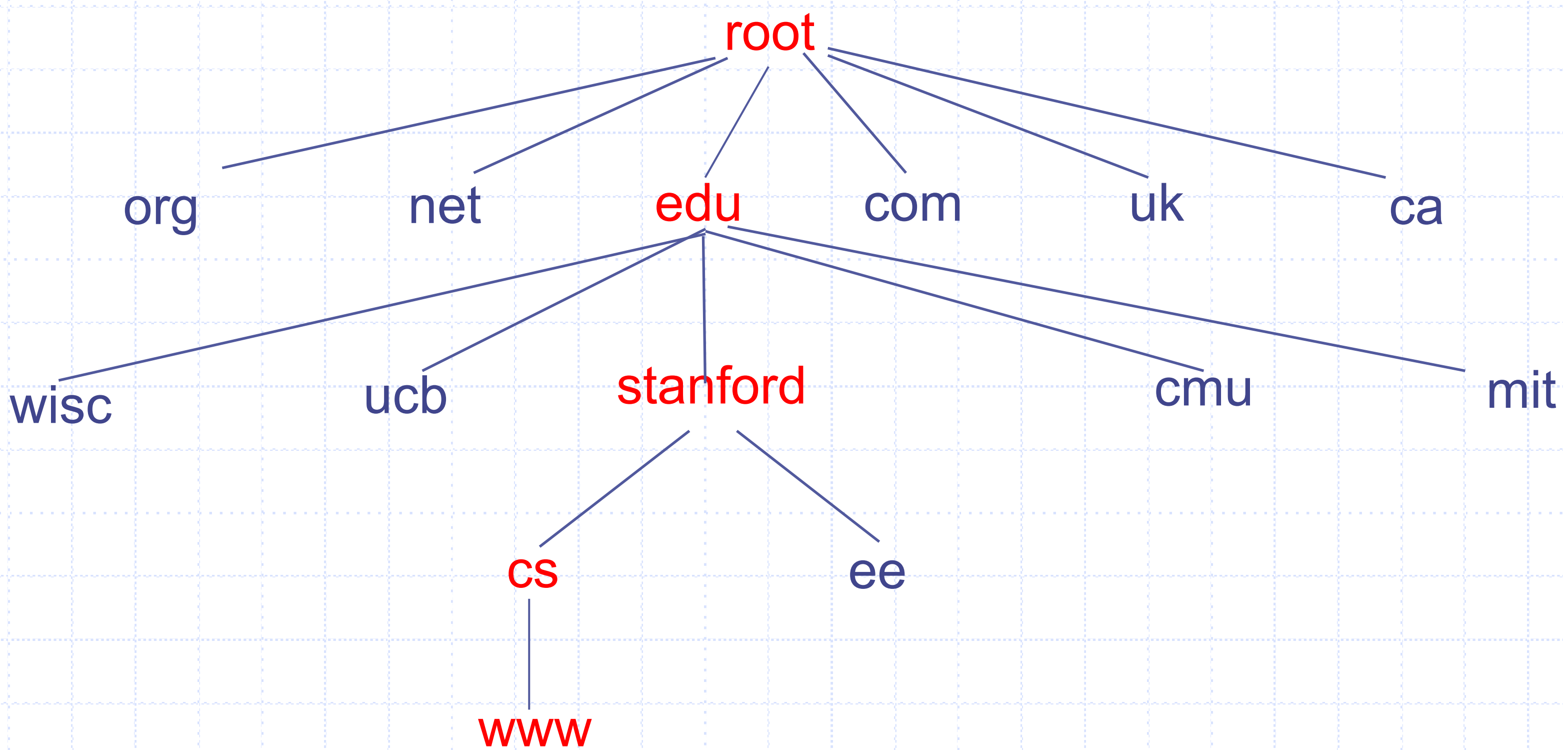
Issues

- Security problems
  - Potential for disruptive attacks
  - BGP packets are un-authenticated
    - Attacker can advertise arbitrary routes
    - Advertisement will propagate everywhere
    - Used for DoS and spam (detailed example in DDoS lecture)
- Incentive for dishonesty
  - ISP pays for some routes, others free

