

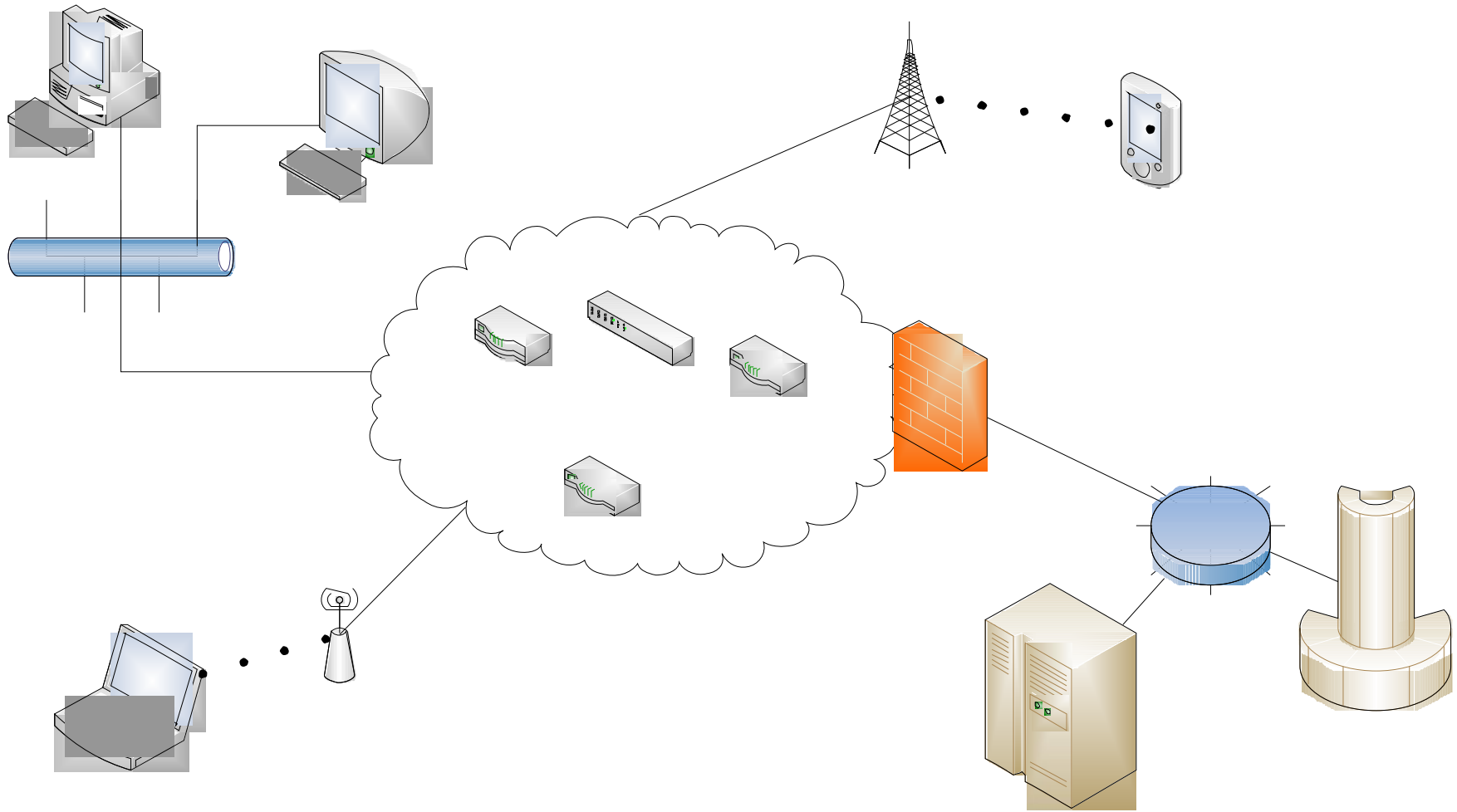
# Network Technology Review and Security Concerns

Computer Security I  
CS461/ECE422  
Fall 2010

# Outline

- Overview Issues and Threats in Network Security
- Review basic network technology
  - TCP/IP in particular
  - Attacks specific to particular technologies

# Security Issues in Networks



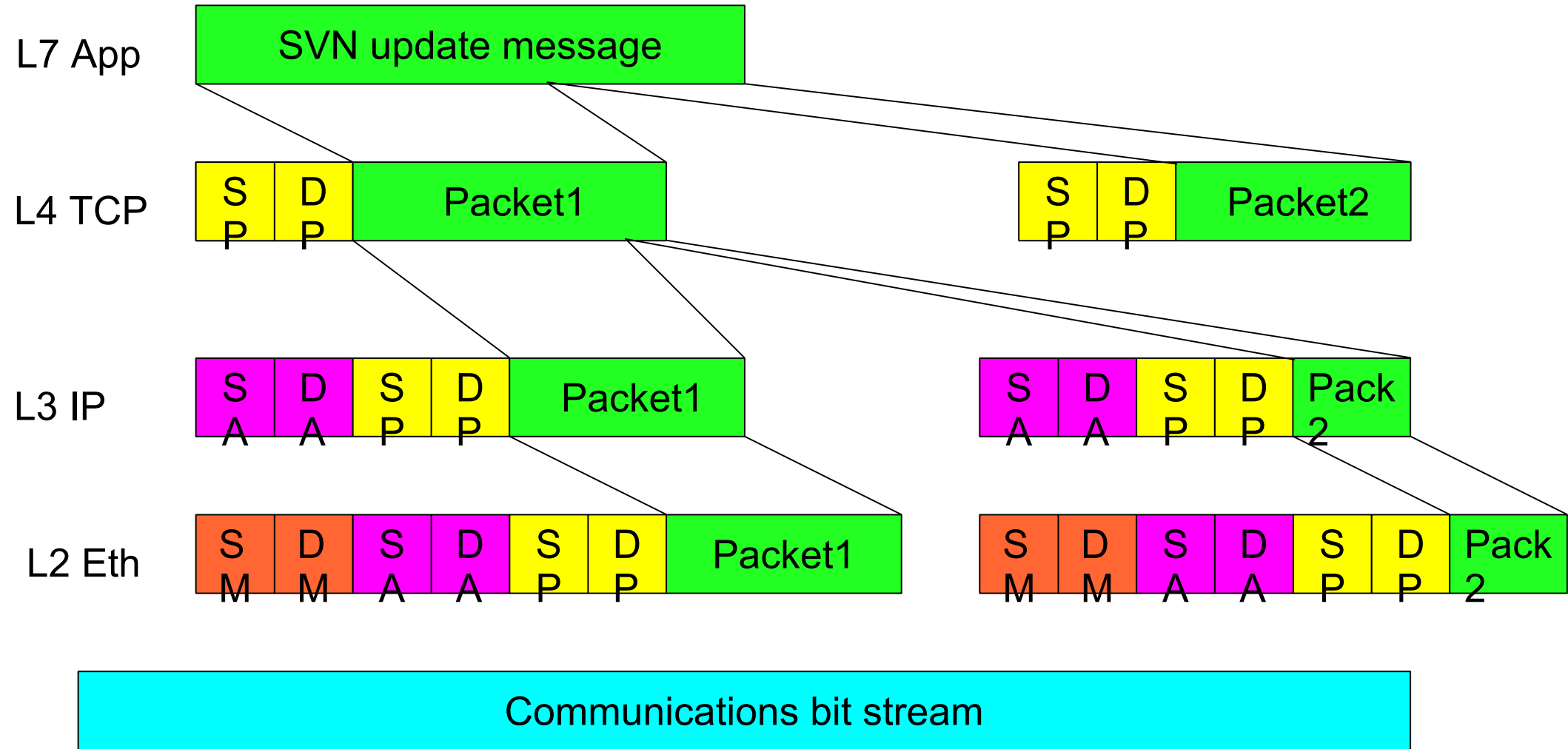
# Increased Security Complexity

- Different operating systems
  - Computers, Servers, Network Devices
- Multiple Administrative Domains
- Need to open access
- Multiple Paths and shared resources
- Anonymity

