CS-630: Cyber and Network Security

Lecture # 13: Security Problems in Network Protocols and Defense Mechanisms

Prof. Dr. Sufian Hameed

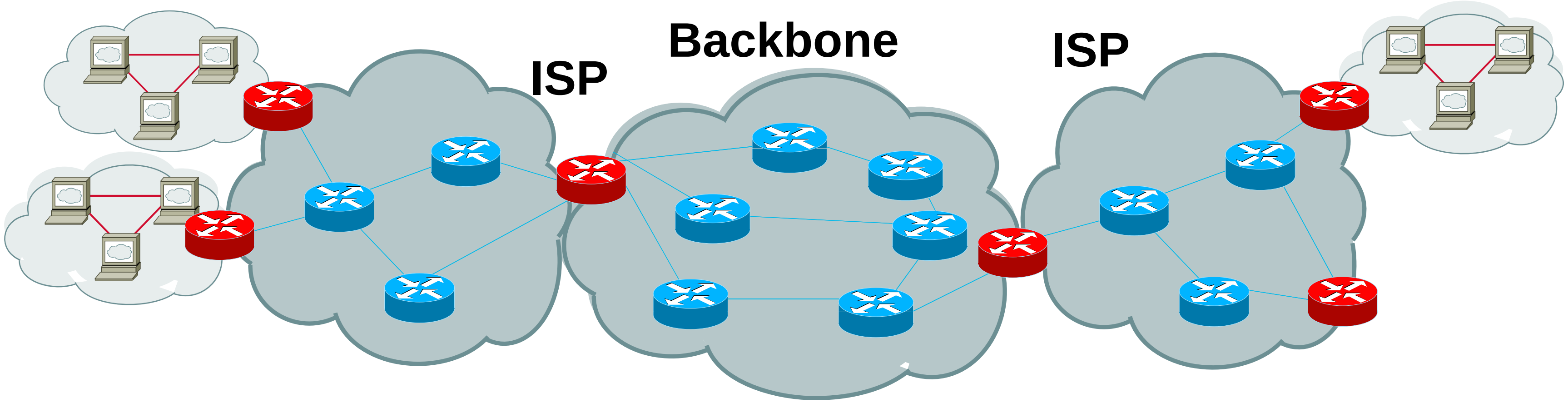
Department of Computer Science

FAST-NUCES

Overview

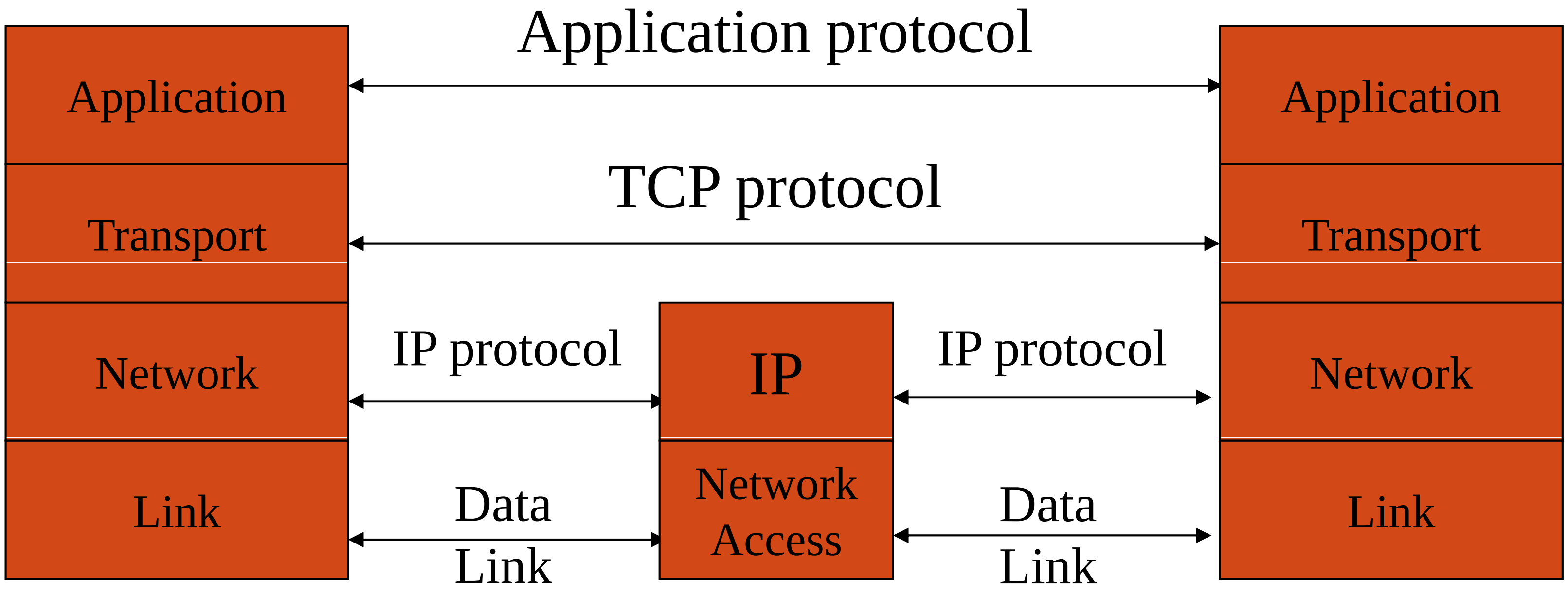
- *How the Internet works and some basic vulnerabilities*
- *Network protocol security*
  - ***IPSEC***
  - *BGP instability*
  - *DNS rebinding and DNSSEC*
- *Standard network defenses*
  - *Firewall*
    - *Packet filter (stateless, stateful), Application layer proxies*
  - *Traffic shaping*
  - *Intrusion detection*
    - *Anomaly and misuse detection*

Internet Infrastructure

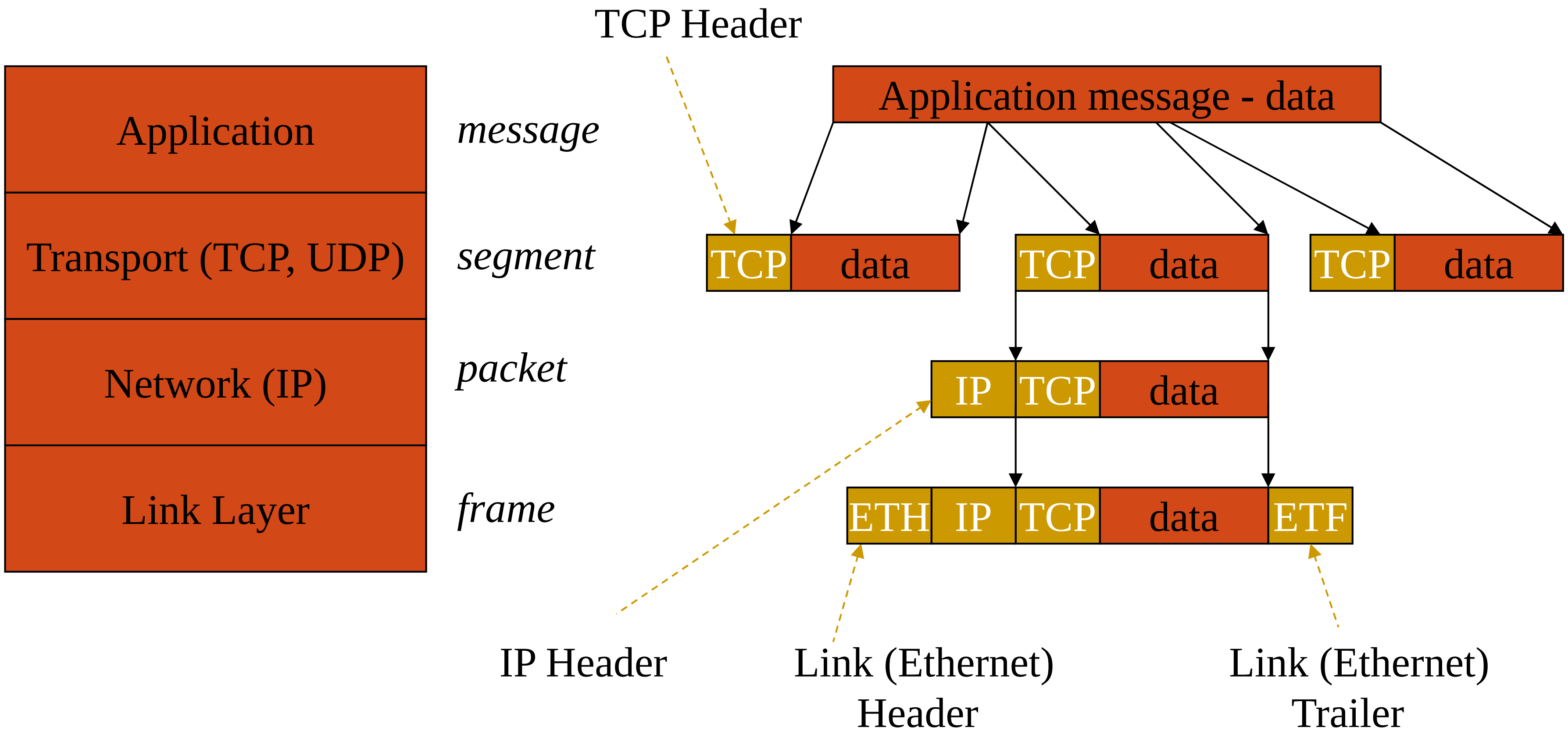


- Local and interdomain routing
  - TCP/IP for routing and messaging
  - BGP for routing announcements
- Domain Name System
  - Find IP address from symbolic name (www.khi.nu.edu.pk)

TCP Protocol Stack

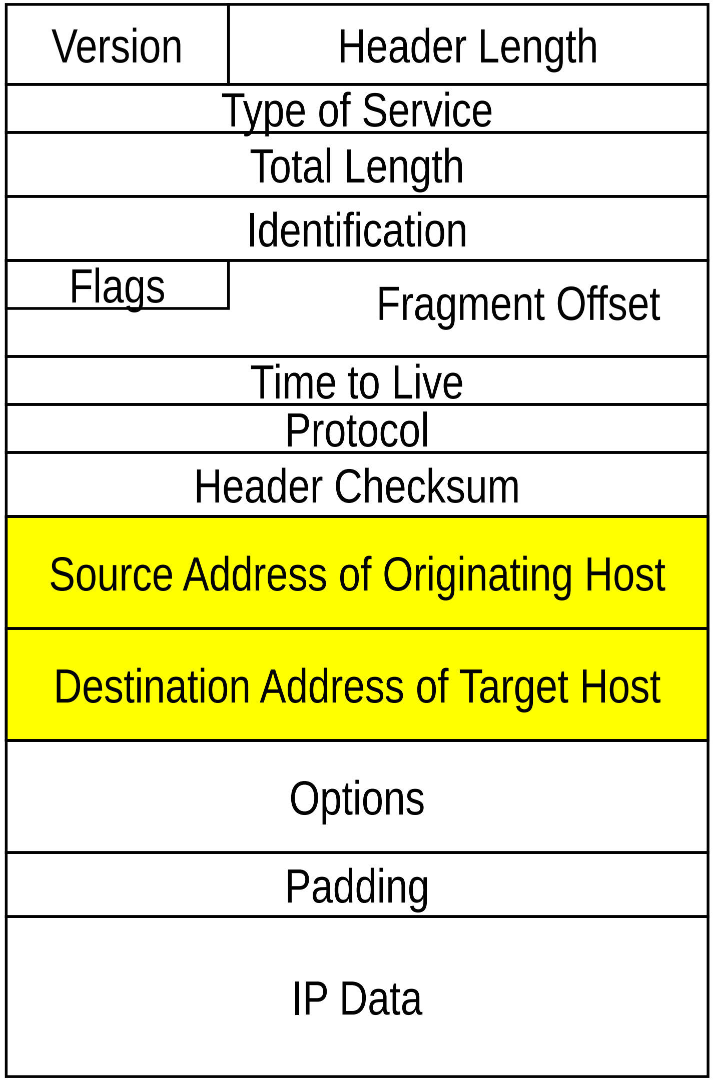


Data Formats



Internet Protocol

- Connectionless
  - Unreliable
    - No central monitoring or performance measurement facility exists that tracks or maintains the state of the network
  - Best effort
    - Users get unspecified variable bit rate and delivery time, depending on the current traffic load
- 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

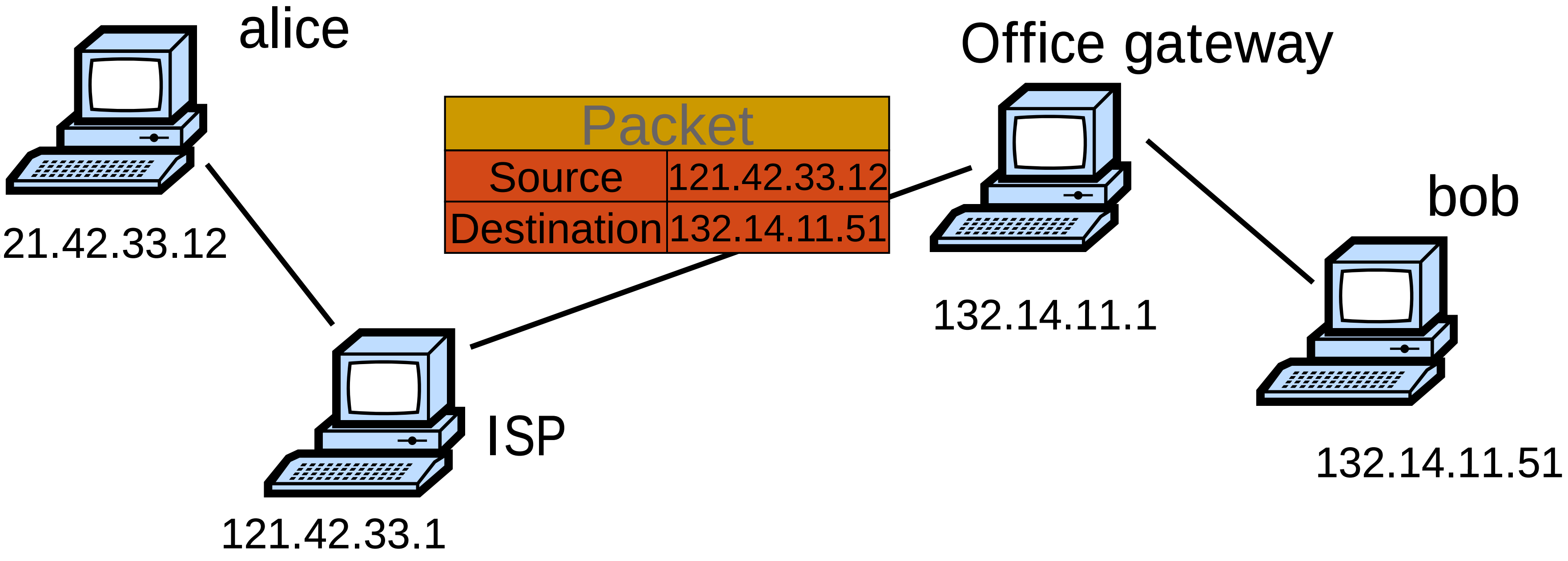
Destination Address of Target Host

Options

Padding

IP Data

IP Routing



Packet	
Source	121.42.33.12
Destination	132.14.11.51

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

IP Protocol Functions (Summary)

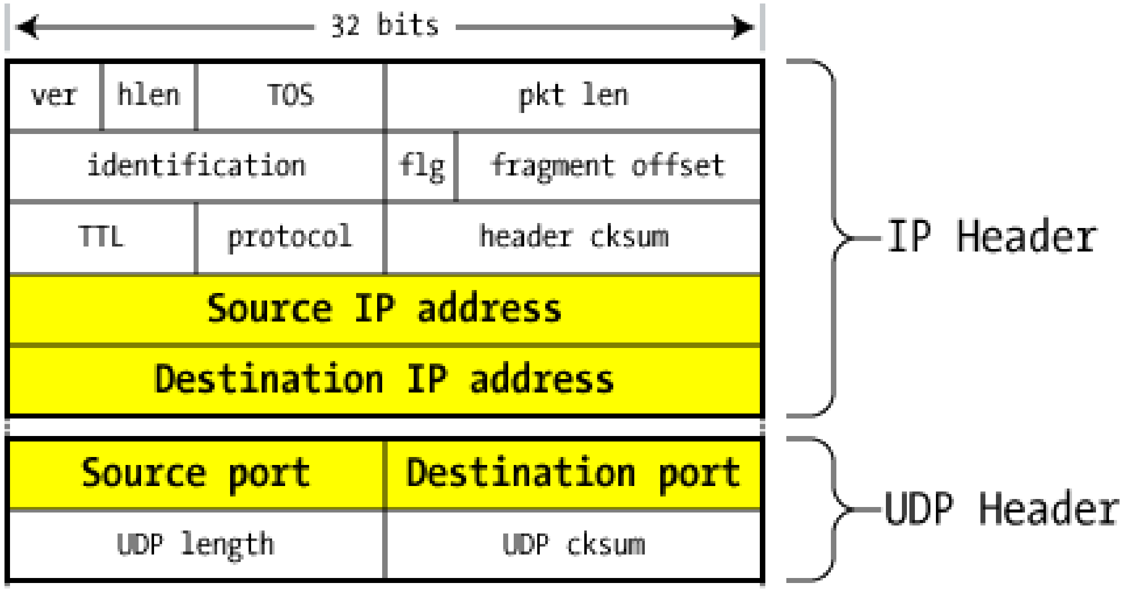
- Routing
  - IP host knows location of router (gateway)
  - IP gateway must know route to other networks
- Fragmentation and reassembly
  - If max-packet-size less than the user-data-size
- Error reporting
  - ICMP packet to source if packet is dropped
- TTL field: decremented after every hop
  - Packet 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
    - a **raw socket** is an internet socket that allows direct sending and receiving of Internet Protocol packets without any protocol-specific transport layer formatting.
  - **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:
  - Anonymous DoS attacks;
  - Anonymous infection attacks (e.g. slammer worm)

User Datagram Protocol (UDP)

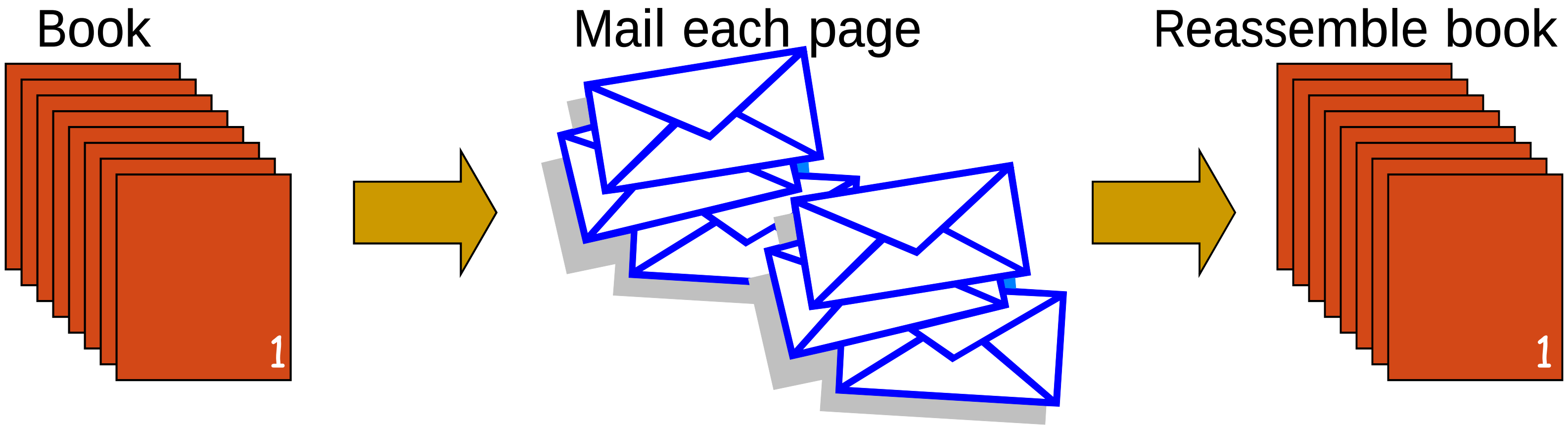
- Unreliable transport on top of IP:
  - No acknowledgment
  - No congestion control
  - No message continuation



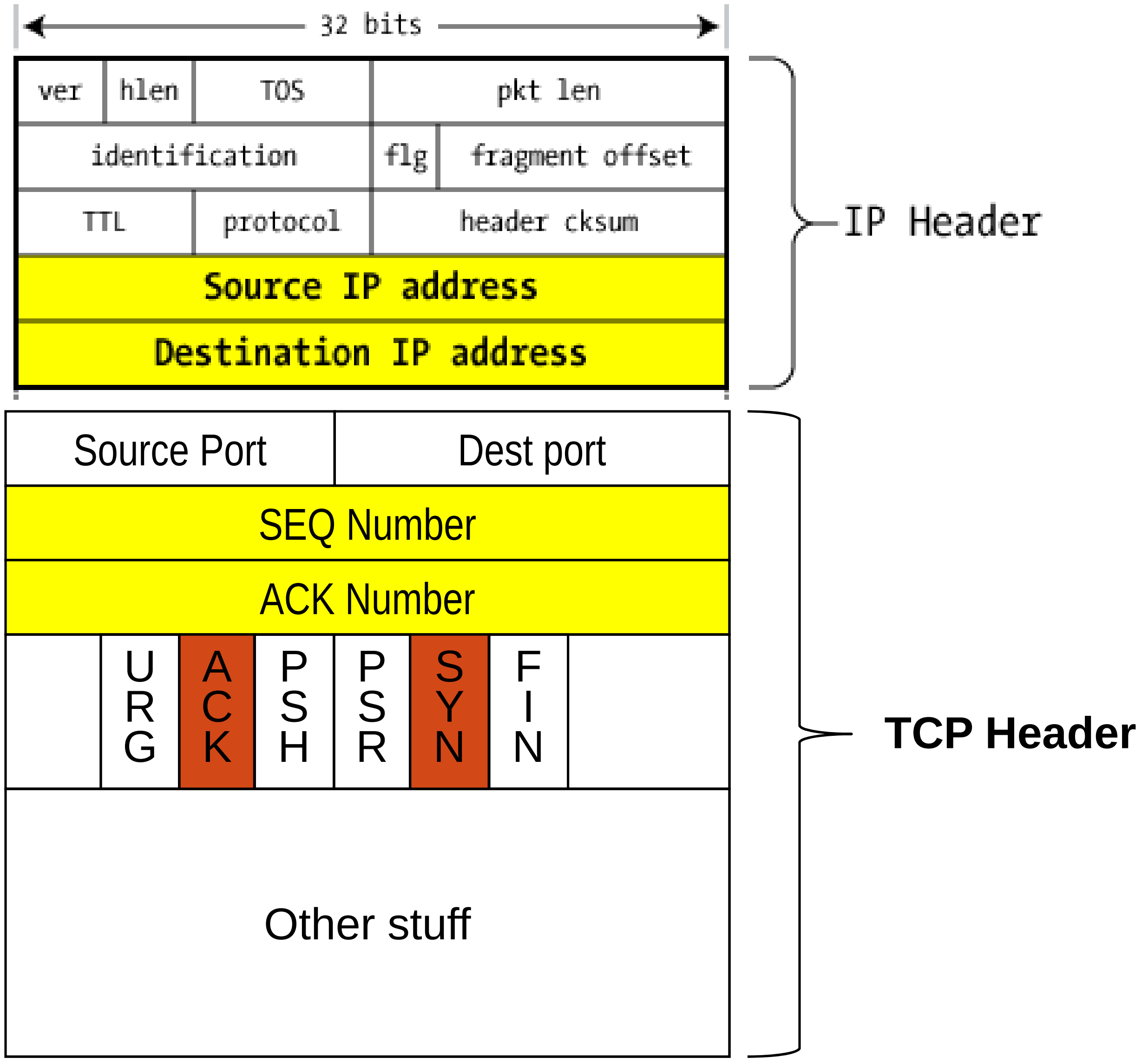
ver	hlen	TOS	pkt len
identification		flg	fragment offset
TTL		protocol	header cksum
Source IP address			
Destination IP address			
Source port		Destination port	
UDP length		UDP cksum	

