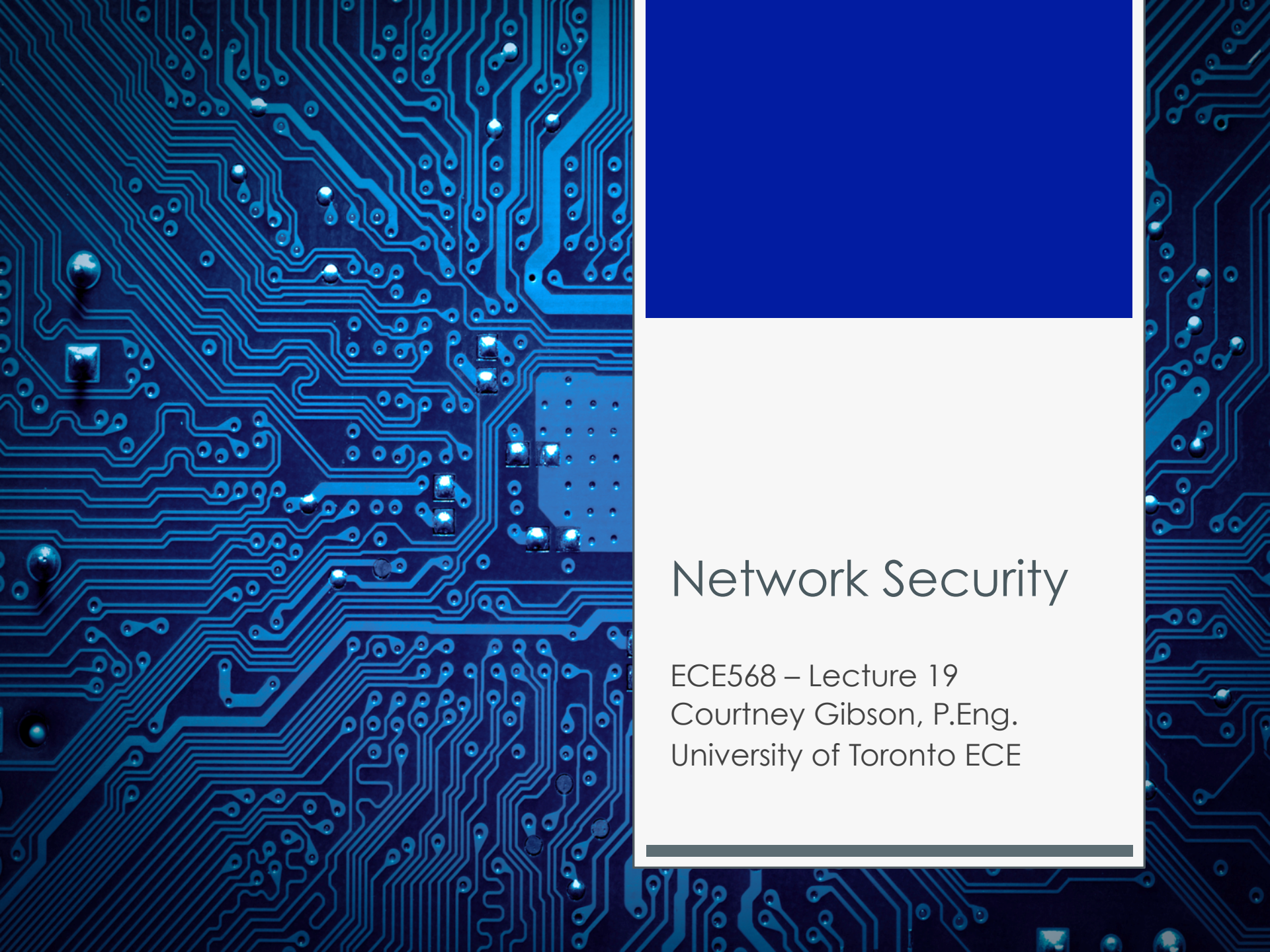


Network Security

ECE568 – Lecture 19

Courtney Gibson, P.Eng.
University of Toronto ECE

Outline

Overview of Internet protocols

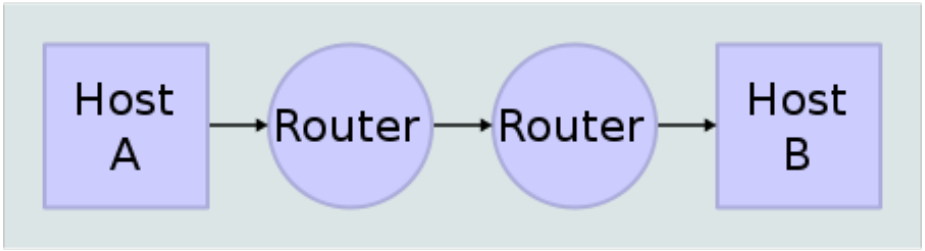
Common attacks on Internet protocols

Defenses

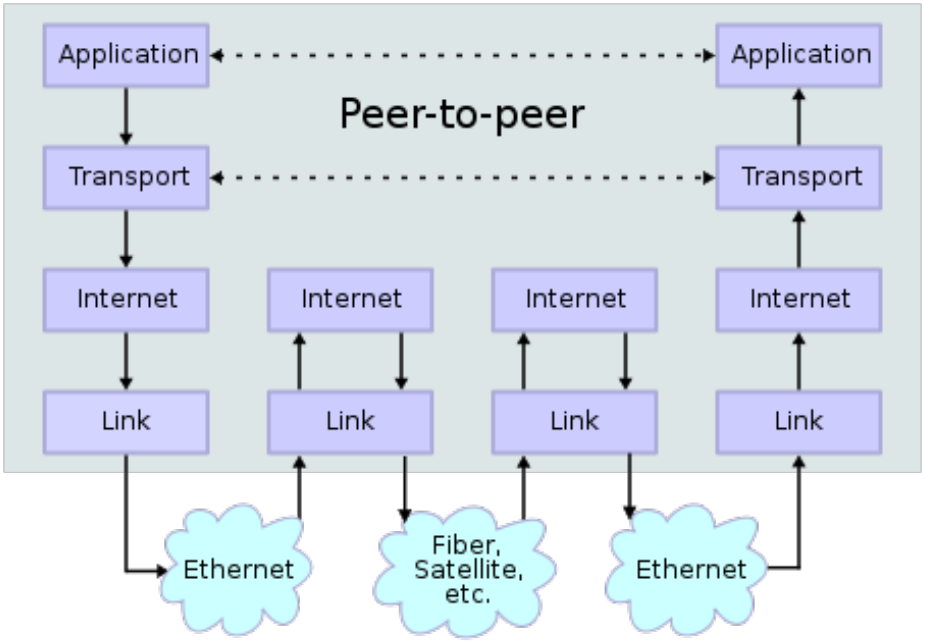
- Cryptographic Protocols
  - IPSec
- Firewalls
  - IPTables
  - DMZ deployment

Overview of Internet Protocols

Network Connections



Stack Connections



DNS, BGP, SMTP, NTP

TCP, UDP

IP, ICMP, IPSec

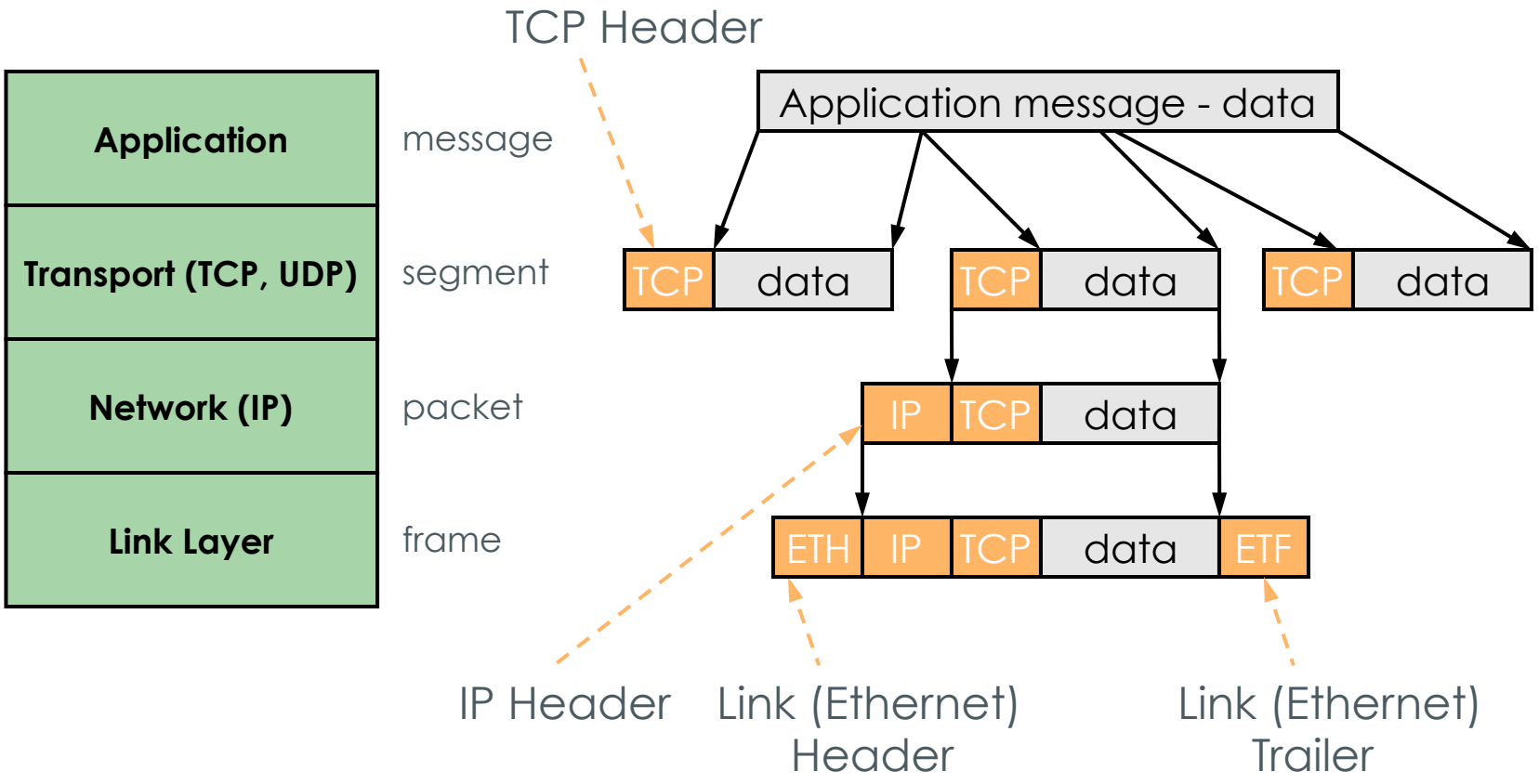
ARP

Internet Protocols

Some of the key protocols of the Internet architecture are (from lowest to highest layers):

