CSE 433S: Introduction to Computer Security

Lecture 1: Security Fundamentals and Network Security through the Lens of Attackers

Washington University in St. Louis

Security Concepts

What is security?

- Keeping something (information, system in some case) secure against stealing & changing & destroying & forging

- Traditionally provided by physical (e.g., cabinets with locks) and administrative means (e.g., personal screening procedures)

Why is security hard ?

- Security game is hard, because we have a negative goal

- Secure means nobody can break our system
  - Who is nobody ?
  - What weapons do they have ?


Three Elements of Security

Achieve some ***goal*** against some ***adversary***

- System Goal / Security Service / Policy
- Threat models
- Mechanism

Security Goal / Services / Policy

- Confidentiality
  - Information can only be accessed by authorized entity
- Integrity
  - Information has not been tampered with
- Availability
  - Information is available to the authorized entities

Other policy goals

- Authenticity
- Accountability
- Non-repudiation
- Attack surface
- Vulnerability
- Exploitation

Policy went wrong – Sarah Palin yahoo account

Img src: wikipedia

Threat models

- Who are the attackers
- What are the attackers capable of?

Threat models go wrong

Img src: Wikipedia.org

Threat models go wrong

VS

Img src: amazon.com

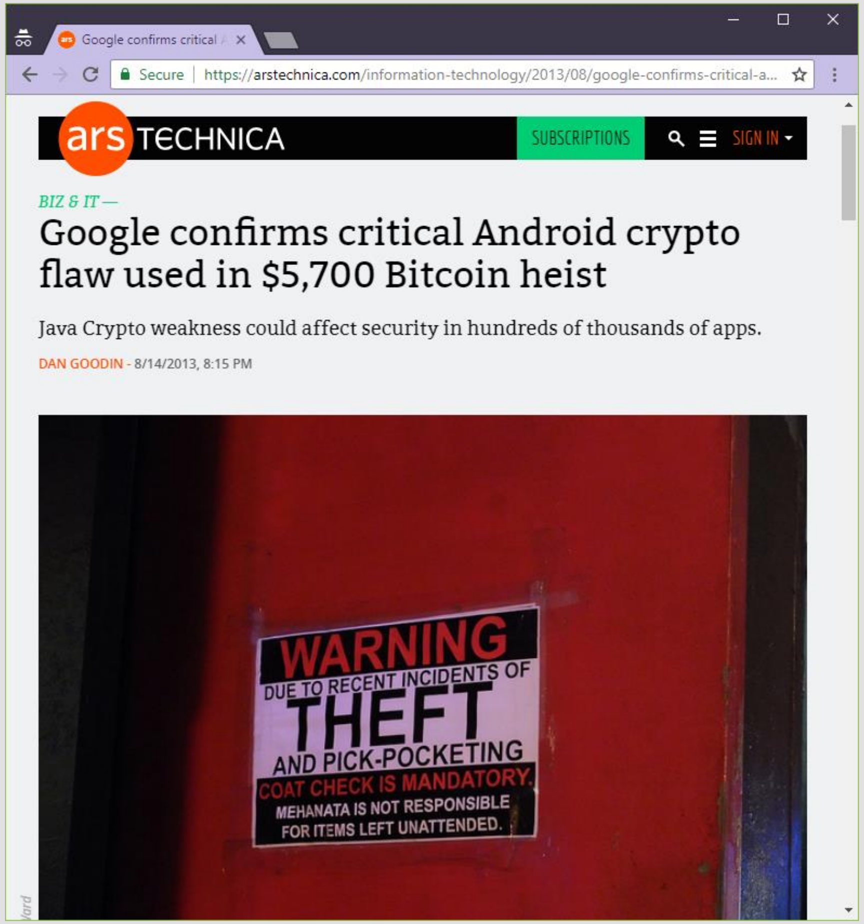
Is this code secure

```
main (char * i){
  char s[128];
  memset(s,0,128);
  strncpy(s, i, 127);
  printf(“%s”,s);
}
```

Mechanism

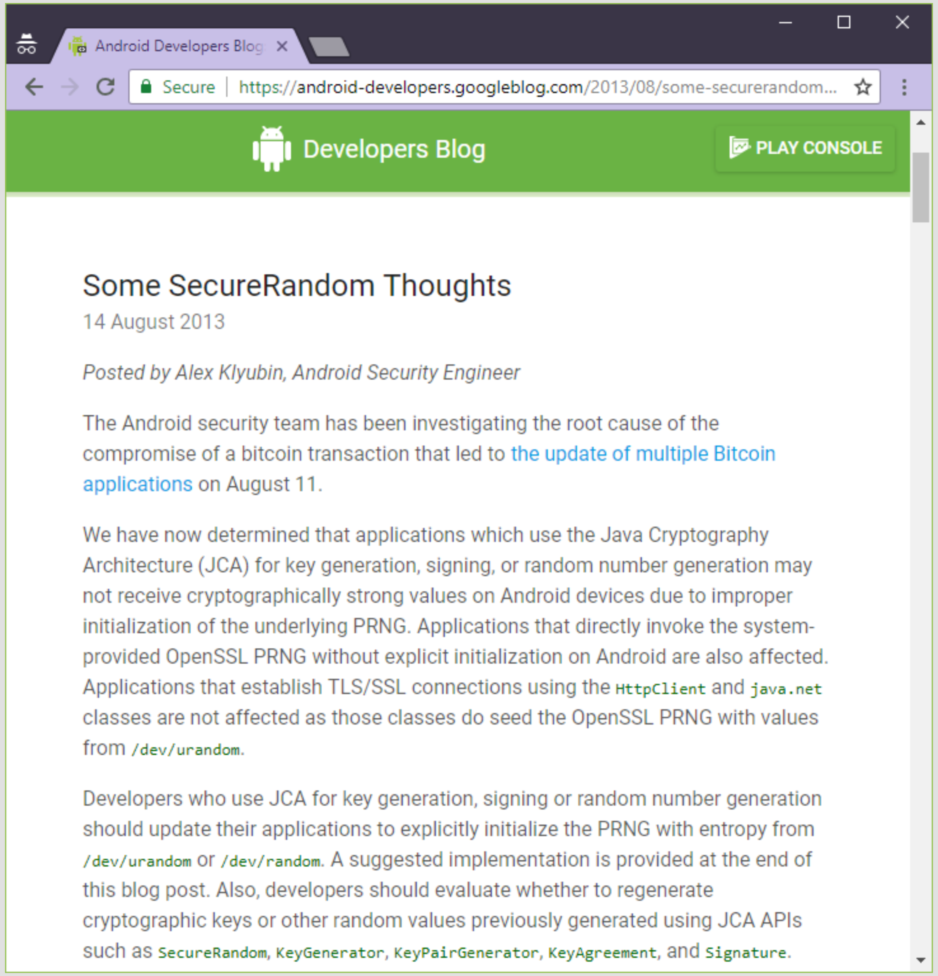
- What is the system composed of ?
  - Software
  - Hardware
  - Design
  - Implementation

Mechanisms go wrong
When random is no longer random


Google confirms critical Android crypto flaw used in $5,700 Bitcoin heist

Java Crypto weakness could affect security in hundreds of thousands of apps.

DAN GOODIN - 8/14/2013, 8:15 PM


Some SecureRandom Thoughts

14 August 2013

Posted by Alex Klyubin, Android Security Engineer

The Android security team has been investigating the root cause of the compromise of a bitcoin transaction that led to the update of multiple Bitcoin applications on August 11.

We have now determined that applications which use the Java Cryptography Architecture (JCA) for key generation, signing, or random number generation may not receive cryptographically strong values on Android devices due to improper initialization of the underlying PRNG. Applications that directly invoke the system-provided OpenSSL PRNG without explicit initialization on Android are also affected. Applications that establish TLS/SSL connections using the `HttpClient` and `java.net` classes are not affected as those classes do seed the OpenSSL PRNG with values from `/dev/urandom`.

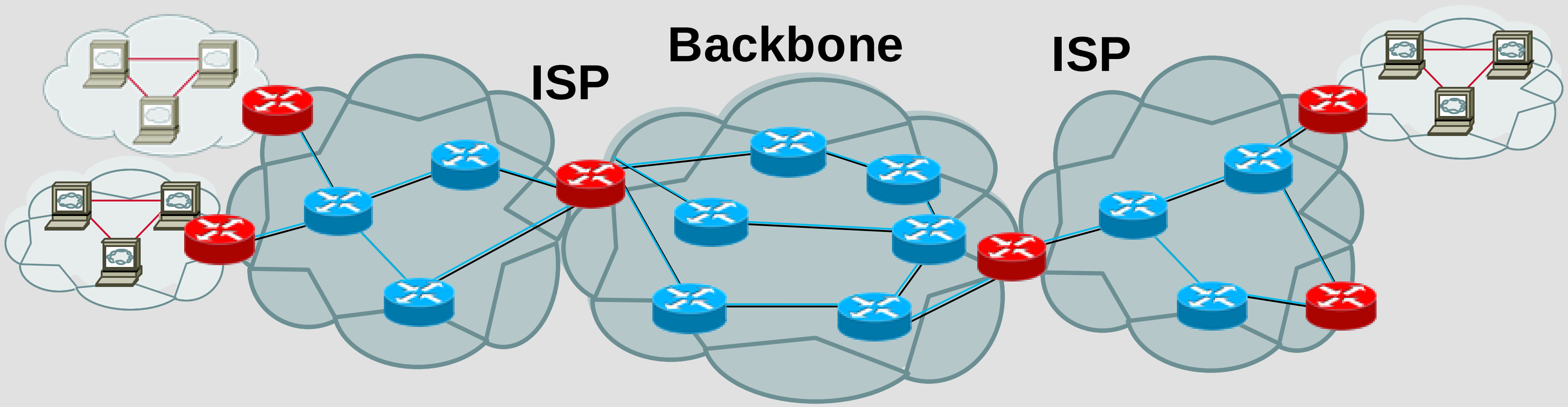
Developers who use JCA for key generation, signing or random number generation should update their applications to explicitly initialize the PRNG with entropy from `/dev/urandom` or `/dev/random`. A suggested implementation is provided at the end of this blog post. Also, developers should evaluate whether to regenerate cryptographic keys or other random values previously generated using JCA APIs such as `SecureRandom`, `KeyGenerator`, `KeyPairGenerator`, `KeyAgreement`, and `Signature`.