Transmission Control Protocol (TCP)

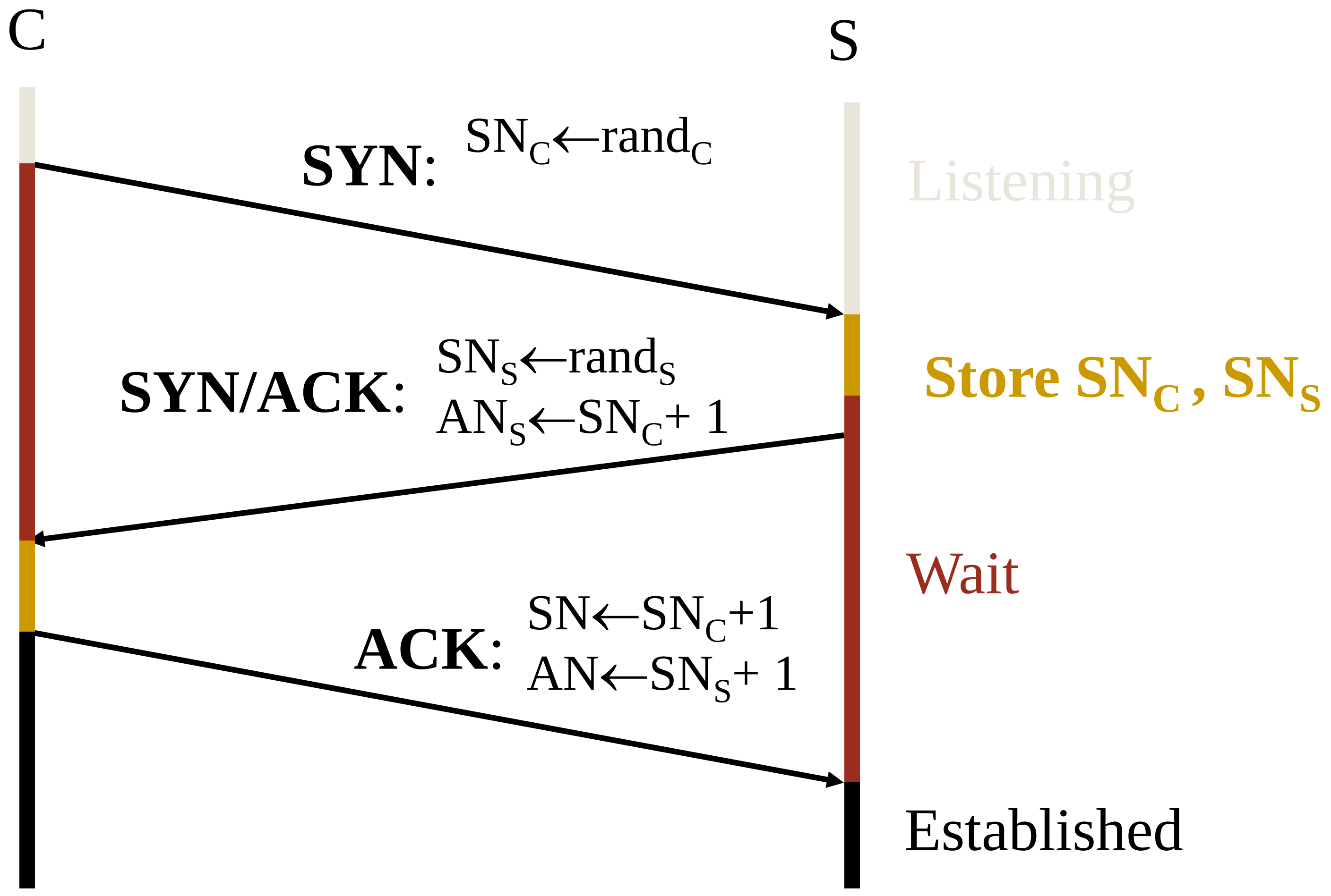
- Connection-oriented, preserves order
  - Sender
    - Break data into packets
    - Attach packet numbers
  - Receiver
    - Acknowledge receipt; lost packets are resent
    - Reassemble packets in correct order



TCP Header


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



C

S

SYN: $SN_C \leftarrow rand_C$

Listening

SYN/ACK: $SN_S \leftarrow rand_S$
$AN_S \leftarrow SN_C + 1$

Store SN_C , SN_S

Wait

ACK: $SN \leftarrow SN_C + 1$
$AN \leftarrow SN_S + 1$

Established

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

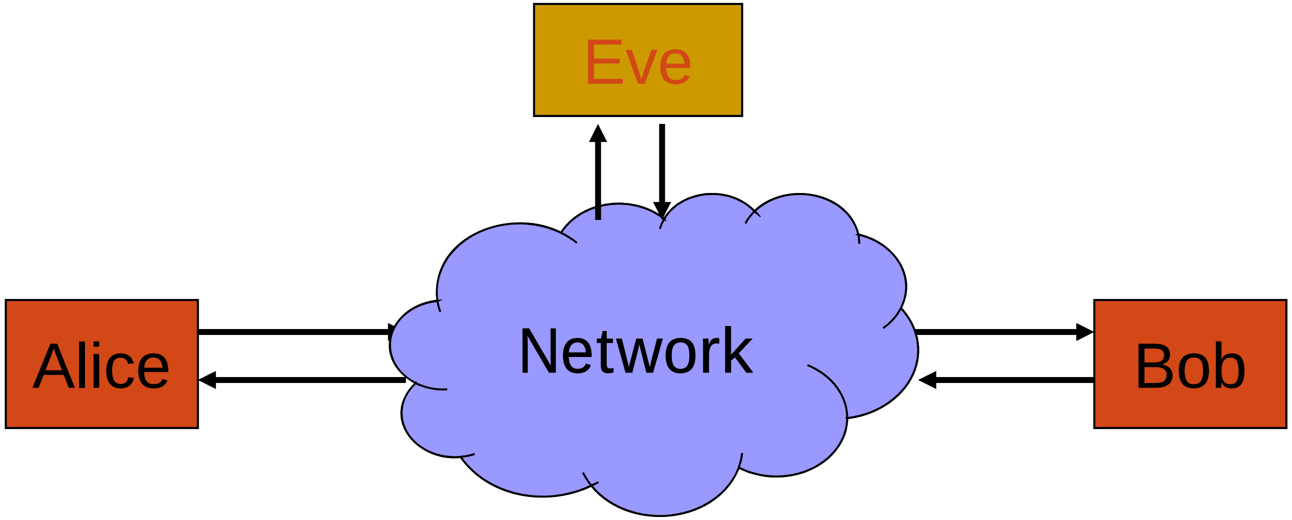
2. TCP state can be easy to guess
   - Enables spoofing and session hijacking

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) may send passwords in clear

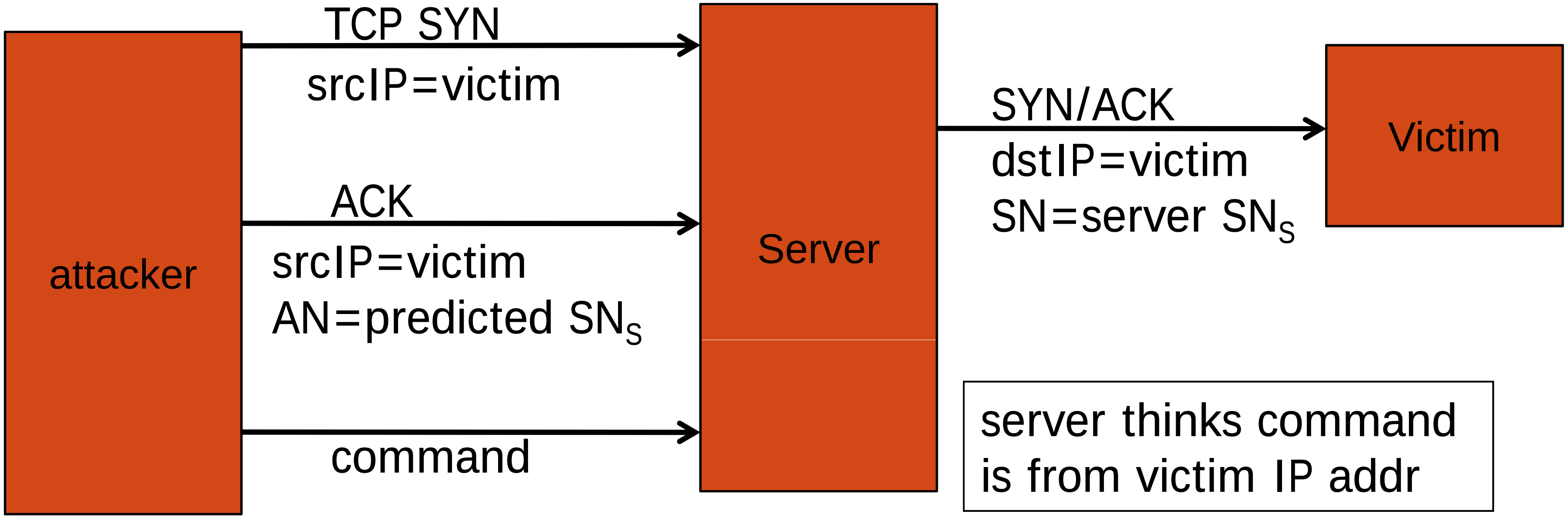


Prevention: Encryption (next lecture: IPSEC)

In **promiscuous mode** the network interface controller (NIC) causes the controller to pass all traffic it receives to the central processing unit (CPU) rather than passing only the frames that the controller is intended to receive.

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)



server thinks command is from victim IP addr

3. 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)

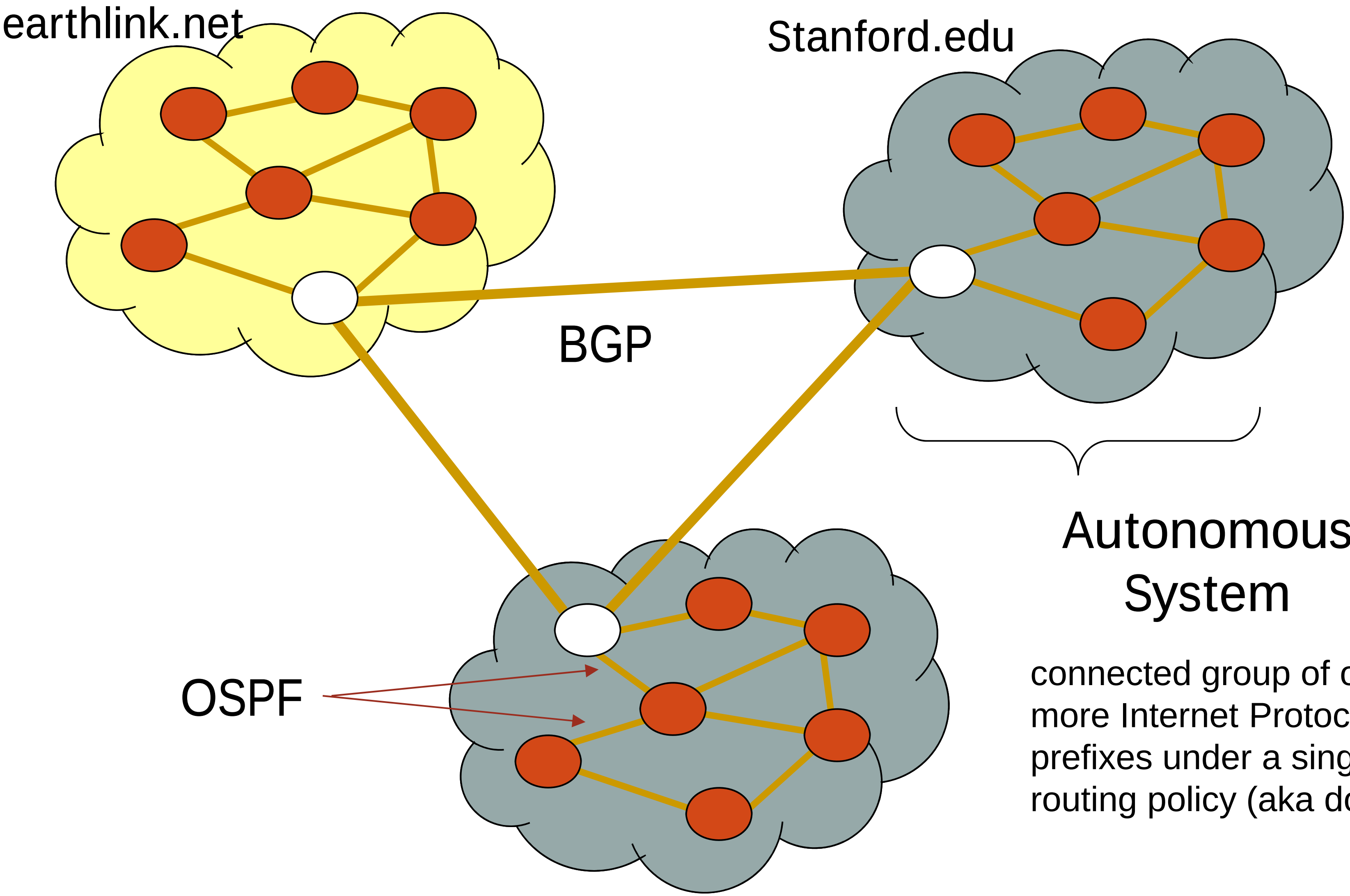
Routing Vulnerabilities

Routing Vulnerabilities

Routing protocols:

- ARP (addr resolution protocol): IP addr ⟶ eth addr
  - Node A can confuse gateway into sending it traffic for B (ARP Spoofing --- Because ARP does not provide methods for authenticating ARP replies on a network)
  - 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.
    - AS will advertise to its provider(s) and/or peer(s) that it can deliver any traffic destined for some other host/AS
  - Example: Youtube mishap (see DDoS lecture)

Interdomain Routing

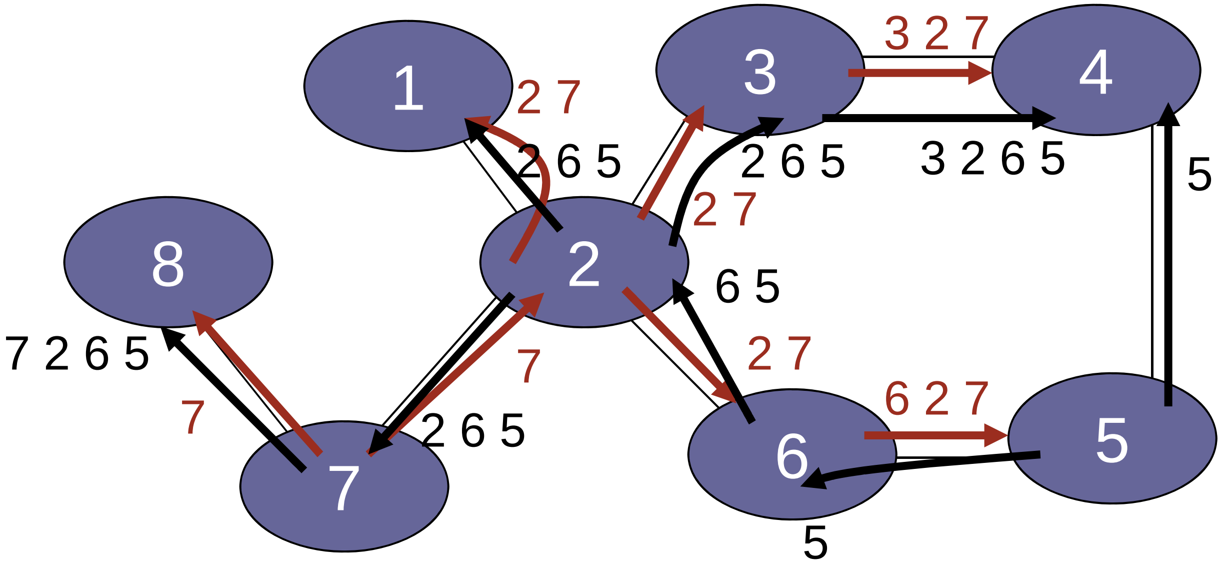


Autonomous System

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

BGP example

[D. Wetherall]



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

Security Issues

BGP packets are un-authenticated

- Attacker can inject advertisements for arbitrary routes
- Advertisement will propagate everywhere
- Used for DoS and spam
    - detailed example in DDoS lecture

Human error problems:

- Mistakes quickly propagate to the entire Internet

OSPF: routing inside an AS

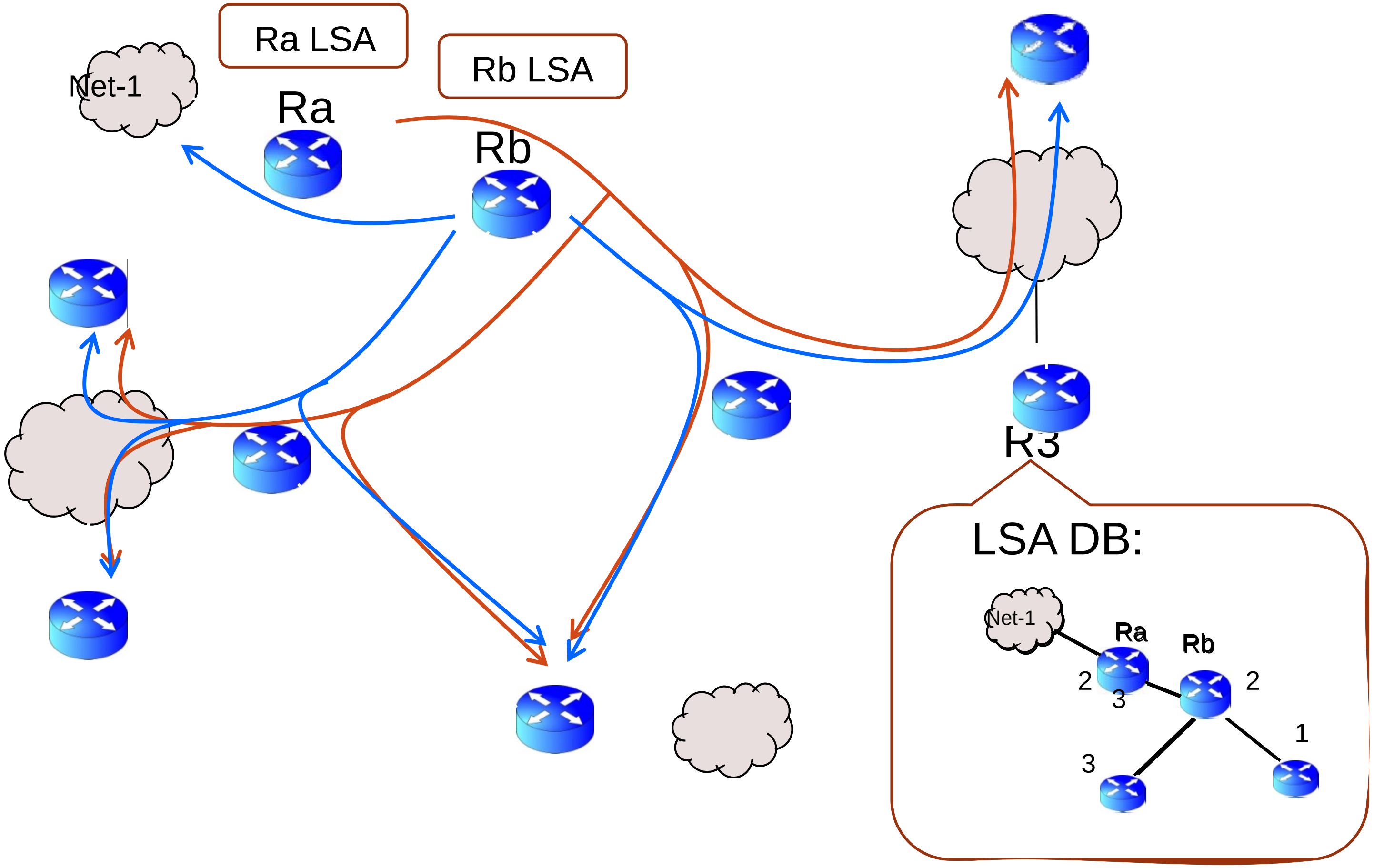
Link State Advertisements (LSA):

- Flooded throughout AS so that all routers in the AS have a complete view of the AS topology
- Transmission: IP datagrams, protocol = 89

Neighbor discovery:

- Routers dynamically discover direct neighbors on attached links --- sets up an “adjacently”
- Once setup, they exchange their LSA databases

Example: LSA from Ra and Rb



Security features

- OSPF message integrity (unlike BGP)
  - Every link can have its own shared secret
  - Unfortunately, OSPF uses an insecure MAC:

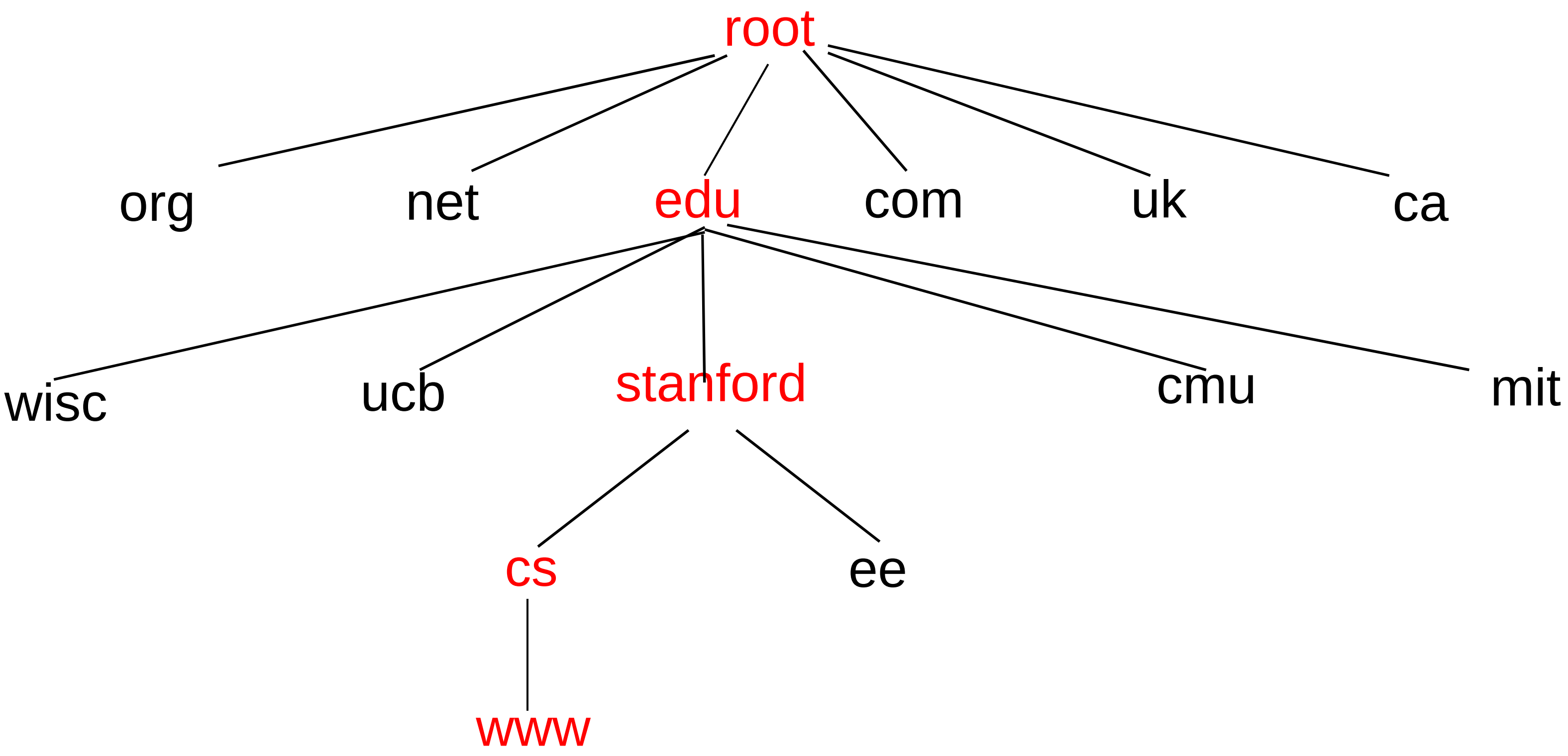
MAC(k,m) = MD5(data ll key ll pad ll len)