- The **Address Resolution Protocol** (ARP) is used for MAC address discovery
  - *e.g.*, 192.168.1.2 -> aa:bb:cc:dd:ee:ff
- The **Internet Control Message Protocol** (ICMP) is used by the IP layer to send error messages to the source host
  - *e.g.*, destination host not available, TTL exceeded, etc.
- The **TCP/IP** protocol for routing packets
  - Transmits data between two connected endpoints
- The **Border Gateway Protocol** (BGP) for route discovery
  - Updates routing information at the intermediate routers
- The **Domain Name System** (DNS) for IP address discovery
  - *e.g.*, www.website.com ➔123.45.67.89

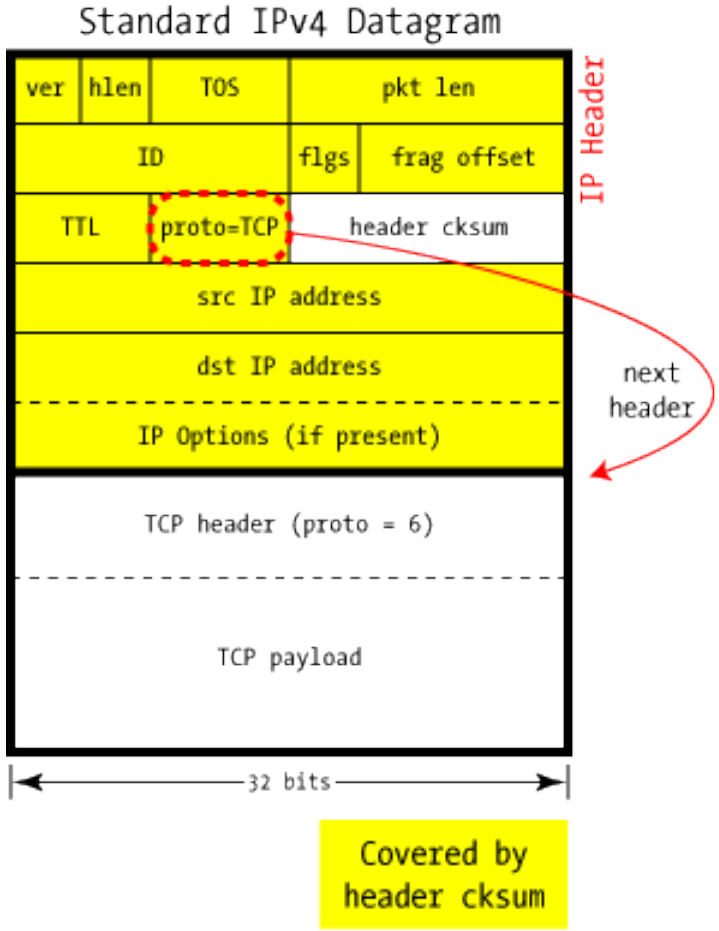
Protocol Packet Formats

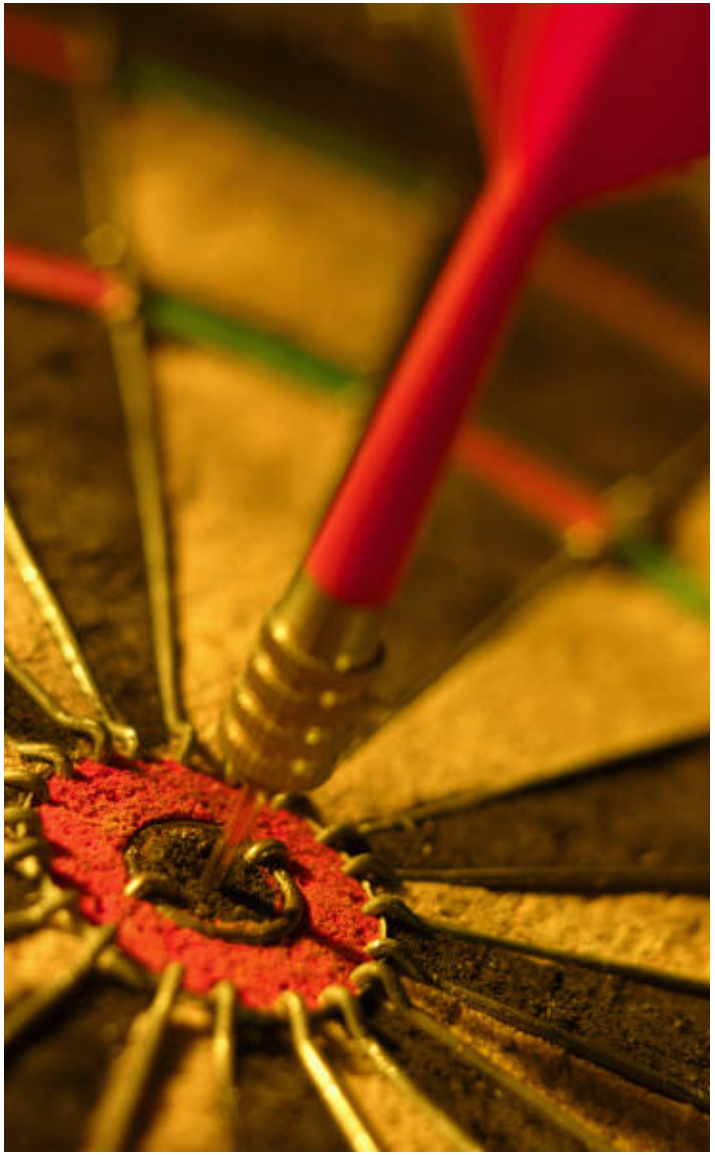


IP Packet Format

The OS creates IP packets, and sets the source and destination addresses in each packet

- Unfortunately, most Internet Protocols trust IP addresses even though they can be easily changed





Common Attacks

Security and Internet Protocols

Many of our Internet protocols were designed assuming that all parties with access to the Internet were trusted

- There was almost no security, and any checks that were instituted were primarily for finding misconfigured systems, rather than dealing with malicious systems

Example: how secure is the SMTP (email) protocol?

- **Integrity:** Anyone can forge an email address (as well as IP address), or alter email contents during transit
- **Confidentiality:** Anyone on a server between the sender and recipient can read the email, since standard email is sent in plain text
- **Availability:** If an email is dropped, neither sender nor receiver will know, and delivery of email is not guaranteed
- **Conclusion:** Email has pretty poor security

Common Attacks on Internet Protocols

Address Resolution Protocol (ARP)

- ARP spoofing

Internet Control Message Protocol (ICMP)

- Smurf/amplification attack

TCP/IP

- TCP connection spoofing, TCP reset attacks, SYN flooding

Border Gateway Protocol (BGP)

- Bad route announcements

Domain Name System (DNS)

- DNS cache poisoning, DNS rebinding

Distributed denial of service

Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP)

- The IP layer uses **IP addresses** for routing packets
- ARP is used to map IP addresses to **MAC addresses** so that IP packets can be sent to the link layer
- Packets are sent to the next hop using MAC (*e.g.*, Ethernet) addresses

ARP is simple and uses broadcasting:

- A host that wants to send a packet to IP address **A**, it will perform an **ARP broadcast** to all devices asking which device owns that IP address
- All hosts ignore the broadcast except the host that has IP address **A**, who will respond with its MAC address
- All packets sent to IP Address **A** are sent using the given MAC address

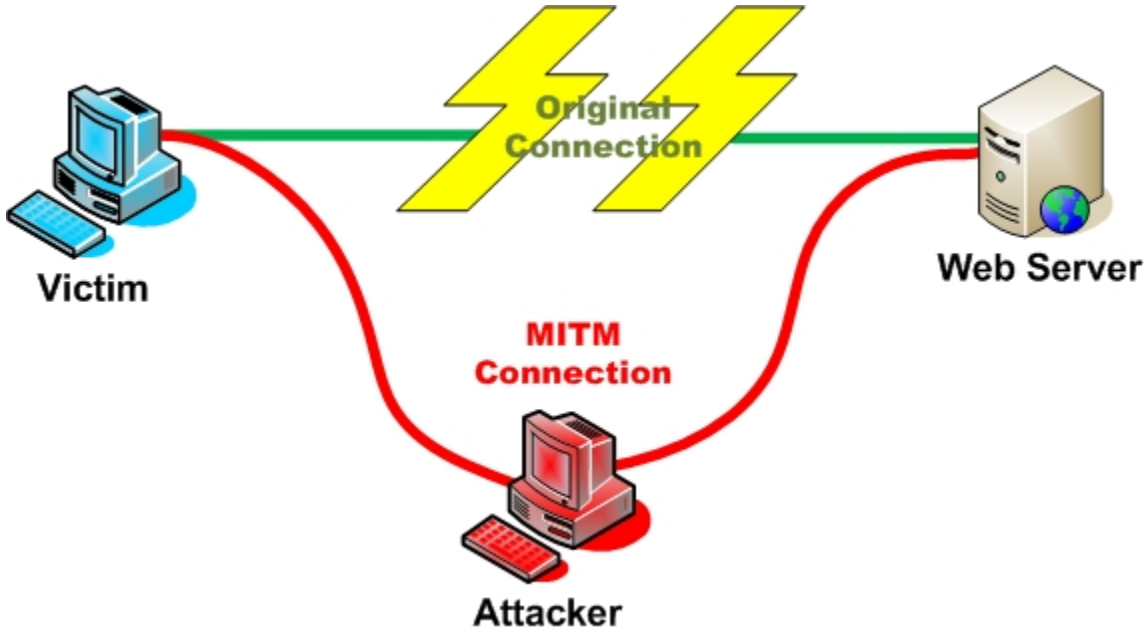
Address Resolution Protocol (ARP)

If an attacker has full access to a host, it is trivial to **spoof ARP requests**

- The attacker makes the hacked machine redirect all traffic to itself by responding to all ARP broadcasts
- Every machine starts to think that the hacked machine owns every IP address
- All LAN traffic is redirected to hacked machine which can start faking services, stealing passwords, etc.