# OSI Reference Model

- The layers
  - 7: Application, e.g., HTTP, SMTP, FTP
  - 6: Presentation
  - 5: Session
  - 4: Transport, e.g. TCP, UDP
  - 3: Network, e.g. IP, IPX
  - 2: Data link, e.g., Ethernet frames, ATM cells
  - 1: Physical, e.g., Ethernet media, ATM media
- Standard software engineering reasons for thinking about a layered design

# Message mapping to the layers



# Confidentiality/Integrity Physical Layer

- Radio waves
  - Just listen
- Microwave
  - Point-to-point sort of
  - Dispersal
- Ethernet
  - Inductance of cables
  - Tapping into ethernet cables
  - Promiscuous sniffing

# Switches

- Original ethernet broadcast all packets
- Layer two means of passing packets
  - Learn or config which MAC's live behind which ports
  - Only pass traffic to the appropriate port
- Span ports
  - Mirror all traffic



# Physical Denial of Service

- Radio
  - Jamming
- Cables
  - Cutting or mutilating

# Network Layer - IP

- Moves packets between computers
  - Possibly on different physical segments
  - Best effort
- Technologies
  - Routing
  - Lower level address discovery (ARP)
  - Error Messages (ICMP)

# IPv4

- See Wikipedia for field details
  - <http://en.wikipedia.org/wiki/IPv4>

Version	IHL	Type of service	Total length		
Identification			DF	MF	Frag Offset
Time to live	Protocol		Header checksum		
Source address					
Destination Address					
0 or more words of options					

# Ipv4 Addressing

- Each entity has at least one address
- Addresses divided into subnetwork
  - Address and mask combination
  - 192.168.1.0/24 or 10.0.0.0/8
  - 192.168.1.0 255.255.255.0 or 10.0.0.0 255.0.0.0
  - 192.168.1.0-192.168.1.255 or 10.0.0.0-10.255.255.255
- Addresses in your network are “directly” connected
  - Broadcasts should reach them
  - No need to route packets to them

# Address spoofing

- Sender can put any source address in packets he sends:
  - Can be used to send unwelcome return traffic to the spoofed address
  - Can be used to bypass filters to get unwelcome traffic to the destination
- Reverse Path verification can be used by routers to broadly catch some spoofers

# Address Resolution Protocol (ARP)

- Used to discover mapping of neighboring ethernet MAC to IP addresses.
  - Need to find MAC for 192.168.1.3 which is in your interface's subnetwork
  - Broadcast an ARP request on the link
  - Hopefully receive an ARP reply giving the correct MAC
  - The device stores this information in an ARP cache or ARP table

# ARP cache poisoning

- Bootstrap problem with respect to security. Anyone can send an ARP reply
  - The Ingredients to ARP Poison,  
<http://www.governmentsecurity.org/articles/TheIngredientsToARPPoison.php>
- Classic Man-in-the-middle attack
  - Send ARP reply messages to device so they think your machine is someone else
  - Better than simple sniffing because not just best effort.
- Solutions
  - Encrypt all traffic
  - Monitoring programs like arpwatch to detect mapping changes
    - Which might be valid due to DHCP

# Basic IPv4 Routing

- Static routing. Used by hosts, firewalls and routers.
  - Routing table consists of entries of
    - Network, Next hop address, metric, interface
  - May have routing table per incoming interface
  - To route a packet, take the destination address and find the best match network in the table. In case of a tie look at the metric
    - Use the corresponding next hop address and interface to send the packet on.
    - The next hop address is on the same link as this device, so you use the next hop's data-link address, e.g. ethernet MAC address
  - Decrement “time to live” field in IP header at each hop. Drop packet when it reaches 0
    - Attempt to avoid routing loops
    - As internet got bigger, TTL fields got set bigger. 255 maximum



# Routing example

- Receive a packet destined to 192.168.3.56 on inside interface
- Local routing table for inside interface
  1. 192.168.2.0/30, 127.0.0.1, 1, outside
  2. 192.168.5.0/29, 127.0.0.1, 1, dmz
  3. 192.168.3.0/24, 192.168.5.6, 1, dmz
  4. 192.168.3.0/24, 192.168.1.2, 3, outside
  5. 0.0.0.0/0, 192.168.1.2, 1, outside
- Entries 3 and 4 tie. But metric for 3 is better
- Entries 1 and 2 are for directly connected networks

# Source Based Routing

- In the IP Options field, can specify a source route
  - Was conceived of as a way to ensure some traffic could be delivered even if the routing table was completely screwed up.
- Can be used by the bad guy to avoid security enforcing devices
  - Most folks configure routers to drop packets with source routes set

# IP Options in General

- Originally envisioned as a means to add more features to IP later
- Most routers drop packets with IP options set
  - Stance of not passing traffic you don't understand
  - Therefore, IP Option mechanisms never really took off
- In addition to source routing, there are security Options
  - Used for DNSIX, a MLS network encryption scheme

# Dynamic Routing Protocols

- For scaling, discover topology and routing rather than statically constructing routing tables
  - Open Shortest Path First (OSPF): Used for routing within an administrative domain
  - RIP: not used much anymore
  - Border Gateway Protocol (BGP): Used for routing between administrative domains. Can encode non-technical transit constraints, e.g. Domain X will only carry traffic of paying customers
    - Receives full paths from neighbors, so it avoids counts to infinity.

# Dynamic Routing

- Injecting unexpected routes a security concern.
  - BGP supports peer authentication
  - BGP blackholing is in fact used as a mechanism to isolate “bad” hosts
  - Filter out route traffic from unexpected (external) points
  - OSPF has MD5 authentication, and can statically configure neighbor routers, rather than discover them.
- Accidents are just as big of a concern as malicious injections