Why are things so broken

- Faulty design
- Buggy Specification
- Implementation Errors
- Side-channel leaks
- Misconfiguration
- Gullible users
- Weak Passwords
- Malicious Insiders
- Physical security Failures
- Reliance on third party software
- Malicious software

Network 101

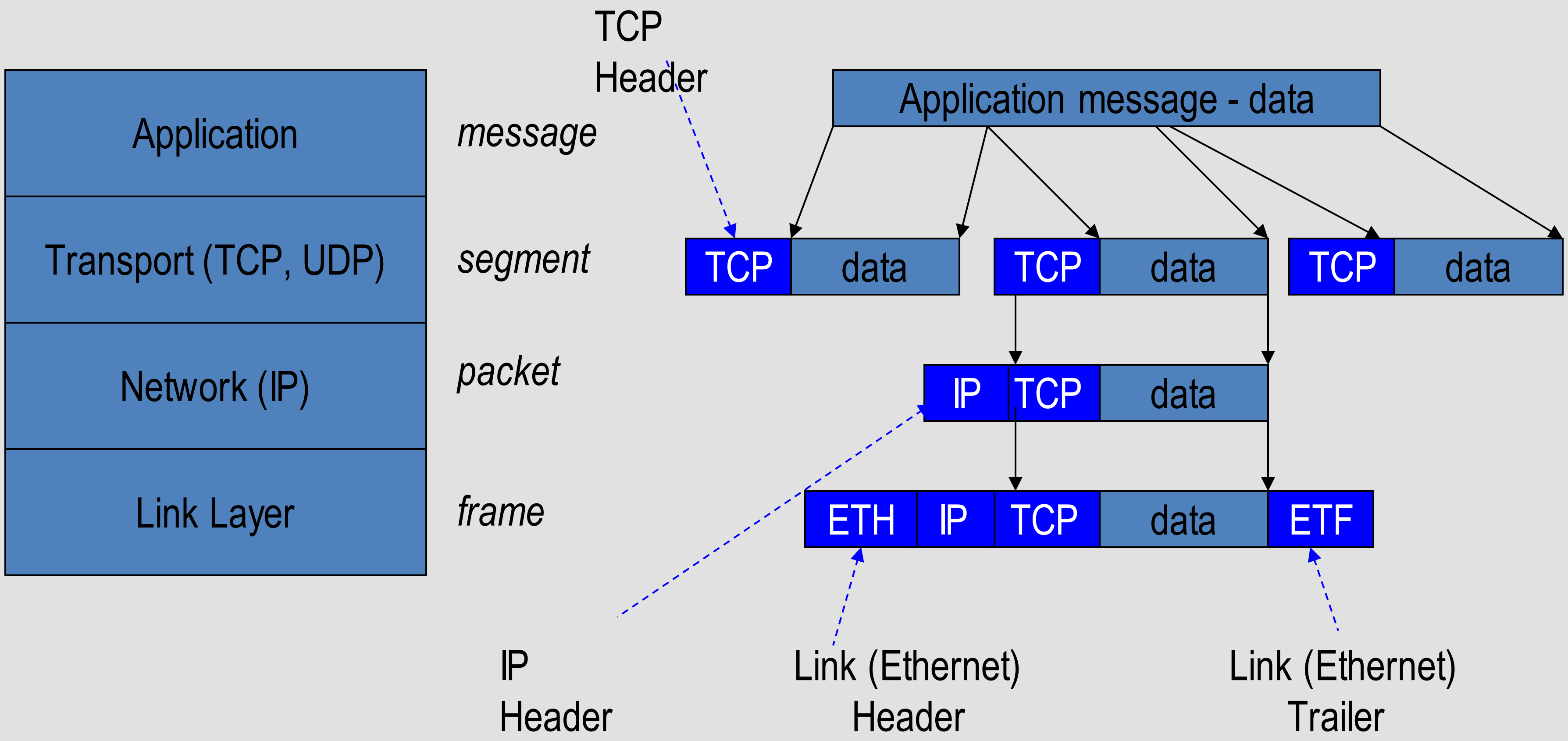
Internet Infrastructure



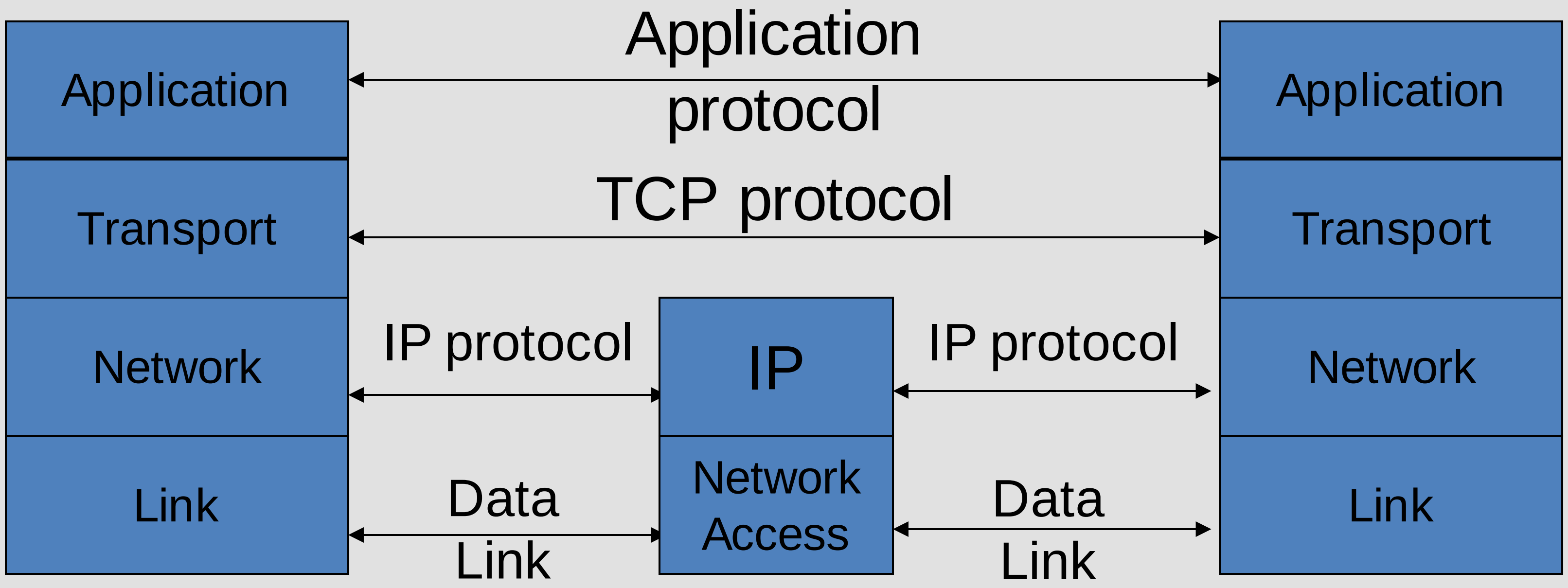
- Local and interdomain routing
  - TCP/IP for routing and messaging
  - BGP for routing announcements

- Domain Name System
  - Find IP address from symbolic name (cse.wustl.edu)

Data Formats



TCP Protocol Stack



Types of Addresses in Internet

- Media Access Control (MAC) addresses in the network access layer
  - Associated w/ network interface card (NIC)
  - 00-50-56-C0-00-01
- IP addresses for the network layer
  - IPv4(32 bit) vs IPv6(128 bit)
  - 128.1.1.3 vs fe80::fc38:6673:f04d:b37b%4
- IP addresses + ports for the transport layer
  - E.g., 10.0.0.2:8080
- Domain names for the application/human layer
  - E.g., www.wustl.edu

Routing and Translation of Addresses (All of them are attack surfaces)

- Translation between IP addresses and MAC addresses
  - Address Resolution Protocol (ARP) for IPv4
  - Neighbor Discovery Protocol (NDP) for IPv6

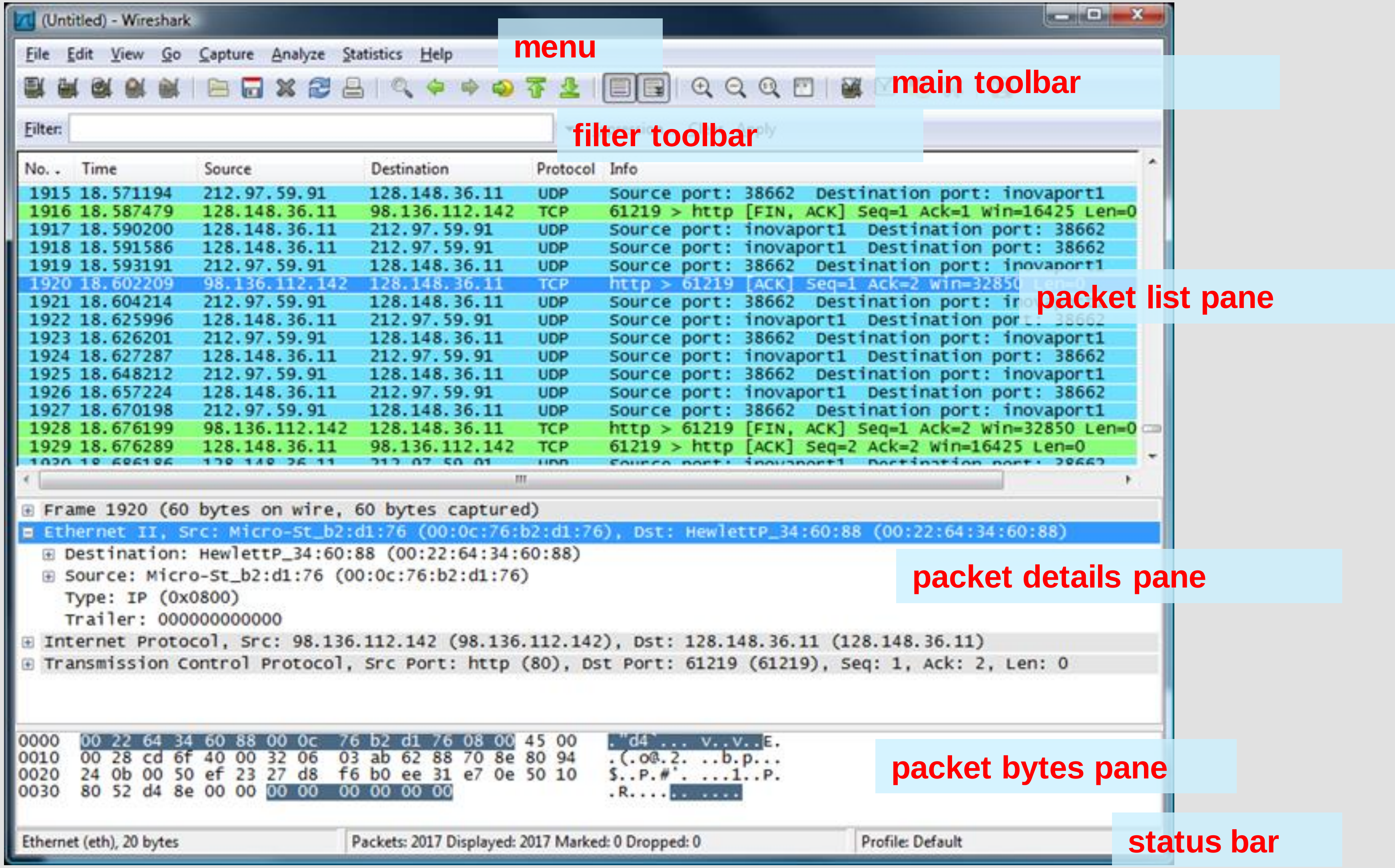
- Routing with IP addresses
  - TCP, UDP for connections, IP for routing packets
  - Border Gateway Protocol for routing table updates

- Translation between IP addresses and domain names
  - Domain Name System (DNS)