- ARP broadcasts are never forwarded outside of a subnet, so the attacker must control a machine on a subnet

ARP Posioning



Reference: http://www.art0.org/security/man-in-the-middle-attacks-with-ettercap

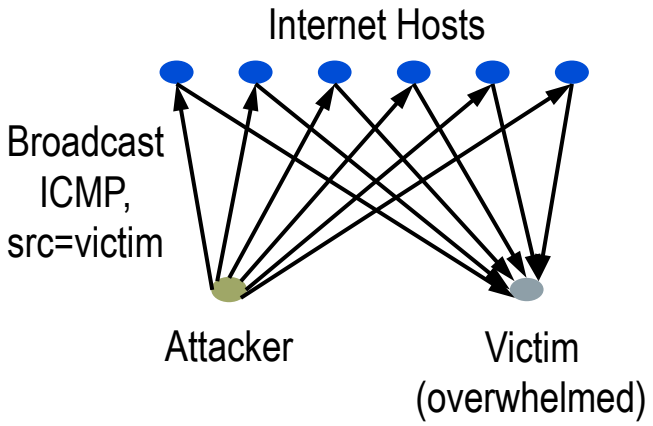
ICMP "Smurf" Attack

A **Smurf attack** generates lots of network traffic, by flooding a victim machine with ICMP packets from many different machines

- An attacker sends a ping stream (ICMP echo requests) to an IP broadcast address with a **spoofed source IP address** of the victim machine
- Every host on target network will generate ping replies (ICMP echo replies) to victim, potentially overloading the victim

Defenses:

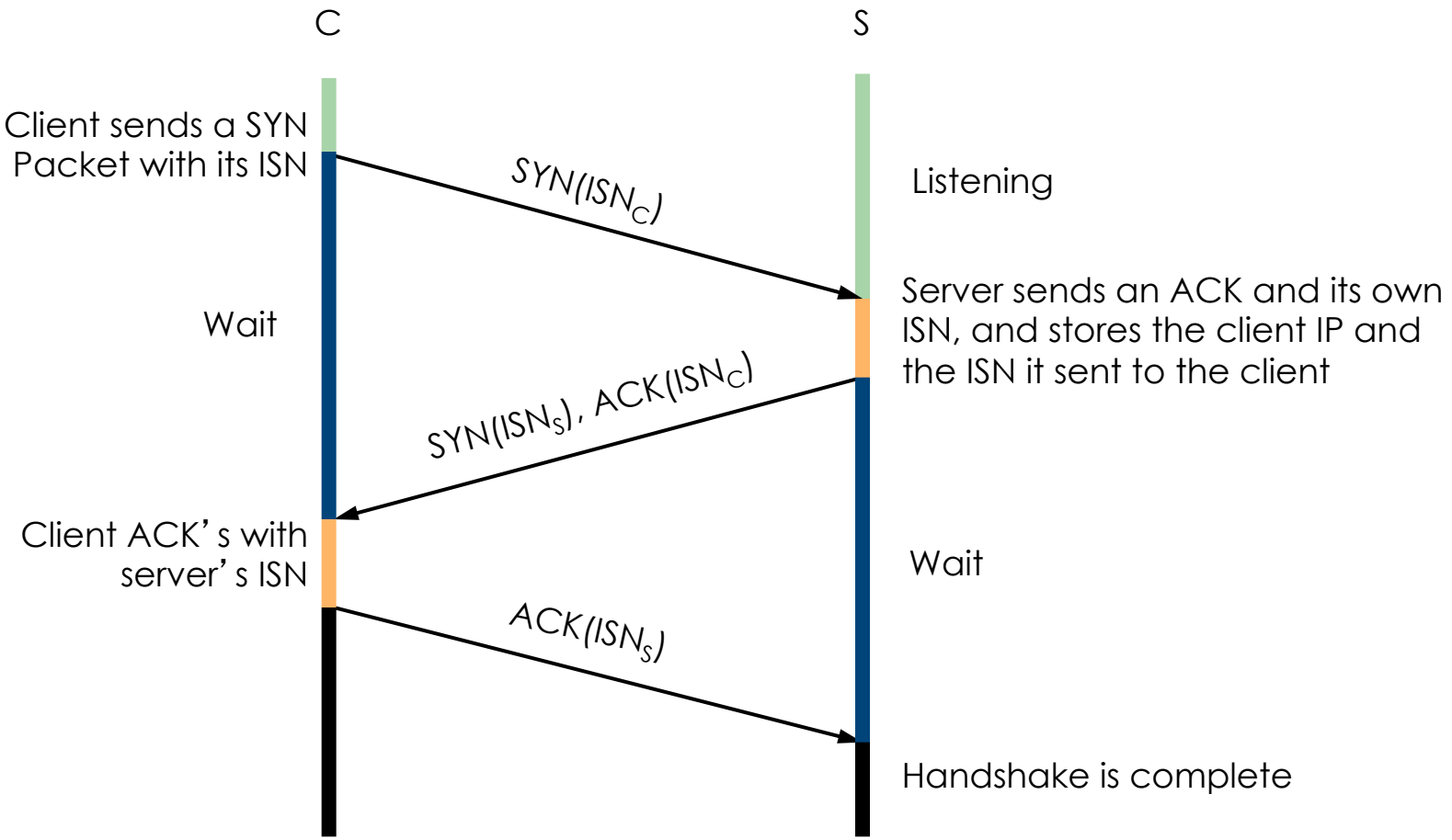
- At Host
  - Disable response to ping broadcast
- At router/switch
  - Disable broadcast forwarding



ICMP “Smurf” Attack

```
cgibson:~ $ ping 10.1.7.255
PING 10.1.7.255 (10.1.7.255): 56 data bytes
64 bytes from 10.1.7.180: icmp_seq=0 ttl=64 time=2.559 ms
64 bytes from 10.1.7.139: icmp_seq=0 ttl=64 time=2.758 ms
64 bytes from 10.1.7.189: icmp_seq=0 ttl=64 time=3.159 ms
64 bytes from 10.1.6.11: icmp_seq=0 ttl=64 time=3.288 ms
64 bytes from 10.1.7.171: icmp_seq=0 ttl=64 time=18.701 ms
64 bytes from 10.1.7.63: icmp_seq=0 ttl=64 time=27.878 ms
64 bytes from 10.1.7.34: icmp_seq=0 ttl=64 time=53.648 ms
64 bytes from 10.1.7.135: icmp_seq=0 ttl=64 time=53.935 ms
64 bytes from 10.1.7.168: icmp_seq=0 ttl=64 time=54.141 ms
64 bytes from 10.1.7.106: icmp_seq=0 ttl=64 time=55.466 ms
64 bytes from 10.1.7.134: icmp_seq=0 ttl=64 time=59.504 ms
64 bytes from 10.1.7.80: icmp_seq=0 ttl=64 time=60.968 ms
64 bytes from 10.1.7.154: icmp_seq=0 ttl=64 time=80.753 ms
64 bytes from 10.1.7.78: icmp_seq=0 ttl=64 time=80.787 ms
64 bytes from 10.1.7.39: icmp_seq=0 ttl=64 time=87.924 ms
64 bytes from 10.1.7.30: icmp_seq=0 ttl=64 time=87.954 ms
64 bytes from 10.1.7.184: icmp_seq=0 ttl=64 time=96.053 ms
64 bytes from 10.1.7.71: icmp_seq=0 ttl=64 time=98.109 ms
64 bytes from 10.1.7.28: icmp_seq=0 ttl=64 time=98.414 ms
64 bytes from 10.1.7.111: icmp_seq=0 ttl=64 time=98.755 ms
64 bytes from 10.1.7.86: icmp_seq=0 ttl=64 time=99.642 ms
64 bytes from 10.1.7.102: icmp_seq=0 ttl=64 time=101.124 ms
64 bytes from 10.1.7.93: icmp_seq=0 ttl=64 time=102.220 ms
64 bytes from 10.1.7.157: icmp_seq=0 ttl=64 time=105.747 ms
64 bytes from 10.1.7.16: icmp_seq=0 ttl=64 time=107.874 ms
64 bytes from 10.1.7.114: icmp_seq=0 ttl=64 time=109.684 ms
^C
--- 10.1.7.255 ping statistics ---
1 packets transmitted, 1 packets received, +25 duplicates, 0.0% packet loss
round-trip min/avg/max/stddev = 2.559/67.348/109.684/36.449 ms
cgibson:~ $
```

TCP Handshake



ISN_C = Client Initial Sequence Number

ISN_S = Server Initial Sequence Number

TCP Connection Spoofing

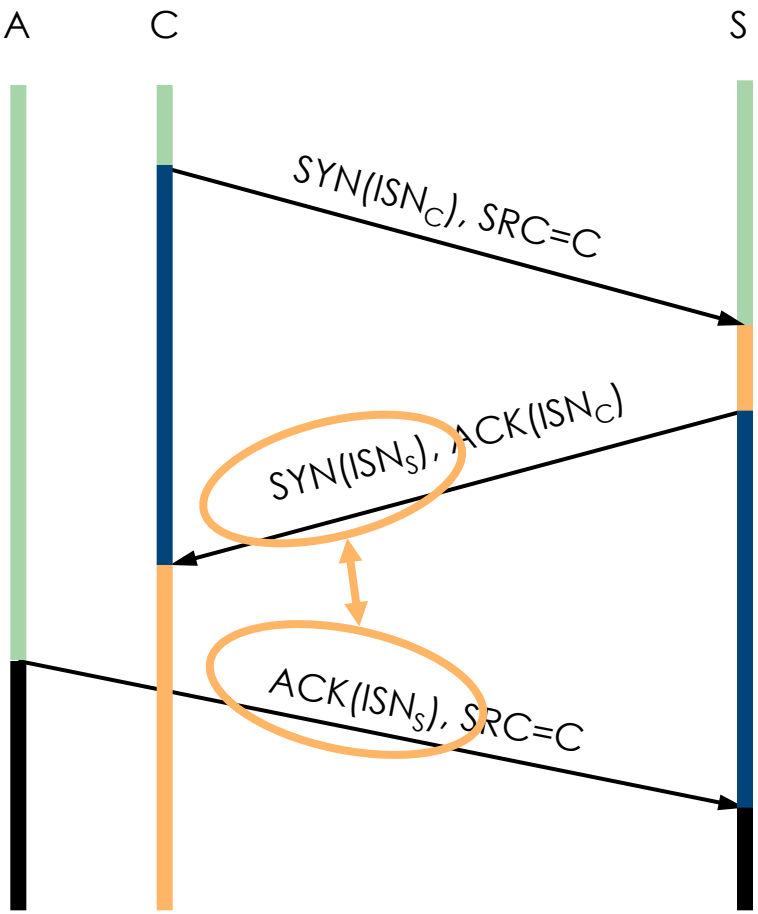
The TCP handshake uses the initial sequence numbers as a weak authenticator

- If these numbers are wrong, the connection is not set up

An attacker can forge a packet with the source IP address set to the client's address

- Forging source IP addresses doesn't allow the attacker to receive packets
- However, if the server's initial sequence number can be guessed, then the attacker can connect to the server as the victim client
- Older systems still use sequential sequence numbers, so these numbers are easy to guess

TCP Connection Spoofing



Here A is able to guess the sequence number ISN_S even though it never sees the value

Now it can inject packets into the connection between C and S

Listening

Server sends an ACK and its own ISN, and stores the client IP and the ISN it sent to the client