- Every LSA is flooded throughout the AS
  - If a single malicious router, valid LSAs may still reach dest.
- The "fight back" mechanism
  - If a router receives its own LSA with a newer timestamp than the latest it sent, it immediately floods a new LSA
- Links must be advertised by both ends

Domain Name Systems

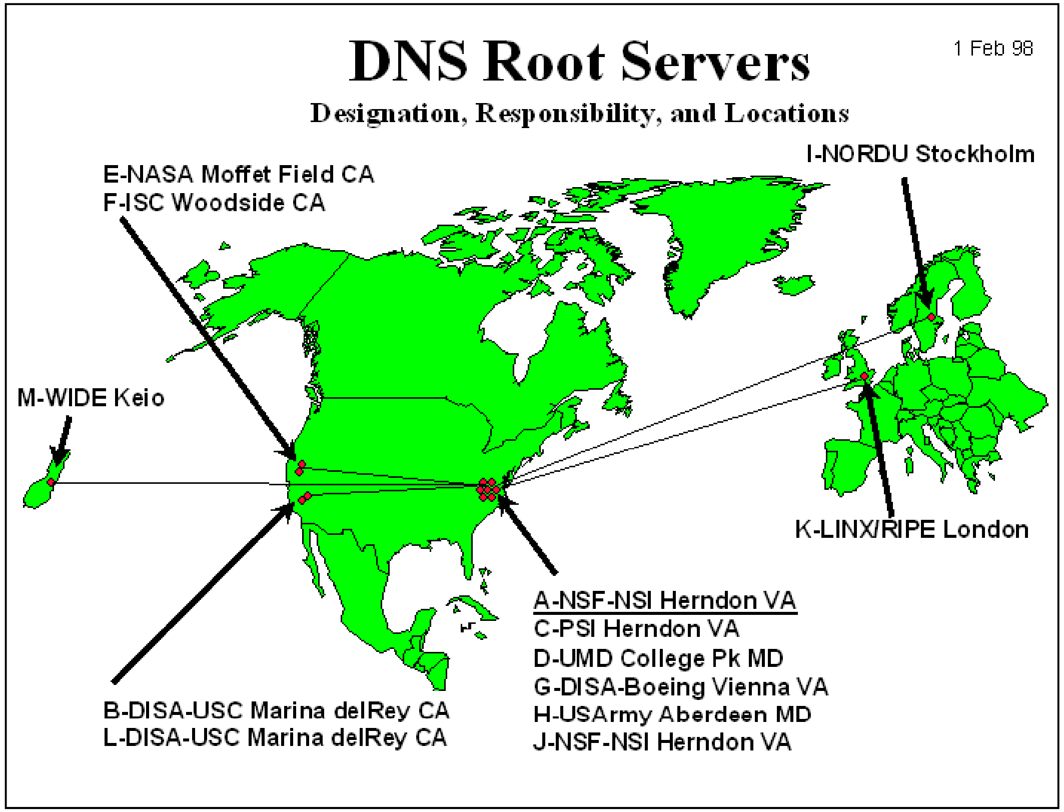
Domain Name System

- Hierarchical Name Space

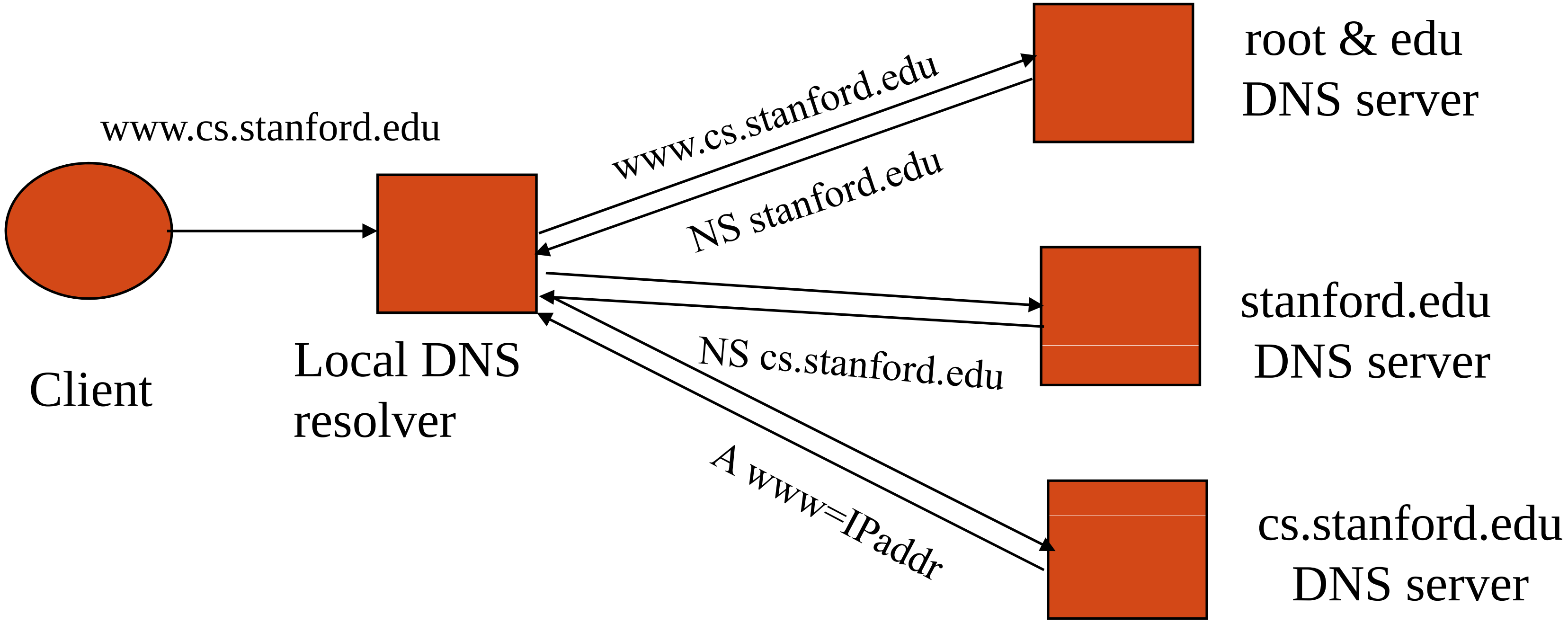


DNS Root Name Servers

- 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



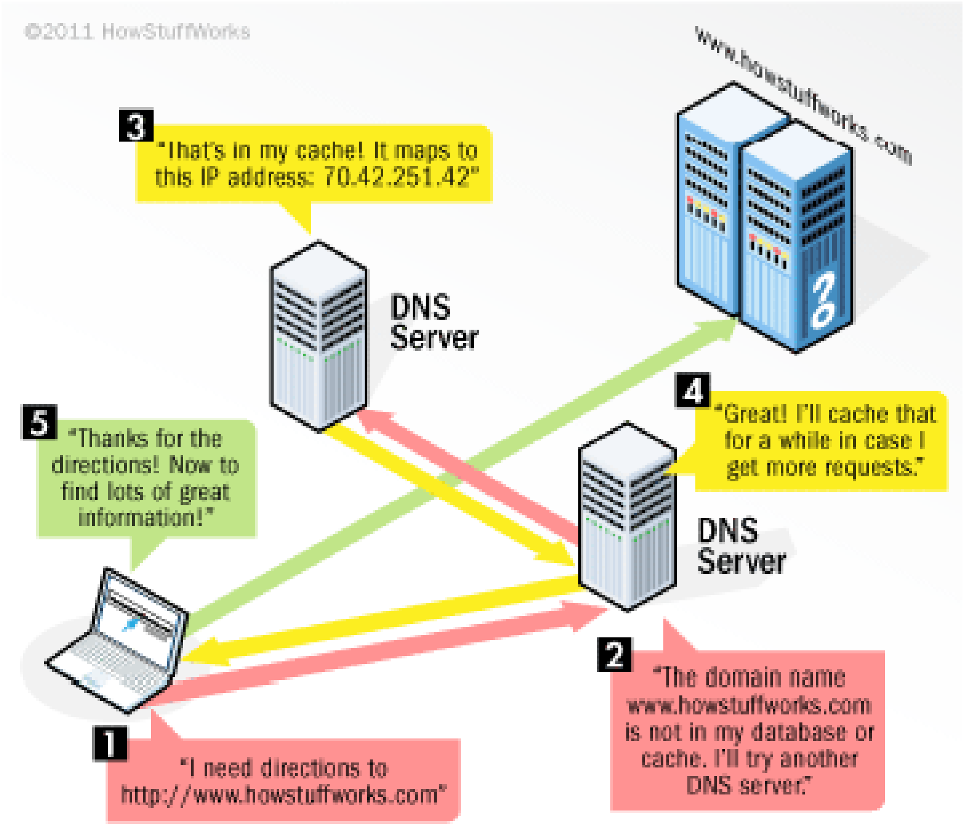
DNS Lookup Example



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))

DNS Lookup Example

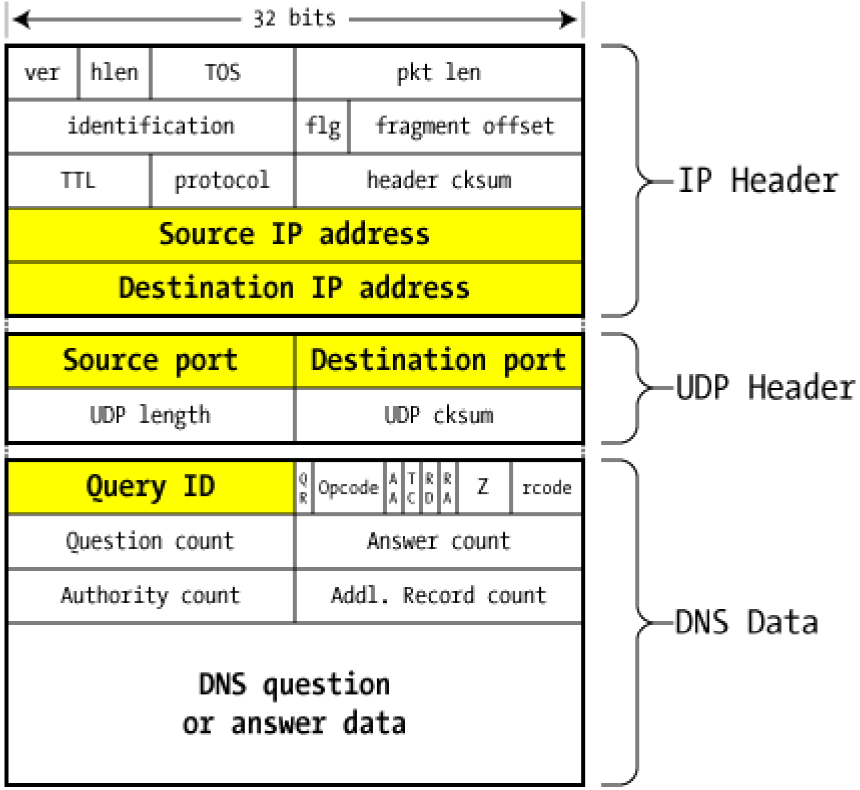

©2011 HowStuffWorks
www.howstuffworks.com
3 "That's in my cache! It maps to this IP address: 70.42.251.42"
DNS Server
5 "Thanks for the directions! Now to find lots of great information!"
4 "Great! I'll cache that for a while in case I get more requests."
DNS Server
1 "I need directions to http://www.howstuffworks.com"
2 "The domain name www.howstuffworks.com is not in my database or cache. I'll try another DNS server."

Caching

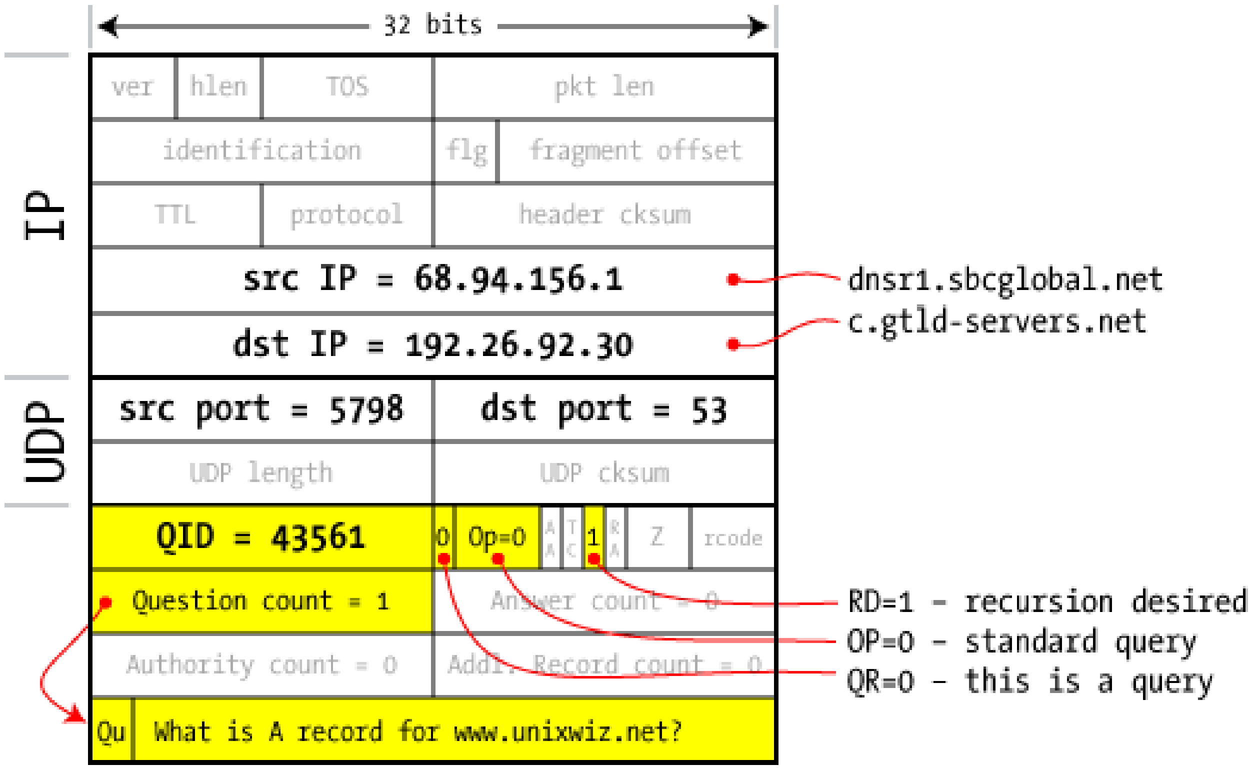
- DNS responses are cached
  - Quick response for repeated translations
  - Useful for finding servers as well as addresses
    - NS records for domains
- DNS negative queries are cached
  - Save time for nonexistent sites, e.g. misspelling
- Cached data periodically times out
  - Lifetime (TTL) of data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

DNS Packet

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



Resolver to NS request



32 bits

IP

ver	hlen	TOS	pkt len
identification		flg	fragment offset
TTL	protocol	header cksum	
src IP = 68.94.156.1			
dst IP = 192.26.92.30			

UDP

src port = 5798	dst port = 53
UDP length	UDP cksum

QID = 43561	0	Op=0	AA	TC	1	RA	Z	rcode
Question count = 1	Answer count = 0							
Authority count = 0	Addl. Record count = 0							
Qu	What is A record for www.unixwiz.net?							

dnsr1.sbcglobal.net

c.gtld-servers.net

RD=1 - recursion desired

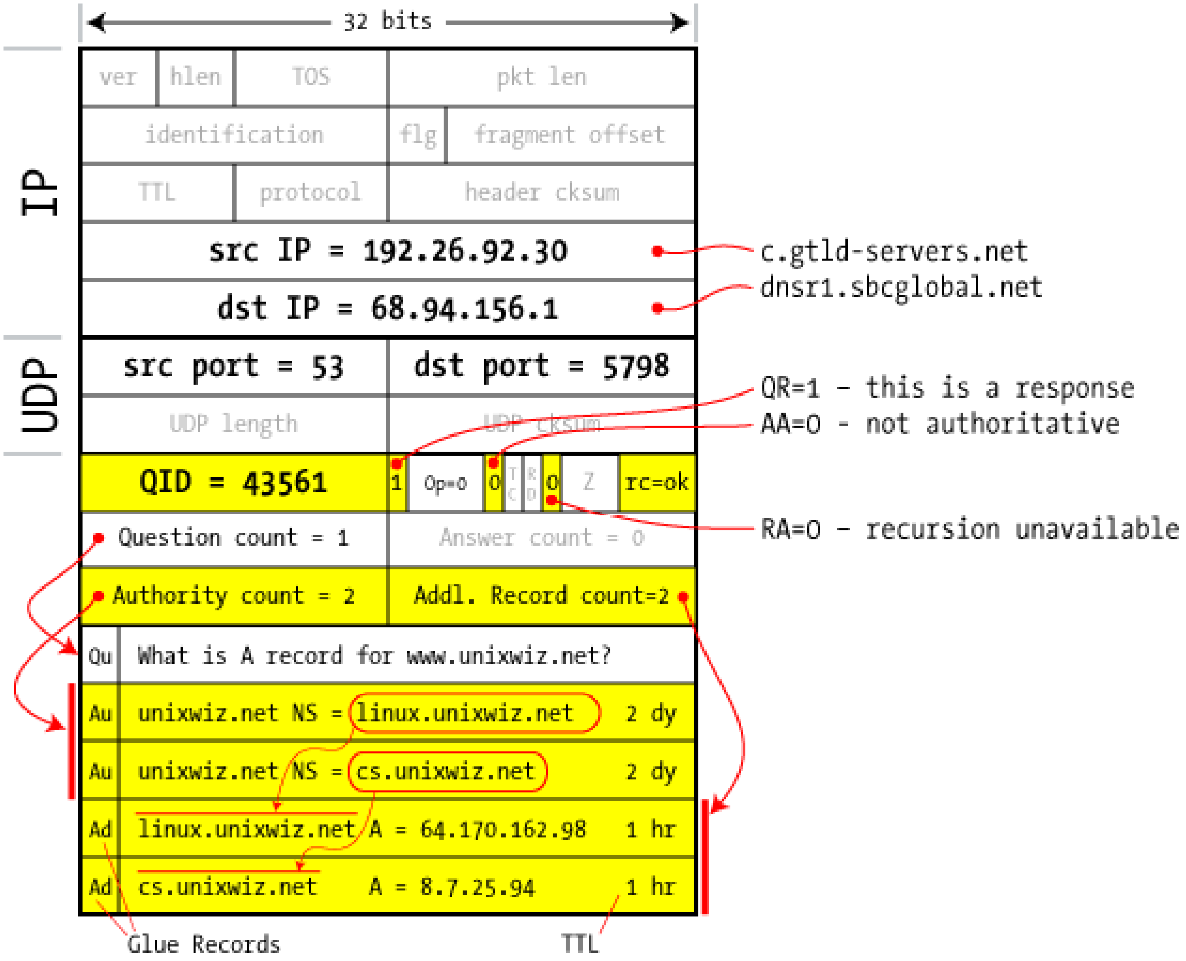
OP=0 - standard query

QR=0 - this is a query

Response to resolver

Response contains IP addr of next NS server (called "glue")

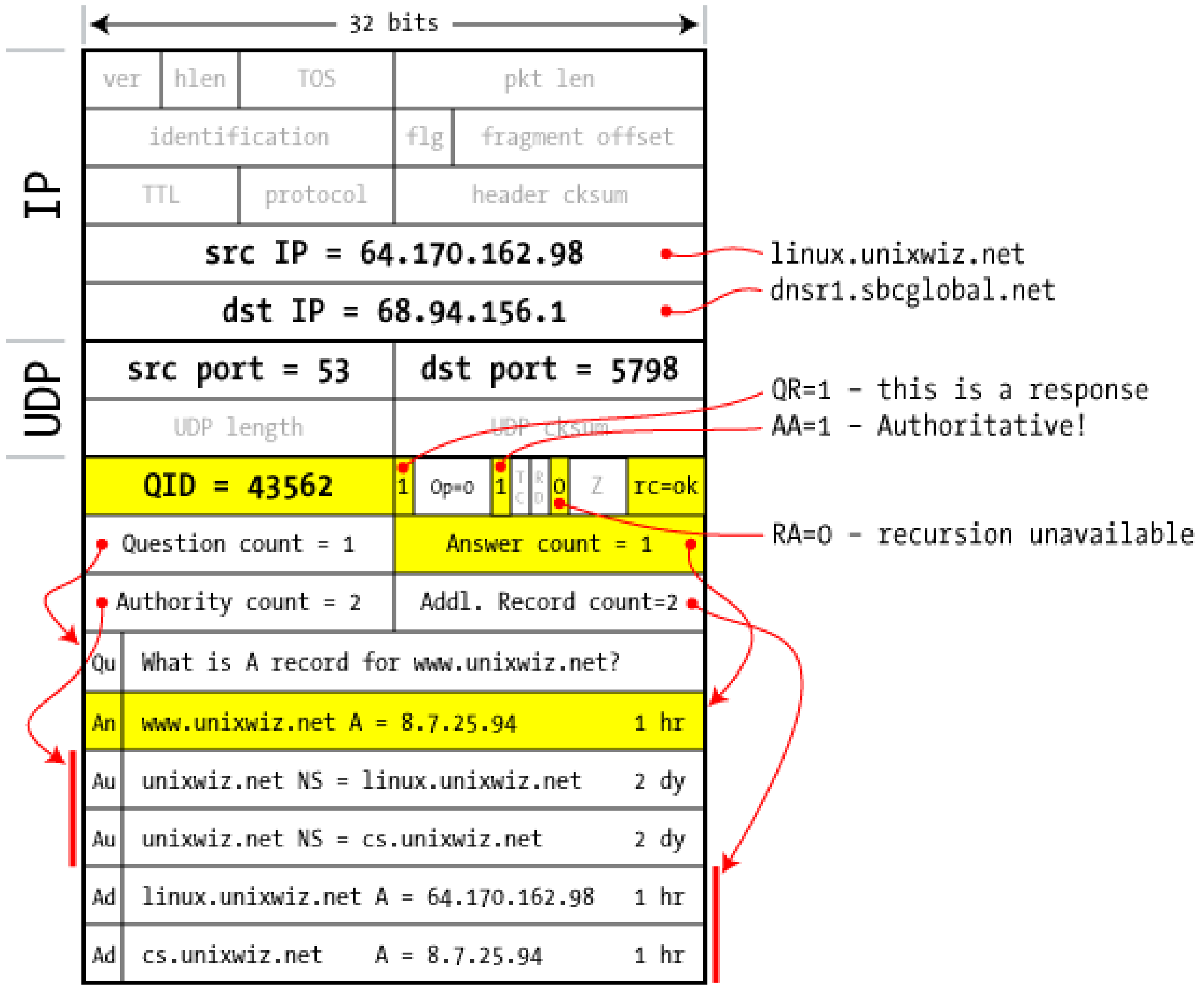
Response ignored if unrecognized QueryID