Domain Name System

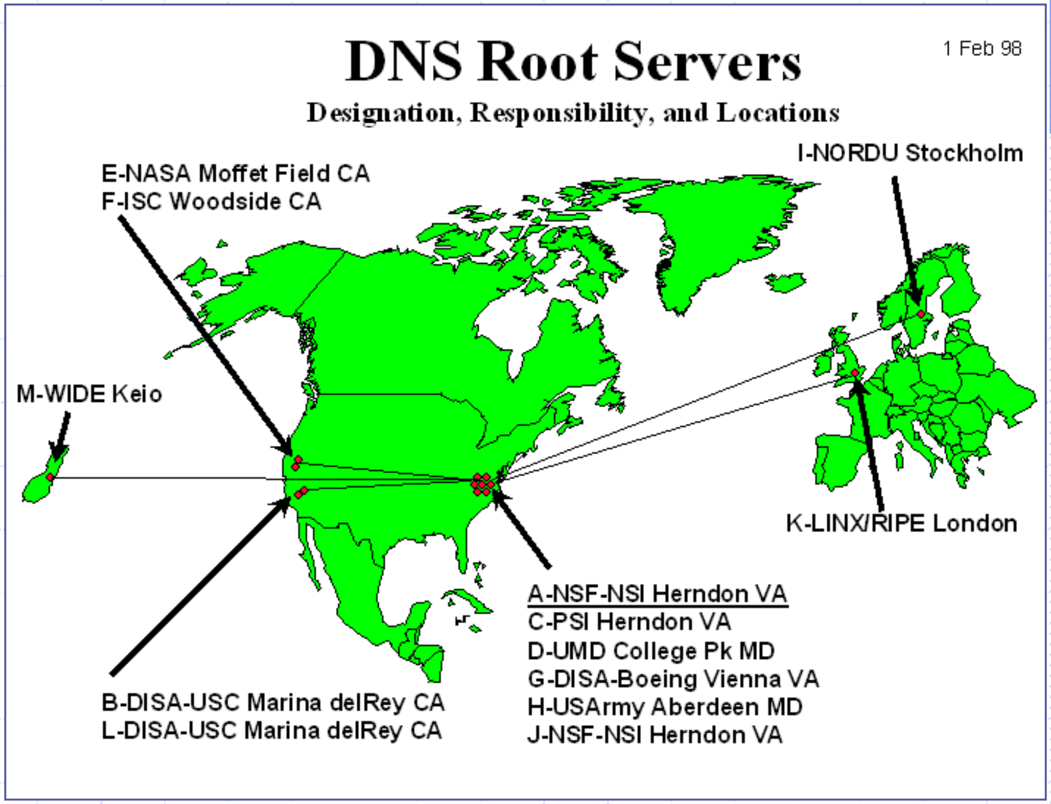
Domain Name System

- Hierarchical Name Space

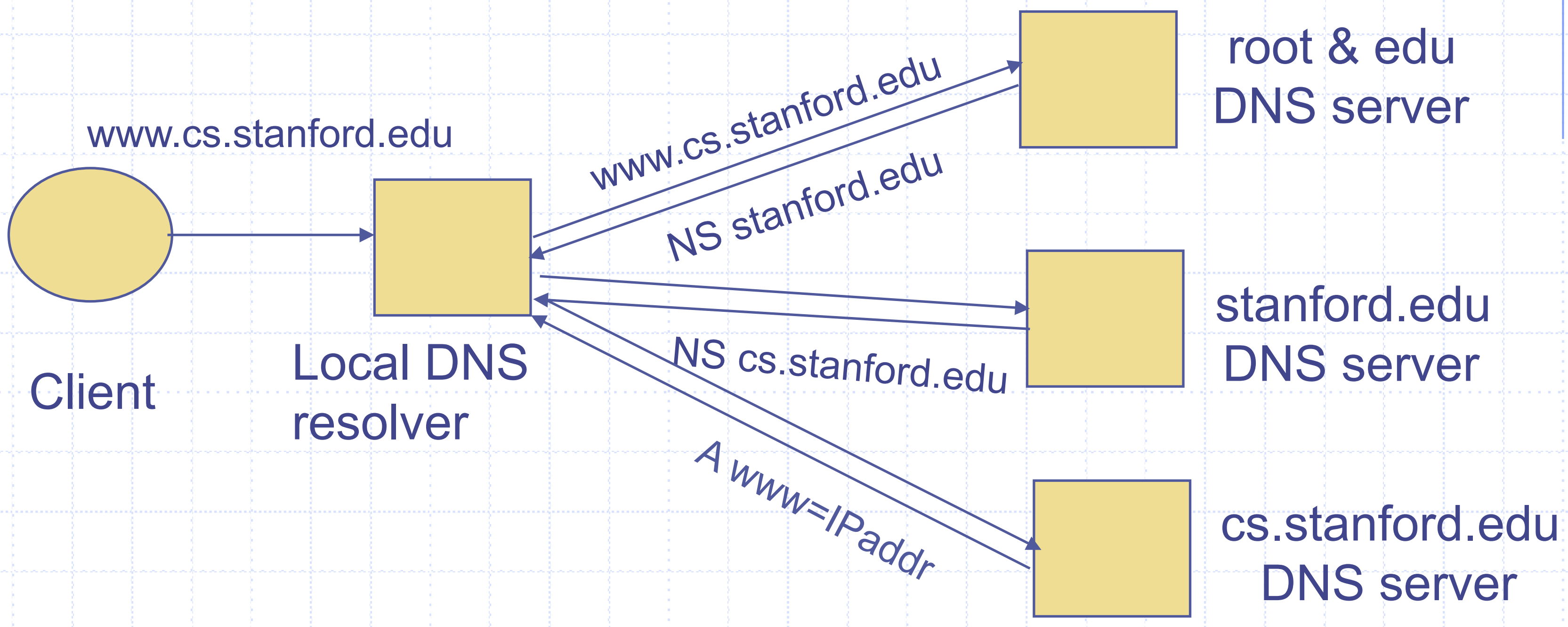


DNS Root Name Servers

- Hierarchical service
  - Root name servers for top-level domains
  - Authoritative name servers for subdomains
  - Local name resolvers contact authoritative servers when they do not know a name



DNS Lookup Example



DNS record types (partial list):

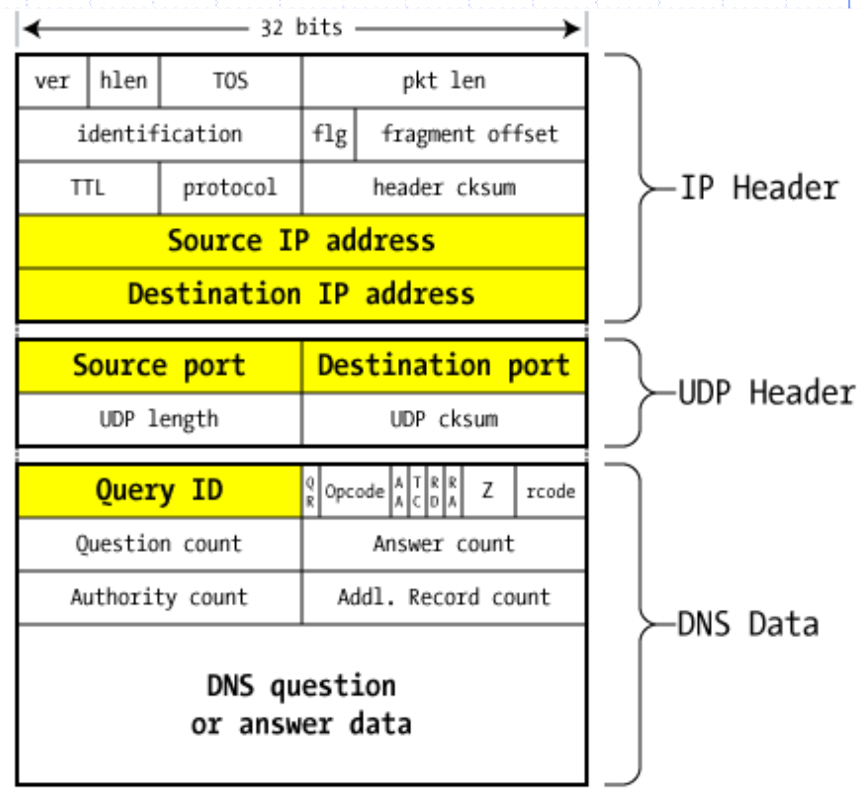
- NS: name server (points to other server)
- A: address record (contains IP address)
- MX: address in charge of handling email
- TXT: generic text (e.g. used to distribute site public keys (DKIM))

Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Useful for finding servers as well as addresses
    - NS records for domains
- DNS negative queries are cached
  - Save time for nonexistent sites, e.g. misspelling
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record