Wait

Handshake is complete

ISN_C = Client Initial Sequence Number
ISN_S = Server Initial Sequence Number

TCP Reset Attack

A variant of a TCP connection spoofing attack is a **TCP reset attack**

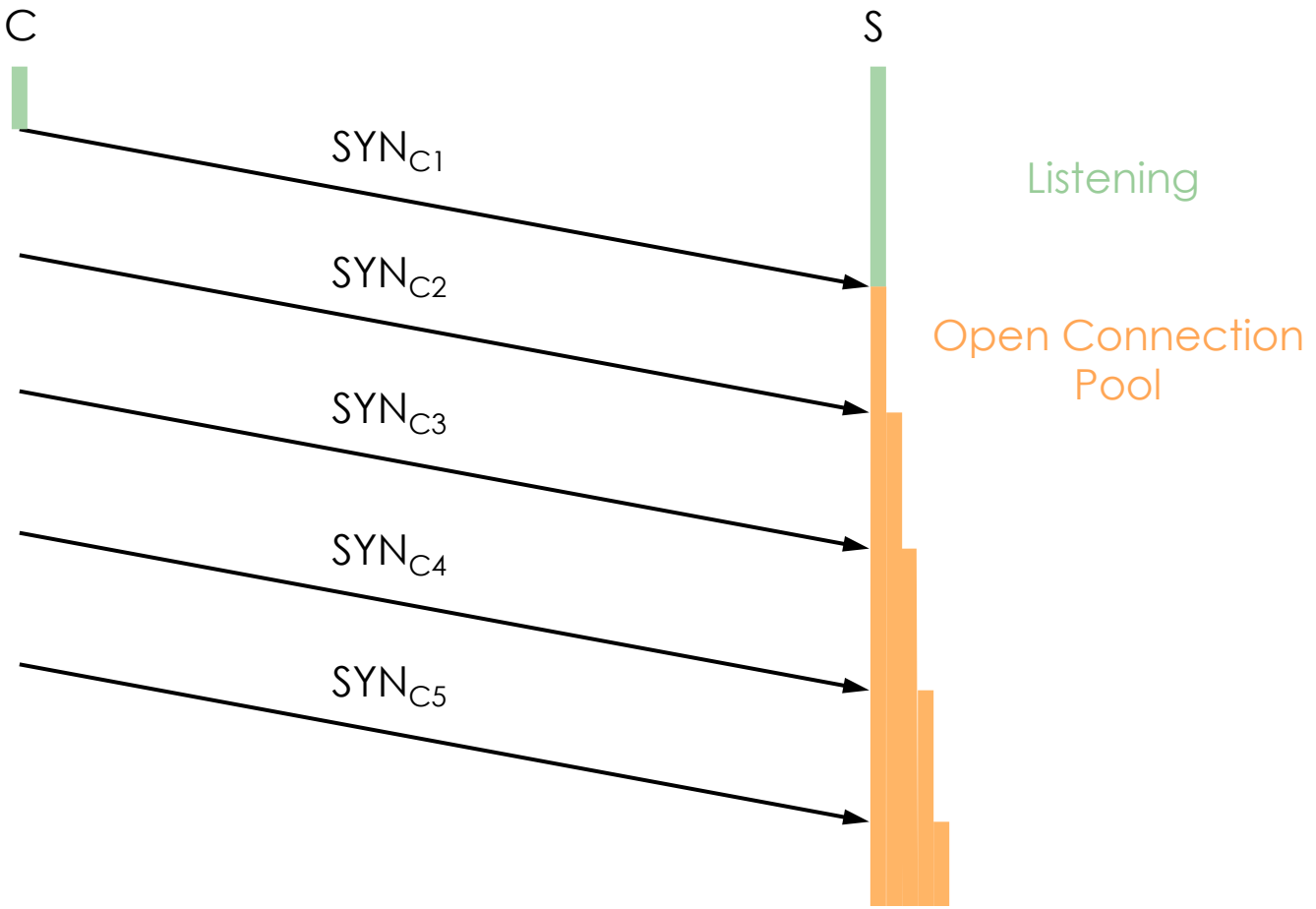
- A TCP reset attack falsely terminates a established TCP connection resulting in denial of service
- The attacker spoofs the sender's connection and sends a RST packet to the receiver
  - Requires IP spoofing and guessing the current packet sequence number
- On receiving a 'valid' RST packet, the receiver immediately terminates the connection
- Defenses are not obvious
  - Requires ignoring bogus RST packets, *e.g.*, multiple packets

TCP SYN Flooding

Attacker sends many connection requests with spoofed IP source addresses

- Victim allocates resources for each request until some timeout
- Typically, OSs have a fixed bound on these **half-open** connections
- Eventually, the half-open connection queue **resource is exhausted**
- Then, no more requests are accepted, leading to denial of service

TCP SYN Flooding



TCP SYN Flooding Defenses

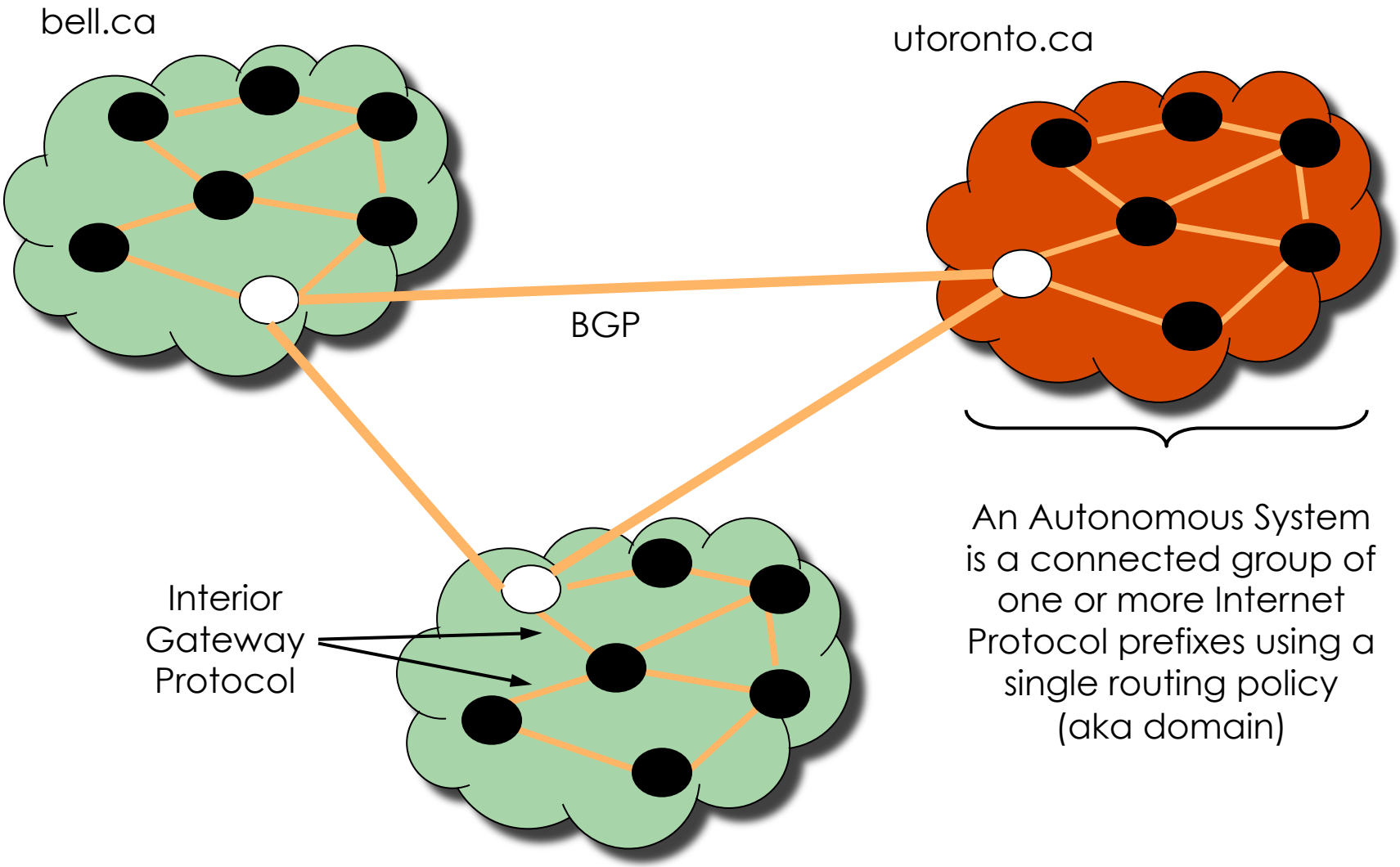
- Reduce half-open connection timeout
- Drop half-open connections randomly
- Use SYN-ACK cookies
  - Client sends SYN
  - Server responds to client with SYN-ACK cookie
    - ISNs = H(src addr, src port, dest addr, dest port, rand)
    - This is a normal response, but server does not save state
  - Honest client responds with ACK(ISNs)
    - No changes required at client
  - Server regenerates ISNs and checks that the client's response matches ISNs
    - rand is derived from a 32-bit time counter
    - Server uses some recent time counter values

Border Gateway Protocol (BGP)

The Internet is broken up into **Autonomous Systems** (AS) that are independently managed and connect to each other via gateways (*e.g.*, an ISP)

- These gateway routers communicate using the **Border Gateway Protocol** (BGP) protocol to update routing information
  - *e.g.*, if a router goes down, BGP detects and updates routing information with alternate routes to destinations
  - Routes are updated in peer-to-peer manner, by asking neighbors if they have a route to a certain destination

Border Gateway Protocol (BGP)



Attacks on BGP

Attackers can compromise the protocol by advertising "good" routes to destinations