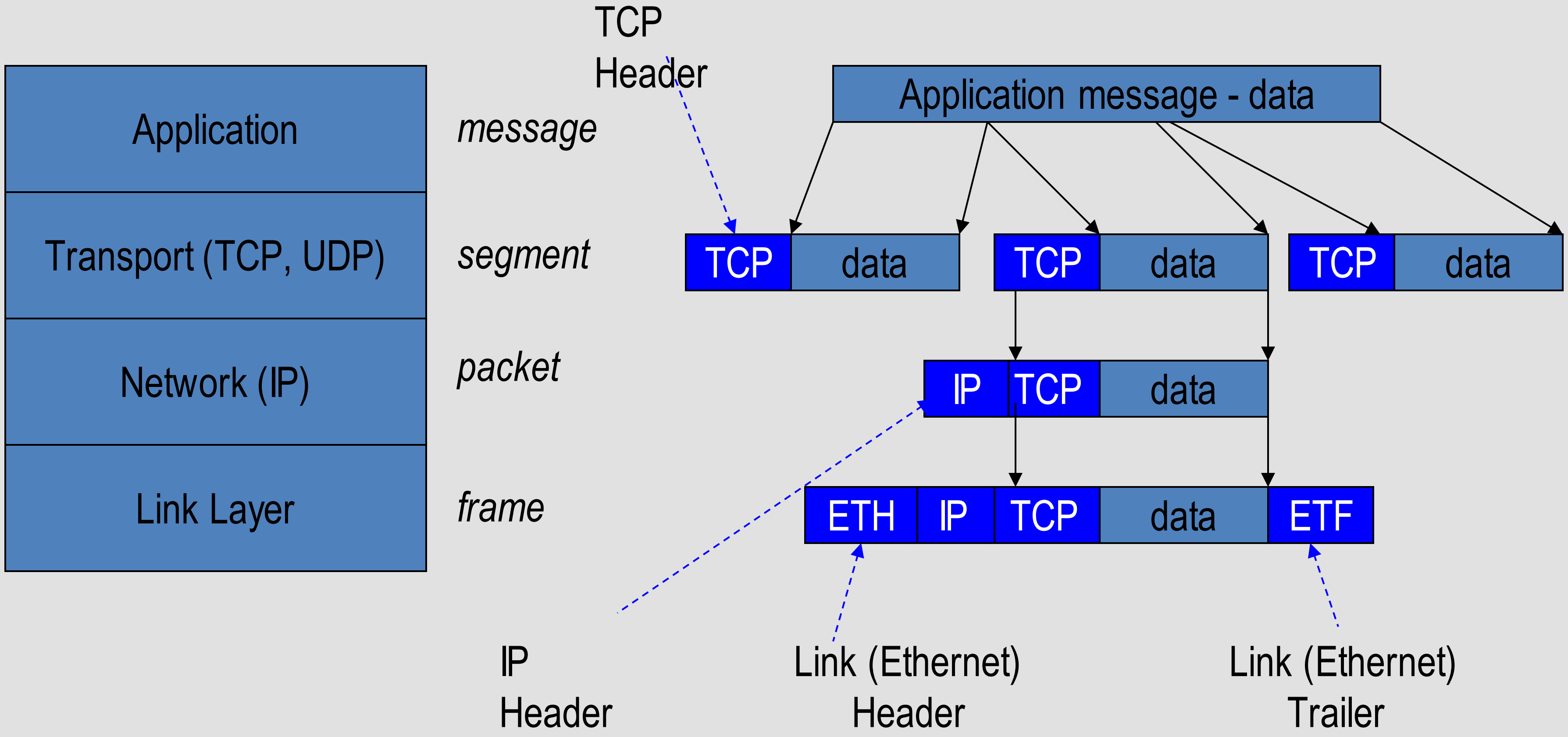
Network Monitoring Tool: Wireshark

- Wireshark is a packet sniffer and protocol analyzer
  - Captures and analyzes frames
  - Supports plugins
- Usually required to run with administrator privileges
- Setting the network interface in promiscuous mode captures traffic across the entire LAN segment and not just frames addressed to the machine
- Freely available on www.wireshark.org

Wireshark GUI



Layer Summary



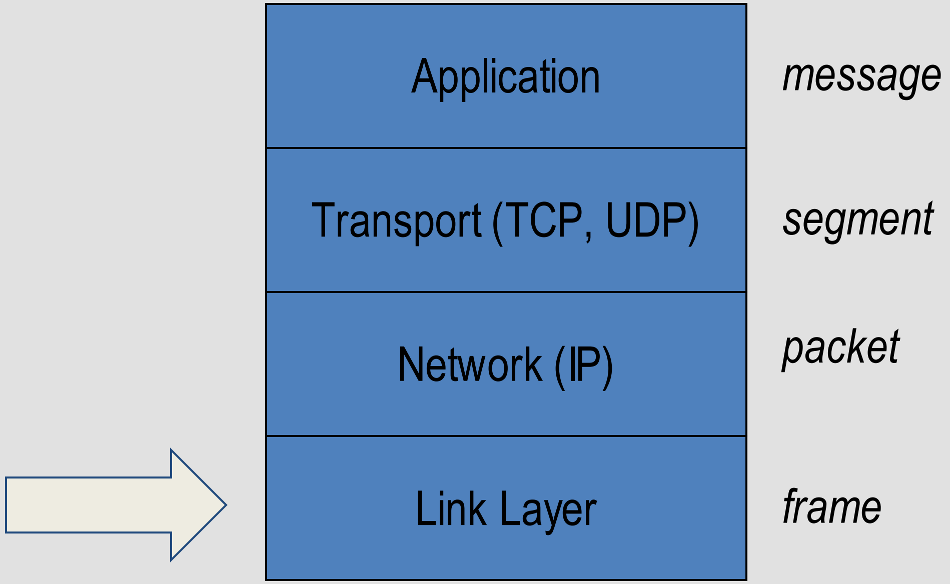
In Class Discussion

- Given the network communication tool you just developed

- How do you attack it?
  - Threat model, target mechanism, properties to violate
  - What assumptions did you violate

- How do you defend it?
  - Threat model, protection mechanism, properties to protect

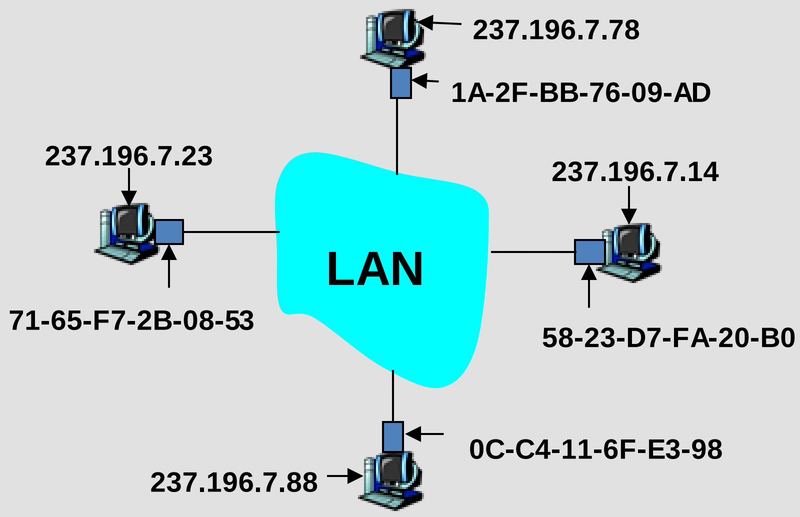
Examining the Link Layer



Routing 101

- When a packet arrives at the destination subnet, MAC address is used to deliver the packet