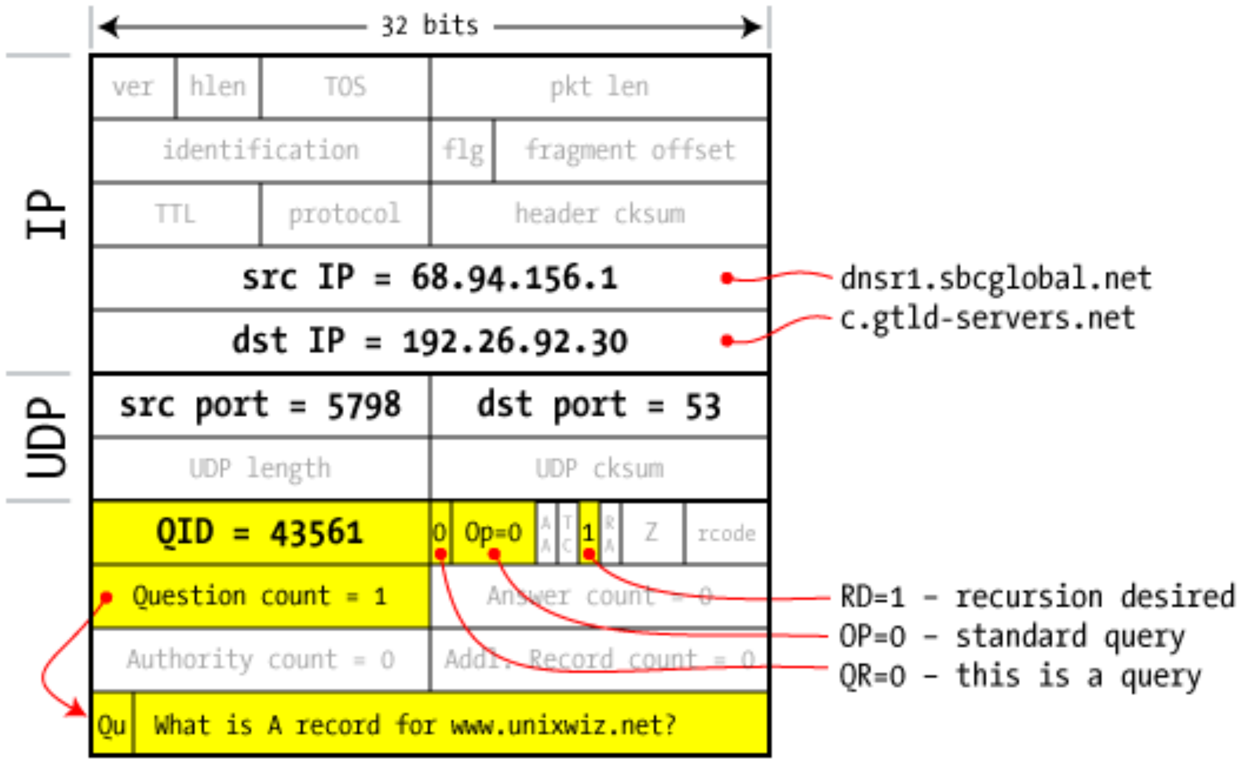
DNS Packet

- Query ID:
  - 16 bit random value
  - Links response to query



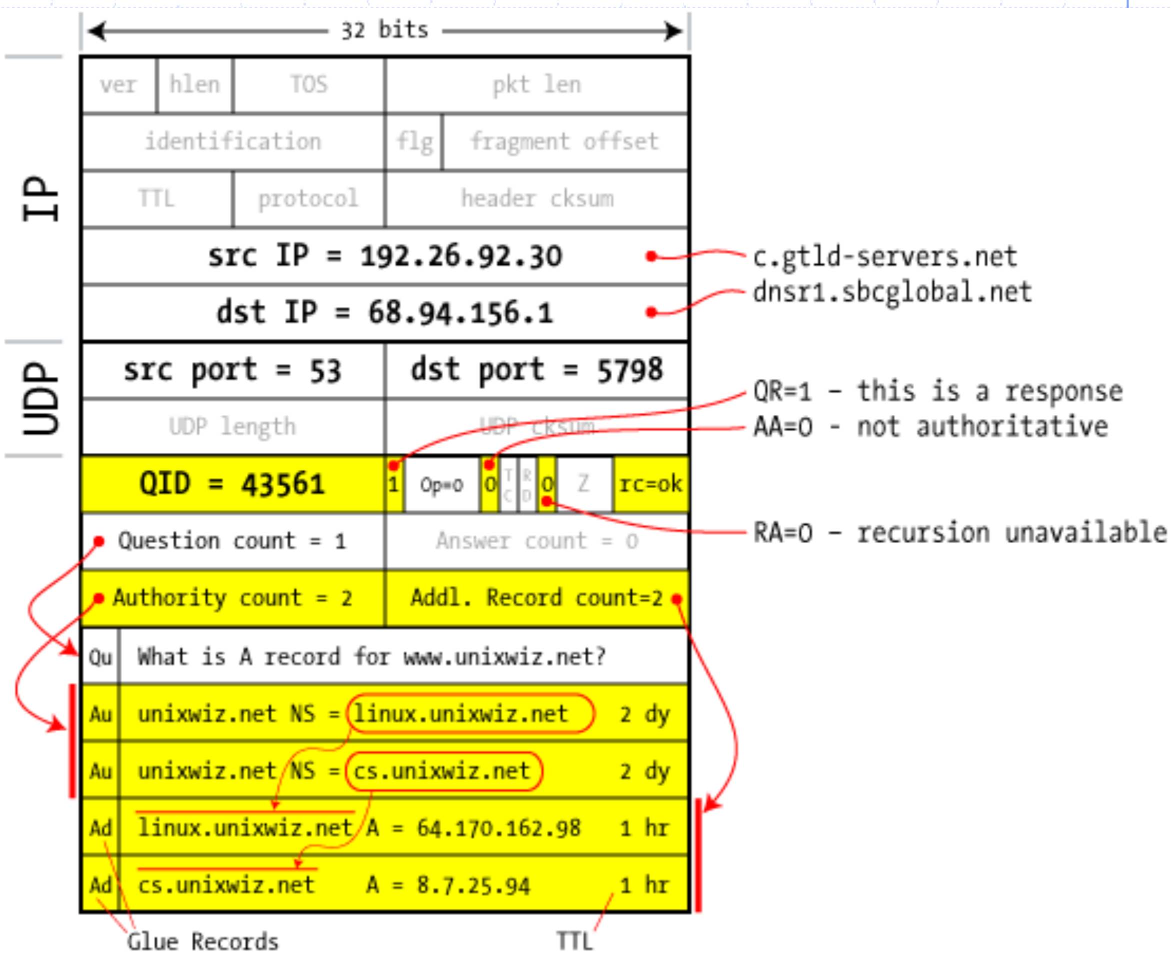
(from Steve Friedl)