- Traffic then gets directed through attacker's gateway

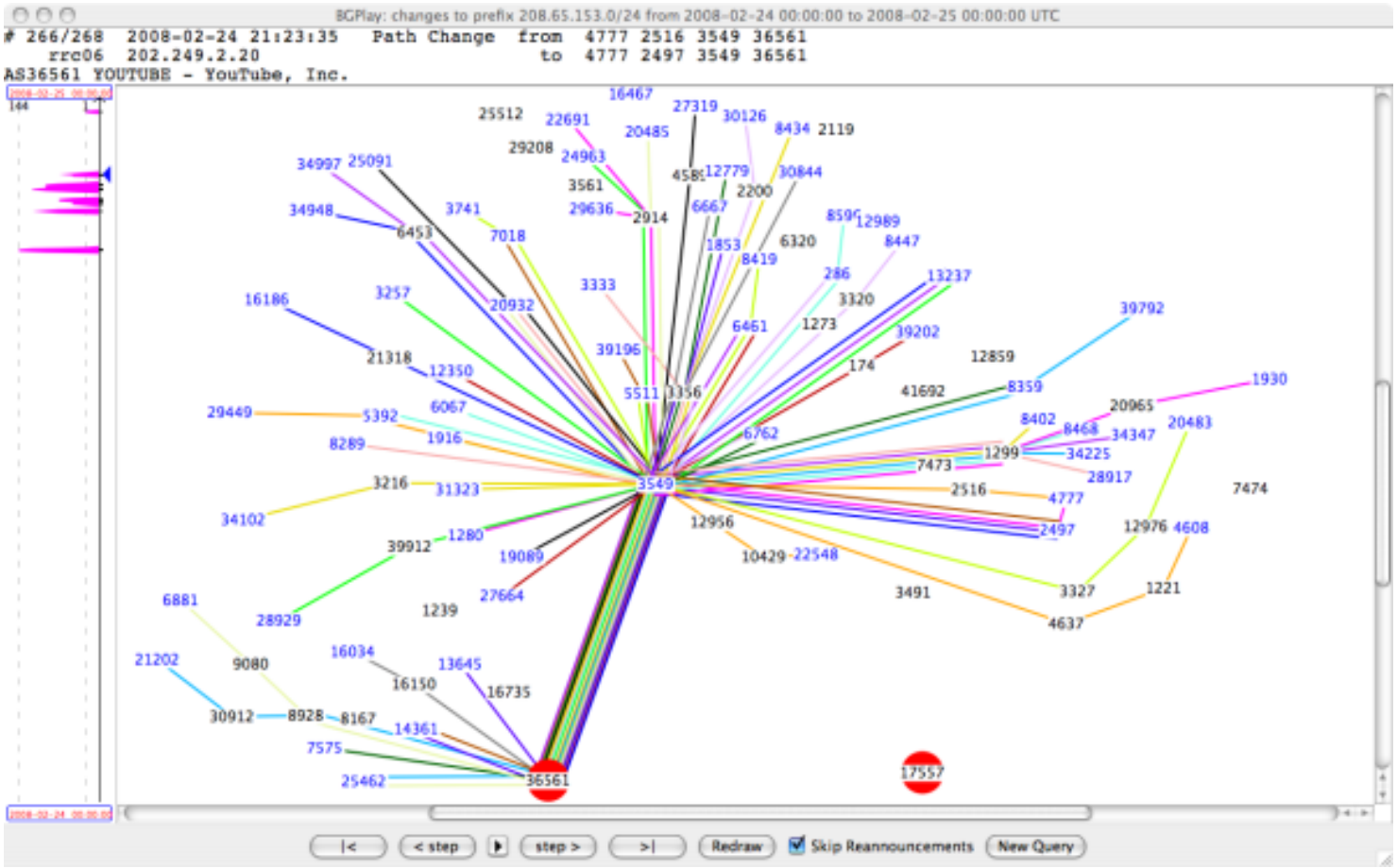
This attack is similar to the ARP attack

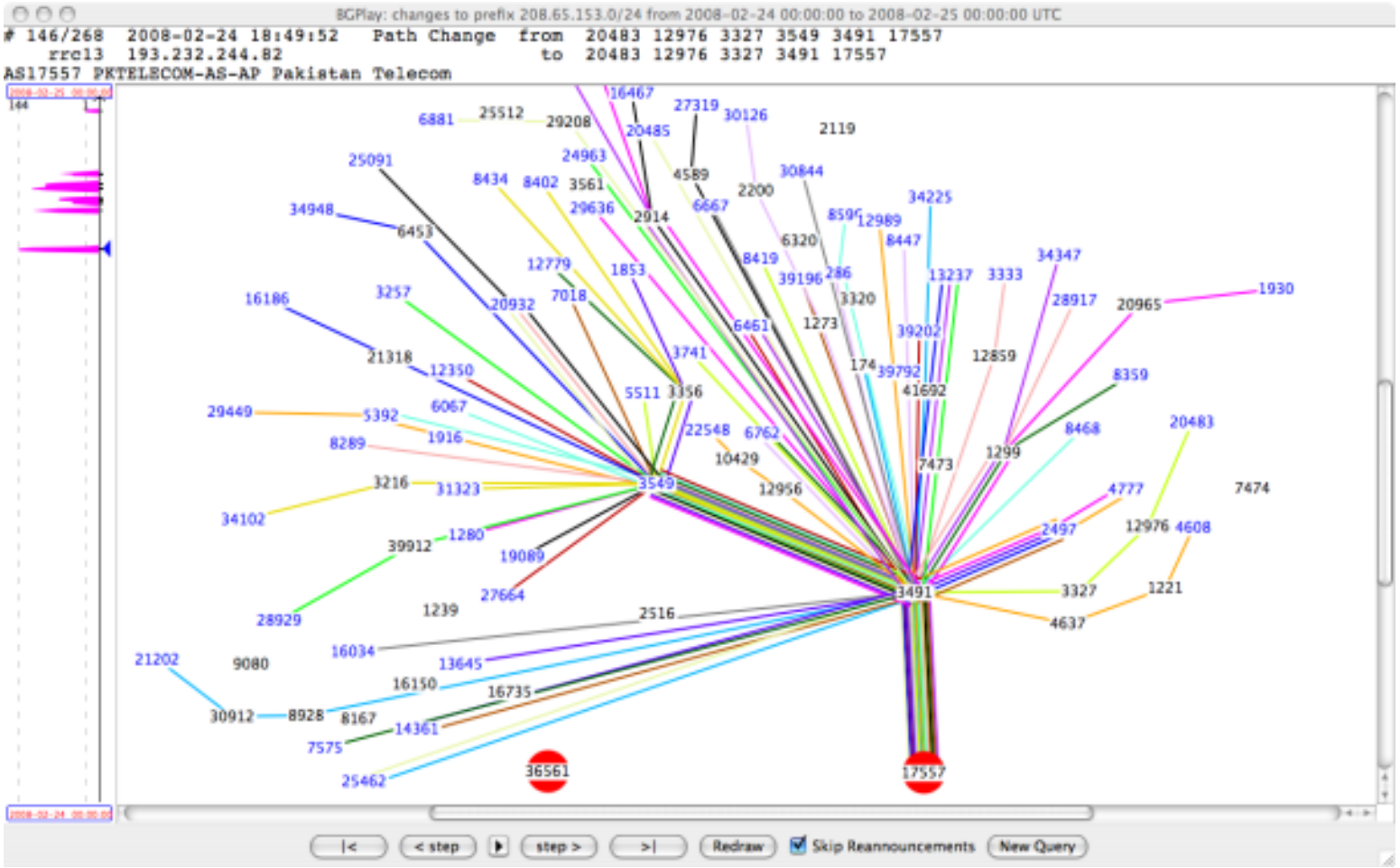
- Routers trust each other and don't authenticate their peers, the BGP packets, or the routes advertised by the peers
- It is assumed that all BGP routers are configured correctly and are not malicious

More information available on course web site

- A Survey of BGP Security

Pakistan Telecom: YouTube Hijack





Reference:

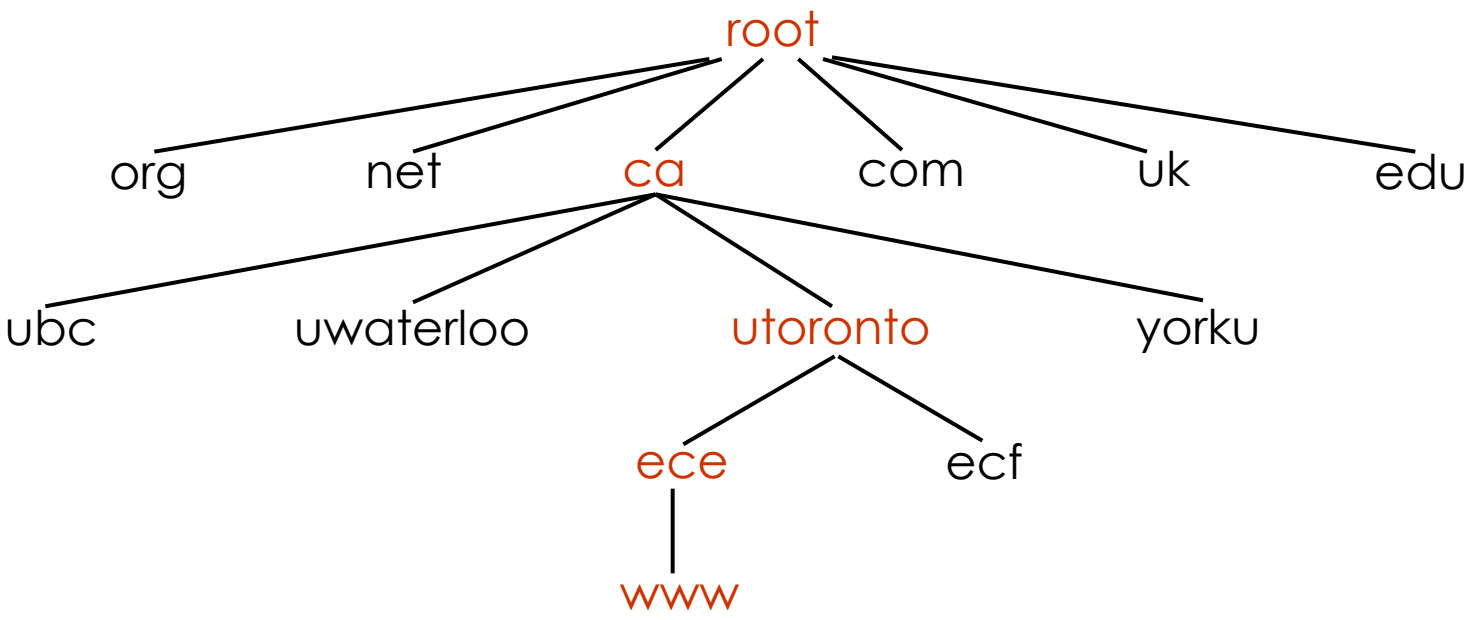
http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study

Domain Name System (DNS)

The Domain Name System (DNS) is a hierarchical naming system for resources on the Internet