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 →



FAST-NUCES

Basic DNS Vulnerabilities

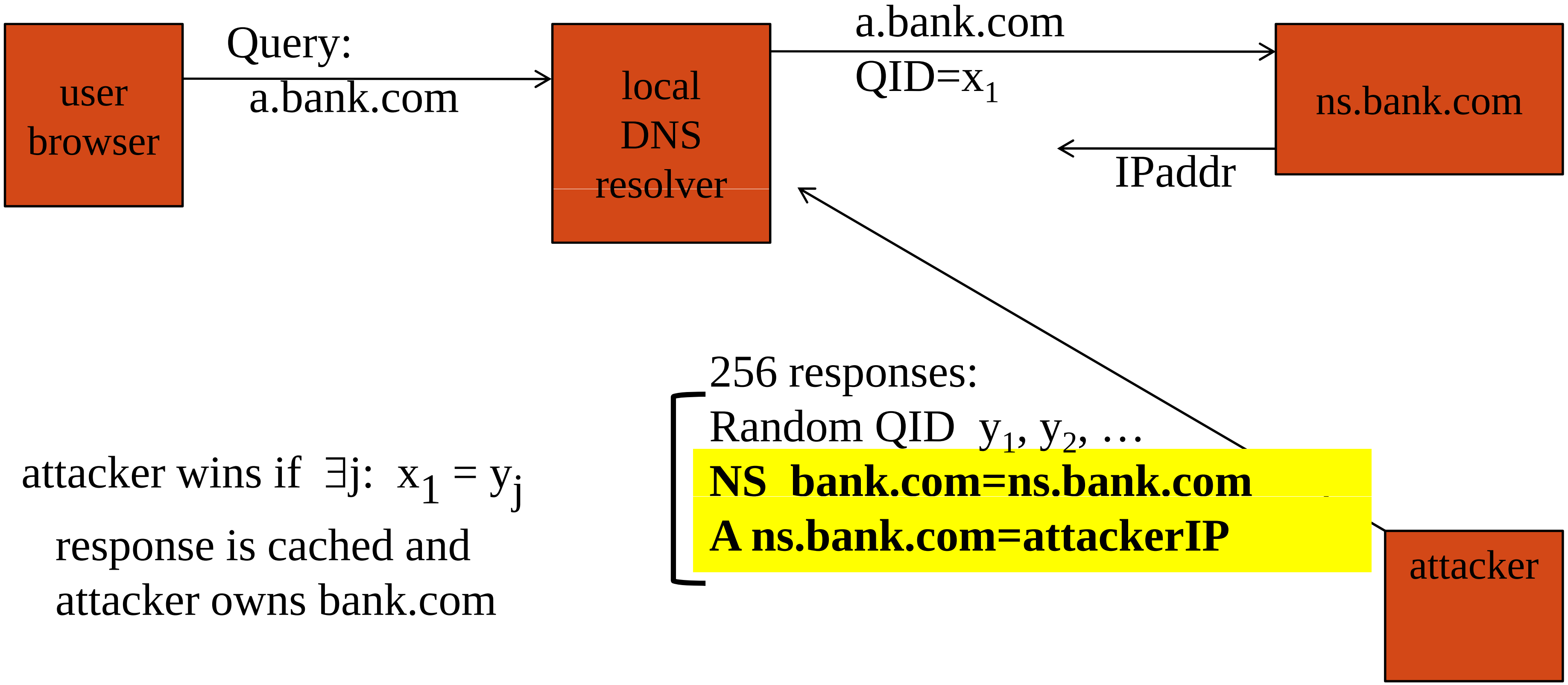
- Users/hosts trust the host-address mapping provided by DNS:
  - Used as basis for many security policies:

    Browser same origin policy (permits scripts running on pages originating from the same site – a combination of scheme, hostname, and port number)

- 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

DNS cache poisoning (a la Kaminsky'08)

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



user browser

Query: a.bank.com

local DNS resolver

a.bank.com QID=x_1

ns.bank.com

IPaddr

256 responses:
Random QID y_1, y_2, …
NS bank.com=ns.bank.com
A ns.bank.com=attackerIP

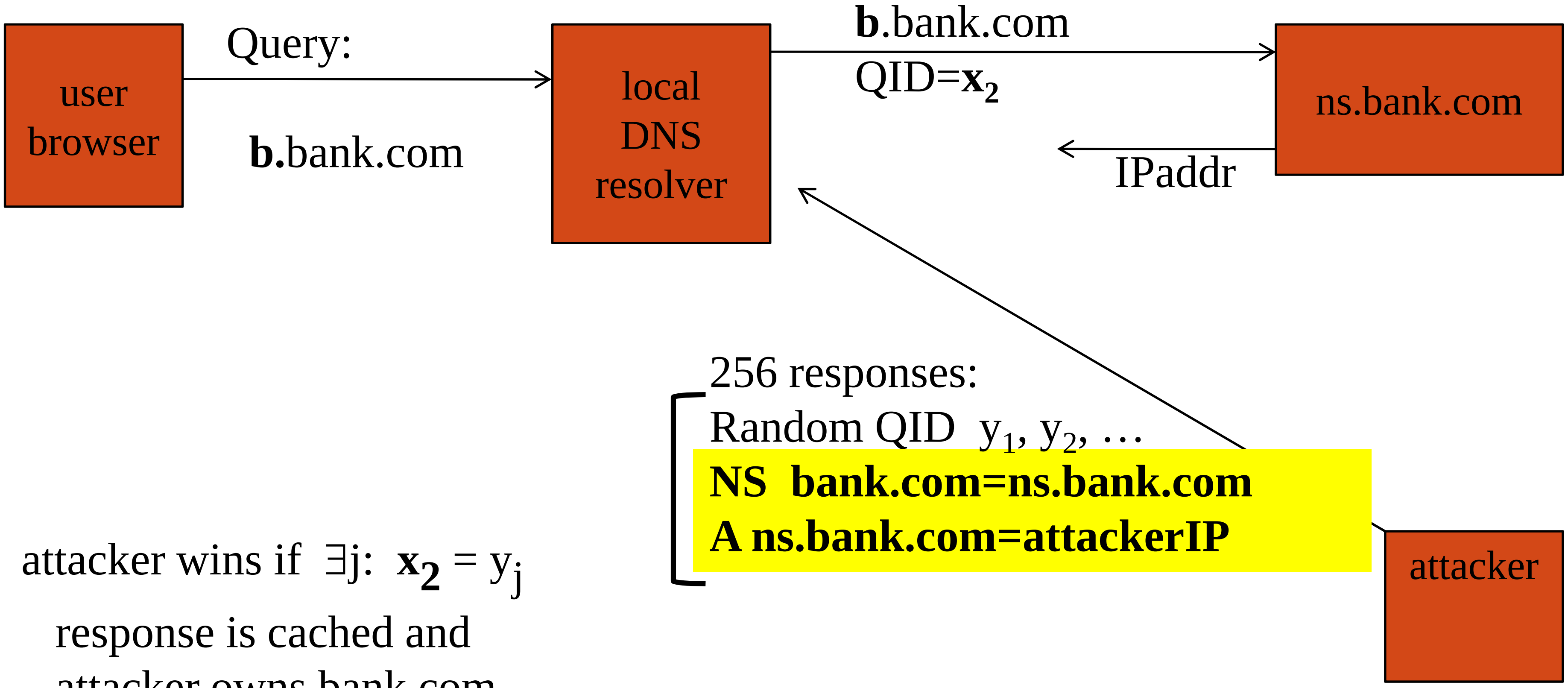
attacker

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



attacker wins if $\exists j$: $\mathbf{x_2} = y_j$

response is cached and attacker owns bank.com

success after ≈ 256 tries (few minutes)

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

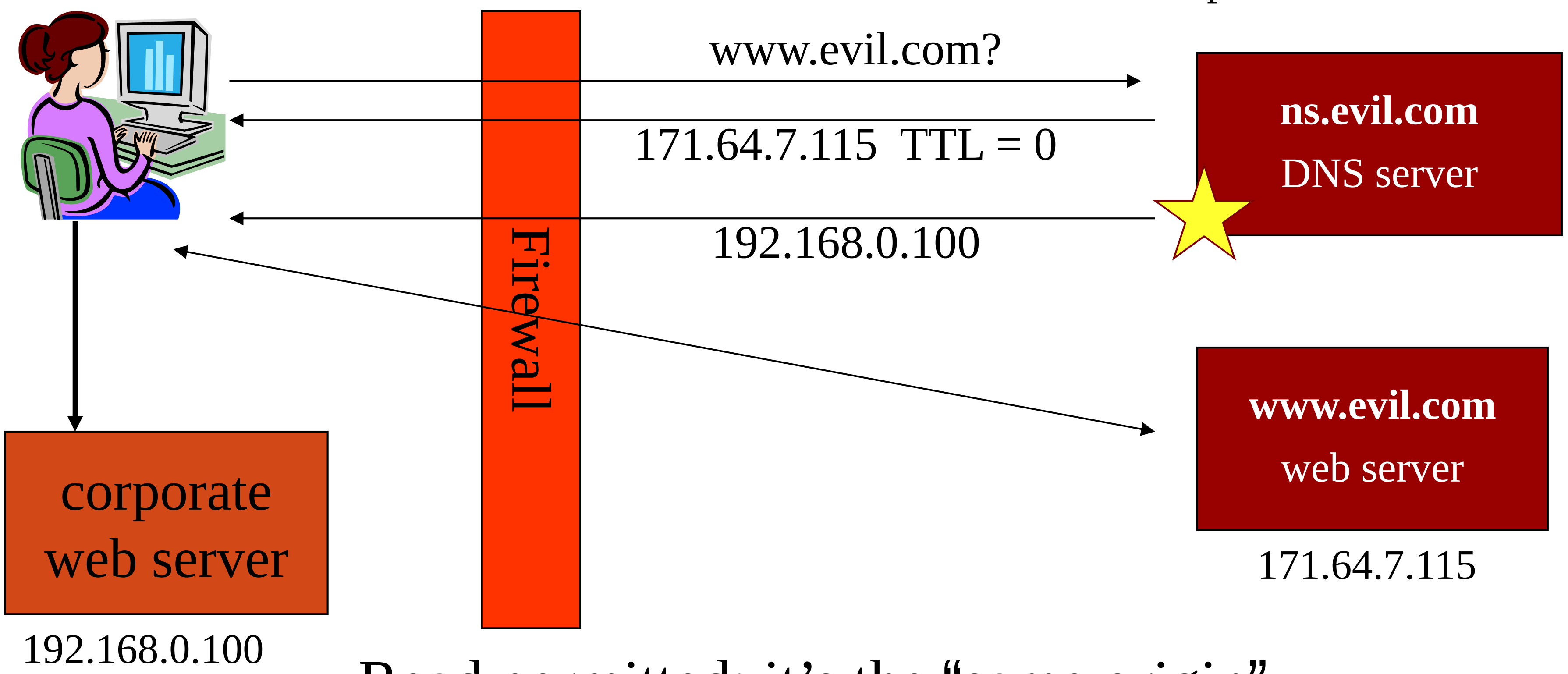
DNS poisoning attacks in the wild

- 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"

DNS Rebinding Attack

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

DNS-SEC cannot
stop this attack



www.evil.com?

ns.evil.com
DNS server

171.64.7.115 TTL = 0

192.168.0.100

Firewall

www.evil.com
web server

171.64.7.115

corporate
web server

192.168.0.100

Read permitted: it's the "same origin"

[DWF'96, R'01]

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

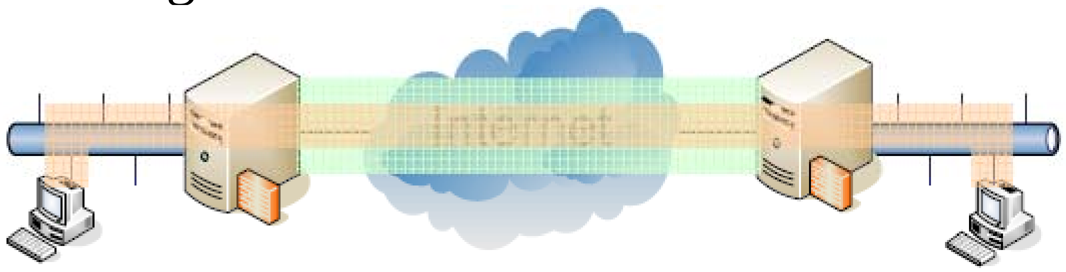
IPSEC

IETF IPSec Working Group

- The basic Internet Protocol (IP) has absolutely no security concepts integrated
  - Data are transferred in plain text
  - Authentication is not possible, everybody can fake the IP sender address or modify the payload of a packet
- Thus: specification of a security architecture for IP that comprises authentication and encryption mechanism
  - Security mechanisms should be algorithm-independent, cryptographic algorithms can be altered without effecting other parts of the protocol
  - Wide variety of security policies should be supported
- Result:
  - Compatibility with the Internet structure as given by IPv4 was needed to introduce security functions earlier than the “rest” of IPv6
  - Thus: in 1992 the *IP Security Protocol (IPSec) was standardized, together with theInternet Key Exchange (IKE)*

IP Security Protocol (IPSec)

- Security design goal: authentication and encryption
- Authentication Header (AH): provides sender authentication and data integrity
- Encapsulation Security Payload (ESP): provides data encryption
- Both mechanism are based on the concept of a *security association (SA)* and may be used together or separately
- Work modes:
  - *Transport mode*
    - Build a point-to-point connection between two hosts
  - *Tunnel mode*
  - Connect two “security gateways”, i.e. whole networks. Security gateways can be integrated with the access routers of the networks

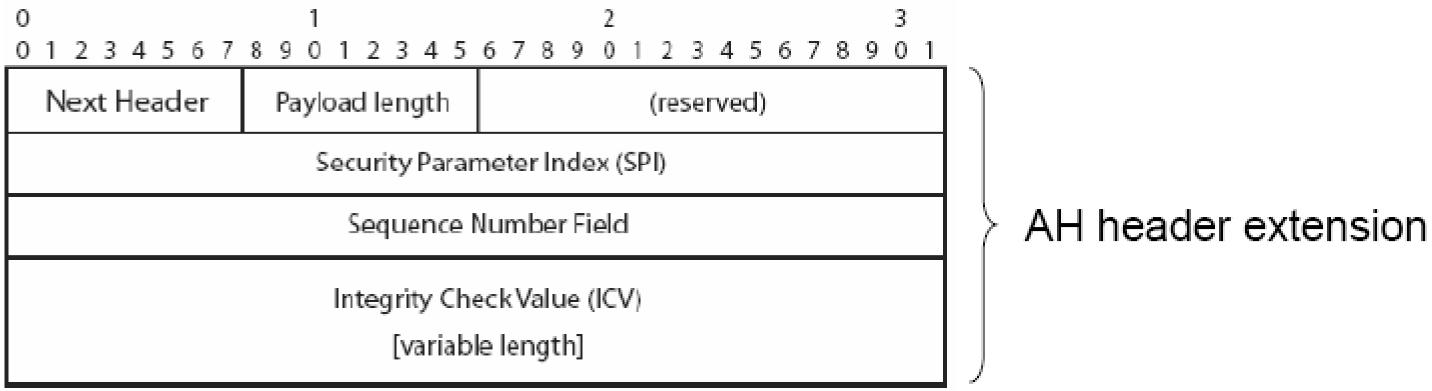


Security Associations (SA)

- A *security association* is an agreement between two or more parties regarding security services that they want to use and how they are going to provide them
- This agreement is established through a common set of security related parameters like
  - Authentication algorithm, mode and keys for the AH
  - Encryption algorithm, mode, and keys for the ESP
  - An initialization vector (IV), if necessary
  - Lifetime of the keys and the SA as a whole
  - Source address
  - Security level of the protected data, such as confidential, secret or unclassified
- When an IP packet is received, it can only be authenticated and/or decrypted if the receiver can link it with an appropriate SA. Hence the IP packet must convey a reference that points to the SA on the receiver's side
- In IPSec this reference is called a *security parameter index (SPI)*
- Each SA is uniquely identified by a destination address and a SPI value negotiated before communication by the *Internet Key Exchange (IKE) protocol*

Authentication Header (AH)

- AH provides *sender authentication* and *data integrity* for IP packets by enhancing the IP header with some additional fields:



- Next Header: type of the following header after the AH (as the protocol field in IPv4)
- Payload Length: length of the integrity check value (ICV) in 32-bit words
- Reserved: for future use, currently set to zero
- SPI: identifies the SA for the IP packet on receiver side
- Sequence number: assign each packet an unique identifier to protect against replay attacks
- ICV: the assigned authentication/integrity value

Integrity Check Value

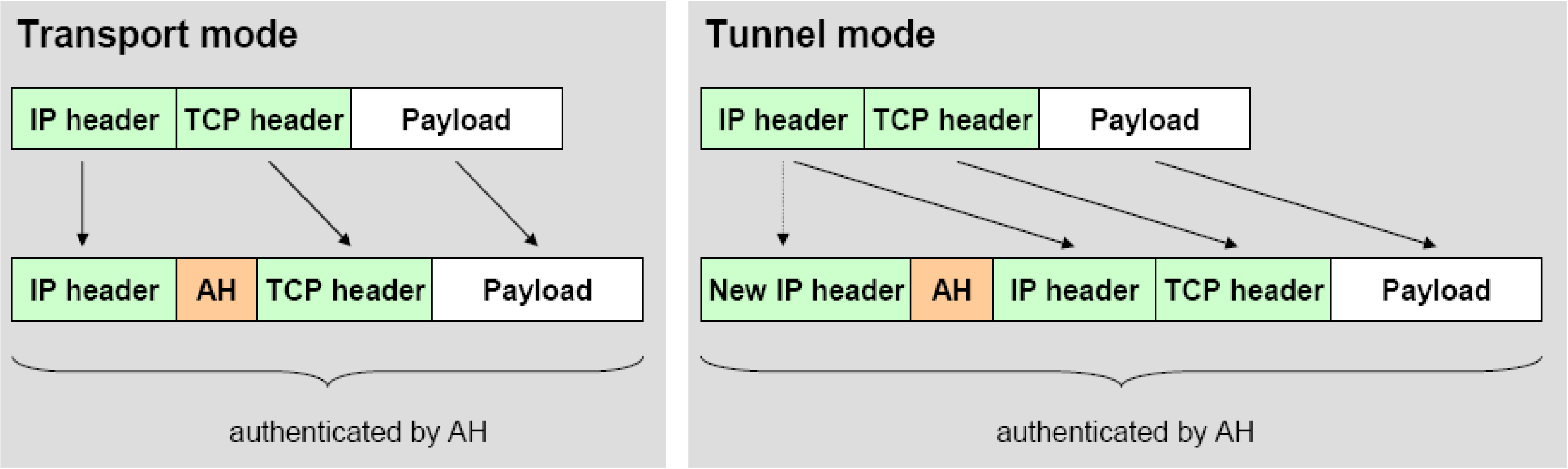
- The authentication data mostly is computed by using a cryptographic authentication algorithm and a corresponding secret key
  - Use a one-way hash function, MD5 is default in IPSec
  - The ICV in case of MD 5 is a 128-bit hash value from the concatenation of the key, the message, and the key again
  - Alternative: use digital signatures from public key cryptography
  - In general, the authentication algorithm is negotiated as part of the SA
  - The receiver uses the same algorithm to test the ICV

- Problem: some fields of the IP packet header change in transit, e.g. the TTL field in IPv4
  - The ICV is computed for a whole IP packet
  - With another TTL, the receiver comes to another ICV as the sender
  - Thus, the receiver has to set such information like TTL back to the initial value before computing and testing the ICV value

Note: AH does not encrypt the payload!

Use of the Authentication Header

AH can be used in two ways:

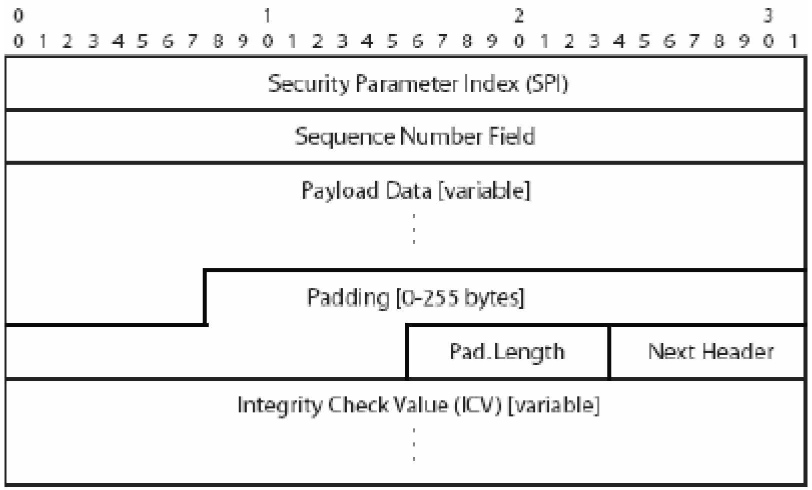


In general, AH has the following characteristics

- Advantage: AH mechanism does not significantly increase implementation costs
- Disadvantage: AH increases IP processing costs and communication latency in participating systems

Encapsulating Security Payload (ESP)

ESP provides data confidentiality by using encryption and encapsulation and adding an ESP header/trailer to an IP packet:



Security Parameter Index (SPI)

Sequence Number Field

Payload Data [variable]

Padding [0-255 bytes]

Pad.Length

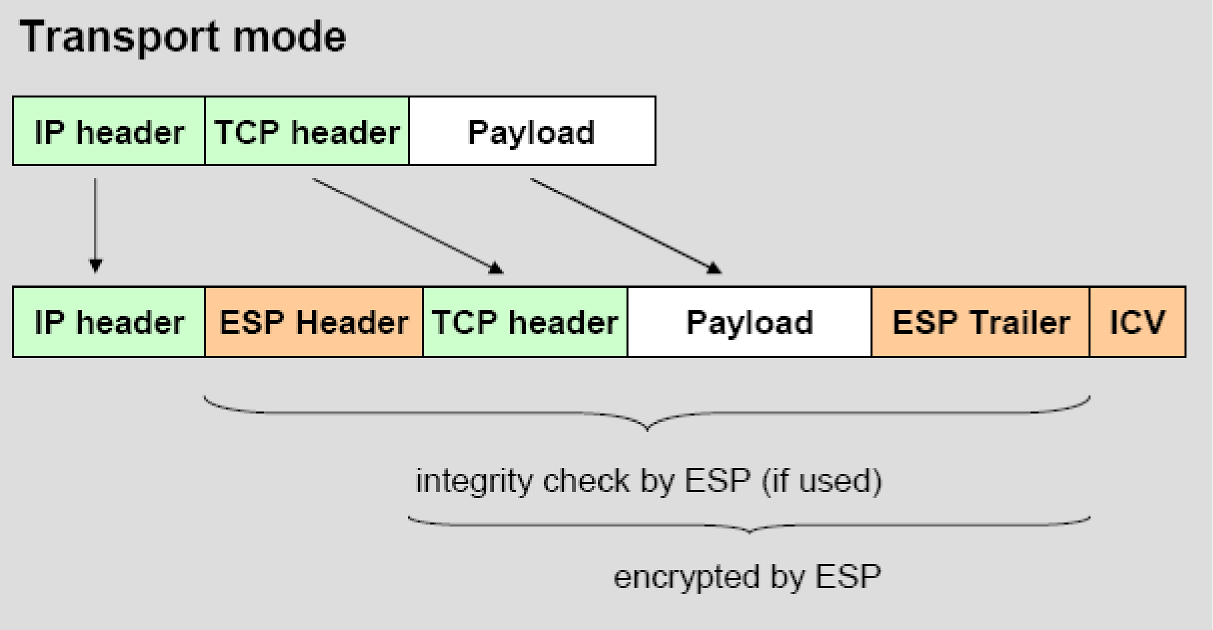
Next Header

Integrity Check Value (ICV) [variable]

- SPI, Sequence Number, and ICV (only if integrity is also checked): as in AH
- Payload data: encrypted with an algorithm defined in the SA (default: DES in CBC mode)
- Padding: filled with random bits
- Padding Length: indicates the total length of the Padding field
- Next header: identifies the following header