ARP: Address Resolution Protocol

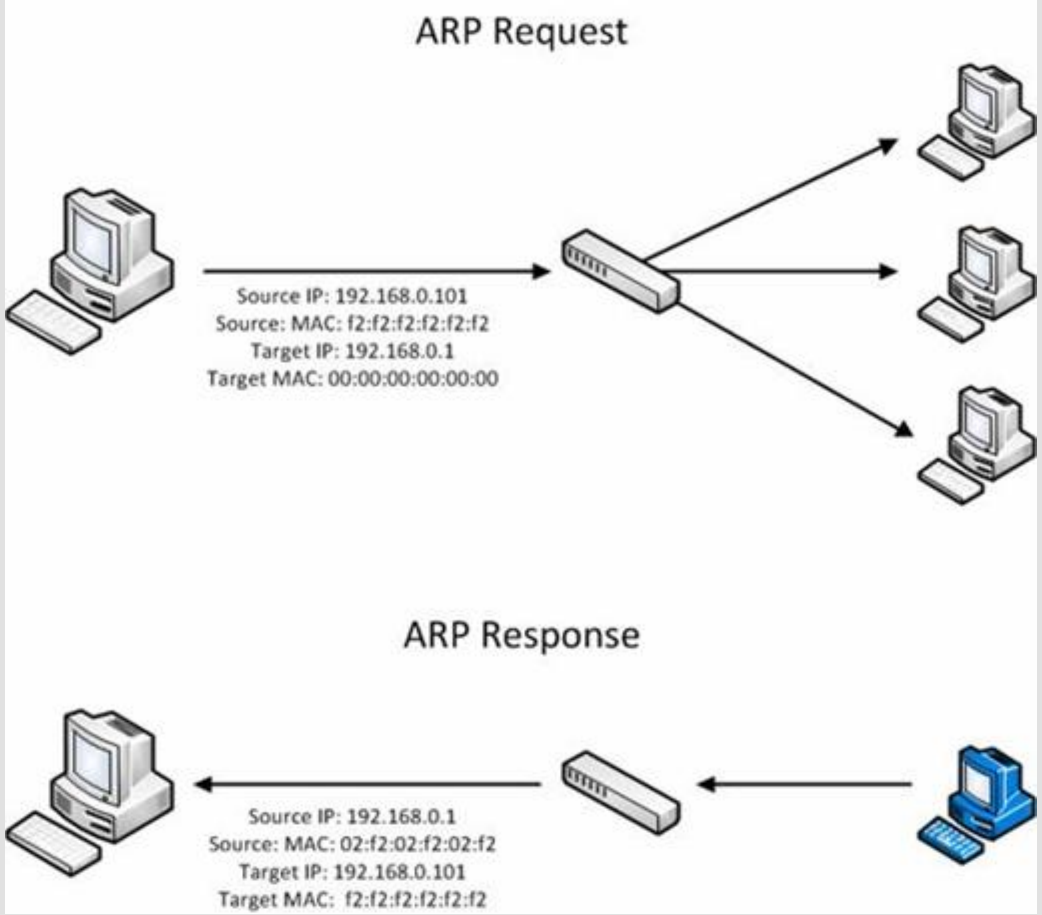


- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>

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

ARP: Address Resolution Protocol

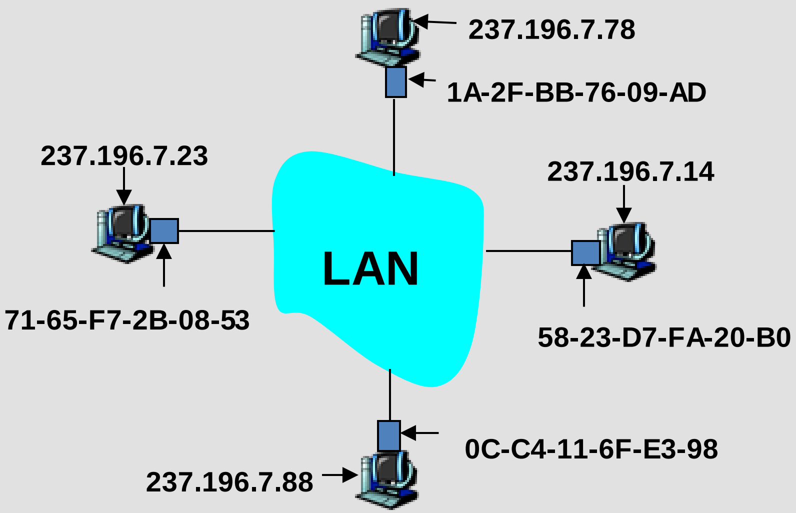


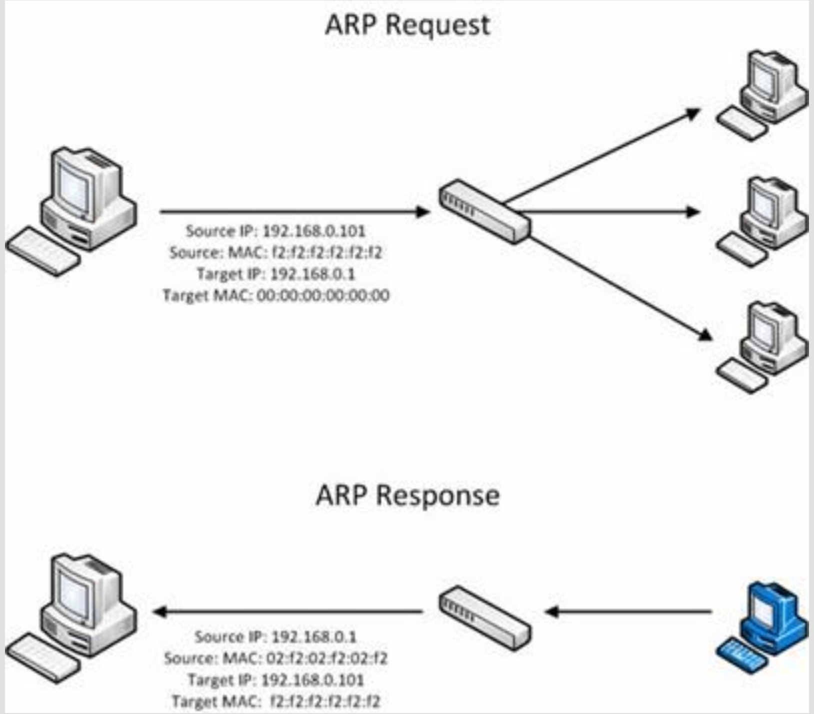
http://www.windowsecurity.com

Discussion

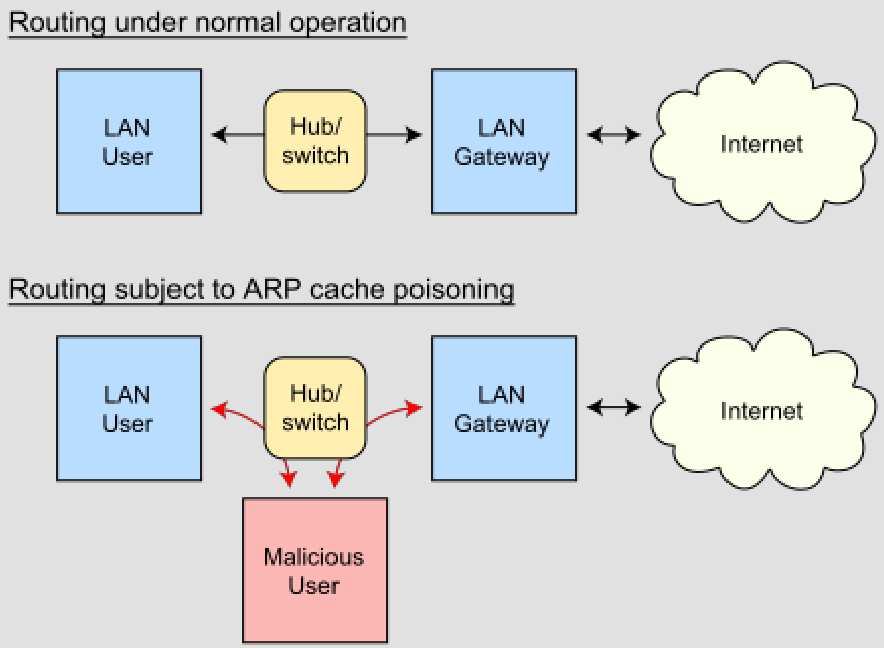
What can go wrong during the IP-to-MAC translation?

- Hint: Try exploiting the ARP request/responses





Problem: Lack of Source Authentication - ARP Spoofing (ARP Poisoning)



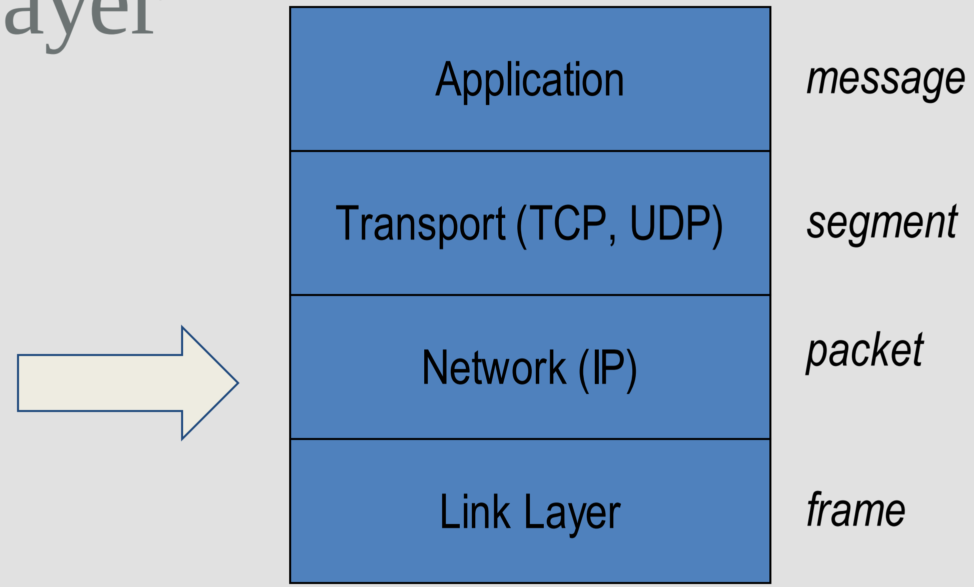
- Send fake or 'spoofed', ARP messages to an Ethernet LAN.
  - To have other machines associate IP addresses with the attacker's MAC
- Legitimate use
  - Implementing redundancy and fault tolerance

ARP Spoofing (Poisoning) Defense

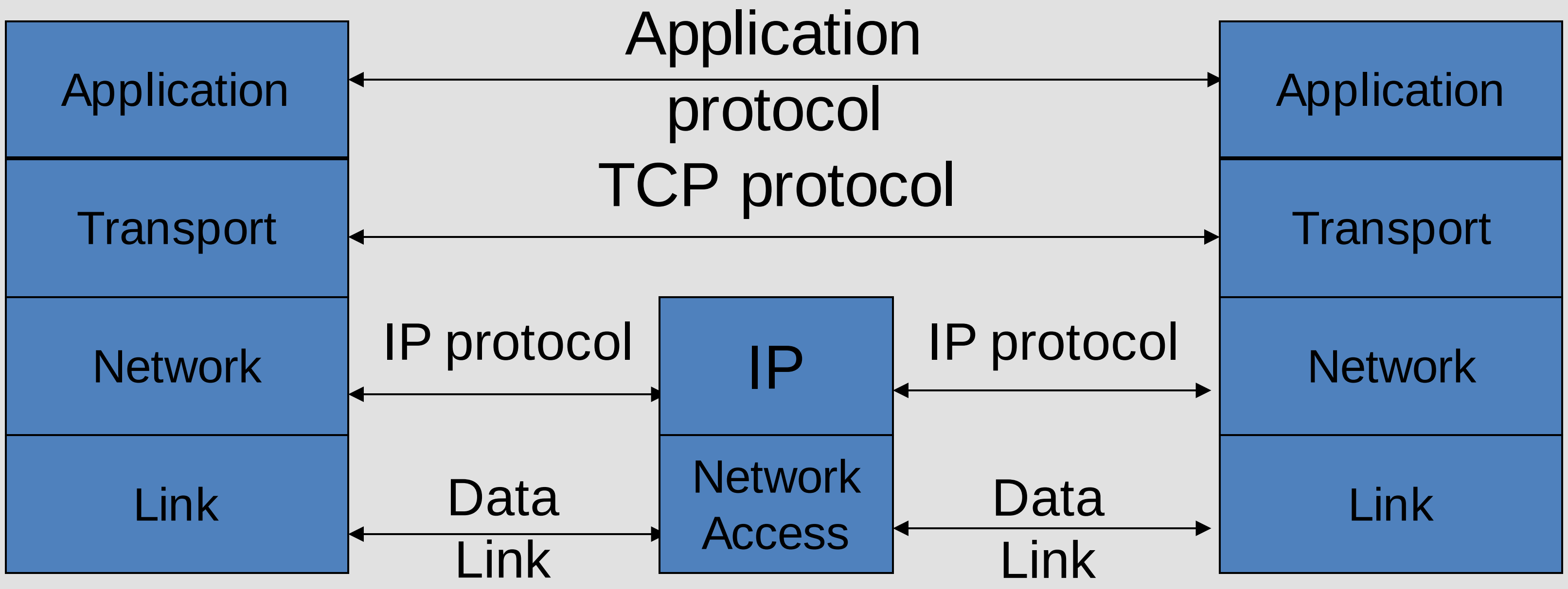
- Prevention
  - Static ARP table
  - DHCP Certification (use access control to ensure that hosts only use the IP addresses assigned to them, and that only authorized DHCP servers are accessible)

- Detection
  - Arpwatch (sending email when updates occur)

Examining the Network Layer



TCP Protocol Stack



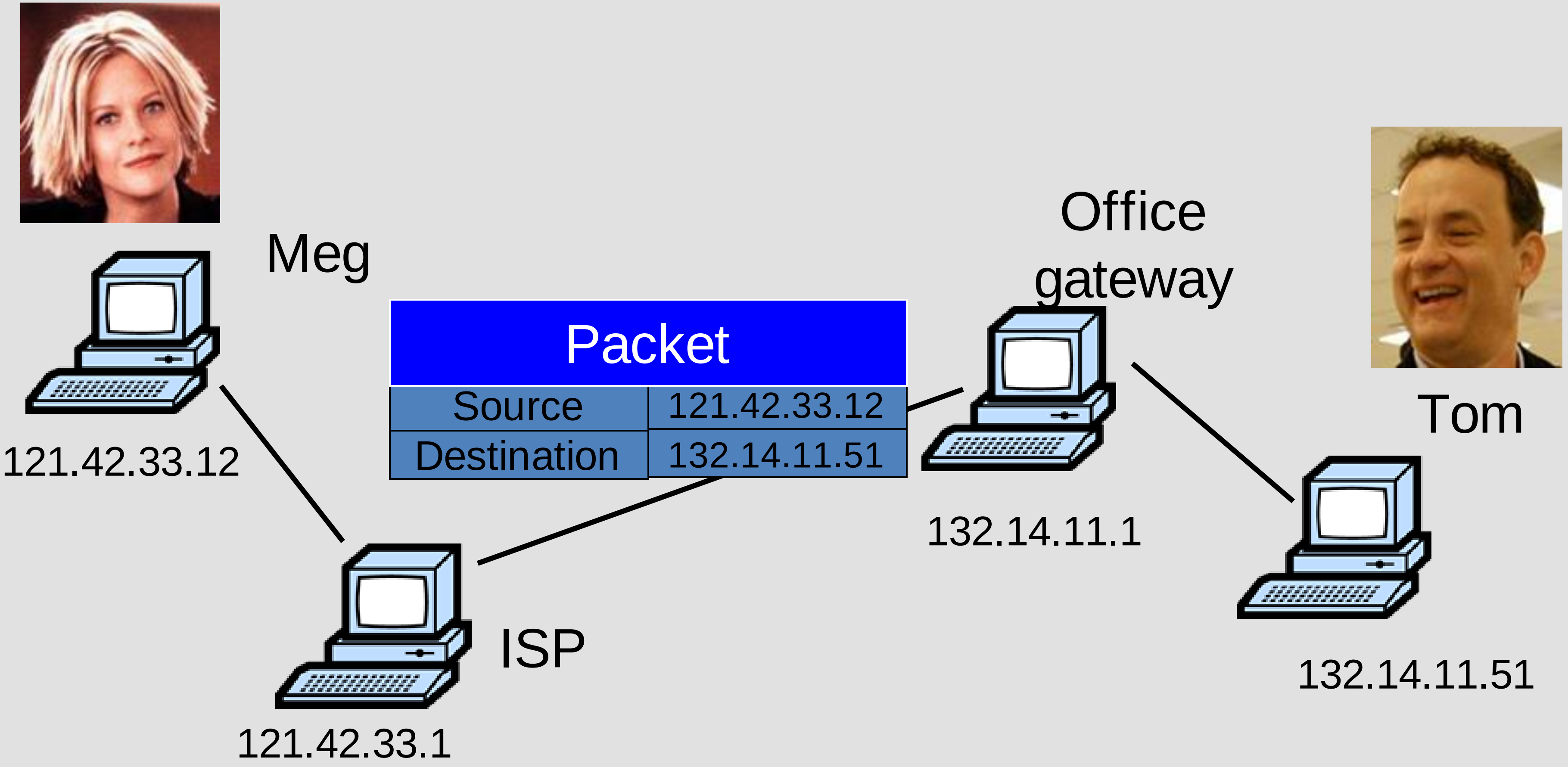
Internet Protocol (IP)

- 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



- Typical route uses several hops
- IP : no ordering or delivery guarantees

Discussion

What can go wrong during IP routing?