- It maps symbolic names to numeric IP addresses

  www.ece.utoronto.ca ↔ 128. 100.131.138



DNS Name Servers

DNS maps names to IP addresses using a set of **authoritative name servers**

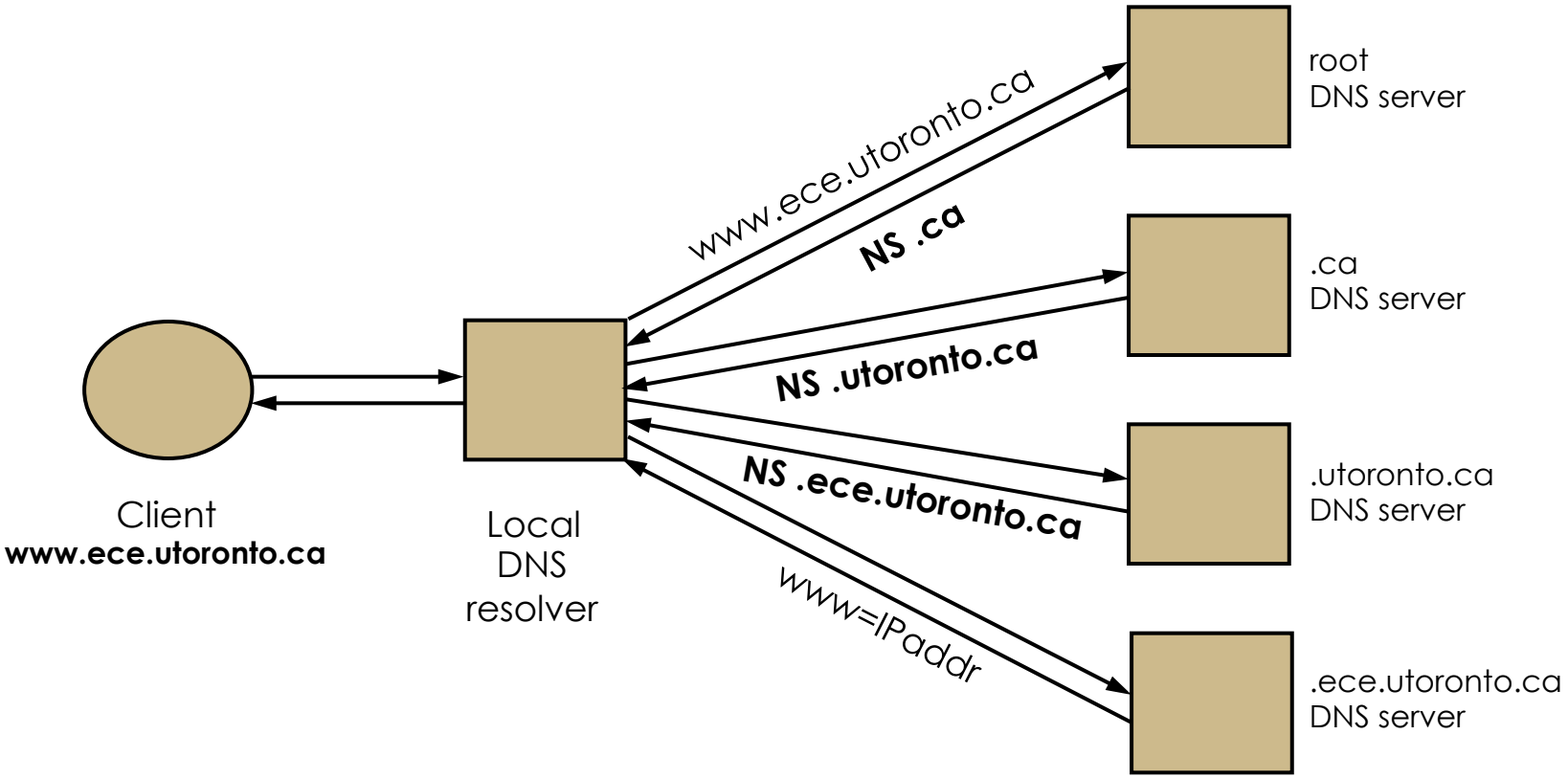
- Each domain has an authoritative name server that is responsible for the DNS mappings for its domain, and in turn can assign other authoritative name servers for their sub-domains
- Example:
  - NS for utoronto.ca is ns1.utoronto.ca
  - NS for ece.utoronto.ca is ugsparc0.eecg.utoronto.ca
- This hierarchy makes DNS distributed, and helps avoid the need for a single central register to be continually consulted and updated

DNS Lookup

A client performs a **DNS lookup** (query) to the local DNS software called a **resolver**

- The resolver starts by querying the name server at the top level of the DNS hierarchy
- Each name server replies with information about the authoritative server (name of the server and possibly IP address) one level down the hierarchy
- The resolver repeats the previous step until the IP address is returned
- Each query has a unique **query id** that helps associate the response with the request

DNS Lookup Example



DNS Caching

- DNS responses are cached at name servers
  - Allows quick response for repeated queries
- DNS negative queries are cached also
  - Saves time for nonexistent sites, e.g. misspelling
- Cached data periodically times out
  - Lifetime (TTL) of data is controlled by owner of data
  - TTL ranges from seconds to days
    - Higher TTL is more efficient
      - Less DNS requests are made
  - Shorter TTL allows better load balancing
    - The same name is mapped to different IP addresses to spread load among web servers in a server farm

DNS Cache Poisoning

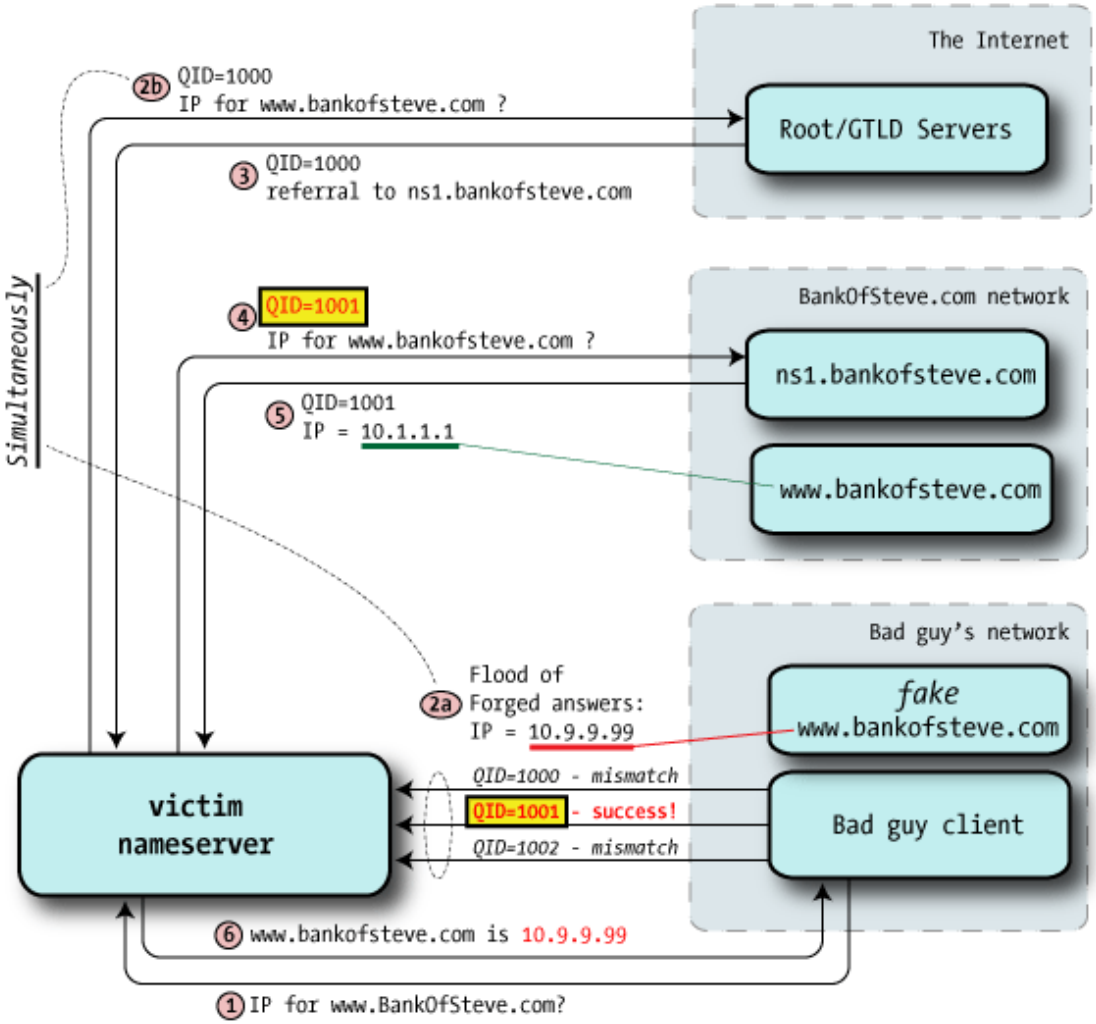
Users/hosts trust DNS mappings at name servers, although these mappings are not authenticated

- If an attacker is able to update a DNS server's cache with bogus mappings, then hosts would be served these **poisoned** mappings
- How is it possible to poison a DNS cache?
  - Exploit vulnerability in DNS software
    - *e.g.*, BIND v4.9 had buffer overflow
  - Spoof DNS response
    - For a single host
    - For an entire domain
    - Let's see how the spoofing works

Spoofing DNS Response: Host

Attacker initiates DNS request for **www.bankofsteve.com** to a victim nameserver

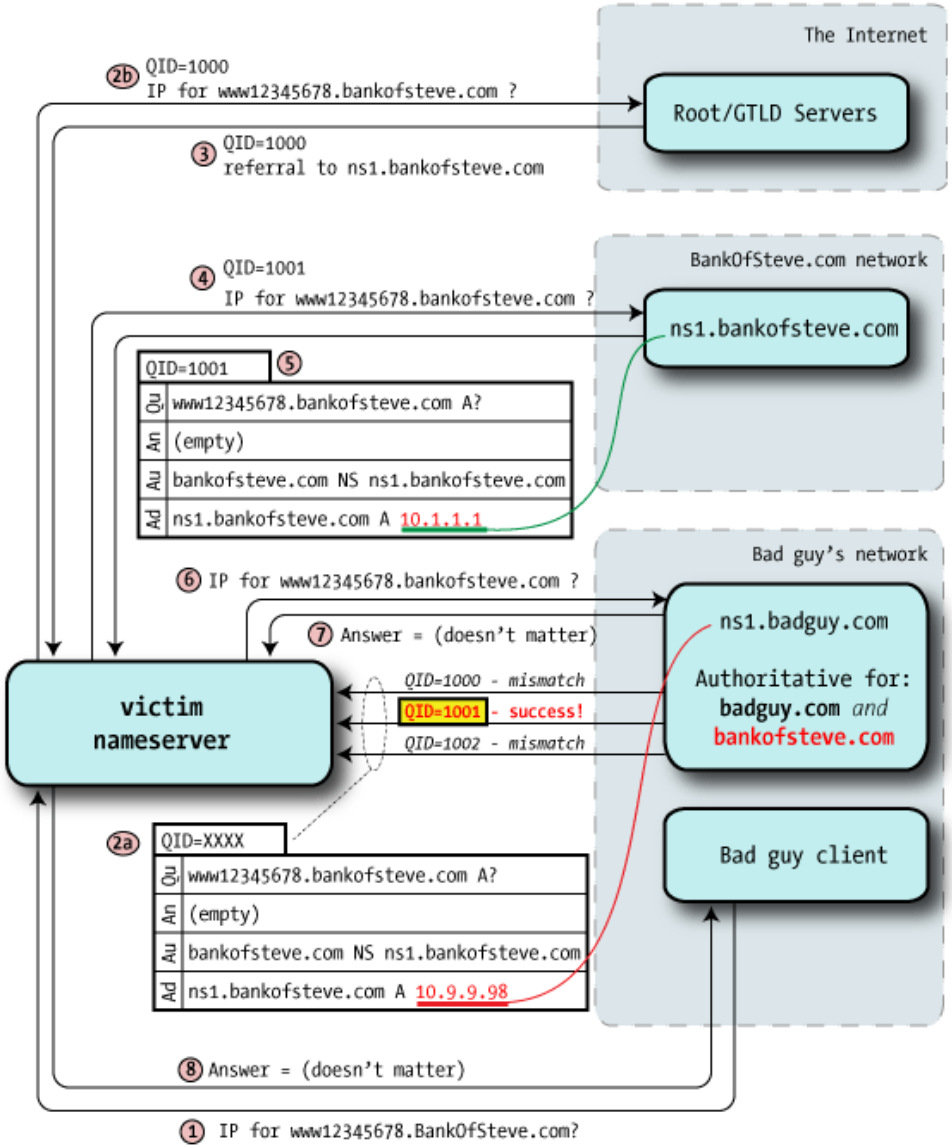
- Then it sends several DNS responses with **spoofed IP addresses** and various guessed **Query IDs**
- If a spoofed response has a correct query id, the victim name server then has a poisoned cache entry for **www.bankofsteve.com**



Spoofing DNS Response: Domain

Previous attack has short window of vulnerability due to caching (TTL). In this attack, the attacker performs several DNS requests for **wwwN.bankofsteve.com** to a victim nameserver where **N** changes with each request

- Then it sends several DNS responses with a **spoofed authoritative name server**
- If correct nameserver is cached, it will be evicted!
- If attack succeeds, attacker controls the entire bankofsteve.com domain



DNS Rebinding Attack

Previous poisoning attacks replace mapping for victim's domain with the attacker's IP

- victim domain -> attacker's IP address

A **DNS rebinding attack** replaces mapping for attacker's domain with victim's IP

- attacker domain -> victim's IP address

- How can that be exploited?
- It allows bypassing the browser's same origin policy because an attacker's IP address and the victim's IP address appear to belong to the same domain!