# Internet Control Message Protocol (ICMP)

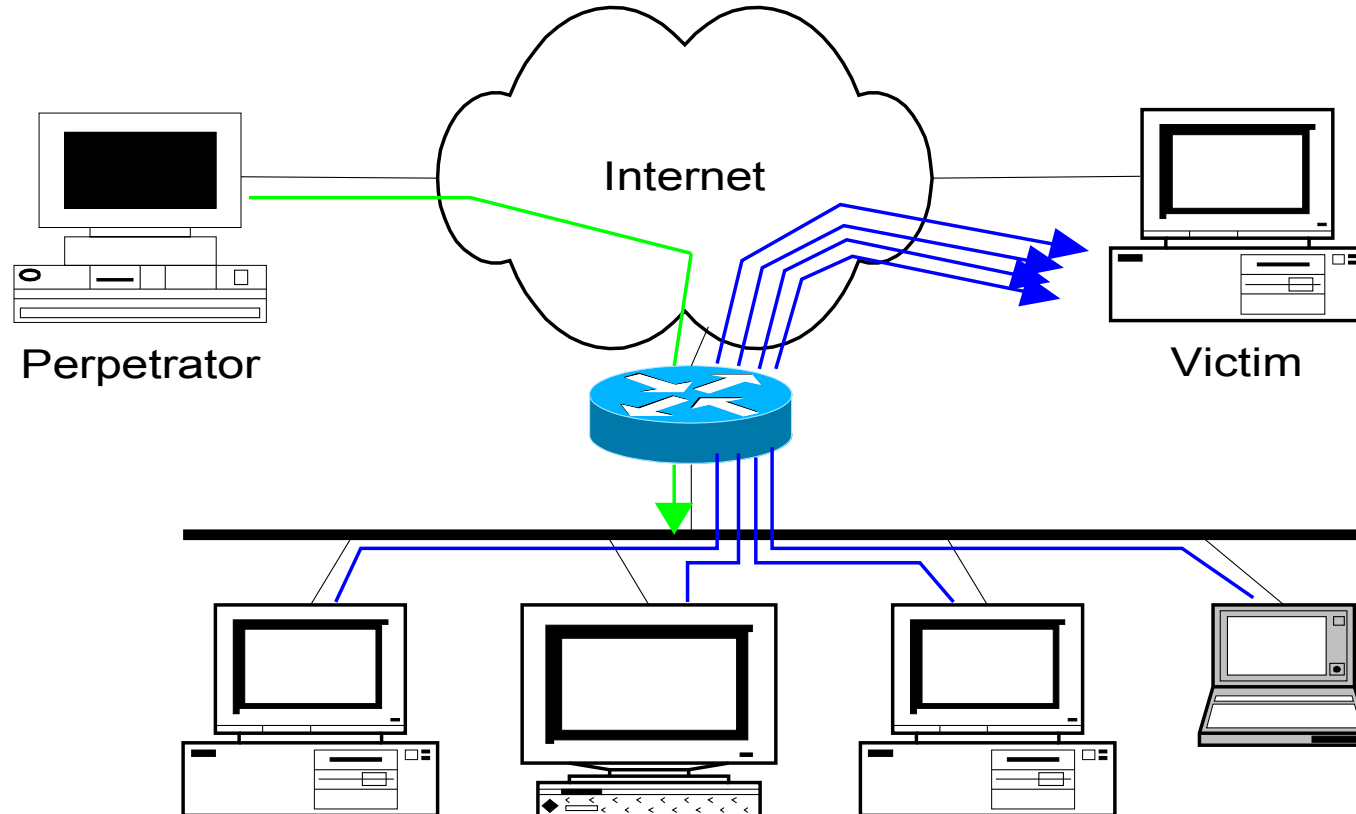
- Used for diagnostics
  - Destination unreachable
  - Time exceeded, TTL hit 0
  - Parameter problem, bad header field
  - Source quench, throttling mechanism rarely used
  - Redirect, feedback on potential bad route
  - Echo Request and Echo reply, ping
  - Timestamp request and Timestamp reply, performance ping
  - Packet too big
- Can use information to help map out a network
  - Some people block ICMP from outside domain

# Smurf Attack

- An amplification DoS attack
  - A relatively small amount of information sent is expanded to a large amount of data
- Send ICMP echo request to IP broadcast addresses. Spoof the victim's address as the source
- The echo request receivers dutifully send echo replies to the victim overwhelming it
- Fraggle is a UDP variant of the same attack

# “Smurf”

- ICMP echo (spoofed source address of victim)  
Sent to IP broadcast address
- ICMP echo reply





# Transport Level – TCP and UDP

- Service to service communication.
  - Multiple conversations possible between same pair of computers
- Transport flows are defined by source and destination ports
- Applications are associated with ports (generally just destination ports)
  - IANA organizes port assignments <http://www.iana.org/>
- Source ports often dynamically selected
  - Ports under 1024 are considered well-known ports
  - Would not expect source ports to come from the well-known range

# Reconnaissance

- Port scanning
  - Send probes to all ports on the target
  - See which ones respond
- Application fingerprinting
  - Analyze the data returned
  - Determine type of application, version, basic configuration
  - Traffic answering from port 8080 is HTTP, Apache or Subversion

# Datagram Transport

- User Datagram Protocol (UDP)
  - A best-effort delivery, no guarantee, no ACK
  - Lower overhead than TCP
  - Good for best-effort traffic like periodic updates
  - No long lived connection overhead on the endpoints
- Some folks implement their own reliable protocol over UDP to get “better performance” or “less overhead” than TCP
  - Such efforts don’t generally pan out
- TFTP and DNS protocols use UDP
- Data channels of some multimedia protocols, e.g., H.323 also use UDP

# UDP Header

Source Port	Destination Port
UDP Length	UDP checksum

# DHCP

- Built on older BOOTP protocol (which was built on even older RARP protocol)
  - Used by diskless Suns
- Enables dynamic allocation of IP address and related information
- Runs over UDP
- No security considered in the design, obvious problems
  - Bogus DHCP servers handing out addresses of attackers choice
  - Bogus clients grabbing addresses
- IETF attempted to add DHCP authentication but rather late in the game to do this.
- Other solutions
  - Physically secure networks
  - Use IPSec