- Hint: How can we direct *all* packets to the victim?

What can we do to prevent the attacks?

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	

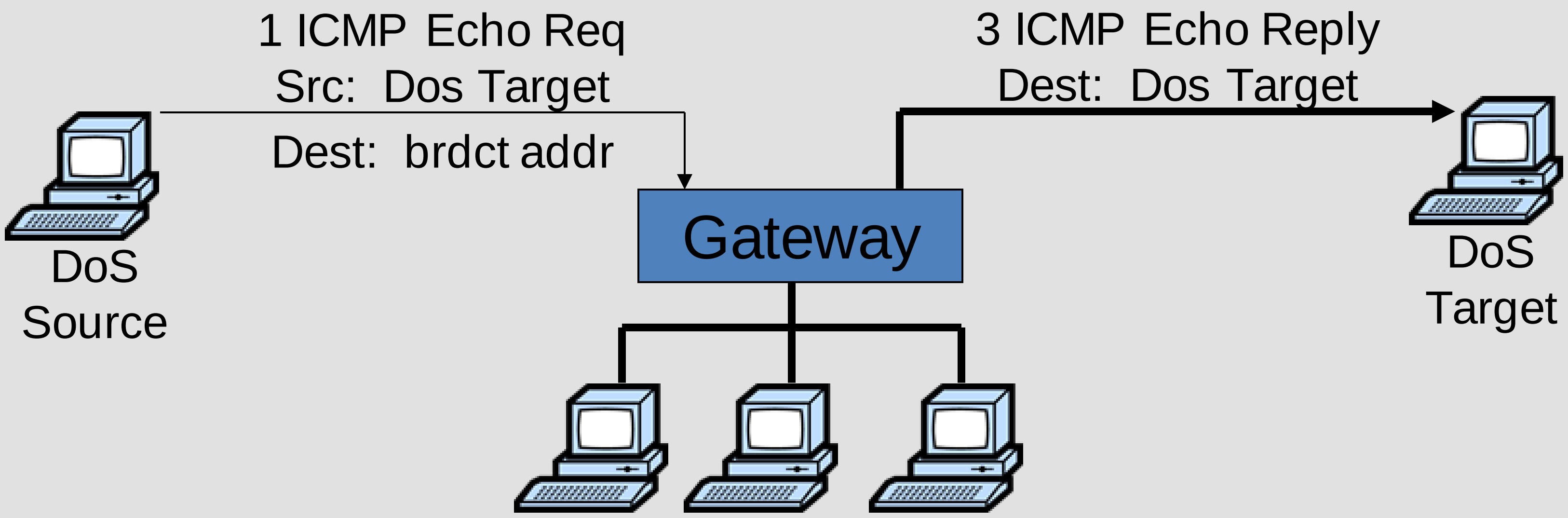
Problem: Lack of Source IP Authentication

- 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
  - **Scapy**: a python library for packet crafting

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

- Implications:
  - Anonymous DoS attacks (e.g. smurf amplification)
  - Anonymous infection attacks (e.g. slammer worm)

Implication: Smurf Amplification DoS attack

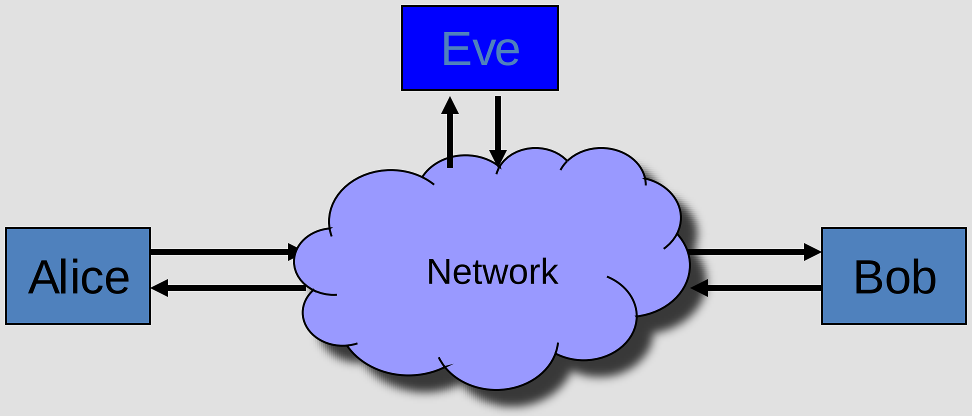


- Send ping request to broadcast addr (ICMP Echo Req)
- Lots of responses:
  - Every host on target network generates a ping reply (ICMP Echo Reply) to victim

Prevention: Reject external packets to broadcast address

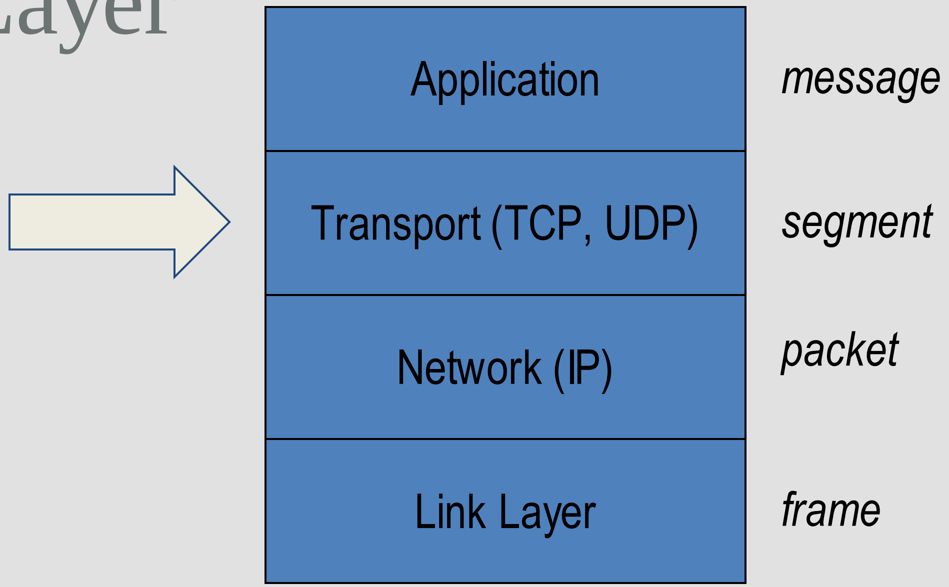
Problem: Lack of Confidentiality Protection - Packet Sniffing

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

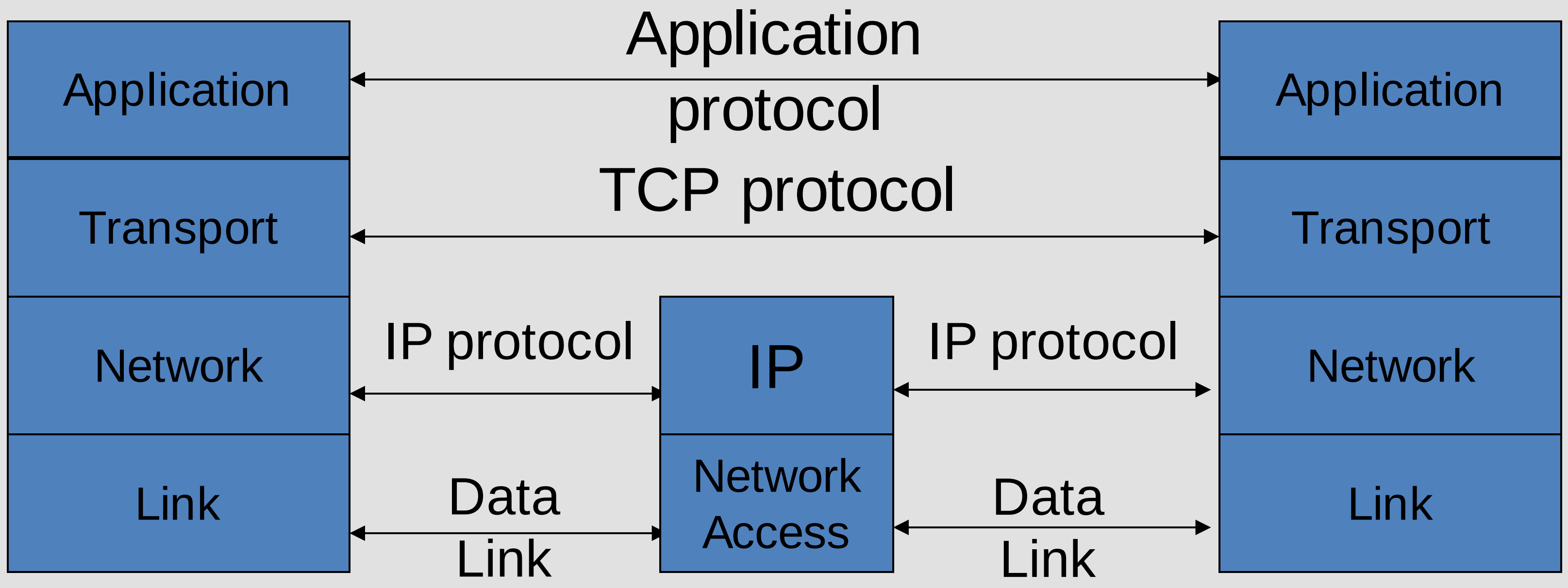


Prevention: Encryption (IPSEC, TLS)