DNS Rebinding Attack

Attack steps

- **Browser:** Attacker gets client to visit attacker's site
- **Attacker:** Attacker controls DNS of site, and returns DNS response with short TTL. Attacker's site serves a web page with malicious script.
- **Browser:** The script makes a request to the attacker's web site. In turn, the browser makes another DNS query, which reaches the attacker's DNS, since the cached entry had a short TTL.
- **Attacker:** This time the attacker's DNS returns the IP address of a victim web server
- **Browser:** Now the victim's web server and the attacker's web server appear to be in the same origin
  - If the browser is located within an Intranet, a rebinding attack may allow accessing internal company machines!

DNS Rebinding Defenses

Why does this attack work?

- Browsers use domains instead of IP addresses to enforce the *same origin policy*
  - The reason browsers use domain names is that many web sites switch IP addresses (*e.g.*, for load balancing)
  - Unfortunately, it is hard to distinguish between IP address switching caused by load balancing vs. DNS rebinding

Possible defenses?

- Browsers can pin DNS/IP mapping to the value in the first DNS response
- Block resolution of external names into local IP addresses at a local nameserver

Distributed Denial of Service

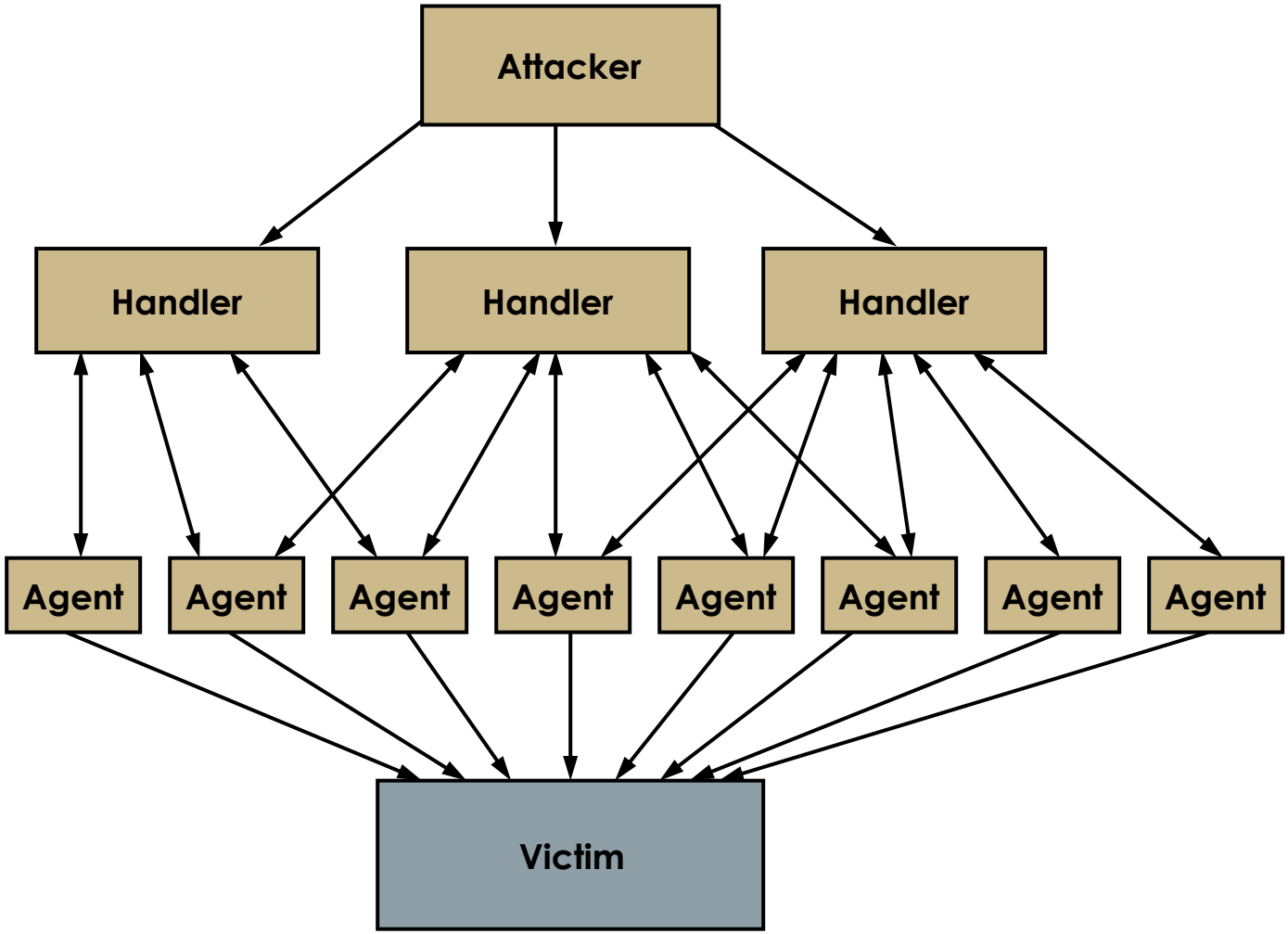
Denial of Service (DoS) simply means making a system inaccessible to legitimate users

- Usually, the attacker's goal is to consume as many resources as possible, so that others can't get service
- Typically, the attacker targets bandwidth or memory
  - What resource did the DoS attacks discussed earlier (Smurf, TCP syn flooding) target?

A bandwidth attack requires flooding the server with packets, so the attacker needs more bandwidth than the victim

- One method used by an attacker is to build up a large number of compromised hosts, and then use them to simultaneously attack a single target
- This is called **Distributed Denial of Service (DDOS)**

Distributed Denial of Service



Common Defenses

Cryptographic Protocols, Firewalls

Cryptographic Protocols

Cryptographic protocols can be used to defend against many of the attacks on Internet protocols

- Protect against spoofing attacks and injected data
- Generally, don't protect against DoS attacks

- **Application layer**
  - SSL, SSH
- **Transport layer**
  - Use cryptographically random ISNs for TCP/IP
- **Network layer**
  - IPSec

We have seen SSL, next we look at IPSec

IPSec

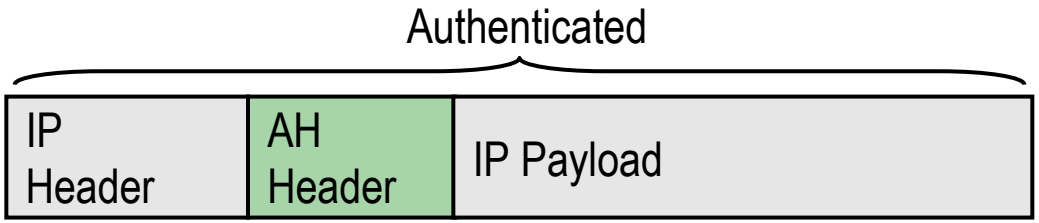
IPSec was developed by the IETF

- Designed to provide the following at IP layer:
  - Message confidentiality
  - Message integrity
  - Source authentication
  - Protection against replay
- IPSec is composed of two protocols:
  - **Authenticated Headers** (AH)
    - Provides all of the above properties except confidentiality
  - **Encapsulating Security Payload** (ESP)
    - Provides all of the above properties

IPSec Protocols

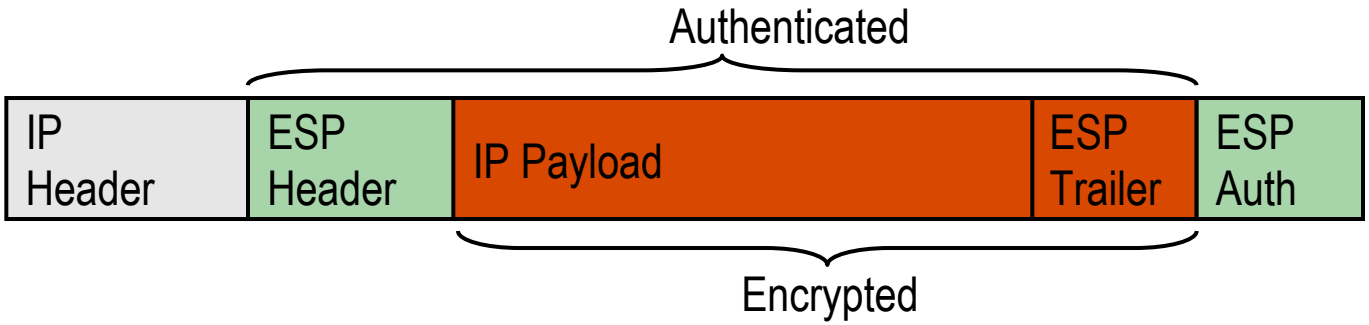
Authenticated Headers

- Protects the IP packet (except IP header fields that are altered during transit) using a MAC stored in AH Header



Encapsulating Security Payload

- Payload is encrypted to protect contents



IPSec Modes

IPSec can be used even when all routers on the Internet are not IPSec enabled. IPSec has two modes:

- **Transport mode**
  - Both endpoints support IPSec, but intermediary routers do not
  - This mode encrypts/authenticates the packet **payload**
  - Similar to SSL, SSH, etc.
- **Tunnel mode**
  - Endpoints do not support IPSec, but endpoint gateways do
  - This mode encrypts/authenticates the packet **header and payload** and encapsulates it in another regular IP packet
  - Similar to SSH tunneling or VPN software