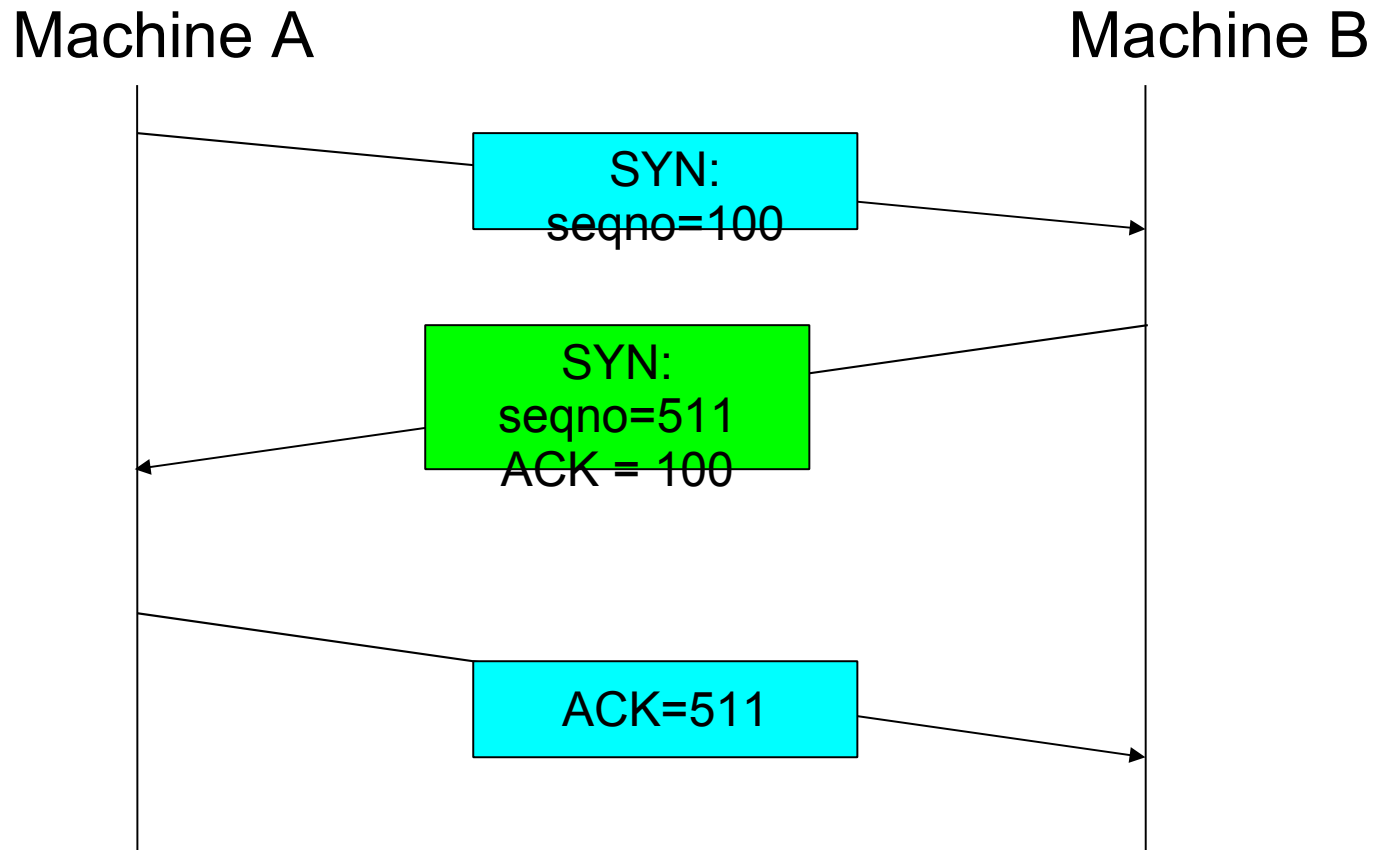
# Reliable Streams

- Transmission Control Protocol (TCP)
  - Guarantees reliable, ordered stream of traffic
  - Such guarantees impose overhead
  - A fair amount of state is required on both ends
- Most Internet protocols use TCP, e.g., HTTP, FTP, SSH, H.323 control channels

# TCP Header

Source Port				Destination Port					
Sequence Number									
Acknowledgement number									
HDR Len		U R G	A C K	P S H	R S T	S Y N	F I N	Window Size	
Checksum				Urgent Pointer					
Options (0 or more words)									

# Three way handshake





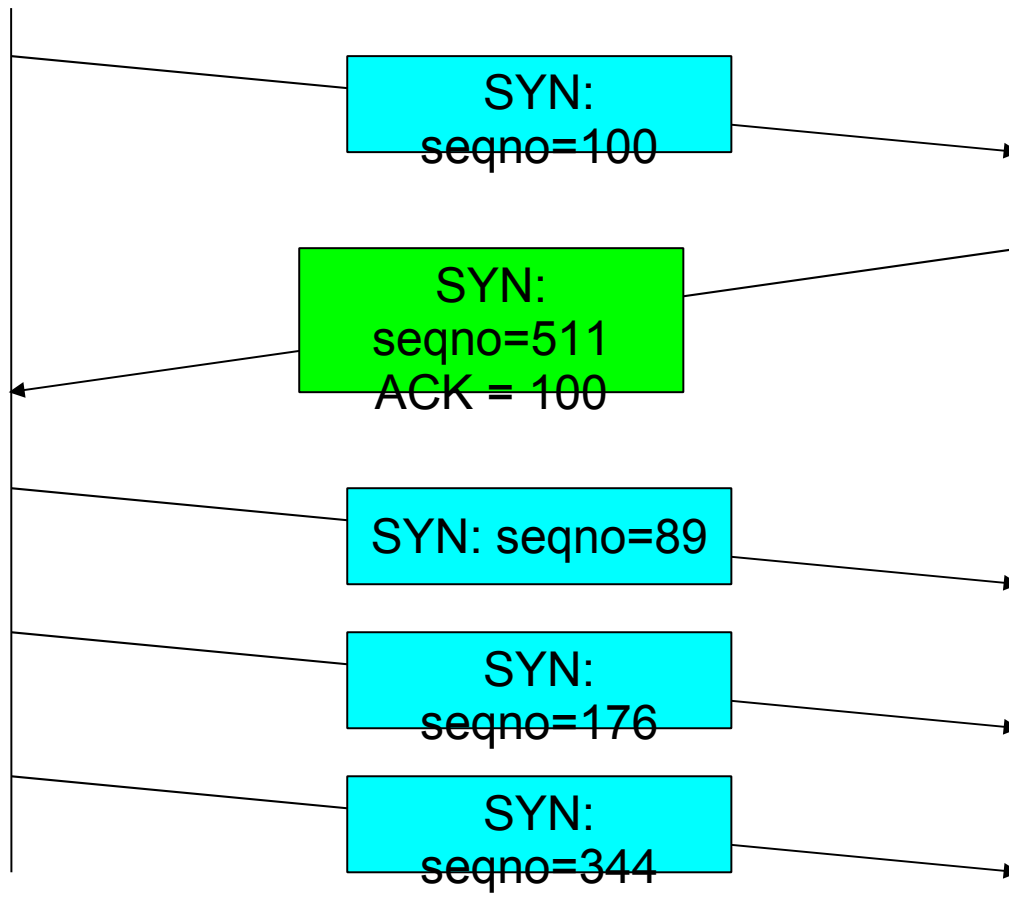
# Syn flood

- A resource DoS attack focused on the TCP three-way handshake
- Say A wants to set up a TCP connection to B
  1. A sends SYN with its sequence number X
  2. B replies with its own SYN and sequence number Y and an ACK of A's sequence number X
  3. A sends data with its sequence number X and ACK's B's sequence number Y
- Send many of the first message to B. Never respond to the second message.
  - This leaves B with a bunch of half open (or embryonic) connections that are filling up memory
  - Firewalls adapted by setting limits on the number of such half open connections.