Examining the Transport Layer

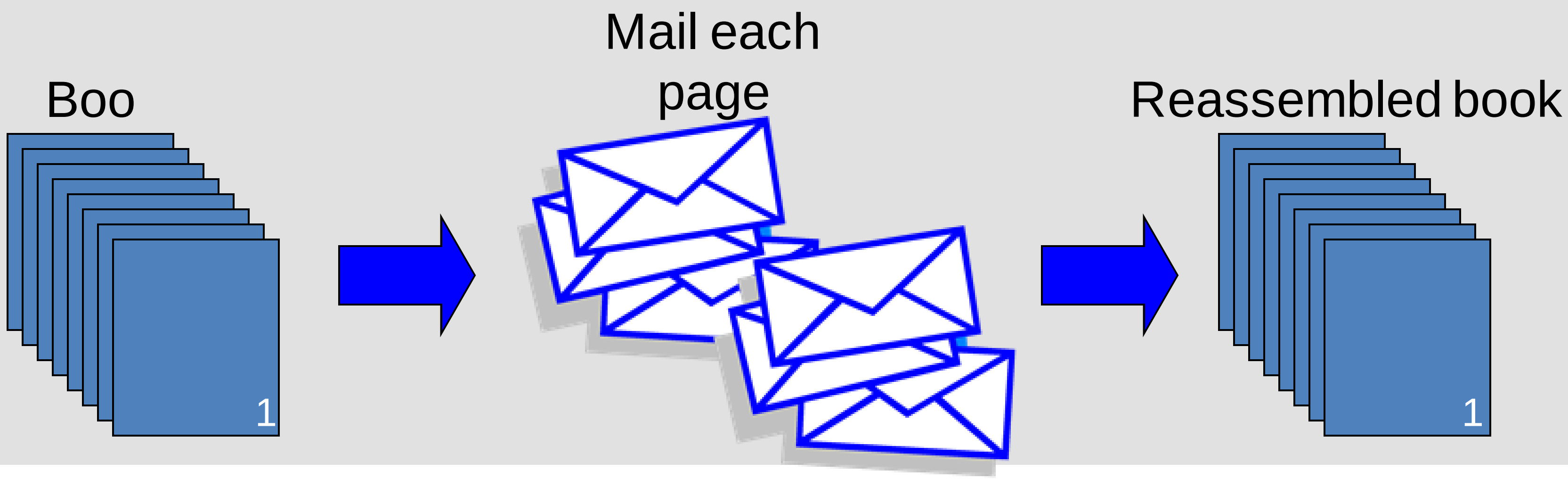


TCP Protocol Stack

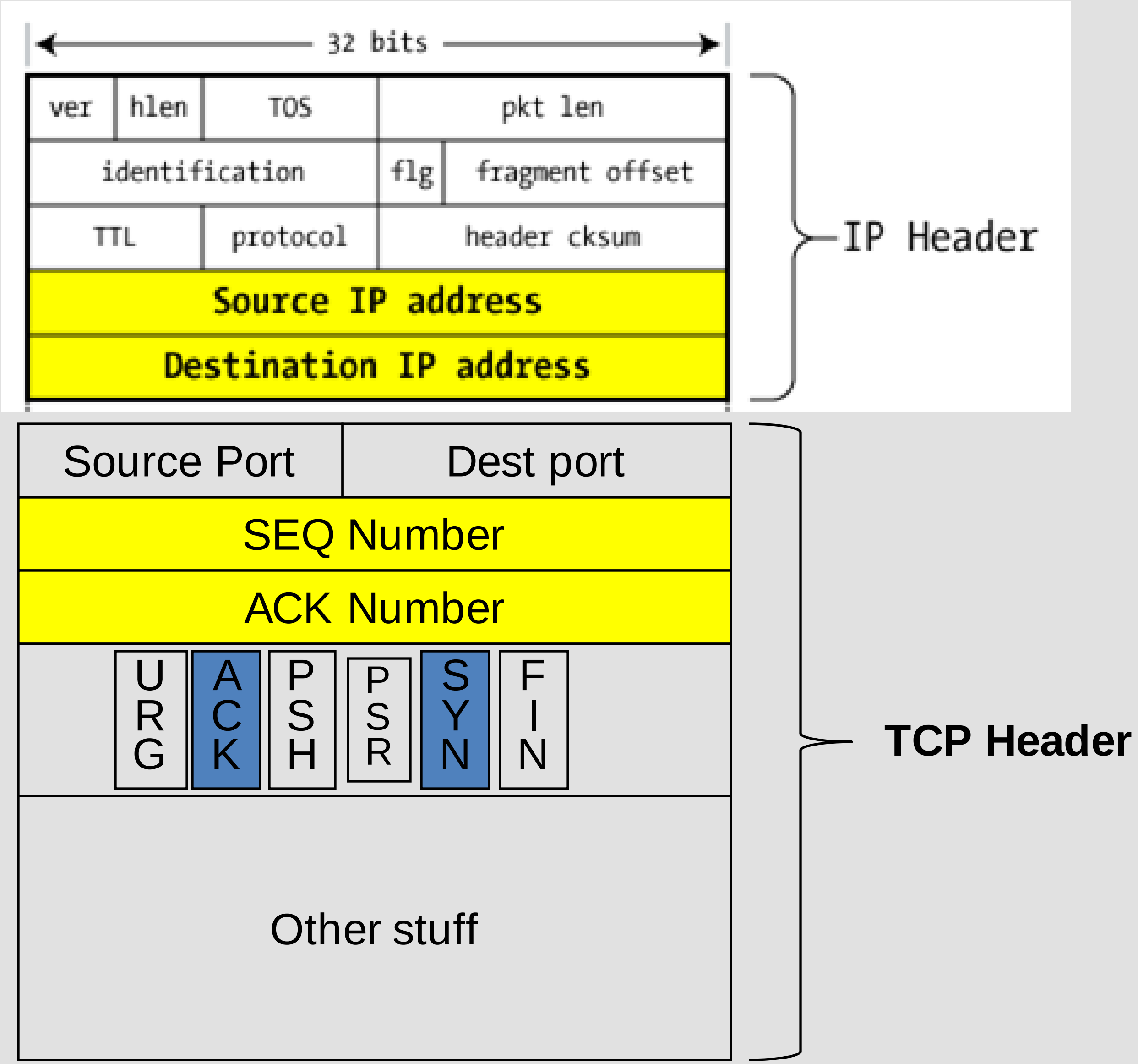


Transmission Control Protocol (TCP)

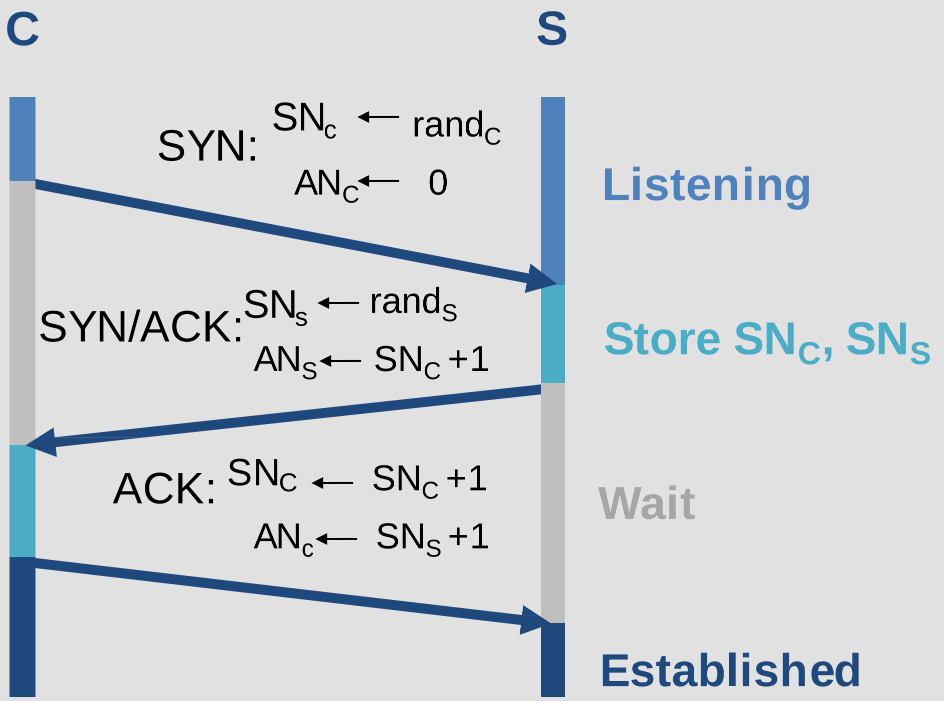
- Connection-oriented, preserves order
  - Sender
    - Break data into packets
    - Attach packet numbers
  - Receiver
    - Acknowledge receipt; lost packets are resent
    - Reassemble packets in correct order



TCP Header (protocol=6)



TCP Handshake

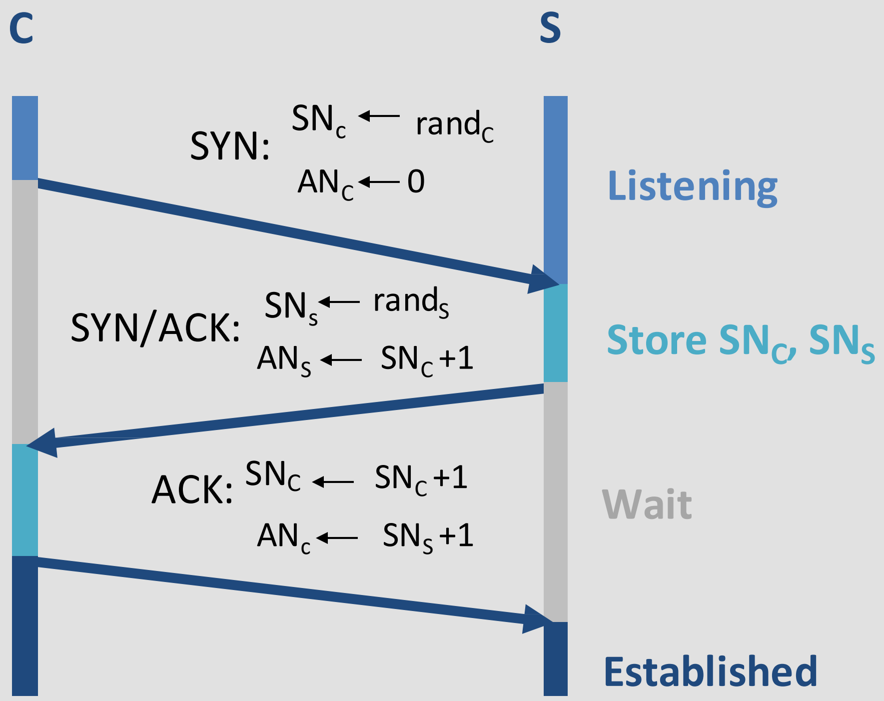


Discussion

What can go wrong during the handshake?
- Hint: “Don’t leave me hanging!”

Problem 2. Denial of Service (DoS) vulnerabilities
(e.g. TCP SYN Flood)

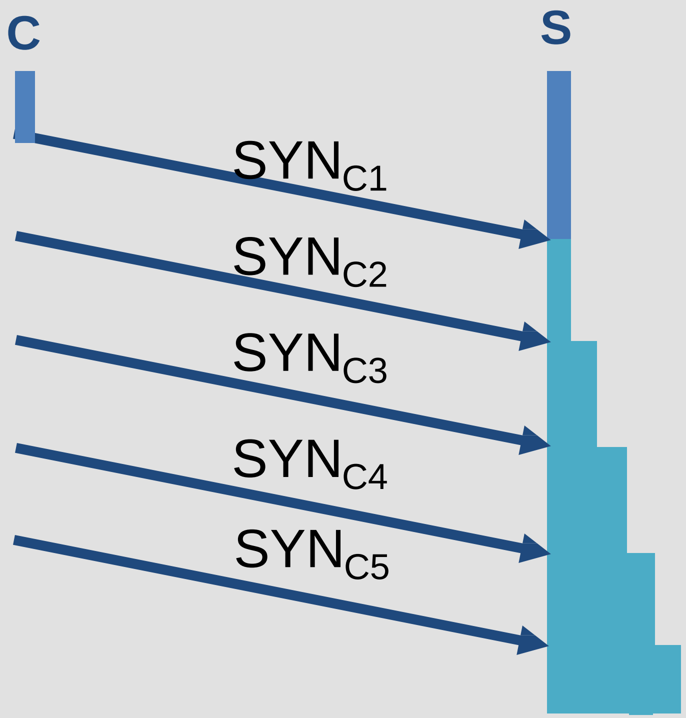
TCP Handshake



Problem: Low Rate TCP SYN Flood

Single machine:

- **SYN Packets with random source IP addresses**
- **Fills up backlog queue on server**
- **No further connections possible**