Resolver to NS request



Response to resolver

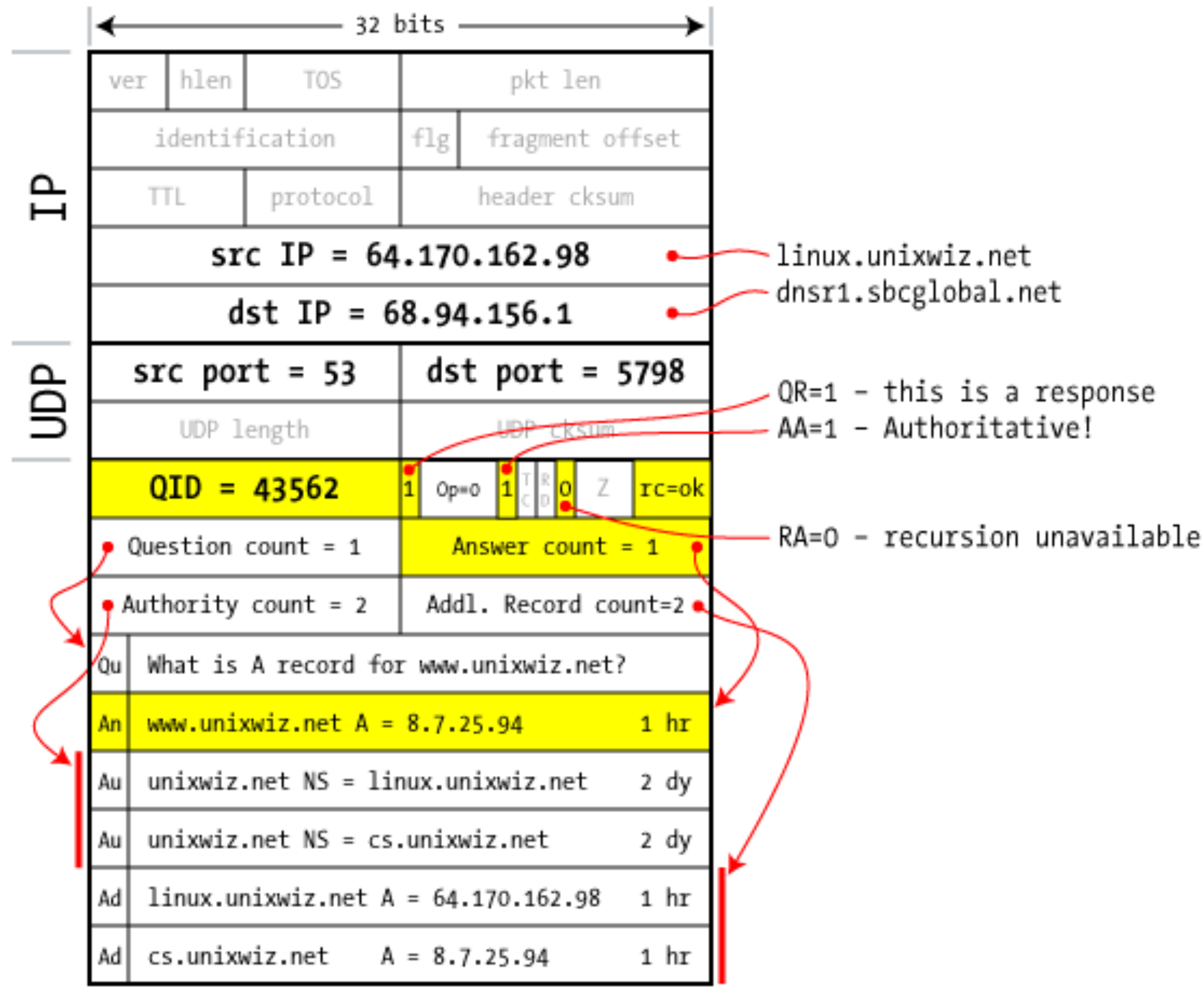
- Response contains IP addr of next NS server (called "glue")
- Response ignored if unrecognized QueryID