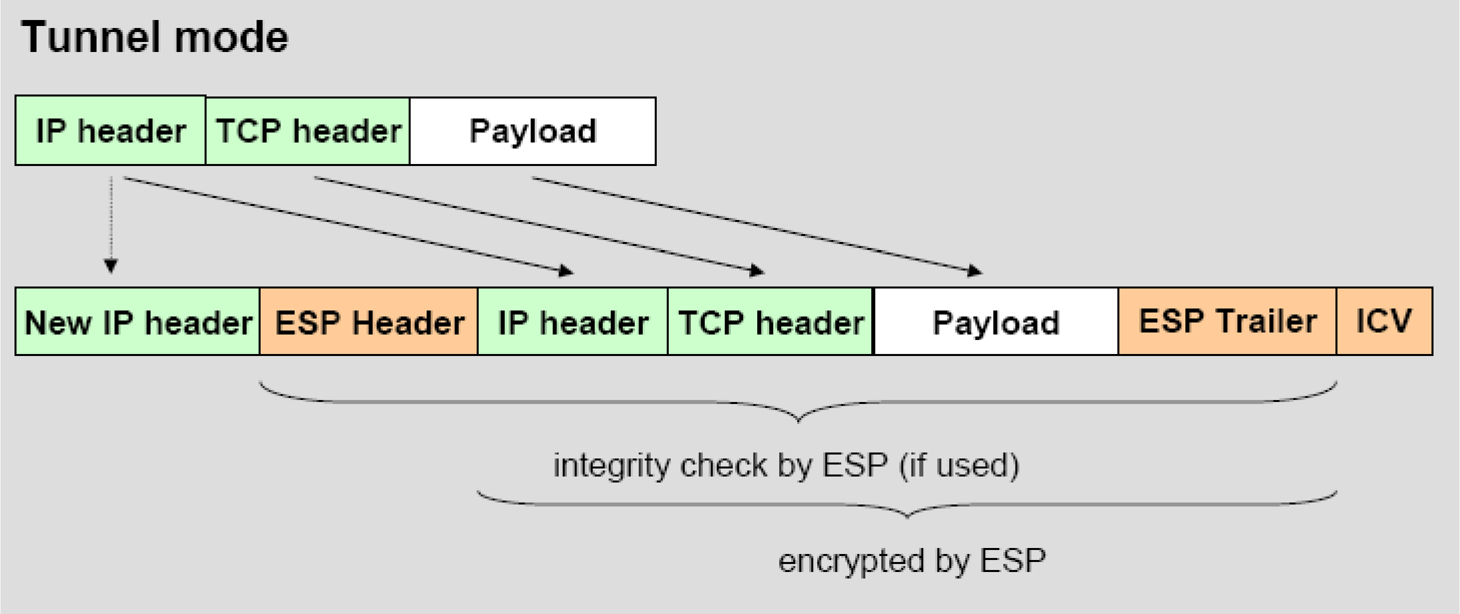
Use of the Encapsulating Security Payload

ESP can be used in two ways, as AH:



- The receiver processes the IP header and plaintext part of the ESP to obtain the SPI value
- This value is used as an index for a local SPI table to find the negotiated SA parameters and cryptographic keys to decrypt the rest of the packet

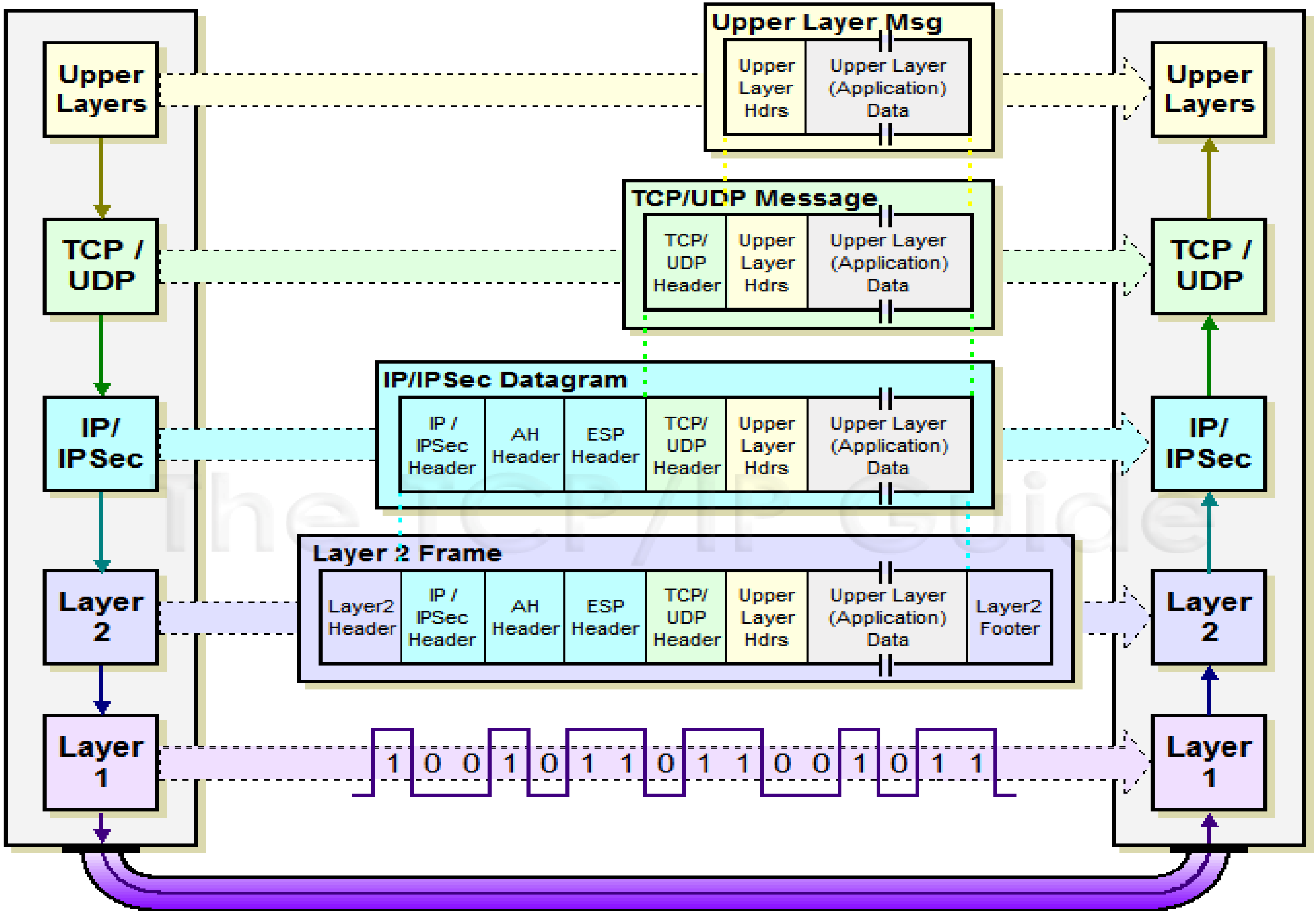
Use of the Encapsulating Security Payload



- In tunnel mode, an encrypted tunnel between two security gateways is installed
- Used e.g. between two LANs which should communicate in a secure manner (as in VPN, Virtual private Network)

Note: AH and ESP mechanism have been designed independently and can be applied separately or together

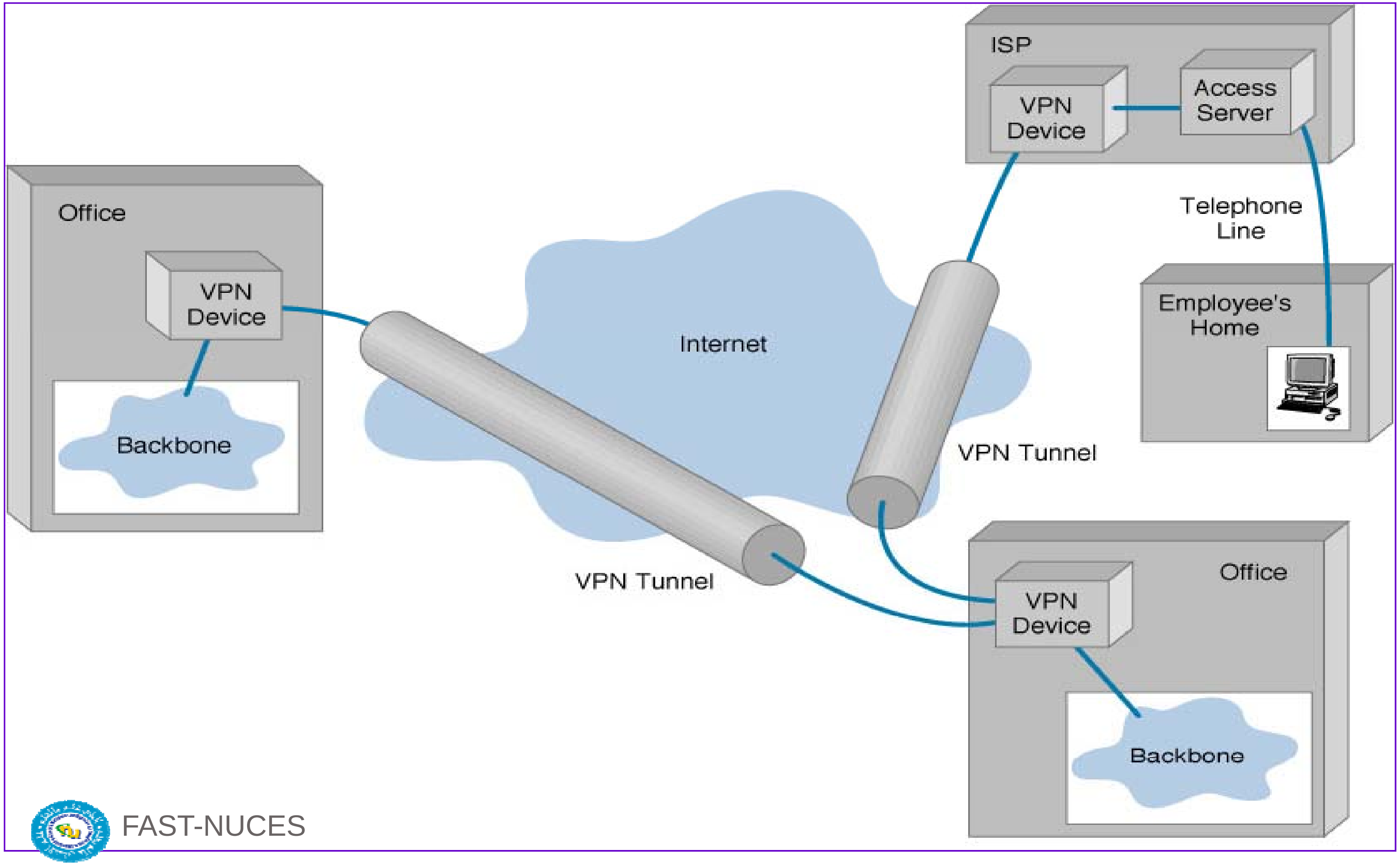
IPSec Transport Mode: IPSEC instead of IP header



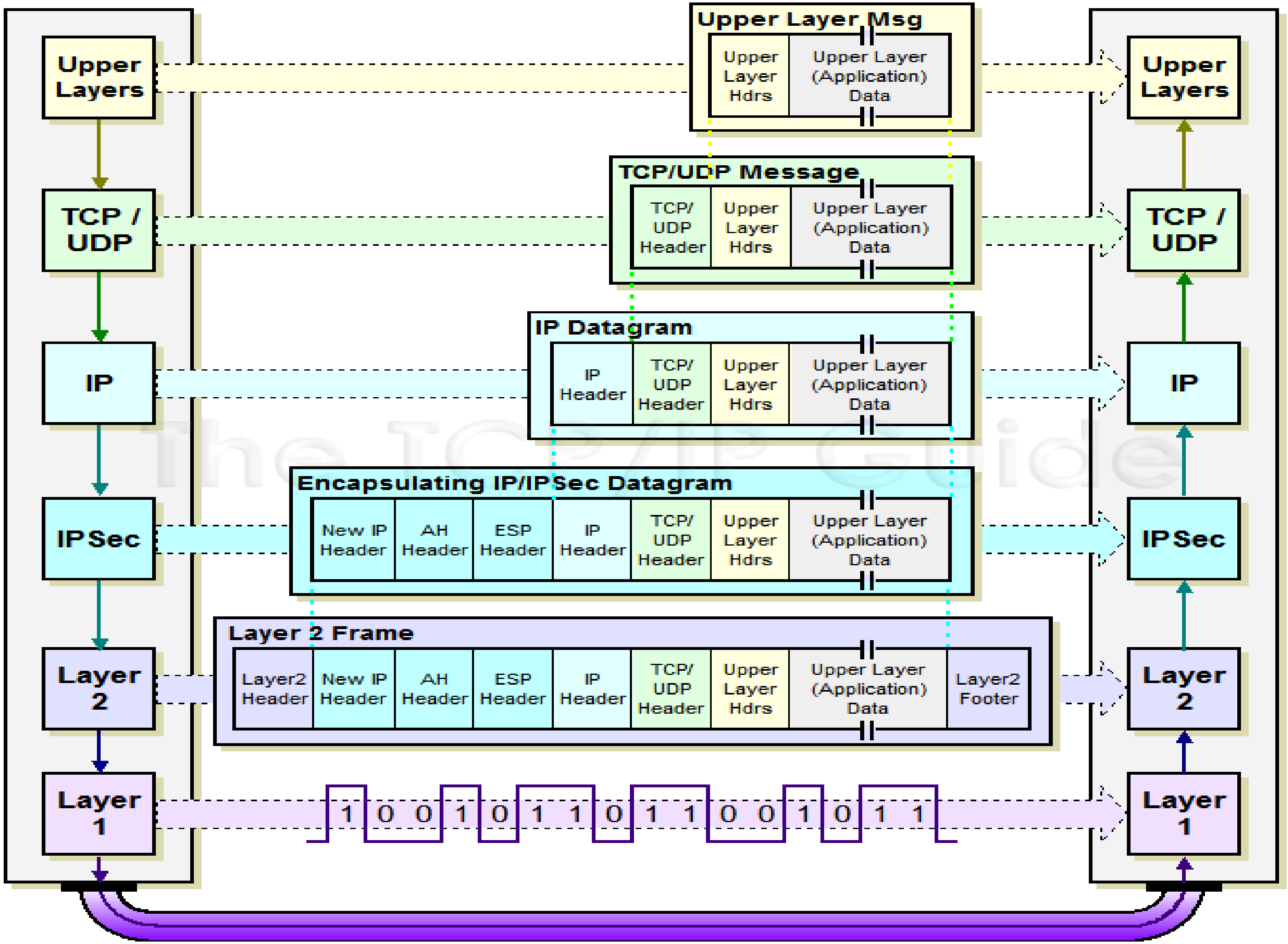
FAST-NUCES

http://www.tcpipguide.com/free/t_IPSecModesTransportandTunnel.htm

IPSEC Tunnel Mode



IPSec Tunnel Mode: IPSEC header + IP header



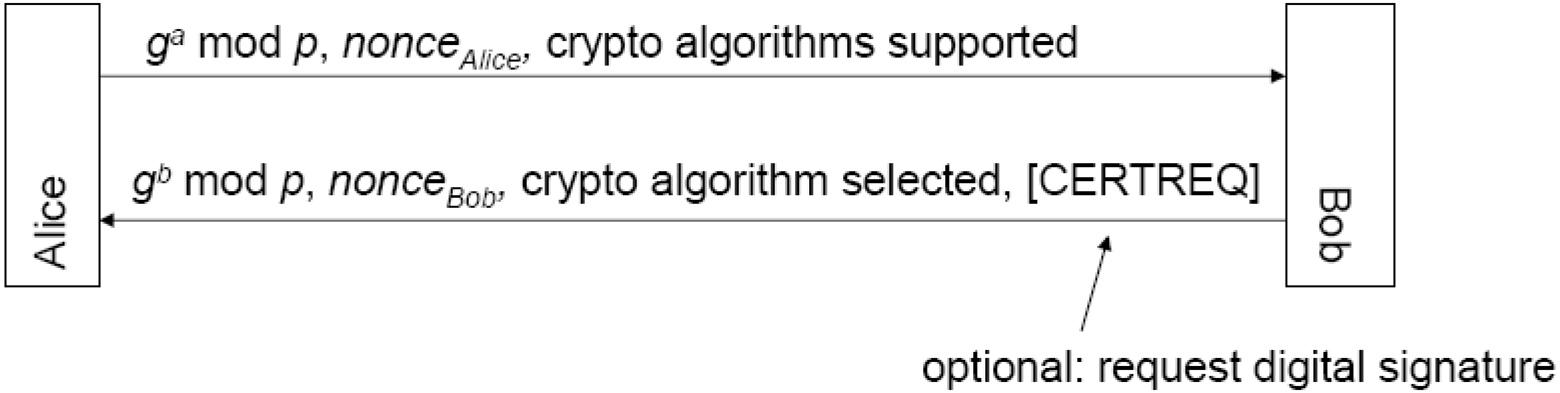
FAST-NUCES

Internet Key Exchange (IKE)

- Before applying AH and/or ESP, a SA has to be established – this is done using the
- Internet Key Exchange (IKEv2) in two phases:
  - Phase 1: negotiate a SA in two steps
    - Purpose: mutual authentication, establish secret keys for phase 2
  - Phase 2: create multiple SAs used for one communication each
    - Purpose: from the results of phase 1, several SAs can be generated, which gives a speedup (if several SAs are needed)

IKE Phases

- First half of phase 1: establish a common secret using Diffie-Hellman key exchange:

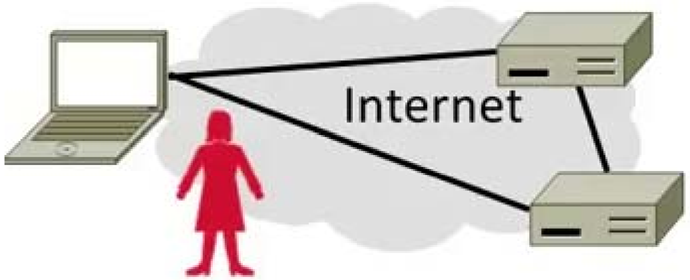

Alice
g^a mod p, $nonce_{Alice}$, crypto algorithms supported
g^b mod p, $nonce_{Bob}$, crypto algorithm selected, [CERTREQ]
Bob
optional: request digital signature

- Second half of phase 1: mutual authentication
  - Using the key negotiated by Diffie-Hellman, now encrypted mutual authentication is done – optionally by using certificates, if requested before
  - In the authentication messages, also IP addresses and TCP ports of communication partners are included
- IKE phase 2: applied after mutual authentication.
  - Use negotiated key to propose/accept SAs – maybe also by doing a new key exchange with every SA proposal to generate new session keys

Virtual Private Networks (VPN)

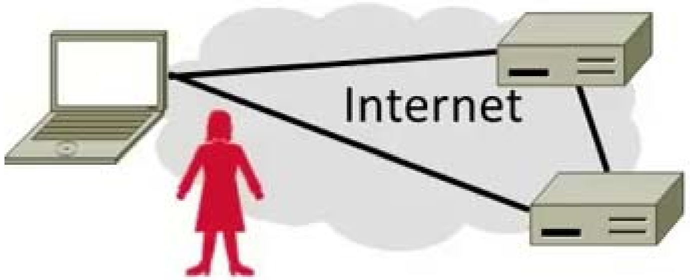
Topic

- Virtual Private Networks (VPNs)
  - Run as closed (private) networks on Internet
  - Use IPSEC to secure messages



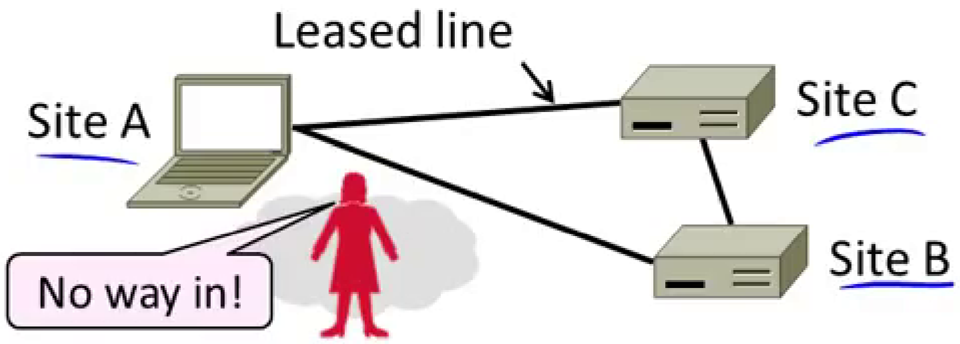
Motivation

- The best part of IP connectivity
  - You can send to any other host
- The worst part of IP connectivity
  - Any host can send packets to you!
  - There's nasty stuff out there …



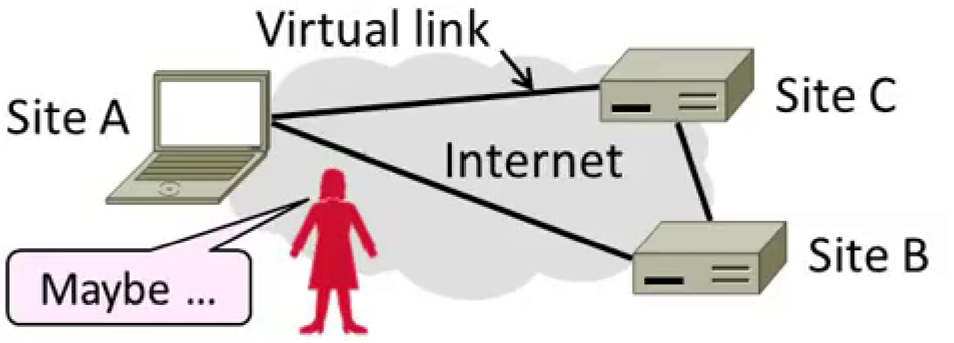
Motivation (2)

- Often desirable to separate network from the Internet, e.g., a company
  - Private network with leased lines
  - Physically separated from Internet



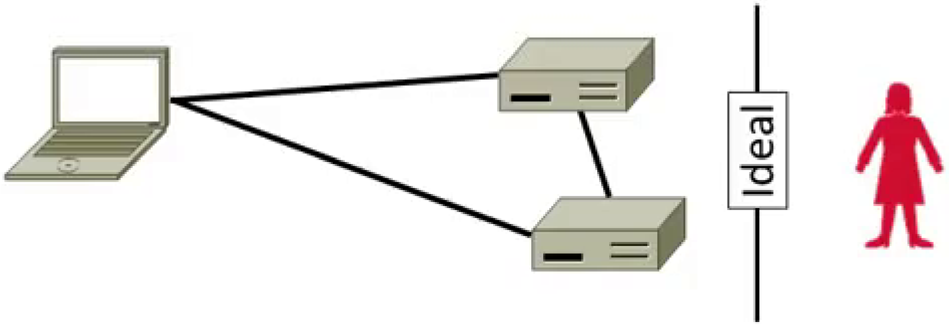
Motivation (3)

- Idea: Use the public Internet instead of leased lines – cheaper!
  - Logically separated from Internet …
  - This is a Virtual Private Network (VPN)



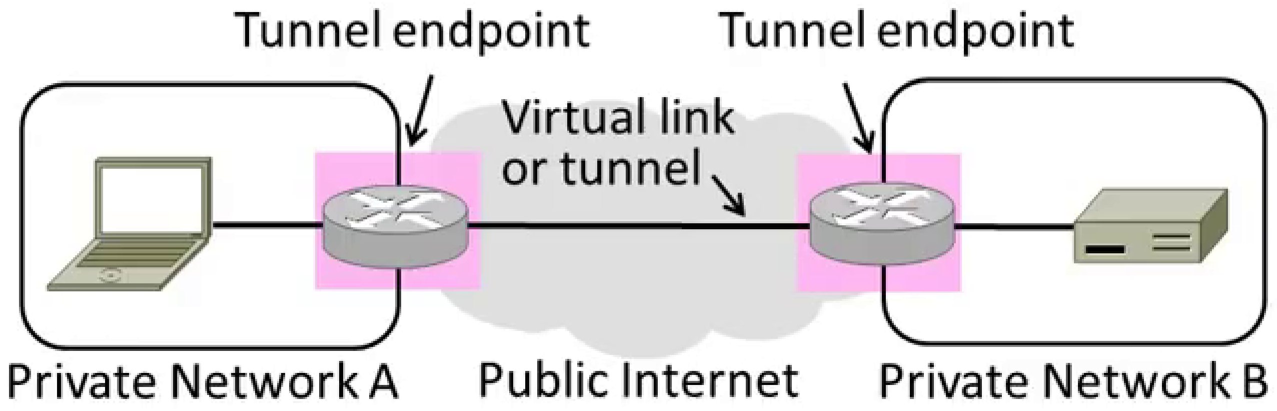
Goal and Threat Model

- Goal is to keep a logical network (VPN) separate from the Internet while using it for connectivity
  - Threat is ***EVE*** may access VPN and intercept or tamper with messages