# SYN Flood

Machine A

Machine B

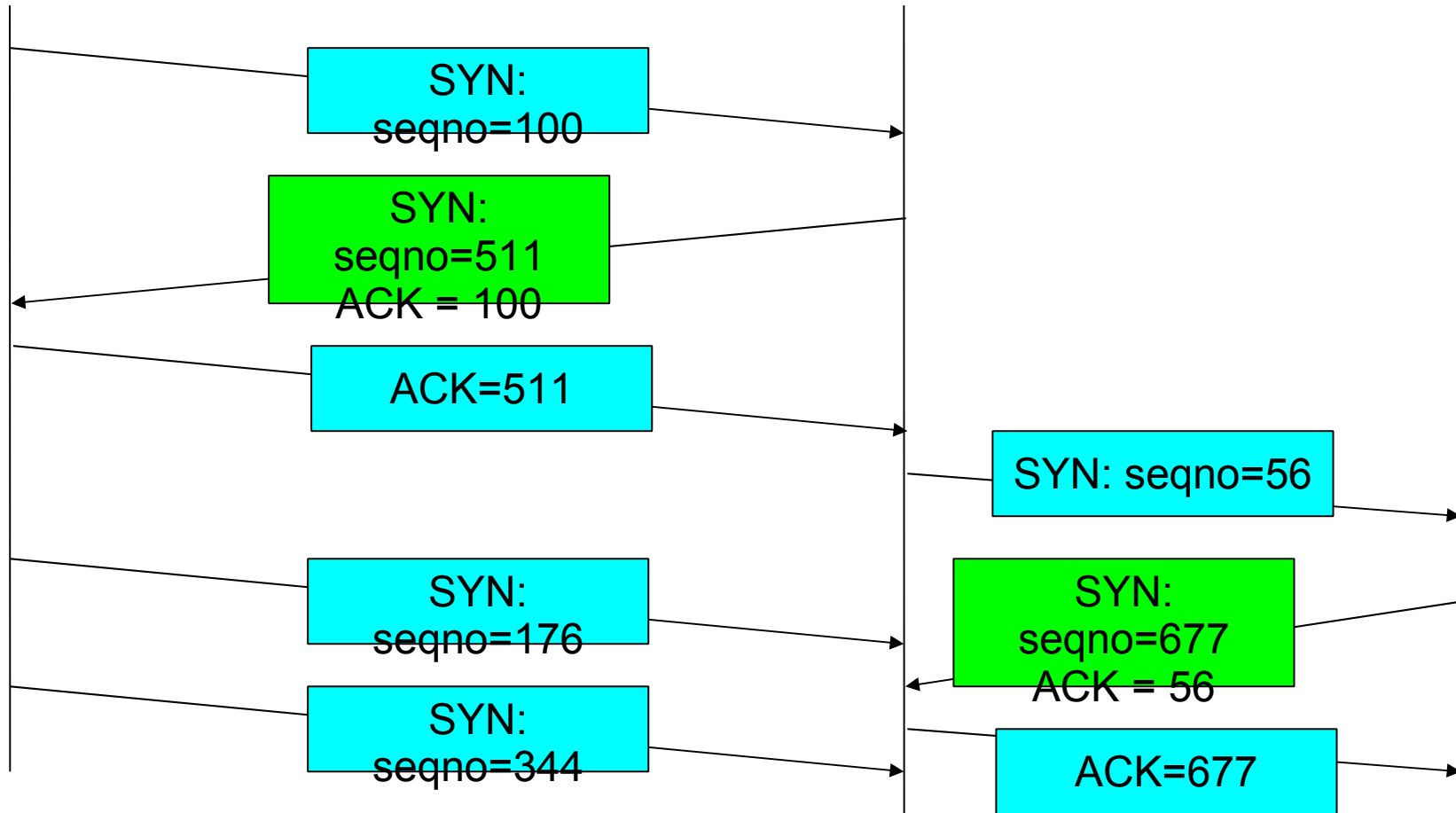


# SYN Flood Constrainer

Machine A

FW

Machine B



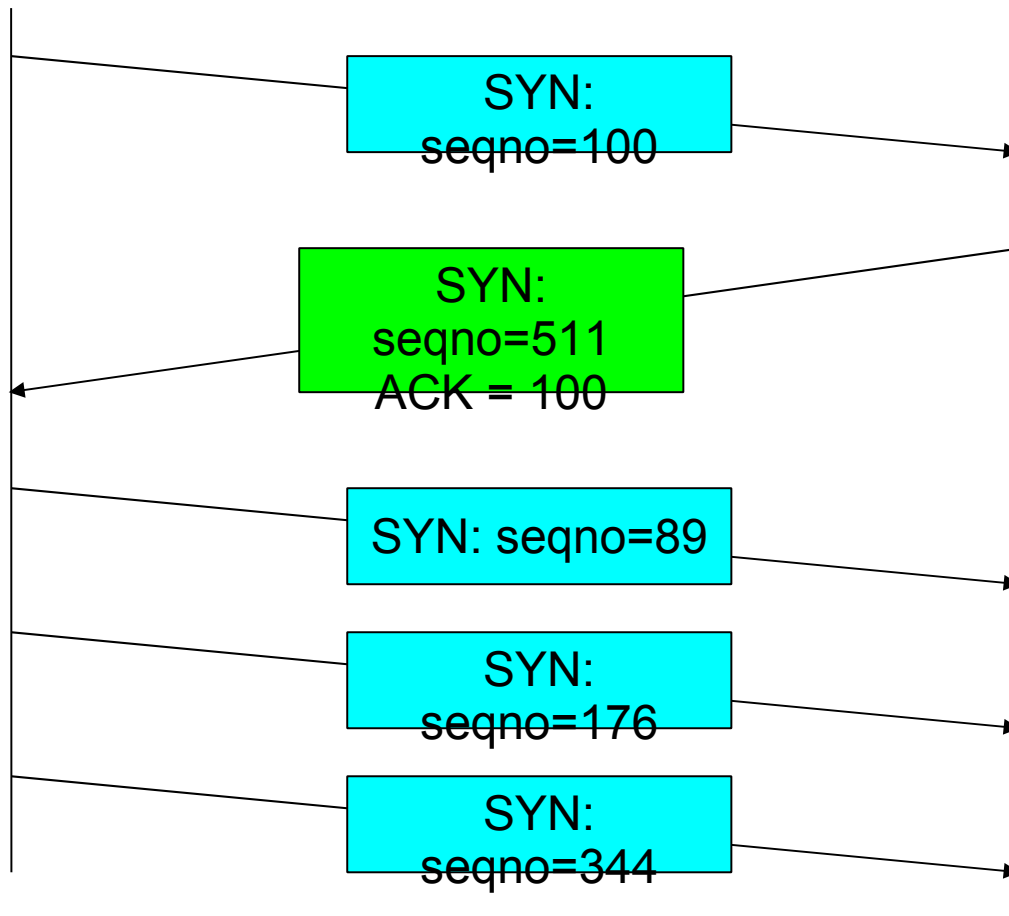
# Another Syn Flood solution: SYN cookie

- Encode information in the sequence number, so receiver does not need to save anything for half open connection
  - $t = \text{counter}$  ,  $m = \text{MSS}$ ,  $s = \text{crypto function}$  computed over IP addresses and server port and  $t$  (24 bits)
  - $\text{Seqno} = (t \bmod 32) \parallel m \text{ encoded in 3 bits} \parallel s$  (24 bits)
- On receiving ACK, get original seqno by subtracting 1
  - Check 1 to verify timeout
  - Recompute  $s$  to verify addresses and ports