Problem: Low Rate TCP SYN Flood

A classic SYN flood example

MS Blaster worm (2003)

- Infected machines at noon on Aug 16th:
- SYN flood on port 80 to windowsupdate.com 50
- SYN packets every second
  - each packet is 40 bytes
- Spoofed source IP: a.b.X.Y where X,Y random

MS Solution

- New name: windowsupdate.microsoft.com

TCP SYN Flood Defense

Can you think of any defense mechanisms?
- Hint: If only I have good memory…

Non-solution:

- Increase backlog queue size or decrease timeout

Correct Solution:

- SYN Cookies: remove state from server
- Small performance overhead

SYN COOKIES

Idea: use secret key and data in packet to generate server SN

Server responds to Client with SYN-ACK cookie:

- T = 5-bit **counter incremented every 64 secs.**
- L = MAC_{key} (SAddr, SPort, DAddr, DPort, SN_C, T) [24 bits]

key: picked at random during boot

- SN_S = (T . mss . L) (|L| = 24 bits)
- **Server does not save state**

Honest client responds with
ACK (AN=SN_S +1, SN=SN_C+1):

- **Server allocates space for socket only if valid SN_S**

Discussion

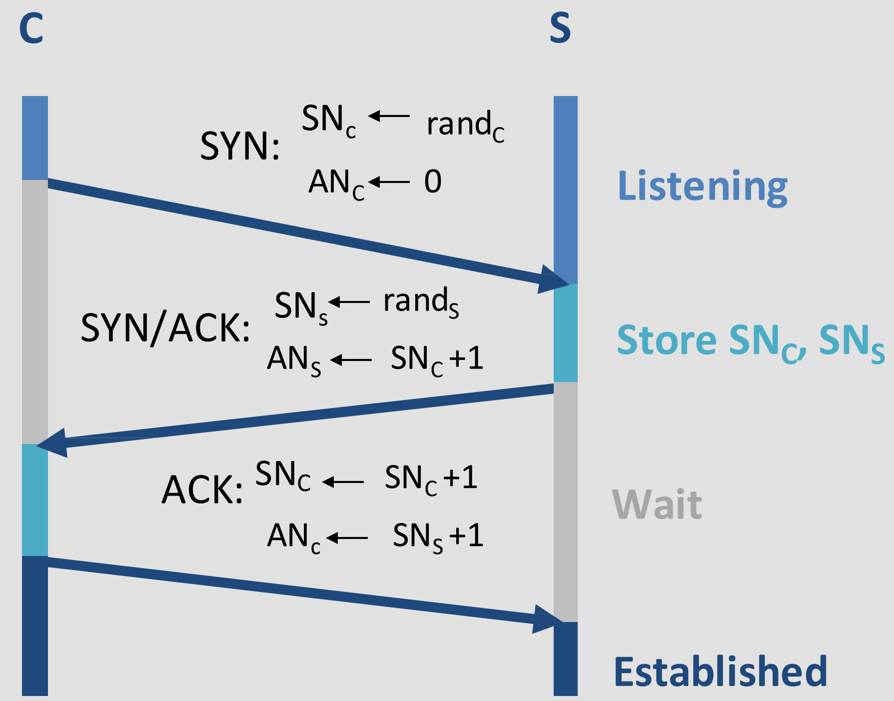
What else can go wrong during the handshake?

- If the seq numbers (SN) are not random…

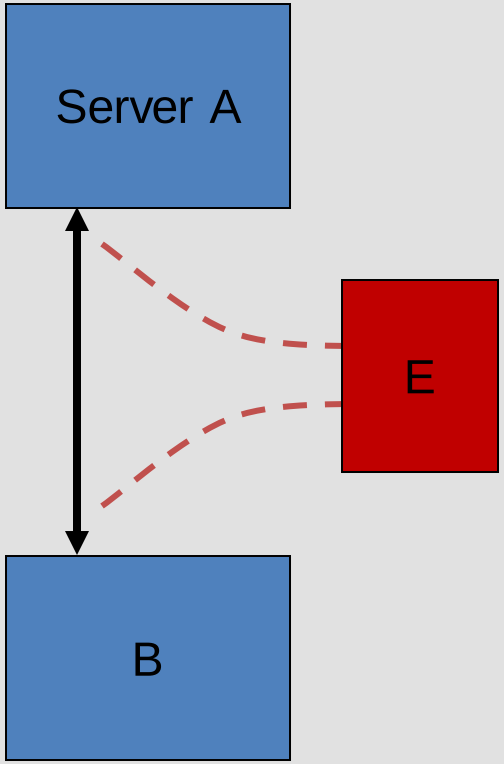
Problem 3. TCP state easily obtained by eavesdropping

– Enables spoofing and session hijacking

TCP Handshake



Problem: Hijacking Existing TCP Connection



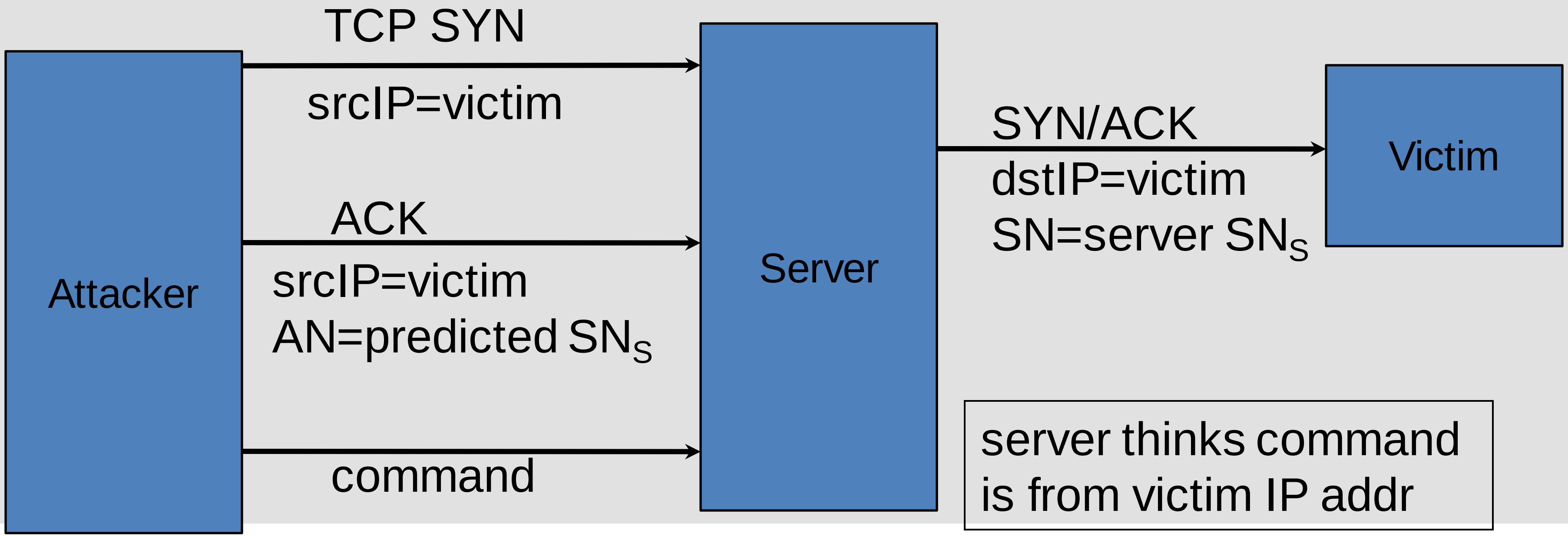
- A, B trusted connection
  - Send packets with **predictable** seq numbers
- E impersonates B to A
  - *DoS B's queue*
  - Sends packets to A that resemble B's transmission
  - E cannot receive, but may execute commands on A

Attack can be blocked if E is outside firewall.

Random Initial Sequence Numbers

Suppose initial seq. numbers (SN_C , SN_S) are predictable:

- Attacker can create TCP session on behalf of forged source IP
- Breaks IP-based authentication (e.g. SPF, /etc/hosts)
  - Random seq. num. do not prevent attack, but make it harder



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.

Let’s take a look at how it is used

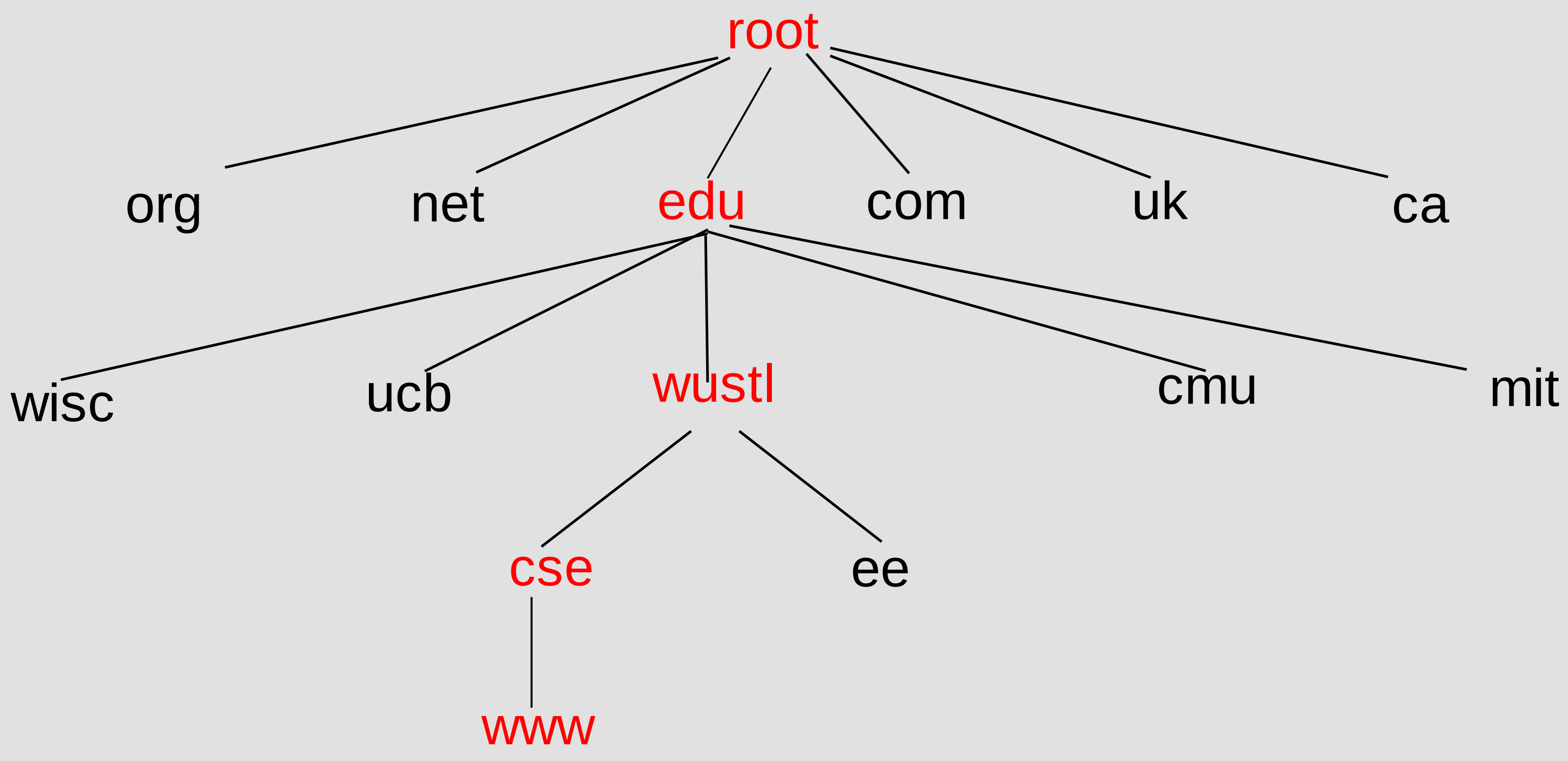
- https://youtu.be/KIWOYkicnlw?t=19m41s

Don’t do this on a public network!