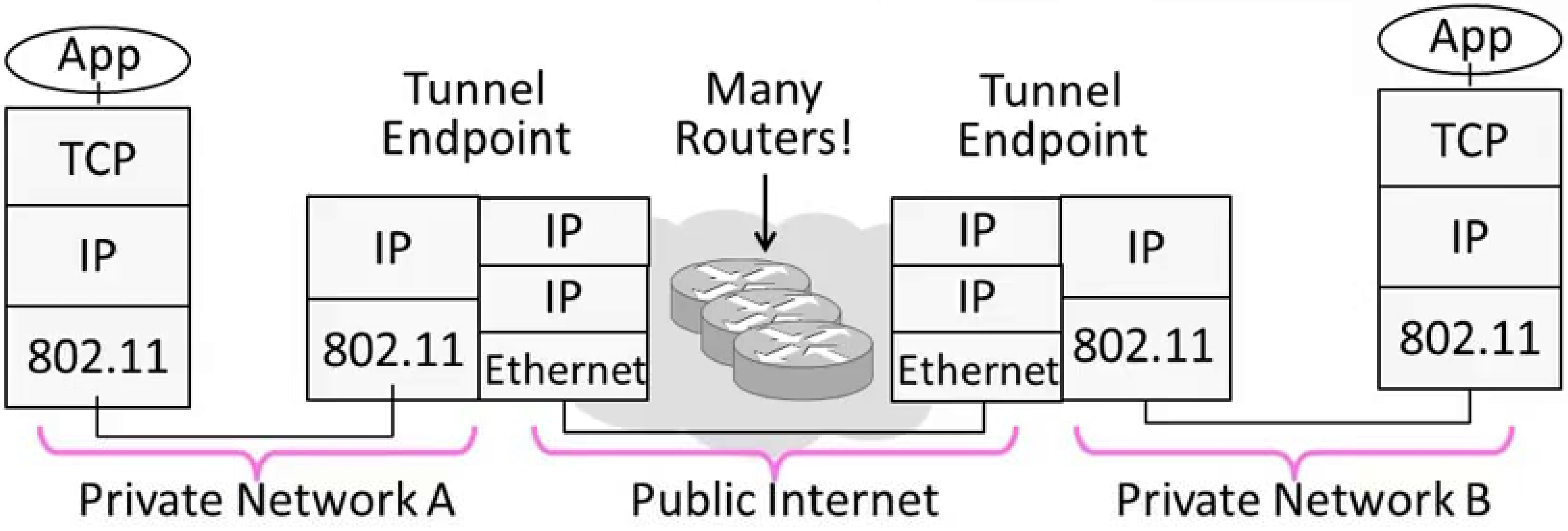
Tunneling

- How can we build a virtual link? With tunneling!
  - Hosts in private network send to each other normally
  - To cross virtual link (tunnel), endpoints encapsulate packet



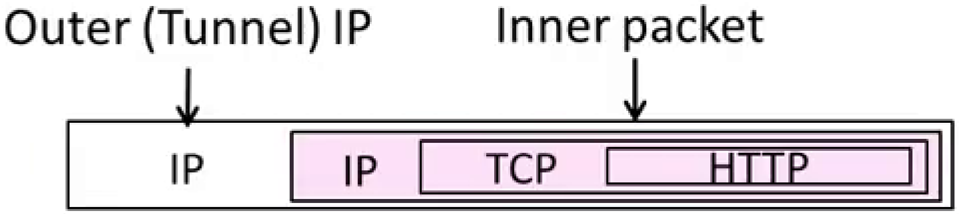
Tunneling (2)

- Tunnel endpoints encapsulate IP packets ("IP in IP")
  - Add/modify outer IP header for delivery to remote endpoint



Tunneling (3)

- Simplest encapsulation wraps packet with another IP header
  - Outer (tunnel) IP header has tunnel endpoints as source/destination
  - Inner packet has private network IP addresses as source/destination

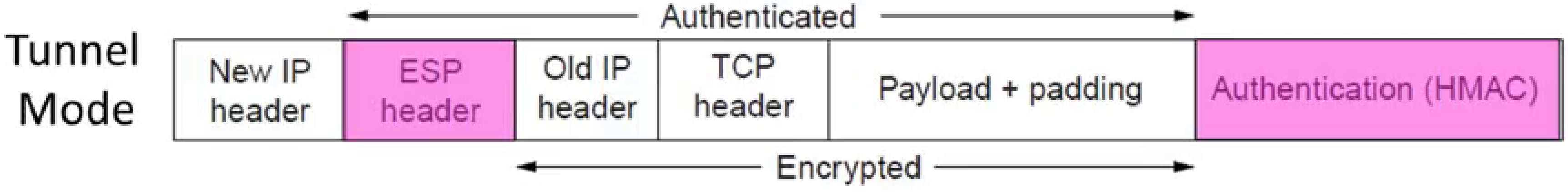


Tunneling (4)

- Tunneling alone is not secure …
  - No confidentiality, integrity/ authenticity
  - ***EVE*** can read, inject her own messages
  - We require cryptographic protections!
- IPSEC (IP Security) is often used to secure VPN tunnels

IPSEC (IP Security)

- Longstanding effort to secure the IP layer
  - Adds confidentiality, integrity/authenticity
- IPSEC operation:
  - Keys are set up for communicating host pairs
  - Communication becomes more connection-oriented
  - Header and trailer added to protect IP packets



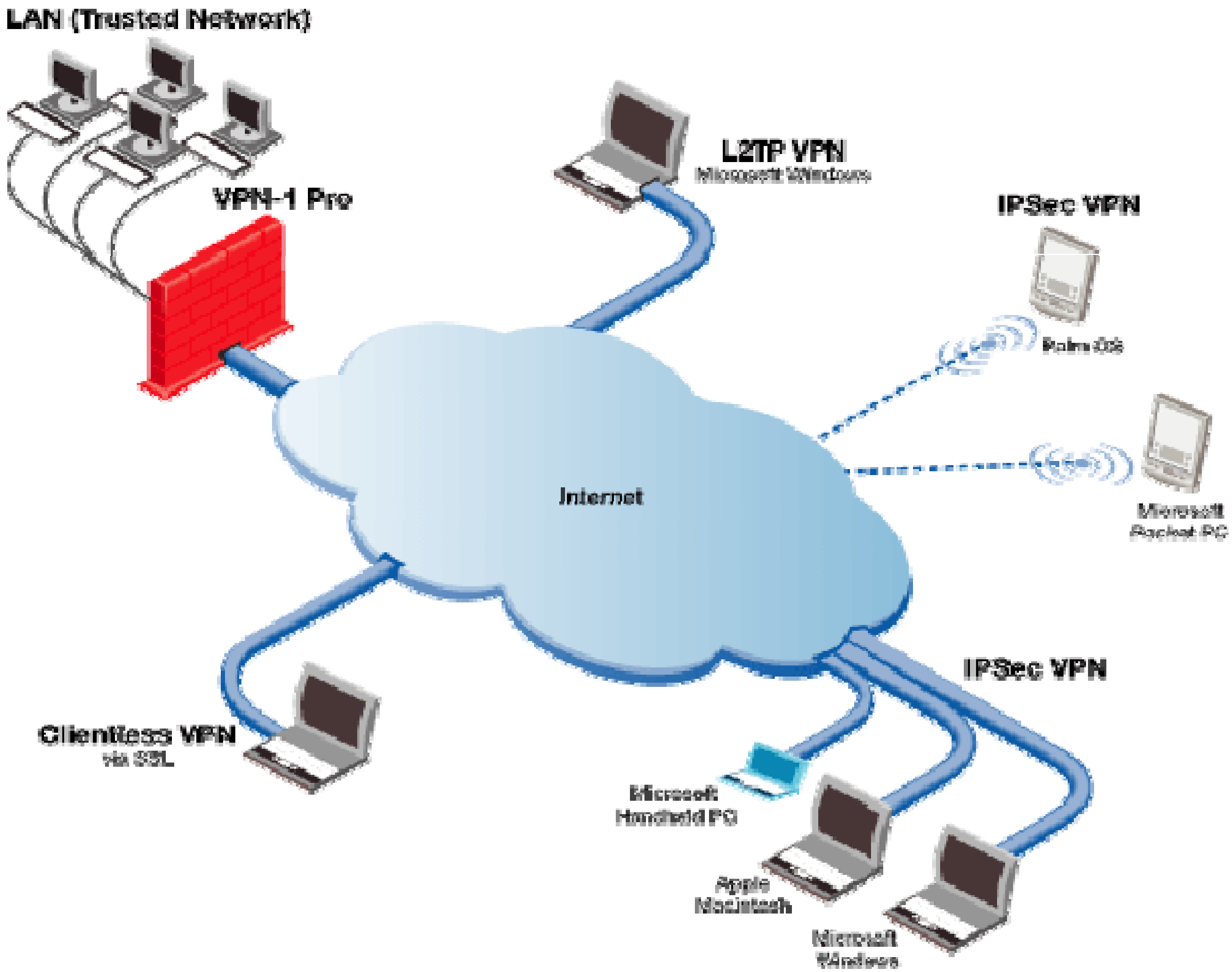
Summary

- VPNs are useful for building networks on top of the Internet
  - Virtual links encapsulate packets
  - Alters IP connectivity for hosts
- VPNs need crypto to secure messages
  - Typically IPSEC is used for confidentiality, integrity/authenticity

Summary (2)

- Three different modes of use:
  - Remote access client connections
  - LAN-to-LAN internetworking
  - Controlled access within an intranet
- Several different protocols
  - PPTP – Point-to-point tunneling protocol
  - L2TP – Layer-2 tunneling protocol
  - IPsec (Layer-3: network layer)

Data layer



Credit: Checkpoint

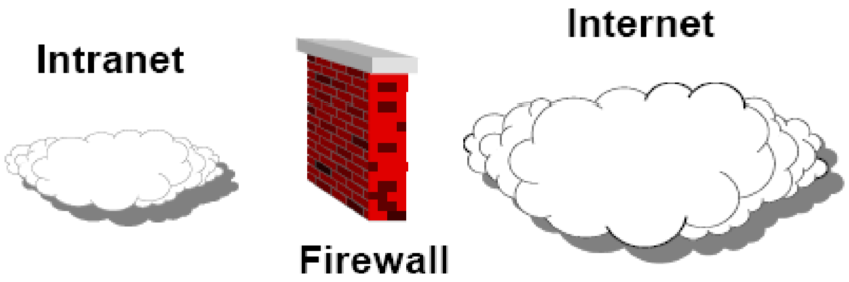
Firewall

Firewall

A *firewall* is any security system protecting the boundary of an Intranet against the Internet

Tasks of a firewall:

- *Access control* based on sender or receiver address or on addressed services (i.e. application layer protocol)
- *Behavior control*, e.g. virus checking on incoming files
- *User control*, i.e. authentication based on the source of traffic
- *Hiding* the internal network, e.g. topology, addresses, etc.
- *Logging* of passing traffic



Two fundamental concepts implemented by firewalls are

- *Packet filter*
- *Proxy server*

Types of Firewalls

1. Packet Filter

- Analyzing of network traffic and filtering due to certain rules on layer 3 and 4. A filtering can use one or a combination of the following information: source address, destination address, used protocol, connection
- If the firewall is realized in combination with a router, it is also called **Screening Router**
- Cheap and simple (all types of connections can be controlled), but filtering rules are hard to define (correctly)

2. Proxy Server (Gateway)

- "Controlled access" to a service: the firewall intercepts a requests up to layer 7 and decides, if to forward it to the receiver
- The proxy is the only computer known to the outer world
- An access control could be done basing on user identity, used protocol, and content
- More possibilities (Logging of detailed information, authentication, ...), but for each application protocol (HTTP, SMTP, FTP, …) an own proxy is needed

Packet Filter

Two possible principles:

- Everything that is not explicitly allowed, is denied
- Everything that is not explicitly denied, is allowed
- E.g. for your SMTP server with address 137.226.12.67 on port 25 you could define

```
From (IP * ), (port *)                 To (IP 137.226.12.67), (port 25)  DENY
From (IP 137.226.12.67), (port 25)     To (IP *), (port *)               ALLOW
```

  (I.e.: your mail server can send mails to everybody, but nobody is allowed to send mails to your mail server)
- In the order of their entry, all rules are applied till a matching one is found

Characteristics:

- Fast processing of packets, but only limited control on address level
- **Static packet filter** only has a fixed set of such rules
- **Dynamic packet filter** also considers a *state*:
  - Deny all packets from outer world
  - Only after a connection establishment from inside (set SYN flag), response packets coming from outside are accepted

Proxy Server

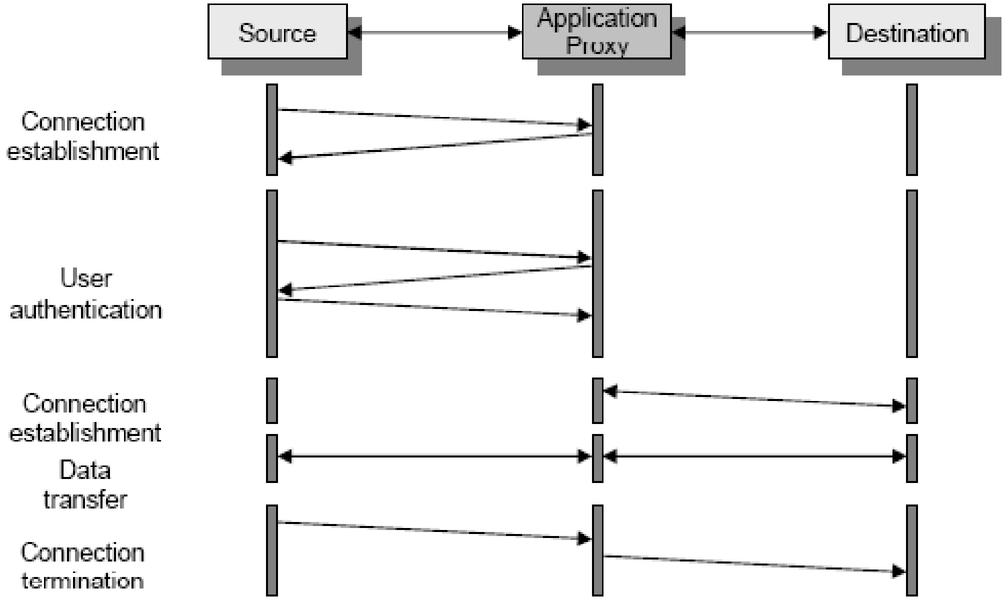
Again two possible types:

Circuit-Level Proxy

- Works on layer 3/4 only (e.g. port numbers)
- Proxy which can be used for each type of application
- The firewall intercepts all connections, thus the network structure is hidden

Application-Level Proxy

- Also checks information on layer 7
- An own proxy is needed for each application protocol (SMTP, FTP, HTTP, ...)
- A user maybe has to authenticate before usage
- Most possibilities, but most expensive



Packet Filter vs Proxy Server

Packet Filter

+ Simple
+ Low cost implementation
- Correctly specifying packet filters is a difficult and error-prone process
- Reordering packet filter rules makes specifying rules correctly even more difficult

Proxy Server

+ User authentication is possible
+ Application protocol control (e.g. virus detection) can be integrated
+ Logging of detailed information
+ Accounting
- Proxy needed for each application protocol (expensive)
- Circuit level proxies are cheaper than application level proxies, but not able to scan application data

Security Architectures

Question: which firewall to install? Where and how to implement it due to the security requirements?

- Personal Firewall
- Dual-Homed Host Firewall
- Screened Hosts Firewall
- Screened Subnet Firewall (Demilitarized Zone)
- Honeypot
- …

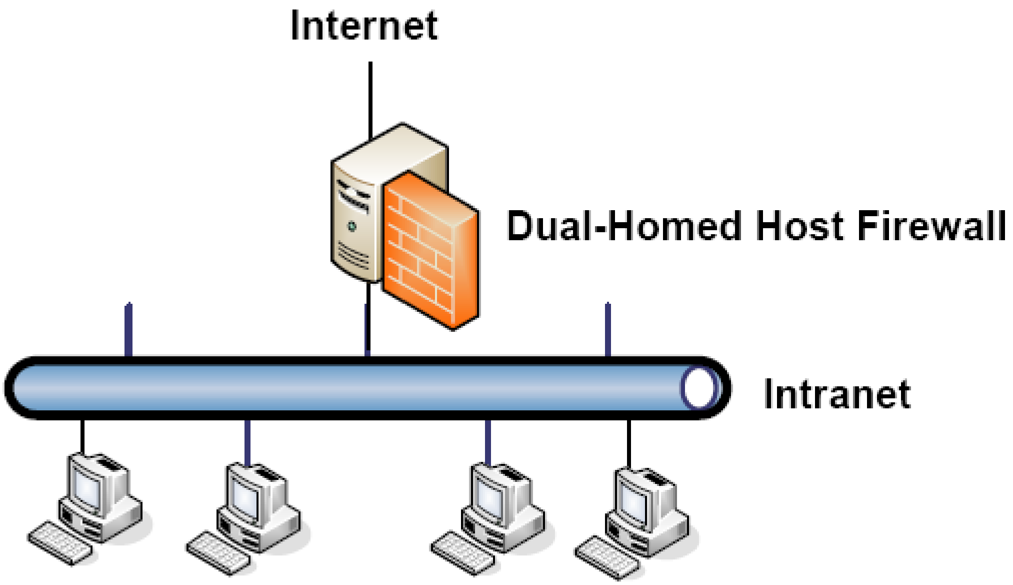
Personal firewall

- Not an own component, but a software installed on a host to protect exactly this host
- Part of operating systems to protect a user's machine at home
- Learning filter which can interact with the user to define filtering rules
- Normally not necessary because even at home the usual DSL router today has an intergrated firewall

Dual-Homed Host Firewall

Simplest implementation: realize packet filter or proxy server as an own machine:

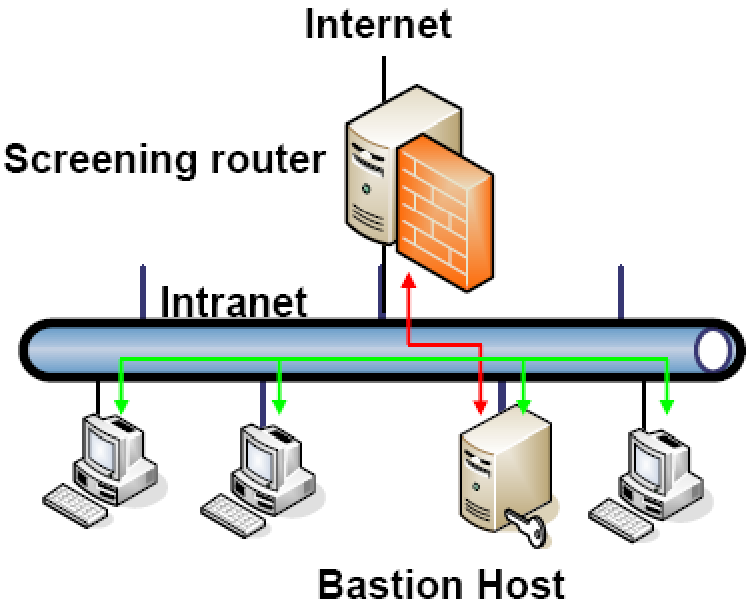
- Machine with two network interfaces
- Routes packets and processes them according to its security rules
- “All-in-one” firewall: can provide packet filter and proxy server
- Clients in the internal network can access services on the Internet either by using a proxy server in the firewall or by logging on to the firewall directly



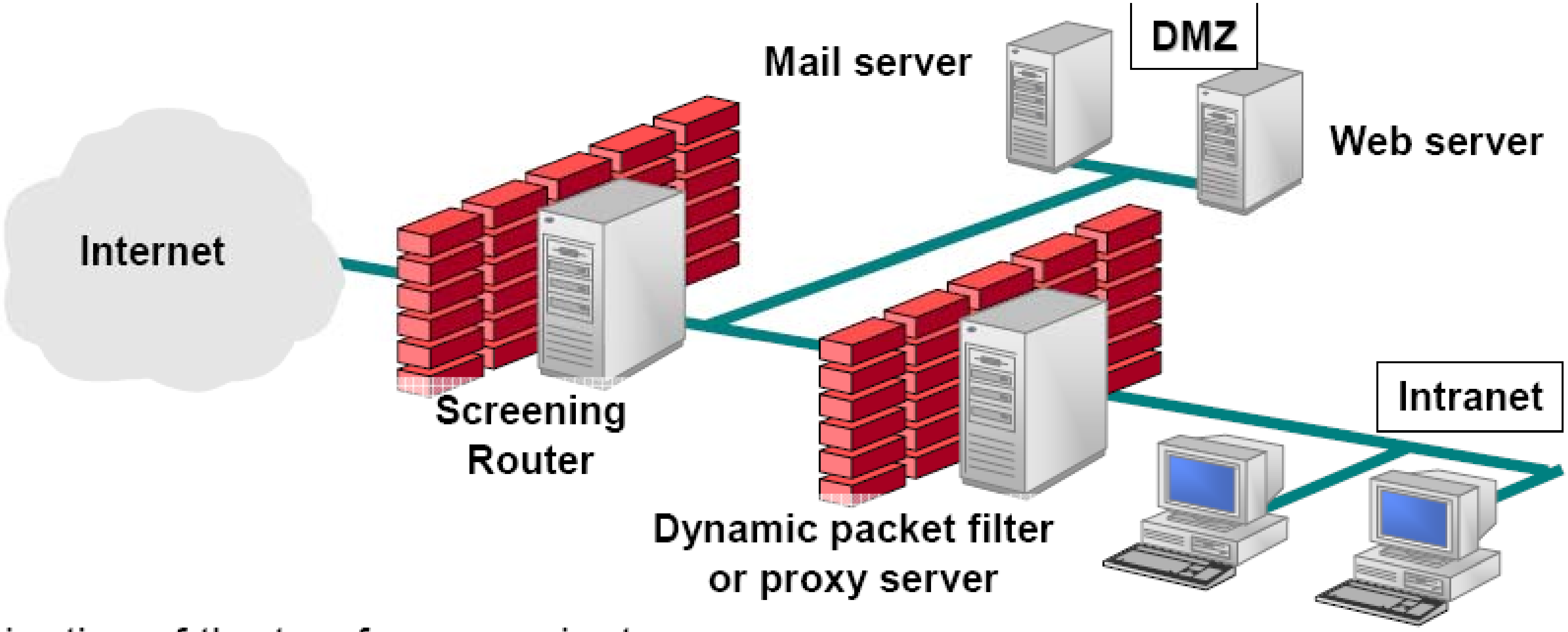
Screened Hosts Firewall

Introduce another special machine:

- Consists of a screening router and a bastion host on the internal network
- *Bastion host*: a single machine which provides all publicly accessible servers (e.g. in principle a less protected machine because we need to allow accesses to it)
- *Screening router* performs packet filtering of incoming Internet traffic
- Screening router sends all permitted incoming traffic to the bastion host, where further access control decision can be made before packets are forwarded to other hosts
- Screening router accepts internal packets only from the bastion host