Authoritative response to resolver

<u>bailiwick checking:</u>
response is cached if it is within the same domain of query (i.e. **a.com** cannot set NS for **b.com**)

final answer



Basic DNS Vulnerabilities

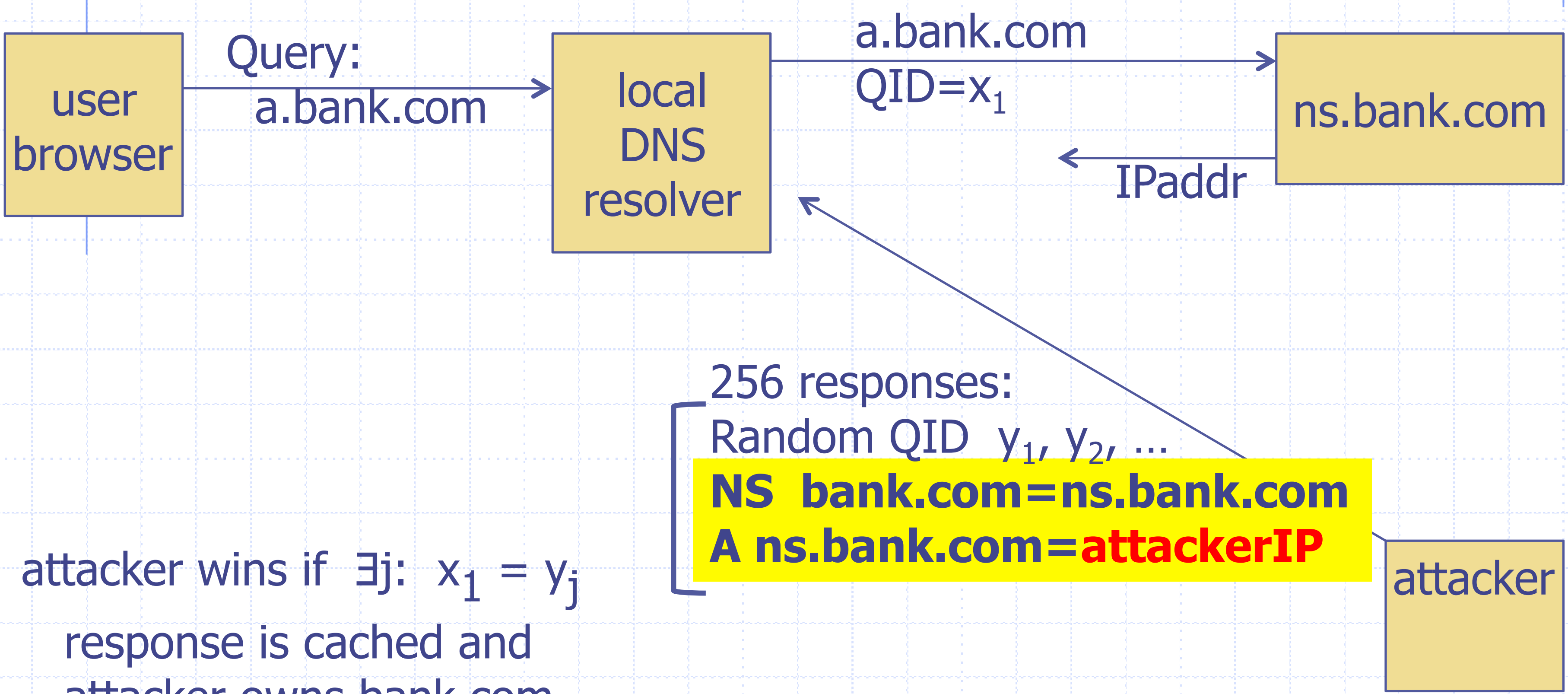
- Users/hosts trust the host-address mapping provided by DNS:
  - Used as basis for many security policies:

    Browser same origin policy, URL address bar

- Obvious problems
  - Interception of requests or compromise of DNS servers can result in incorrect or malicious responses
    - e.g.: hijack BGP route to spoof DNS
  - Solution – authenticated requests/responses
    - Provided by DNSsec … but no one uses DNSsec