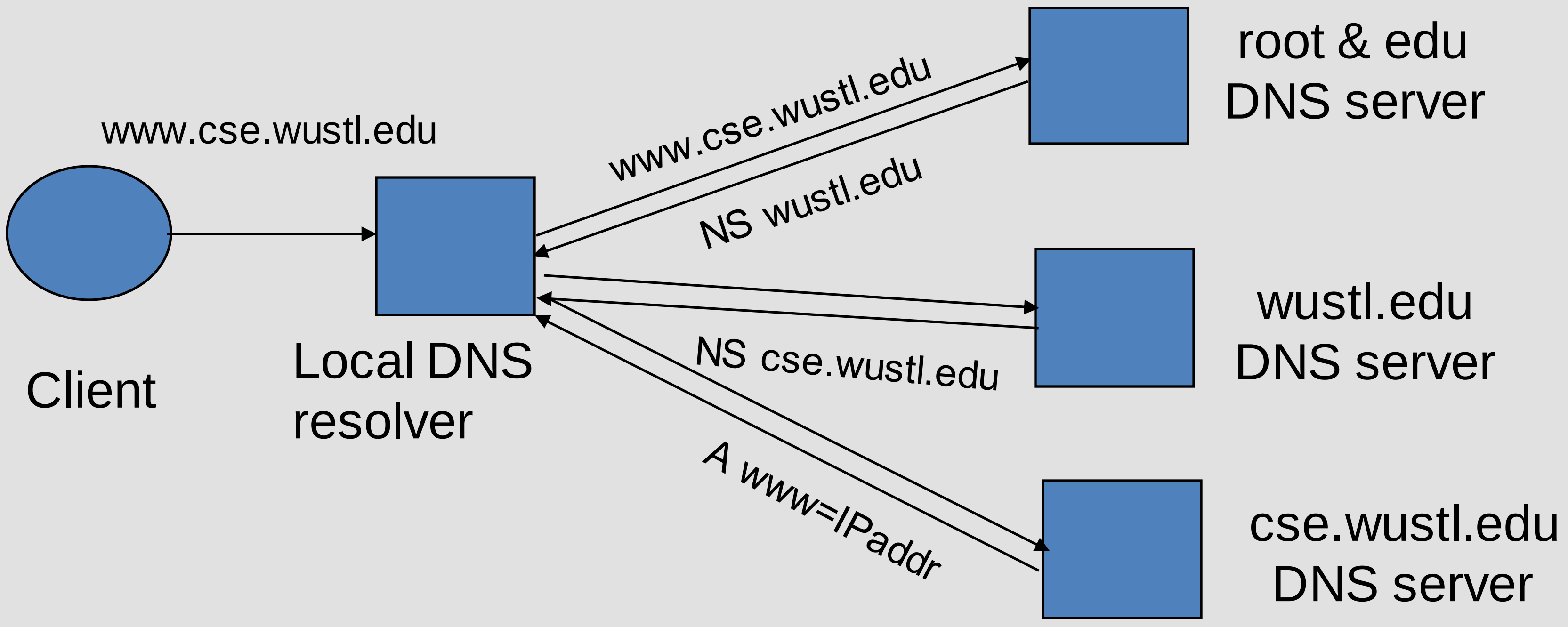
Domain Name System

Domain Name System (DNS)

- Hierarchical Name Space



DNS Lookup Example

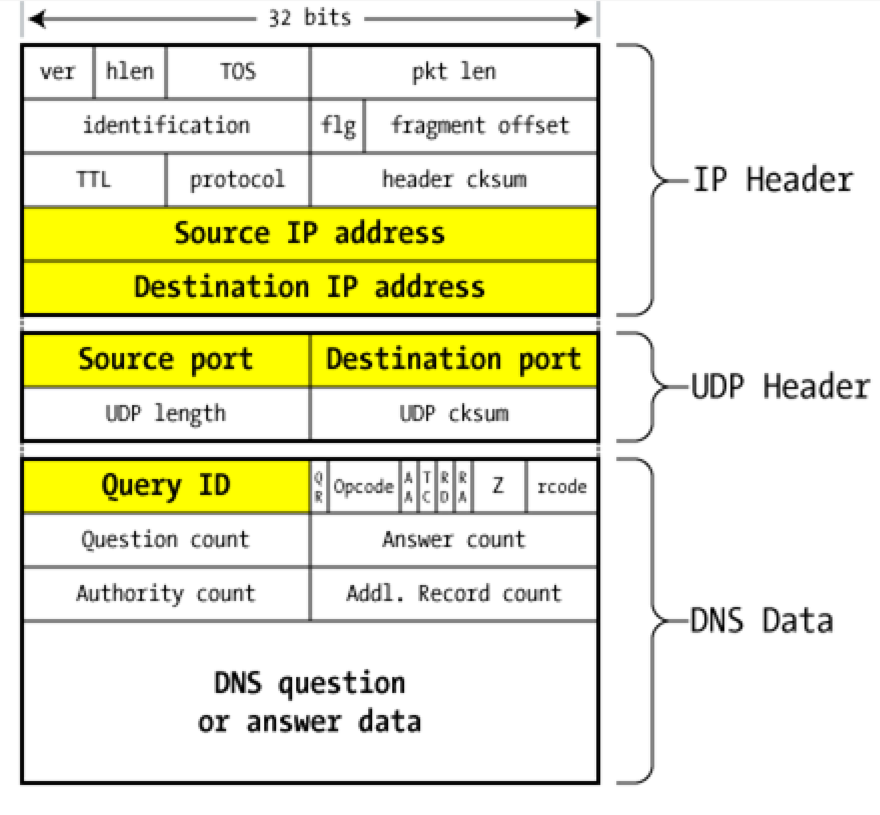


DNS record types (partial list):

- 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)

DNS Packet

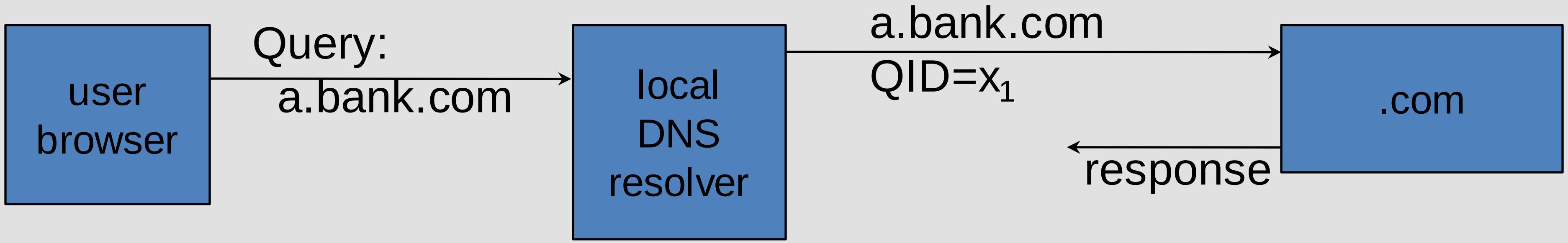
- Query ID:
  - 16 bit random value
  - Links response to query



(from Steve Friedl)

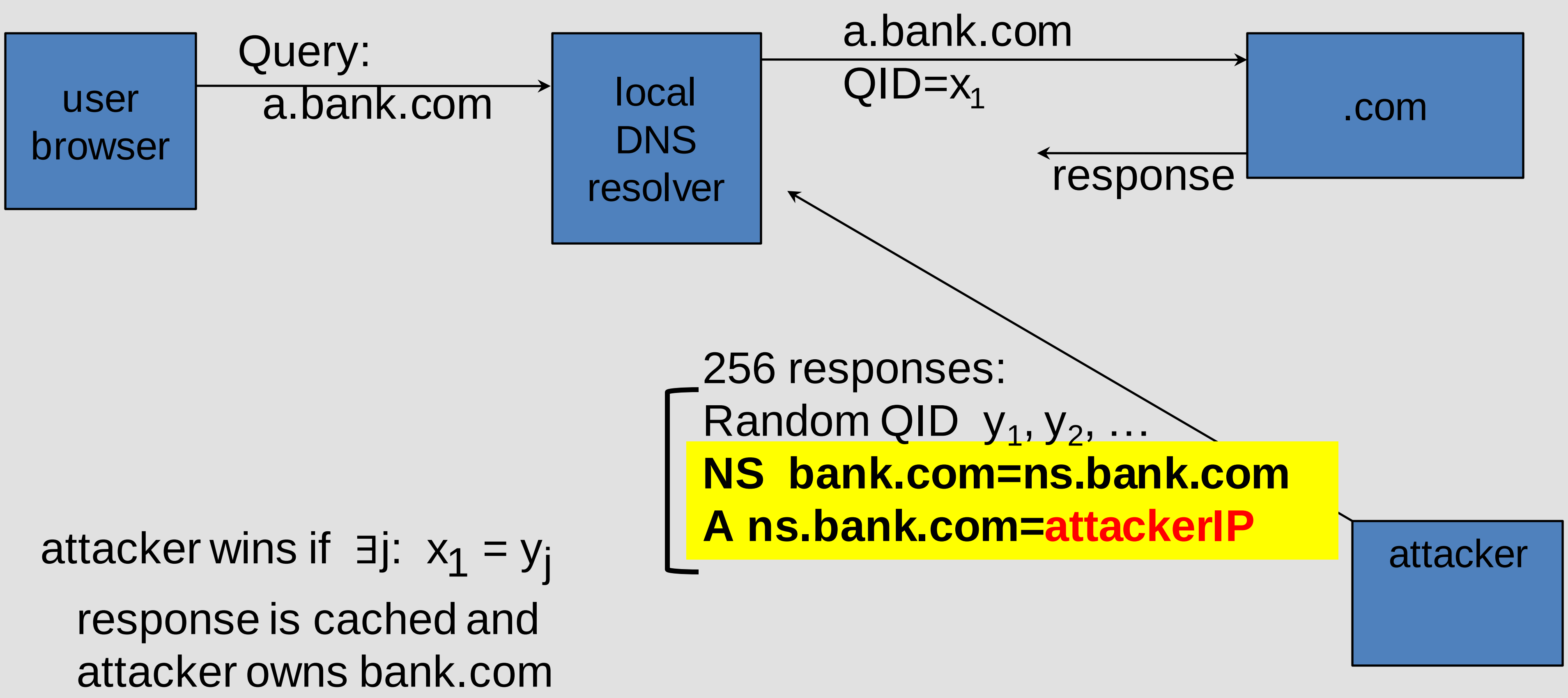
Discussion

How can the attacker hijack this DNS Lookup session?



DNS Cache Poisoning (a la Kaminsky'08)

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



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. malicious access point in a Cafe
  - Solution – authenticated requests/responses
    - Provided by DNSsec … but few use DNSsec

Summary of Threats

- Confidentiality
  - Packet sniffing
- Integrity
  - ARP poisoning
  - UDP spoofing
  - TCP Session hijacking
  - DNS poisoning
- Availability
  - Denial of service attacks
- Common
  - Address translation poisoning attacks (DNS, ARP)
  - Packet Spoofing

Competition

- **Objective:** Destroy other teams' flags

- **Rules:**
  - No physical attack
  - No permanent denial of service
  - No self-replicating or self-propagating malware
  - No attacks against other team's computing infrastructure
  - No attacks against instructor's computing infrastructure

- **Local Network:** Tenda_6CB460, pwd: fillquest448