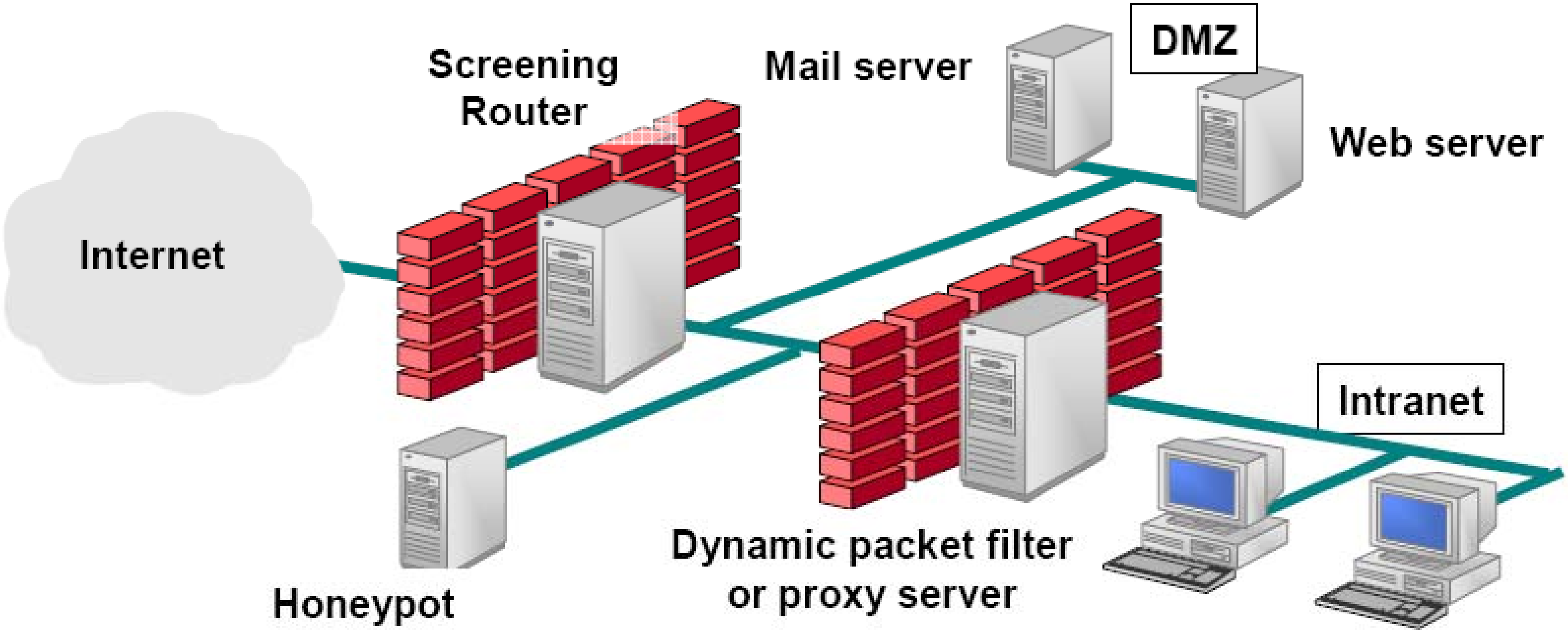
Demilitarized Zone



Combination of the two former variants:

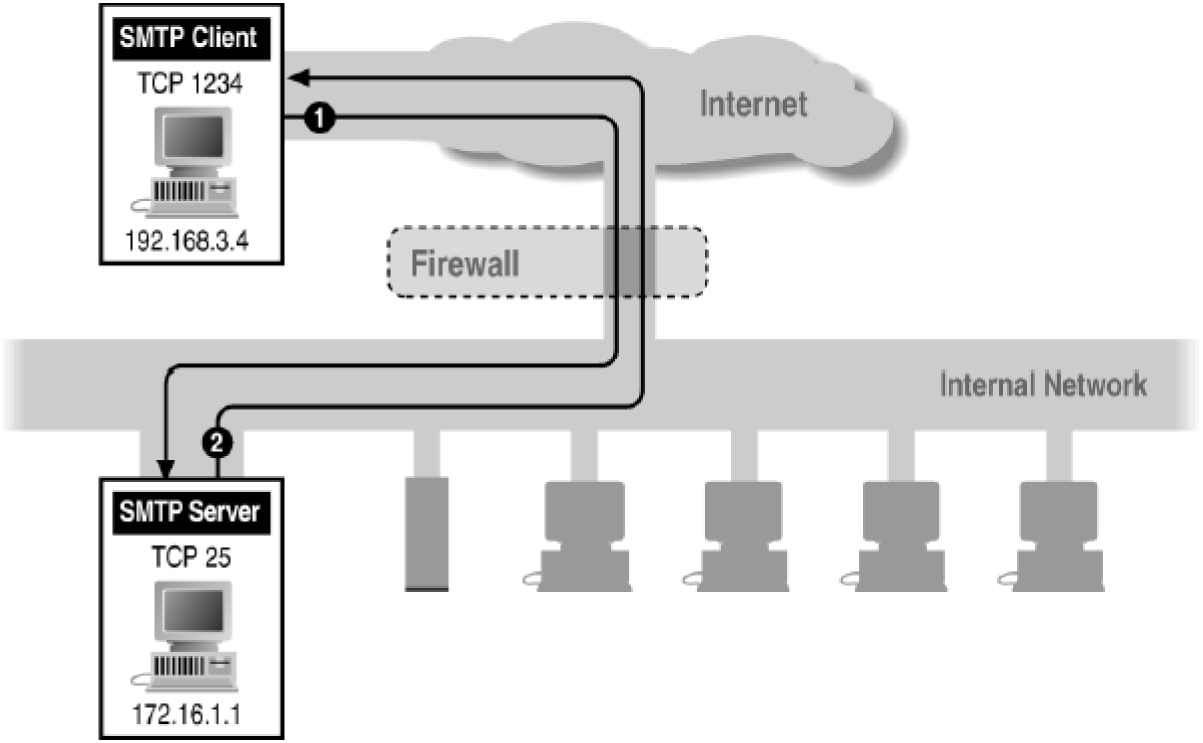
- All resources which have to be contacted from outside (without restrictions) are placed in an own network segment (*DMZ – Demilitarized Zone*) instead on a bastion host
- This segment is protected against the Internet only by a simple firewall (usually a screening router for packet filtering of uncritical systems, e.g. web server)
- The private network is protected by a more powerful firewall (dynamic packet filter and/or application-level proxy)

Additional: Honeypot



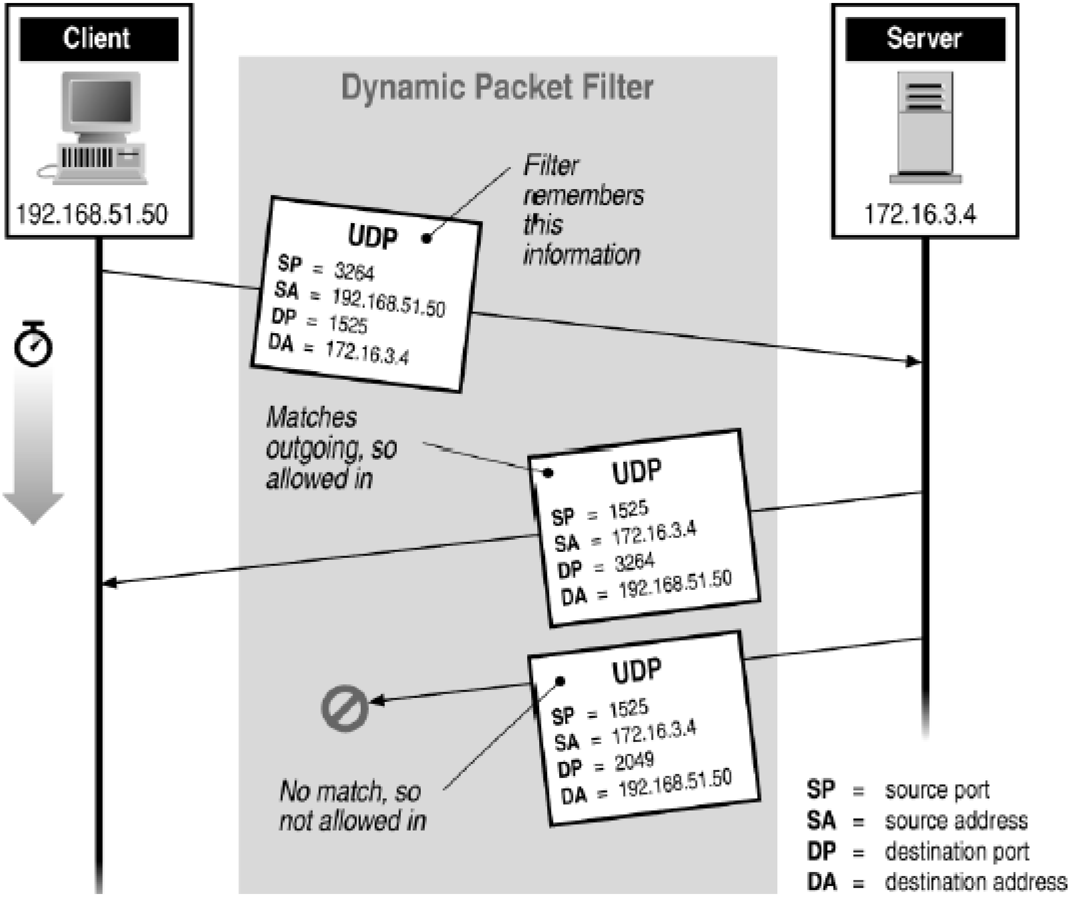
- Although possible: provide a weak faked server in your DMZ to attract attackers
- The honeypot does heavy logging and provides alarm systems instead of the real application services
- Goal: get knowledge about the attackers

Filtering Example: Inbound SMTP

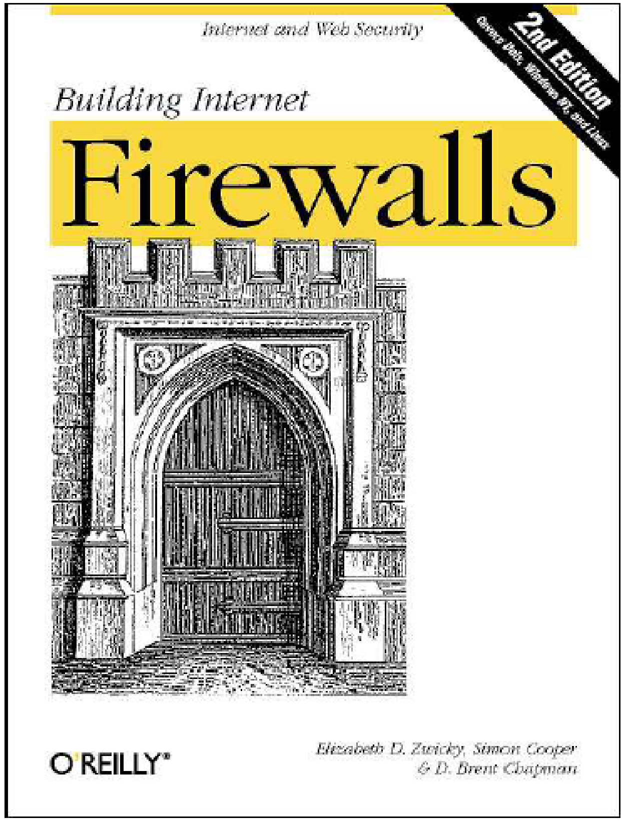


Can block external request to internal server based on port number

Stateful or Dynamic Packet Filtering



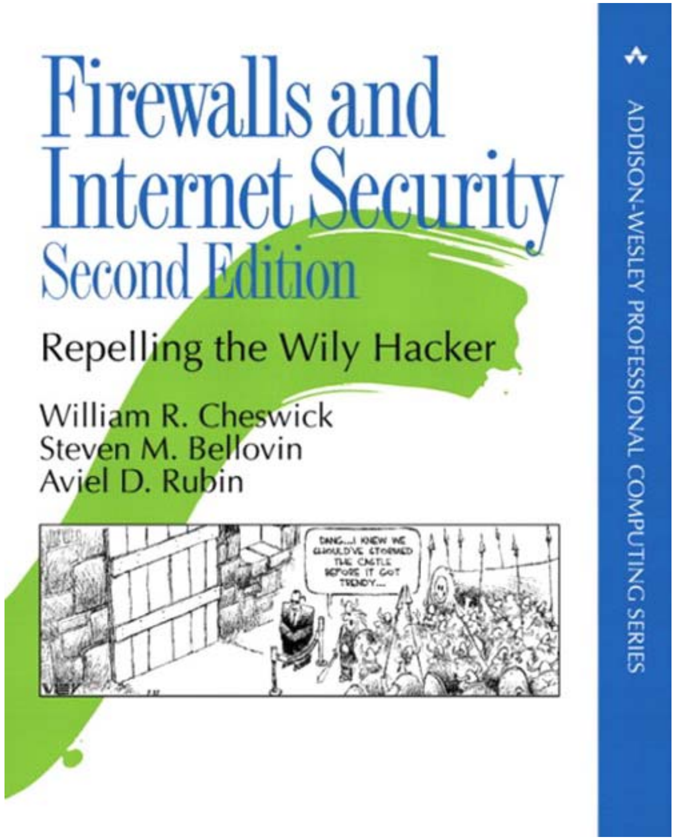
Firewall references


Internet and Web Security
2nd Edition
Covers Unix, Windows NT, and Linux
Building Internet
Firewalls
O'REILLY
Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman

Elizabeth D. Zwicky

Simon Cooper

D. Brent Chapman


Firewalls and Internet Security
Second Edition
Repelling the Wily Hacker
William R. Cheswick
Steven M. Bellovin
Aviel D. Rubin
DANG...I KNEW WE SHOULD'VE STORMED THE CASTLE BEFORE IT GOT TRENDY...
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

William R Cheswick

Steven M Bellovin

Aviel D Rubin

Intrusion Detection

Intrusion Detection

Firewalls…

- do not protect against internal attacks
- do not protect against errors in software
- do not protect against configuration errors
- do not protect against errors of external servers
- do not protect against connection hijacking
- can be eluded

→ **Intrusion Detection** to deal with these problems

Additionally to a firewall, let run an *Intrusion Detection System* (IDS) in your network to detect against attacks

Needed:

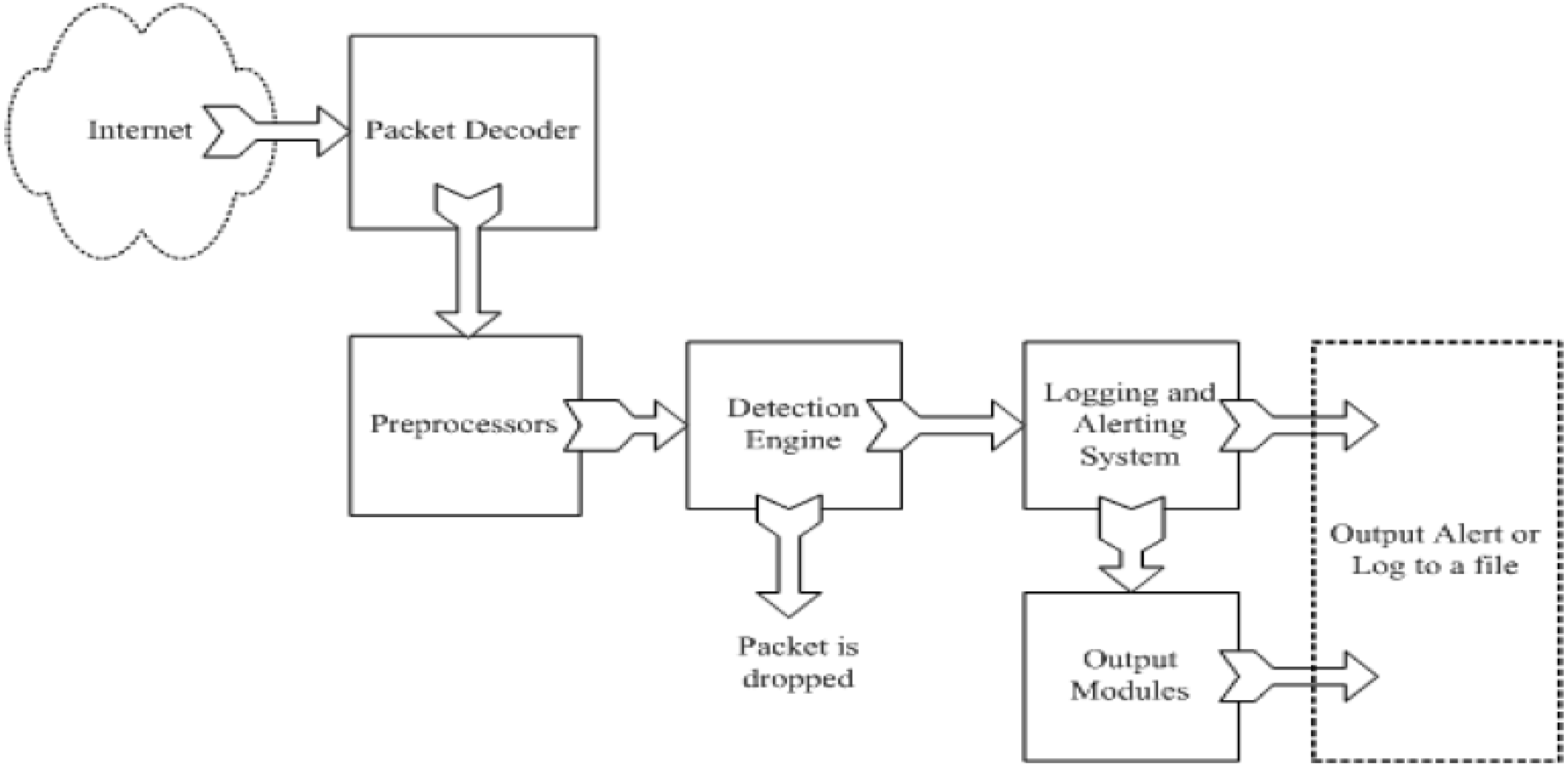
- ➢ Monitoring of the network traffic and generate events if something happens (i.e. constantly process a network audit)
- ➢ Processing of events, generating alarms
- ➢ Defining actions to be taken in presence of certain alarms

Intrusion detection

- Many intrusion detection systems
  - Close to 100 systems with current web pages
  - Network-based, host-based, or combination
- Two basic models
  - Misuse detection model
    - Maintain data on known attacks
    - Look for activity with corresponding signatures
  - Anomaly detection model
    - Try to figure out what is “normal”
    - Report anomalous behavior
- Fundamental problem: too many false alarms

Example: Snort

http://www.snort.org/

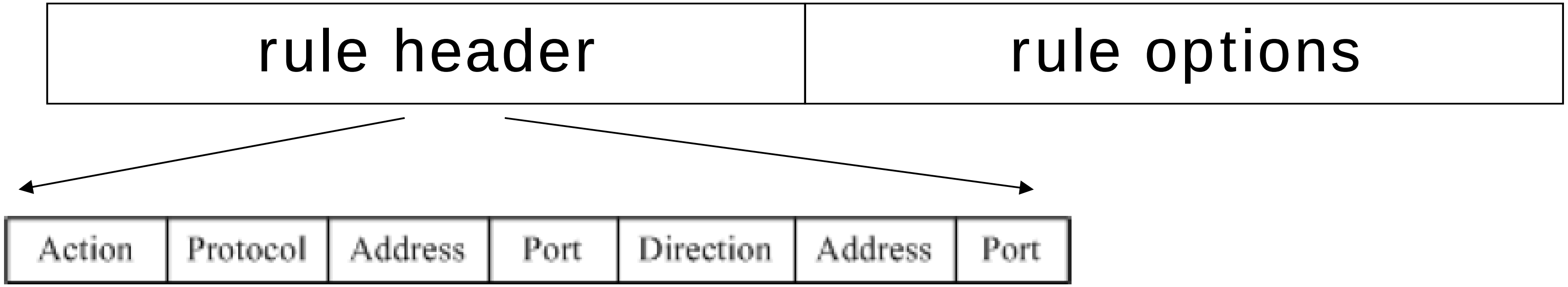


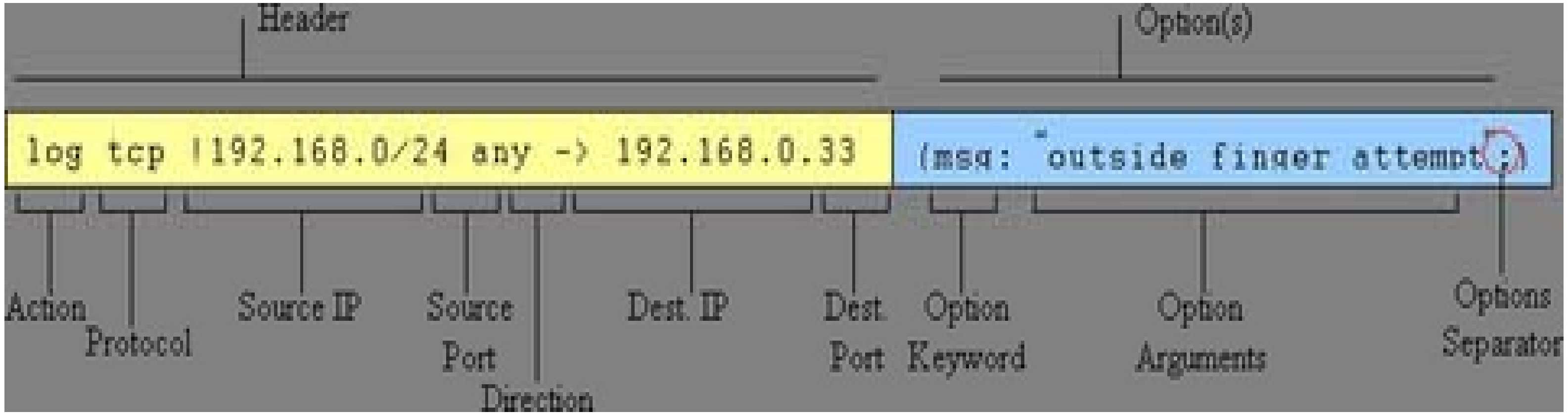
From: Rafeeq Ur Rehman, *Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID.*

Snort components

- Packet Decoder
  - input from Ethernet, SLIP, PPP…
- Preprocessor:
  - detect anomalies in packet headers
  - packet defragmentation
  - decode HTTP URI
  - reassemble TCP streams
- Detection Engine: applies rules to packets
- Logging and Alerting System
- Output Modules: alerts, log, other output

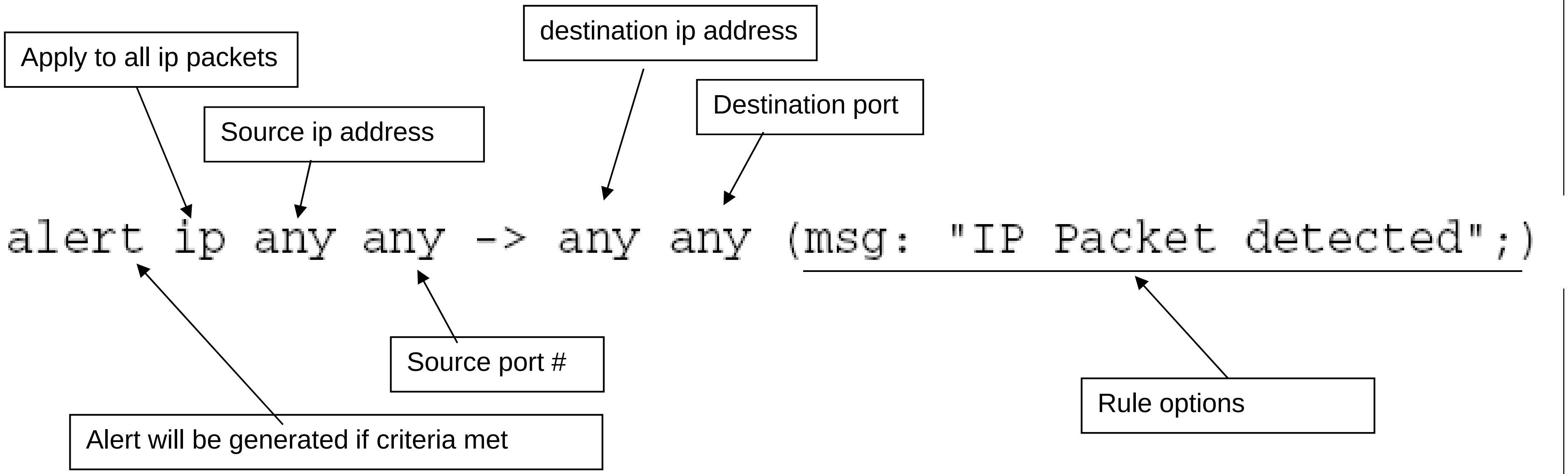
Snort detection rules





```
log tcp !192.168.0/24 any -> 192.168.0.33 (msg: "outside finger attempt";)
```