DNS cache poisoning (a la Kaminsky'08)

- Victim machine visits attacker's web site, downloads Javascript

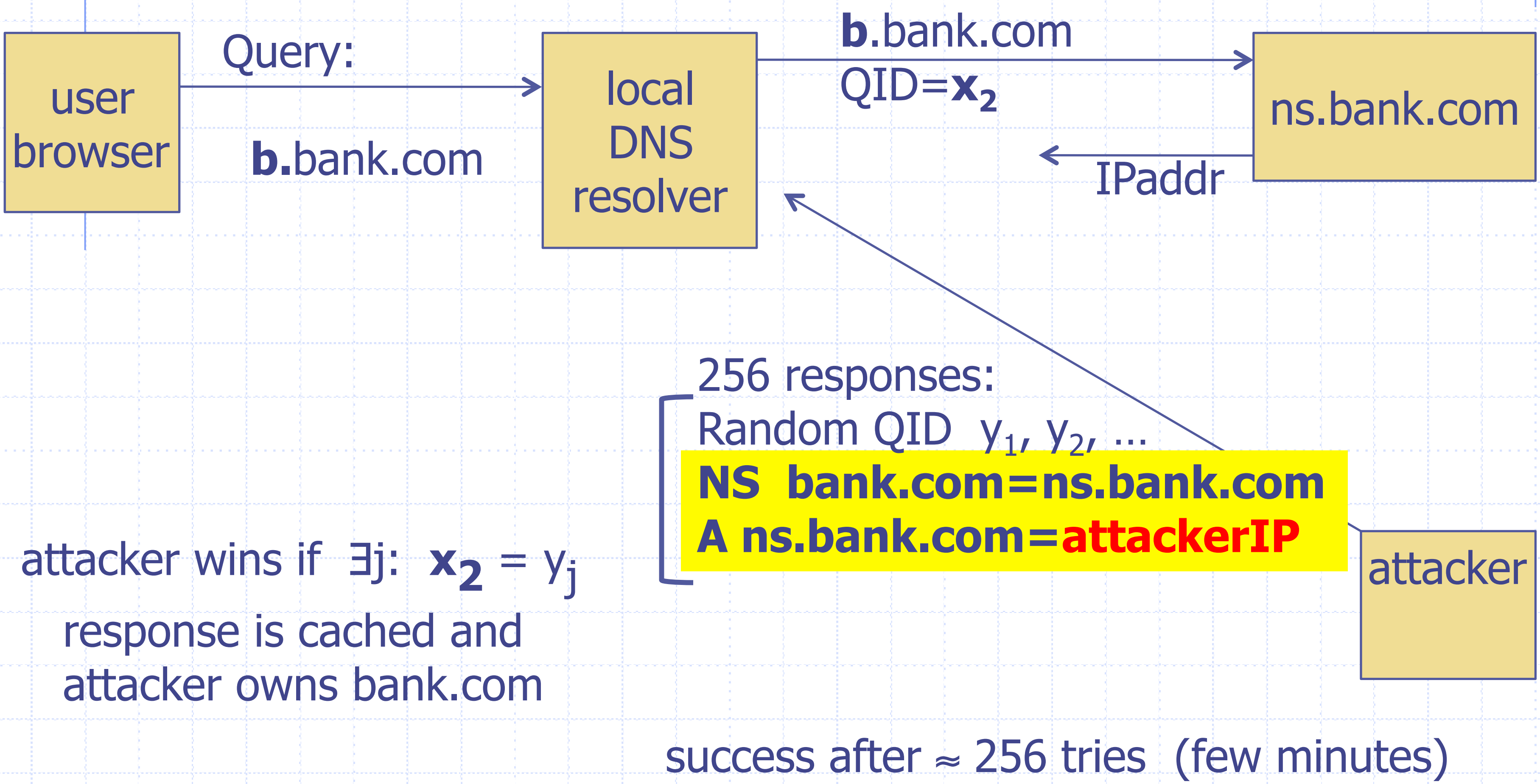


attacker wins if $\exists j$: $x_1 = y_j$

response is cached and attacker owns bank.com

If at first you don't succeed …

- Victim machine visits attacker's web site, downloads Javascript



Defenses

- Increase Query ID size.   How?

a. Randomize src port, additional 11 bits

   Now attack takes several hours

b. Ask every DNS query twice:
   - Attacker has to guess QueryID correctly twice (32 bits)
   - Apparently DNS system cannot handle the load