IPSec vs. SSL

Choosing IPSec or SSL depends on security needs

- If a specific service is required and is supported by SSL it is better to select SSL
- If access to an entire network is required, VPN software/device using IPSec is a good choice

Security

- SSL can provide more security because if one connection is compromised, others are not
- SSL can provide better access control since with IPSec allows access to entire network

IPSec vs. SSL

Compatibility and deployment

- SSL does not require special client software (already in browser) and configuration
- SSL is more compatible with firewalls, unless IPSec and firewall are on same device
- IPSec often doesn't interoperate well (poorly standardized implementations), so both sides of the connection are required to have the same vendor's devices
- IPSec supports pre-shared keys so PKI is not needed

Overhead

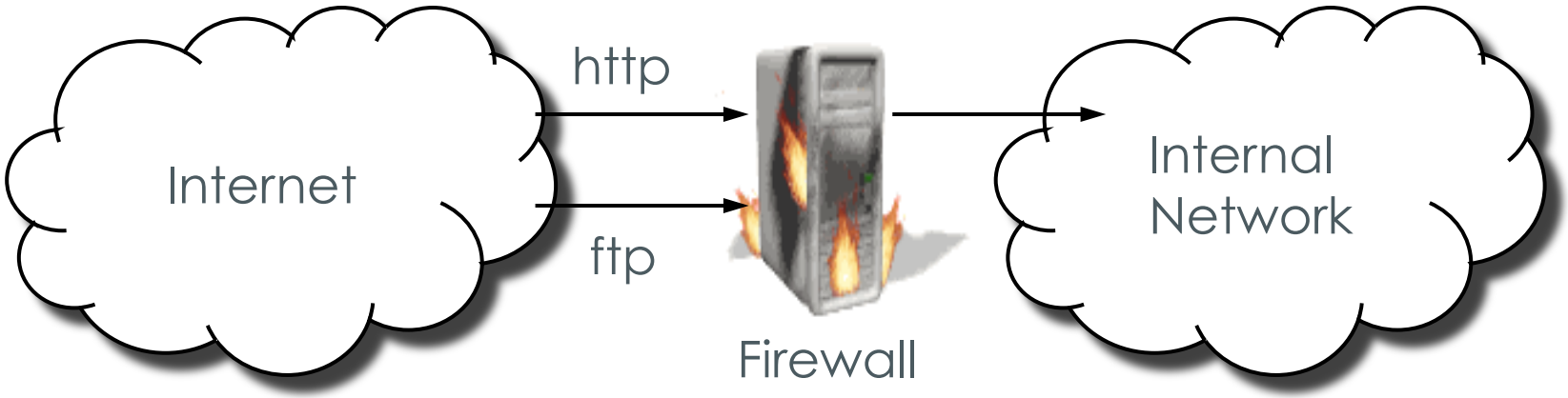
- IPSec has lower overhead because many users can use the same secure channel while SSL requires connection establishment for each channel

For more info see Bruce Schneier's "Criticism/Evaluation of IPSec" on course web site

Firewalls

A firewall is a machine whose function is to control access to an internal network

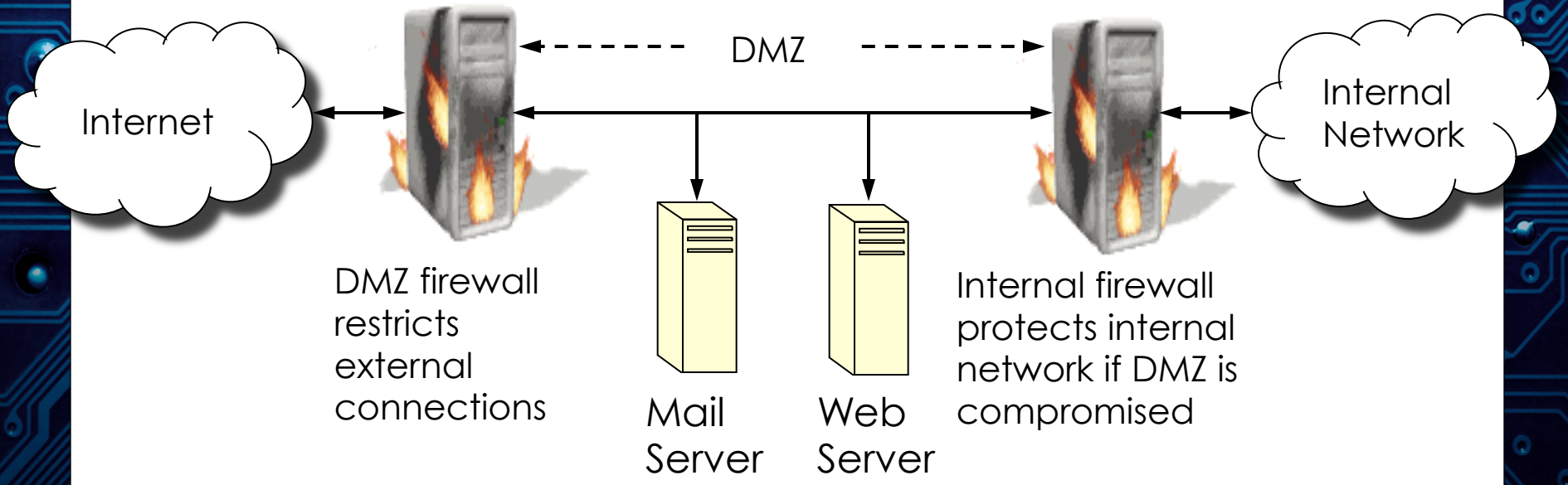
- Some types of connections are allowed while other are not
- Access policy is usually determined by the port number which indicates the type of service being accessed



Firewall Deployment

A firewall is normally placed at entry points between an internal and an external network

- Sometimes certain machines need to be accessible both externally and internally, requiring an internal firewall and a Demilitarized Zone (DMZ) firewall



Firewall Example

Home routers provide simple firewall functionality

- Filter based on port number, and possibly source address

Sophisticated firewalls filter on numerous criteria:

- Protocol, frequency of packets, etc.
- Allowing incoming packets only when an initial outgoing connection has been established, etc.

Example: a Linux system can be configured with firewall rules with the **iptables** command

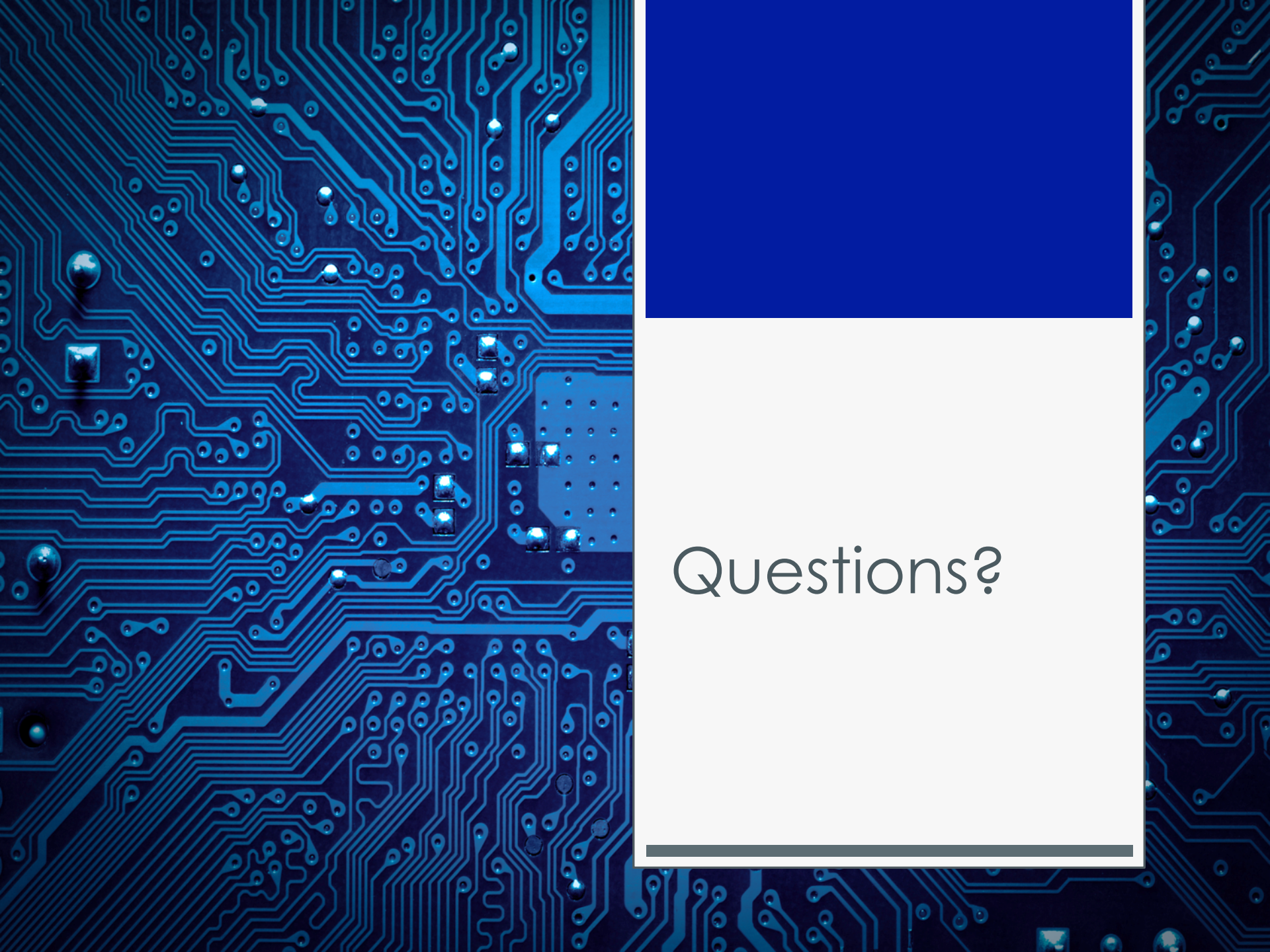
- Each rule has a set of criteria that the kernel matches with each packet (*i.e.*, src addr, dest addr, protocol, etc.)
- Each rule also has an target (accept, drop, log, etc.)
- The firewall tables are composed of chains of rules
- The OS tries each rule in the chain until one matches
- If a match is found, the target is executed

IPTables Example

```
% iptables -L
…
Chain INPUT (policy DROP)
target     prot opt source          destination
ACCEPT     all  --  anywhere        anywhere     state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere        anywhere     state NEW tcp dpt:http
ACCEPT     tcp  --  anywhere        anywhere     state NEW tcp dpt:ssh
ACCEPT     tcp  --  128.100.8.51    anywhere     state NEW tcp dpt:smtp
```

The INPUT chain determines whether hosts outside of the firewall can make new connections

- Rule 1 says that all packets on already established connections are allowed (to come in)
- Rules 2-4 allow http, ssh and smtp connections to be initiated from the outside, but smtp connections can only come from 128.100.8.51


Questions?