Additional examples



Apply to all ip packets

Source ip address

destination ip address

Destination port

```
alert ip any any -> any any (msg: "IP Packet detected";)
```

Source port #

Rule options

Alert will be generated if criteria met

```
alert tcp $TELNET_SERVERS 23 -> $EXTERNAL_NET any (msg:"TELNET
  Attempted SU from wrong group"; flow:
from_server,established; content:"to su root"; nocase;
  classtype:attempted-admin; sid:715; rev:6;)
```

Snort challenges

- Misuse detection – avoid known intrusions
  - Database size continues to grow
    - Snort version 2.3.2 had 2,600 rules
  - Snort spends 80% of time doing string match

- Anomaly detection – identify new attacks
  - Probability of detection is low

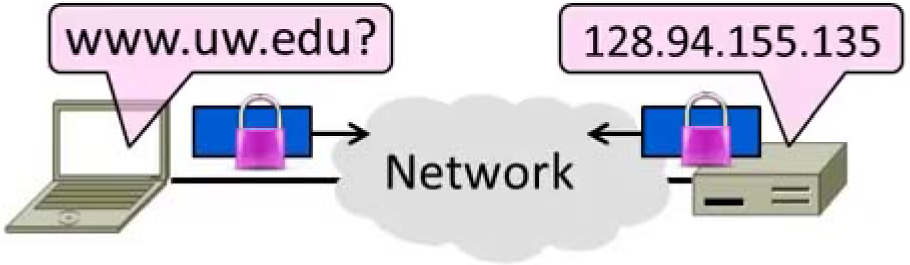
Difficulties in anomaly detection

- Lack of training data
  - Lots of “normal” network, system call data
  - Little data containing realistic attacks, anomalies
- Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally
- Main characteristics not well understood
  - By many measures, attack may be within bounds of “normal” range of activities
- False identifications are very costly
  - Sys Admin spend many hours examining evidence

DNSSEC

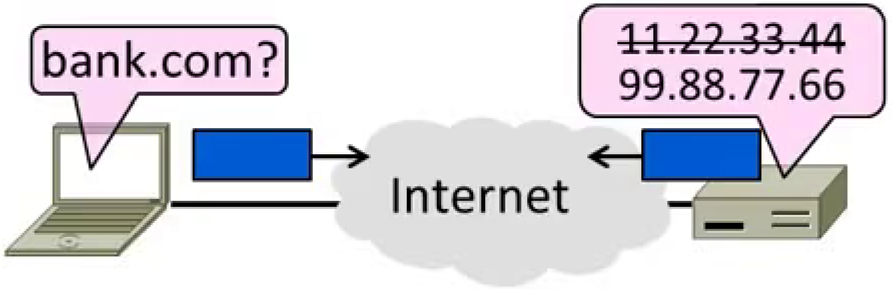
Topic

- Securing Internet naming
  - DNS security extensions (DNSSEC)



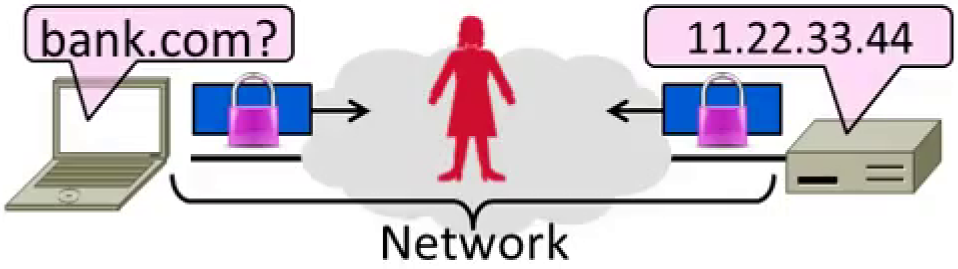
Goal and Threat Model

- Naming is a crucial Internet service
  - Binds host name to IP address
  - Wrong binding can be disastrous …



Goal and Threat Model (2)

- Goal is to secure the DNS so that the returned binding is correct
  - Integrity/authenticity vs confidentiality
- Attacker (Trudy) can intercept/tamper with messages on the network

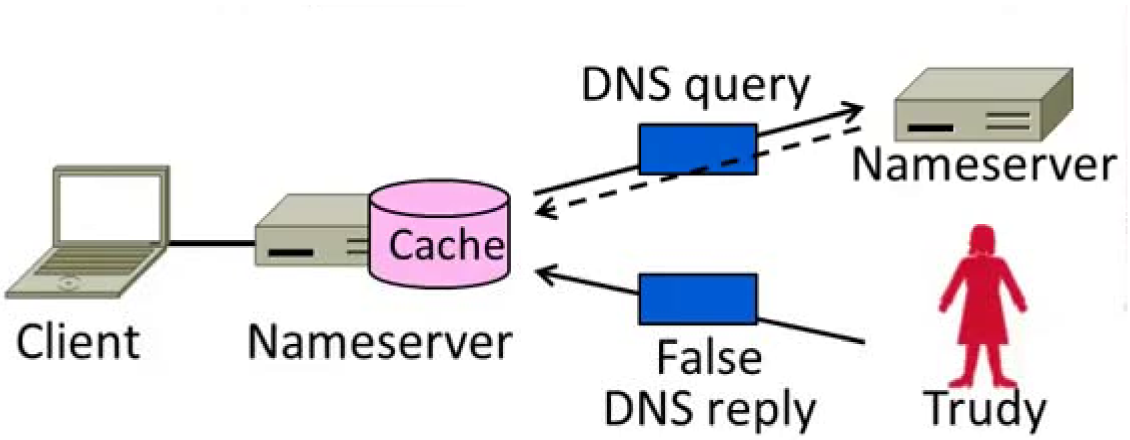


DNS Spoofing

- Hang on – how can a network attacker corrupt the DNS?
- Trudy can trick a nameserver into caching the wrong binding
  - By using the DNS protocol itself
  - This is called DNS spoofing

DNS Spoofing (2)

- To spoof, Trudy returns a fake DNS response that appears to be true
  - Fake response contains bad binding



DNS Spoofing (3)

- Lots of questions!
  1. How does Trudy know when the DNS query is sent and what it is for?
  2. How can Trudy supply a fake DNS reply that appears to be real?
  3. What happens when the real DNS reply shows up?

- There are solutions to each issue …

DNS Spoofing (4)

1. How does Trudy know when the query is sent and what it is for?

- Trudy can make the query herself!
  - Nameserver works for many clients
  - Trudy is just another client

DNS Spoofing (5)

2. How can Trudy supply a fake DNS reply that appears to be real?

- A bit more difficult. DNS checks:
  - Reply is from authoritative nameserver (e.g., .com)
  - Reply ID that matches the request
  - Reply is for outstanding query
- • (Nothing about content though …)

DNS Spoofing (6)

2. How can Trudy supply a fake DNS reply that appears to be real?

- Techniques:
  - Put IP of authoritative nameserver as the source IP address
  - ID is 16 bits (64K). Send many guesses! (Or if a counter, sample to predict.)
  - Send reply right after query
- Good chance of succeeding!

DNS Spoofing (7)

3. What happens when the real DNS reply shows up?

- Likely not be a problem
  - There is no outstanding query after fake reply is accepted
  - So real reply will be discarded

DNSSEC (DNS Security Extensions)

- Extends DNS with new record types
  - RRSIG for digital signatures of records
  - DNSKEY for public keys for validation
  - DS for public keys for delegation
  - First version in ‘97, revised by ’05
- Deployment requires software upgrade at both client and server
  - Root servers upgraded in 2010
  - Followed by uptick in deployment

DNSSEC (2) – New Records

- As well as the usual A, NS records
- RRSIG
  - Digital signatures of domain records
- DNSKEY
  - Public key used for domain RRSIGs (for validation of signatures)
- DS
  - Public keys for delegated domain
- NSEC/NSEC3
  - Authenticated denial of existence (answer from an authoritative NS that really there is no domain)

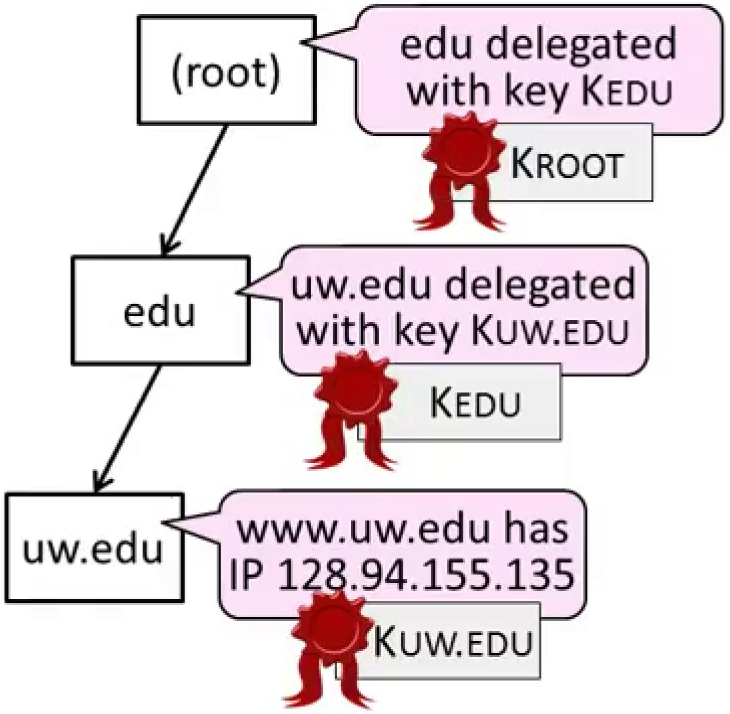
DNSSEC (3) – Validating Replies

- Clients query DNS as usual, then validate replies to check that content is authentic
- Trust anchor is root public keys
  - Part of DNS client configuration
- Trust proceeds down DNS hierarchy
  - Similar concept to SSL certificates

DNSSEC (4) – Validating Replies

- Client queries www.uw.edu as usual
  - Replies include signatures/keys
- Client validates answer:

1. KROOT is a trust anchor
2. Use KROOT to check KEDU
3. Use KEDU to check KUW.EDU
4. Use KUW.EDU to check IP


(root)
edu delegated with key KEDU
KROOT
edu
uw.edu delegated with key KUW.EDU
KEDU
uw.edu
www.uw.edu has IP 128.94.155.135
KUW.EDU

DNSSEC (5)

- Other features too:
  - Authoritative answers a domain record doesn’t exist (NSEC/NSEC3)
  - Optional anti-spoofing to bind query and reply
  - Flags related to deployment …

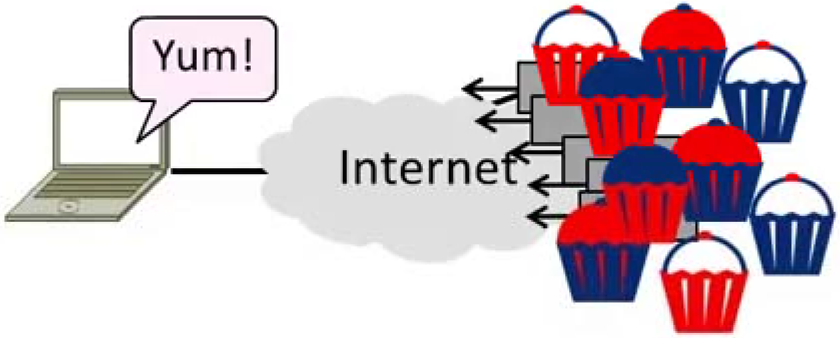
Summary

- DNS spoofing is possible without added security measures
  - Large problem in practice!
- DNSSEC adds authentication (only) of replies to the DNS
  - Using a hierarchy of public keys

DDoS

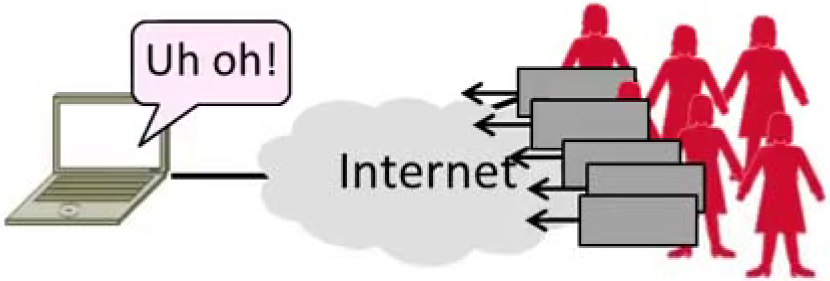
Topic

- Distributed Denial-of-Service (DDOS)
  - An attack on network availability



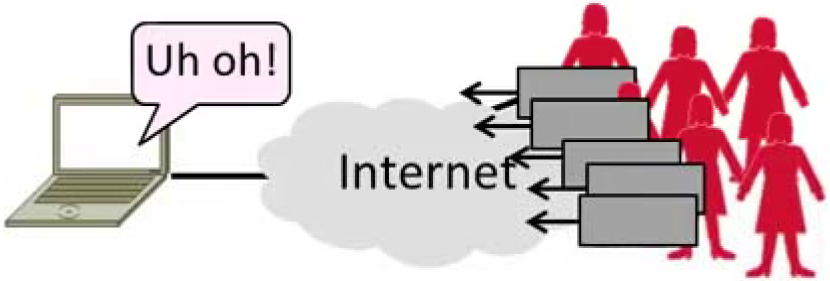
Topic

- Distributed Denial-of-Service (DDOS)
  - An attack on network availability



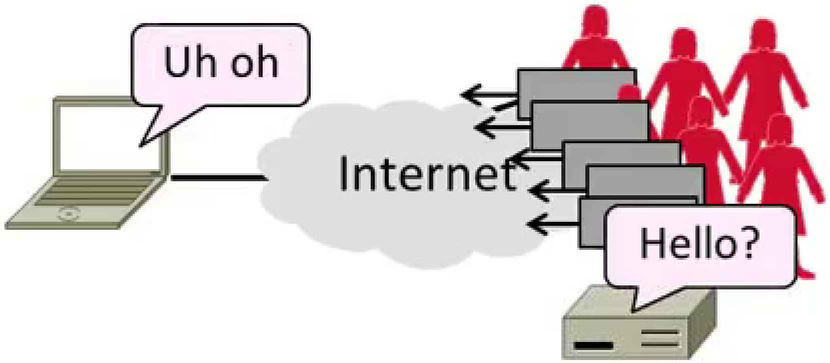
Motivation

- The best part of IP connectivity
  - You can send to any other host
- The worst part of IP connectivity
  - Any host can send packets to you!



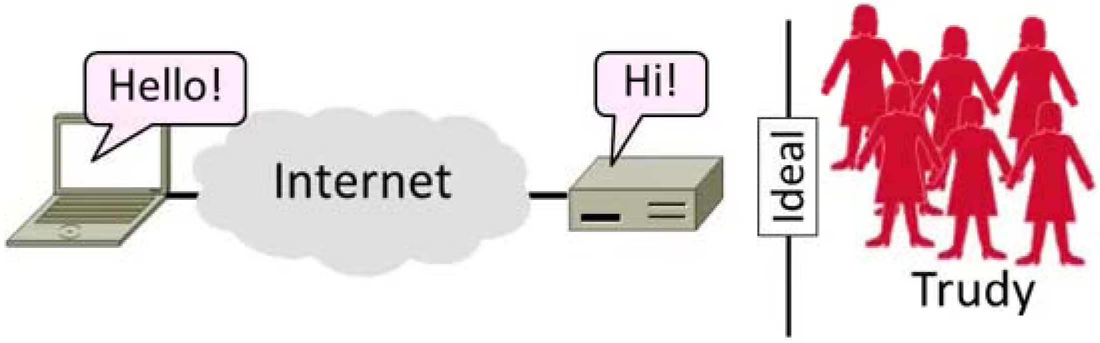
Motivation (2)

- Flooding a host with many packets can interfere with its IP connectivity
  - Host may become unresponsive
  - This is a form of denial-of-service



Goal and Threat Model

- Goal is for host to keep network connectivity for desired services
  - Threat is Trudy may overwhelm host with undesired traffic



Internet Reality

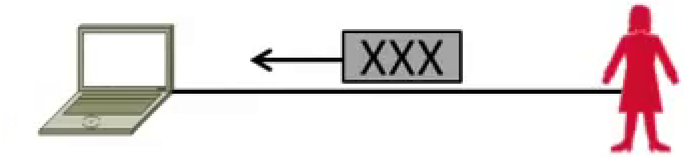
- Distributed Denial-of-Service is a huge problem today!
  - Akamai Q3-12 reports DDOS against US banks peaking at 65 Gbps of traffic flooding the bank
- There are no great solutions
  - CDNs, network traffic filtering, and best practices all help

Denial-of-Service

- Denial-of-service means a system is made unavailable to intended users
  - Typically because its resources are consumed by attackers instead
- In the network context:
  - “System” means server
  - “Resources” mean bandwidth (network) or CPU/memory (host)

Host Denial-of-Service

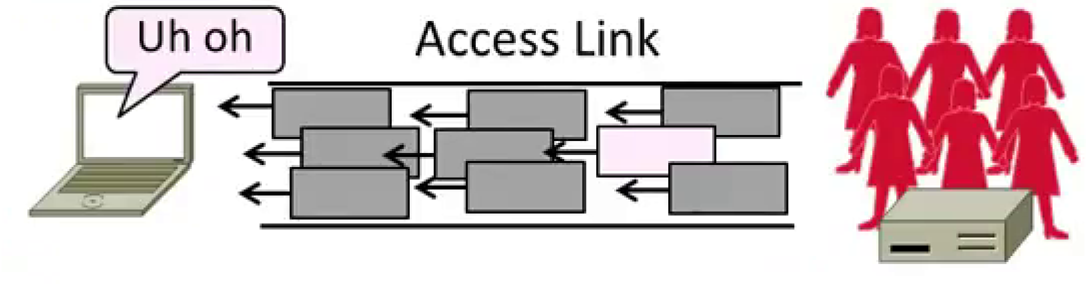
- Strange packets can sap host resources!
  - “Ping of Death” malformed packet (bug the kernel and system crash)
  - “SYN flood” sends many TCP connect requests and never follows up
  - Few bad packets can overwhelm host



- Patches exist for these vulnerabilities
  - Read about “SYN cookies” for interest

Network Denial-of-Service

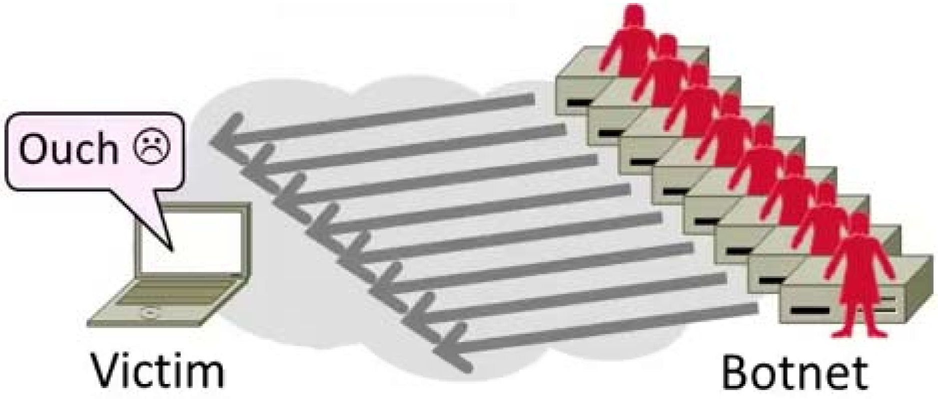
- Network DOS needs many packets
  - To saturate network links
  - Causes high congestion/loss



- Helpful to have many attackers or Distributed Denial-of-Service

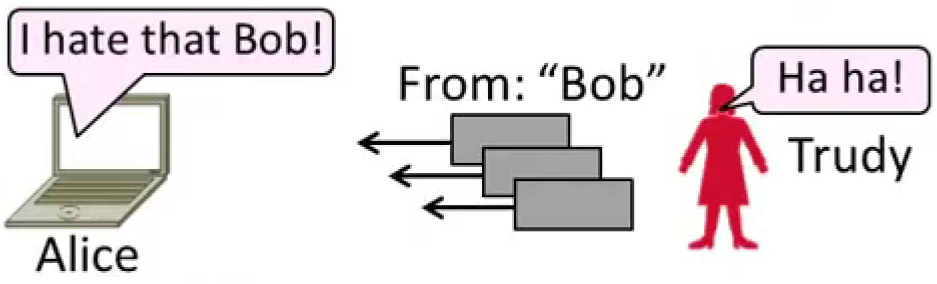
Distributed Denial-of-Service (DDoS)

- Botnet provides many attackers in the form of compromised hosts
  - Hosts send traffic flood to victim
  - Network saturates near victim



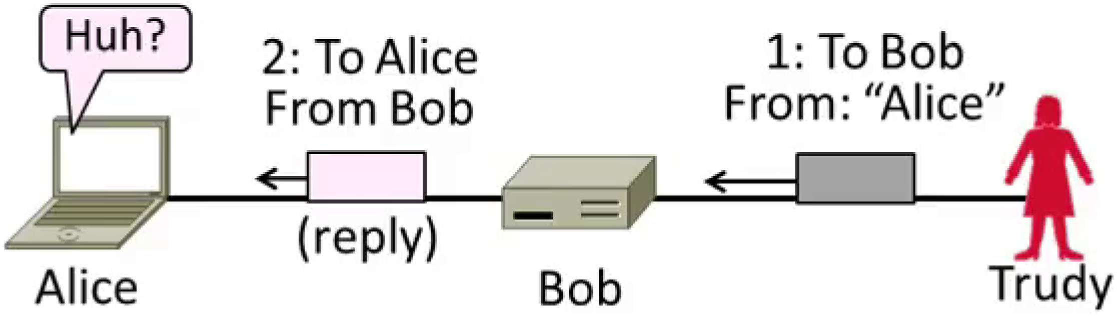
Complication: Spoofing

- Attackers can falsify their IP address
  - Put fake source address on packets
  - Historically network doesn't check
  - Hides location of the attackers
  - Called IP address spoofing



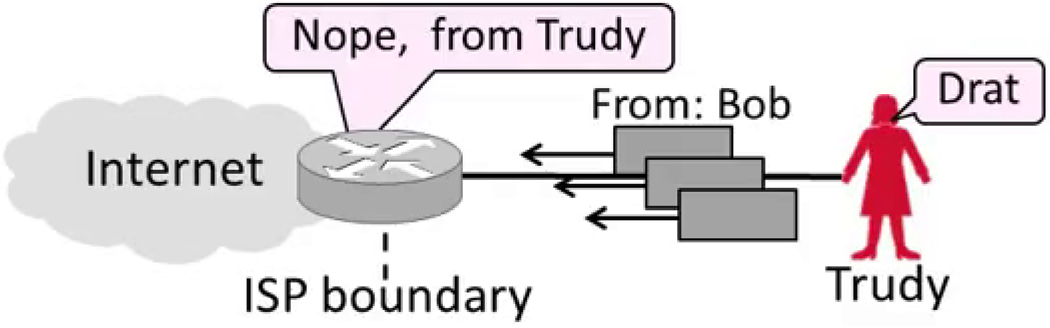
Spoofing (2)

- Actually, it’s worse than that
  - Trudy can trick Bob into really sending packets to Alice
  - To do so, Trudy spoofs Alice to Bob



Best Practice: Ingress Filtering

- Idea: Validate the IP source address of packets at ISP boundary (Duh!)
  - Ingress filtering is a best practice, but deployment has been slow



Flooding Defenses

1. Increase network capacity around the server; harder to cause loss
   - Use a CDN for high peak capacity
2. Filter out attack traffic within the network (at routers)
   - The earlier the filtering, the better
   - Ultimately what is needed, but ad hoc measures by ISPs today

Acknowledgements

Material in this lecture are taken from the slides prepared by:

- Prof. Dan Boneh (Standford)
- Prof. O. Spaniol (RWTH Aachen)
- Prof. David Wetheral (University of Washington)