# SYN Flood

Machine A

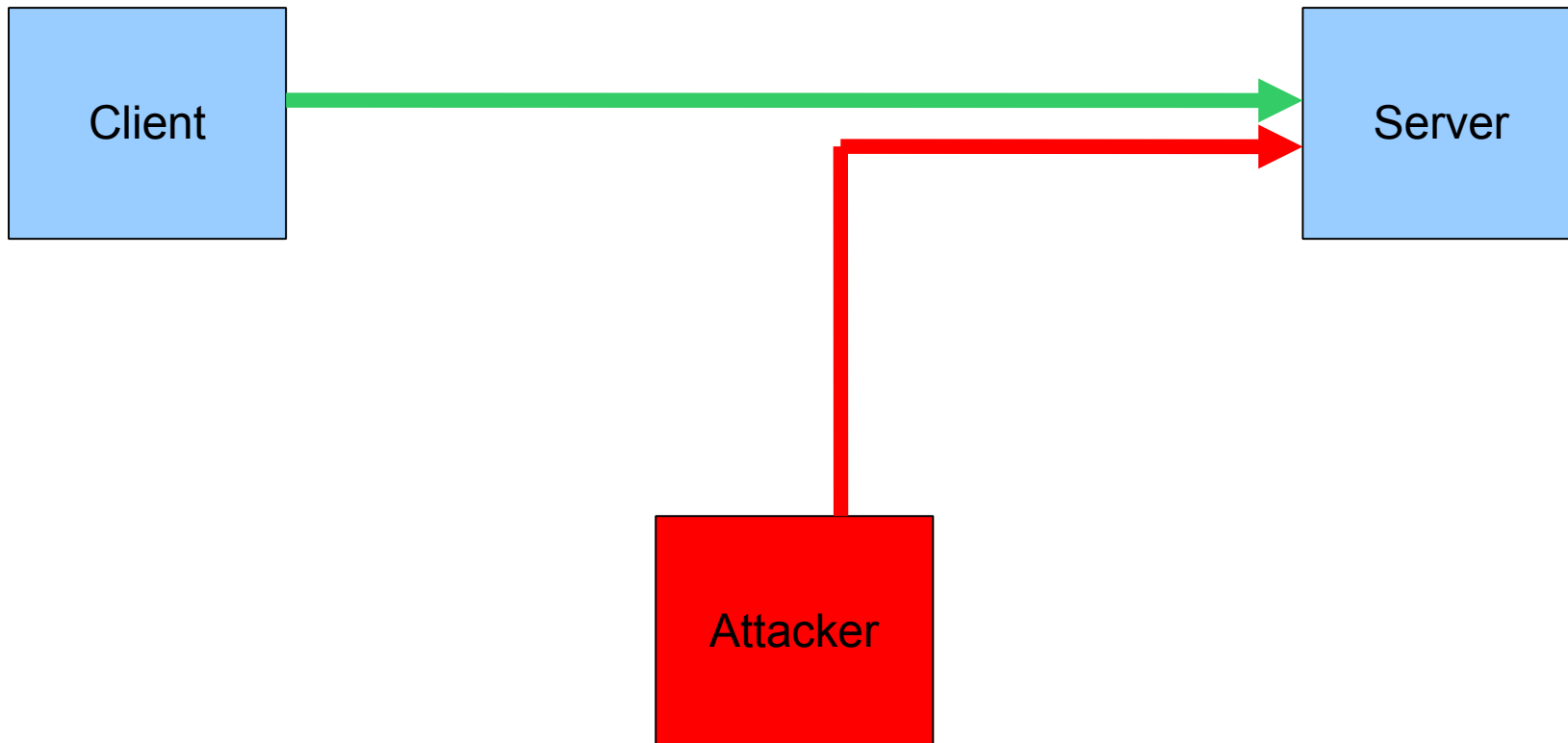
Machine B



# Session Hijacking

- Take over a session after the 3 way handshake is performed
  - After initial authentication too
- Local
  - Can see all traffic.
  - Simply inject traffic at a near future sequence number
- Blind
  - Cannot see traffic
  - Must guess the sequence number

# Session Hijacking



# Application Protocols

- Single connection protocols
  - Use a single connection, e.g. HTTP, SMTP
- Dynamic Multi-connection Protocols, e.g. FTP and H.323
  - Have a well known control channel
  - Negotiate ports and/or addresses on the control channel for subsidiary data channels
  - Dynamically open the negotiated data channels
- Protocol suites, e.g. Netbios and DNS



# Spooftng Applications

- Often times ridiculously easy
- Fake Client
  - Telnet to an SMTP server and enter mail from whoever you want
  - Authenticating email servers
    - Require a password
    - Require a mail download before server takes send requests
- Fake server
  - Phishing: misdirect user to bogus server

# Default Settings

- Many applications installed with default users and passwords
  - Wireless routers, SCADA systems
- Default passwords for many of these systems are easily found on the Internet
  - <http://www.cirt.net/cgi-bin/passwd.pl>

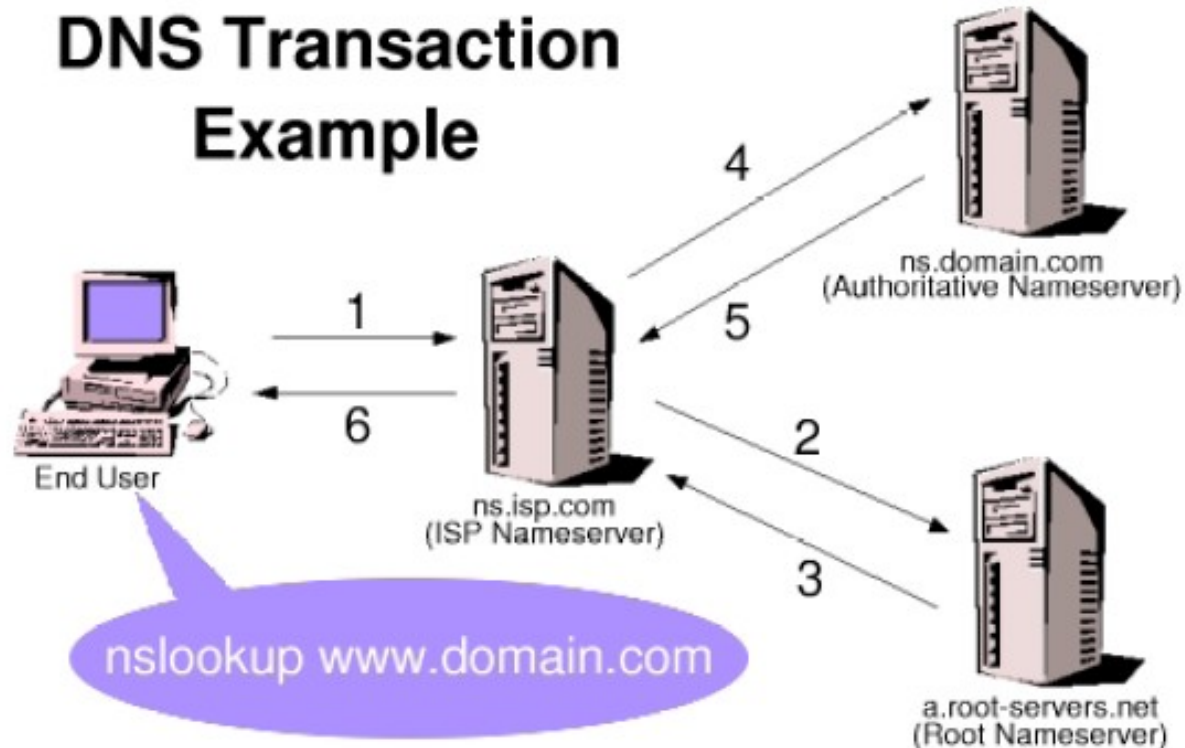
# Domain Name System (DNS)