Pharming

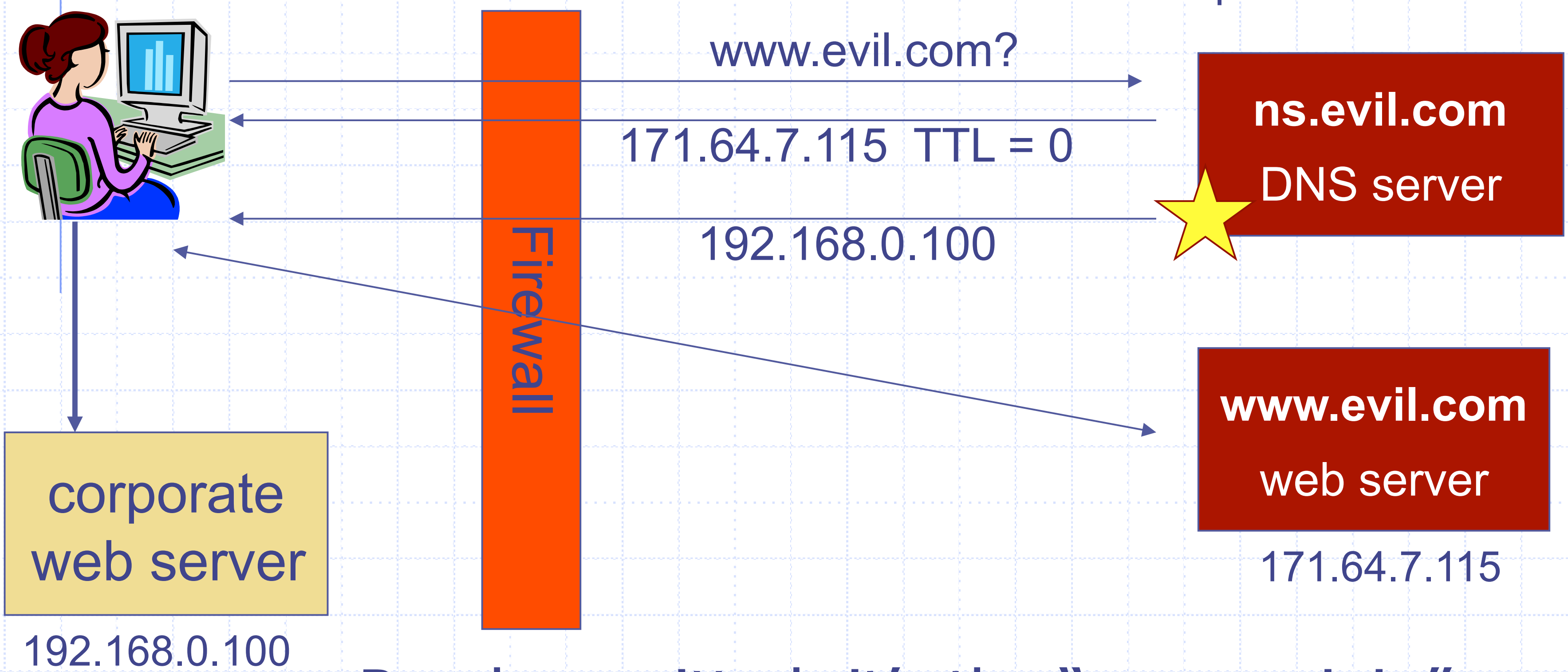
- DNS poisoning attack (less common than phishing)
  - Change IP addresses to redirect URLs to fraudulent sites
  - Potentially more dangerous than phishing attacks
  - No email solicitation is required
- DNS poisoning attacks have occurred:
  - January 2005, the domain name for a large New York ISP, Panix, was hijacked to a site in Australia.
  - In November 2004, Google and Amazon users were sent to Med Network Inc., an online pharmacy
  - In March 2003, a group dubbed the "Freedom Cyber Force Militia" hijacked visitors to the Al-Jazeera Web site and presented them with the message "God Bless Our Troops"

[DWF'96, R'01]

DNS Rebinding Attack

<iframe src="**http://www.evil.com**">

DNS-SEC cannot stop this attack



Read permitted: it's the "same origin"

DNS Rebinding Defenses

- Browser mitigation: DNS Pinning
  - Refuse to switch to a new IP
  - Interacts poorly with proxies, VPN, dynamic DNS, …
  - Not consistently implemented in any browser
- Server-side defenses
  - Check Host header for unrecognized domains
  - Authenticate users with something other than IP
- Firewall defenses
  - External names can't resolve to internal addresses
  - Protects browsers inside the organization

Summary

- Core protocols not designed for security
  - Eavesdropping, Packet injection, Route stealing, DNS poisoning
  - Patched over time to prevent basic attacks

    (e.g. random TCP SN)

- More secure variants exist (next lecture) :

  IP -> IPsec

  DNS -> DNSsec

  BGP -> SBGP