- Hierarchical service to resolve domain names to IP addresses.
  - The name space is divided into non-overlapping zones
  - E.g., consider shinrich.cs.uiuc.edu.
  - DNS servers in the chain. One for .edu, one for .uiuc.edu, and one for .cs.uiuc.edu
- Can have primary and secondary DNS servers per zone. Use TCP based zone transfer to keep up to date
- Like DHCP, no security designed in
  - But at least the DNS server is not automatically discovered
  - Although this information can be dynamically set via DHCP

# DNS Problems

- DNS Open relays
  - Makes it look like good DNS server is authoritative server to bogus name
  - Enables amplification DoS attack
    - [http://www.us-cert.gov/reading\\_room/DNS-recursion033006.p](http://www.us-cert.gov/reading_room/DNS-recursion033006.p)
- DNS Cache Poisoning
  - Change the name to address mapping to something more desirable to the attacker
    - <http://www.secureworks.com/research/articles/cachepoisoning>
  - Dan Kaminsky raised issue again last summer
    - <http://www.linuxjournal.com/content/understanding-kaminskys>

# DNS Transaction

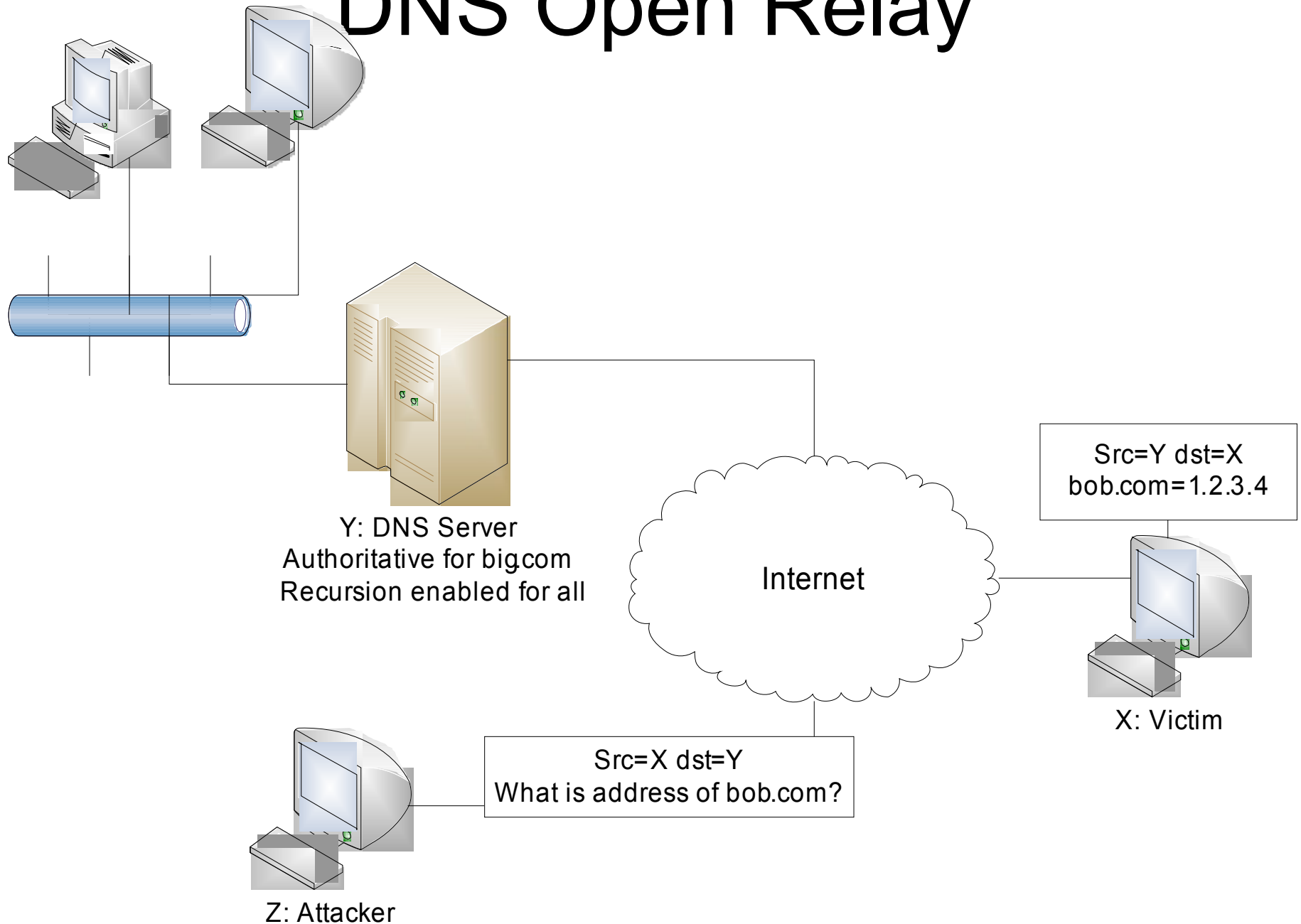


- Step 1 - User asks ISP nameserver to look up the IP address of `www.domain.com`
- Step 2 - ISP nameserver queries root nameserver to find out who is authoritative for `domain.com`
- Step 3 - Root nameserver answers: `ns.domain.com` is authoritative for `domain.com`
- Step 4 - ISP nameserver queries `ns.domain.com` for IP address of `www.domain.com`
- Step 5 - `ns.domain.com` answers "`www.domain.com` is at 1.2.3.4"
- Step 6 - ISP nameserver sends reply to user - "`www.domain.com` is at 1.2.3.4"

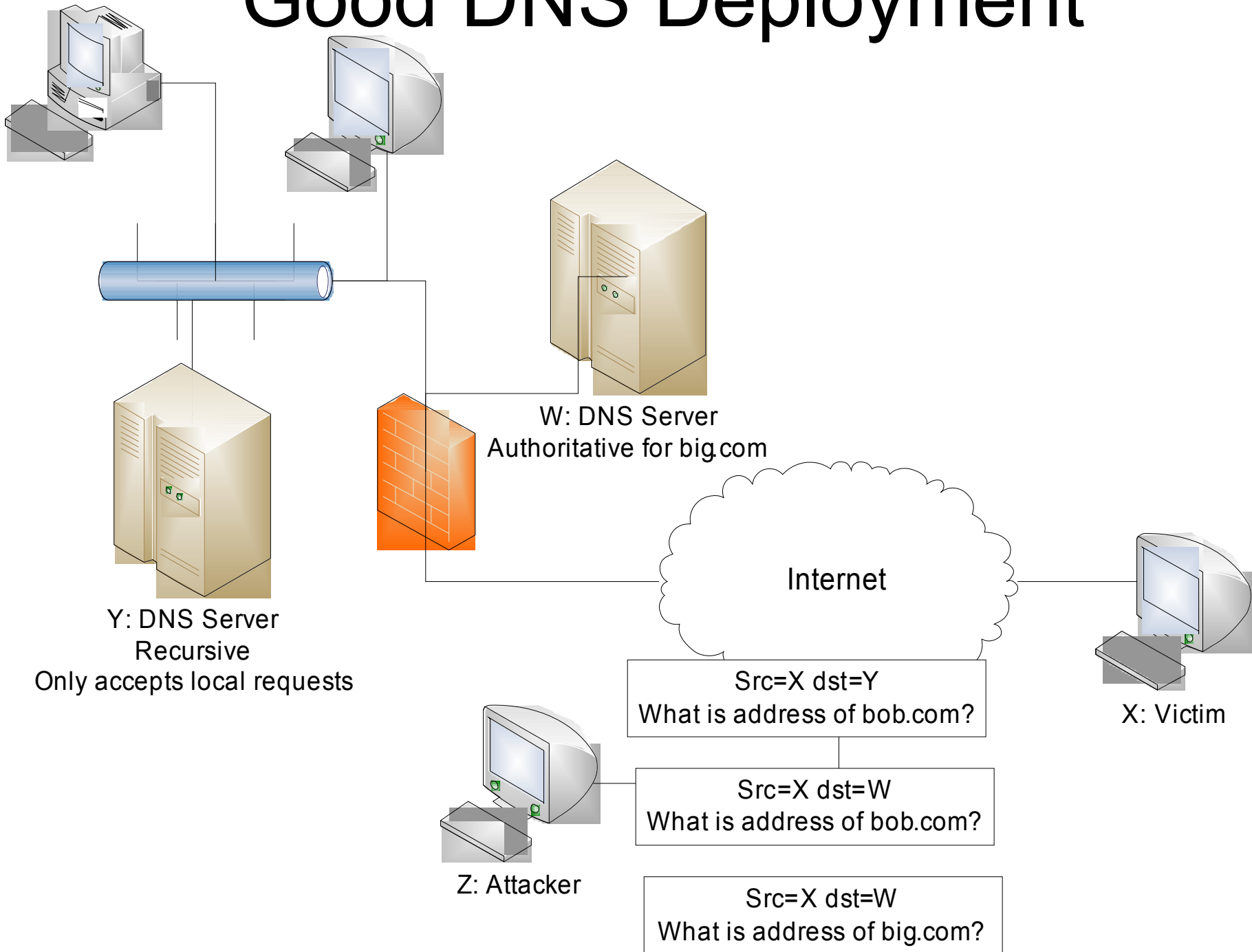
# DNS Communication

- Use UDP
- Requests and responses have matching 16 bit transaction Ids
- Servers can be configured as
  - Authoritative Nameserver
    - Officially responsible for answering requests for a domain
  - Recursive
    - Pass on requests to other authoritative servers
  - Both (this can be the problem)

# DNS Open Relay



# Good DNS Deployment





# DNS Cache Poisoning

- Older implementations would just accept additional information in a reply
  - e.g. A false authoritative name server
  - Fixed by bailiwick checking. Additional records only include entries from the requested domain
- Now to spoof a reply must anticipate the correct transaction ID
  - Only 16 bits
  - Random selection of ID isn't always the greatest

# Bailiwick Checks

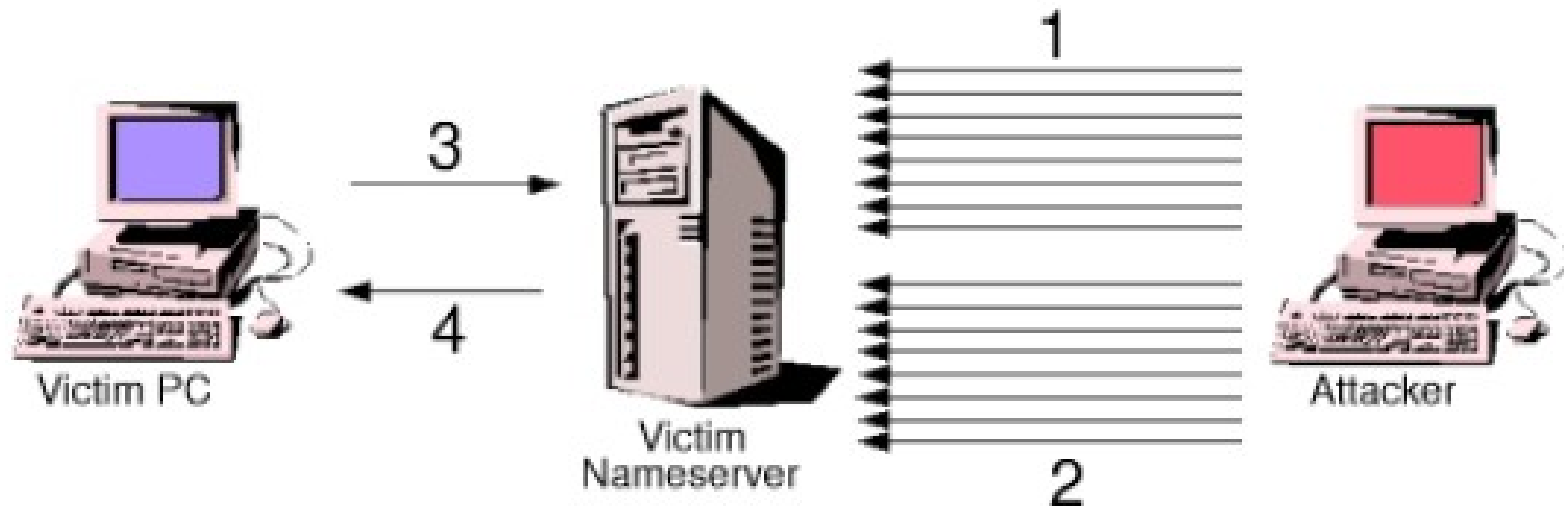
```
$ dig @ns1.example.com www.example.com
;; ANSWER SECTION:
www.example.com. 120 IN A 192.168.1.10

;; AUTHORITY SECTION:
example.com. 86400 IN NS ns1.example.com.
example.com. 86400 IN NS ns2.example.com.

;; ADDITIONAL SECTION:
ns1.example.com. 604800 IN A 192.168.2.20
ns2.example.com. 604800 IN A 192.168.3.30
www.linuxjournal.com. 43200 IN A 66.240.243.113
```

# Tricking the Transaction ID's

## The BIND Birthday Attack



Step 1 - Attacker sends a large number of queries to the victim nameserver, all for the same domain name

Step 2 - Attacker sends spoofed replies giving fake answers for the queries it made

Step 3 - At a later time, victim PC sends a request for the spoofed domain name

Step 4 - Victim nameserver returns fake information to victim PC

# Kaminsky's Observations

- Most implementations don't randomize source ports (making the TID collision more likely)
- Try to poison through the additional information (side stepping the bailiwick check)

```
$ dig doesnotexist.example.com
;; ANSWER SECTION:
doesnotexist.example.com. 120 IN A 10.10.10.10

;; AUTHORITY SECTION:
example.com. 86400 IN NS www.example.com.

;; ADDITIONAL SECTION:
www.example.com. 604800 IN A 10.10.10.20
```

# DNSSEC

- Seeks to solve the trust issues of DNS
- Uses a key hierarchy for verification
- Has been under development for over a decade and still not really deployed
  - This year articles say root servers for .edu, .org, and .com will be deployed in 2010, 2011 timeframe.
- Provides authentication, not confidentiality
- DNS Threat Analysis in RFC 3833.

# Key Points

- Network is complex and critical
- Many flaws have been simple implementation problems
- Poor configuration also can cause widespread problems
- Other guys problems can affect me
- Next, what can you do about it?