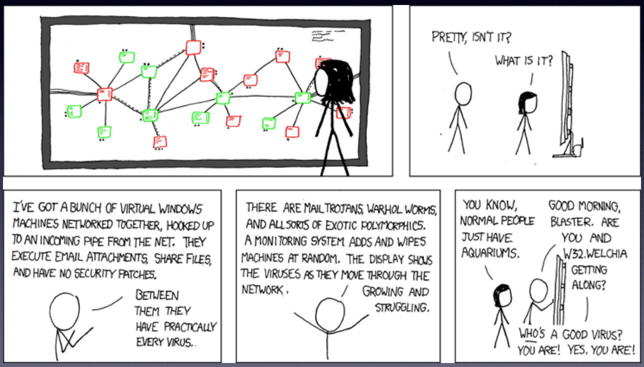
Network Security

CMSC 426 - Computer Security

Network Insecurity


PRETTY, ISN'T IT?
WHAT IS IT?
I'VE GOT A BUNCH OF VIRTUAL WINDOWS MACHINES NETWORKED TOGETHER, HOOKED UP TO AN INCOMING PIPE FROM THE NET. THEY EXECUTE EMAIL ATTACHMENTS, SHARE FILES, AND HAVE NO SECURITY PATCHES.
BETWEEN THEM THEY HAVE PRACTICALLY EVERY VIRUS..
THERE ARE MAIL TROJANS, WARHOL WORMS, AND ALL SORTS OF EXOTIC POLYMORPHICS. A MONITORING SYSTEM ADDS AND WIPES MACHINES AT RANDOM. THE DISPLAY SHOWS THE VIRUSES AS THEY MOVE THROUGH THE NETWORK,
GROWING AND STRUGGLING.
YOU KNOW, NORMAL PEOPLE JUST HAVE AQUARIUMS.
GOOD MORNING, BLASTER. ARE YOU AND W32.WELCHIA GETTING ALONG?
WHO'S A GOOD VIRUS? YOU ARE! YES, YOU ARE!

Overview

- Internet protocol layers - TCP/IP model
- Details of specific layers
  - Link Layer
  - Internet Layer
  - Transport Layer

TCP/IP Layer Model

- *Physical Layer* - wires, fiber, radios, etc.
- *Link Layer* - local / point-to-point communications
- *Internet Layer* - host-to-host communications
- *Transport Layer* - application-to-application communications (via ports)
- *Application Layer* - high-level protocols to provide useful network functions

Link Layer

- Connection of machines on a local network, e.g. on the same wire or AP.
- Common link layer technologies:
  - *Ethernet* (wired)
  - *802.11* (wifi)
- Extending the network: hubs and switches

Media Access Control

- Devices on the network are identified by 48-bit *Media Access Control* (MAC) address.
- Written as six bytes, e.g. `00:1b:63:07:1c:c1`.
- MAC addresses are assigned by vendors; meant to be unique, but easily changed
- Ethernet frame includes MACs, payload, CRC-32 checksum

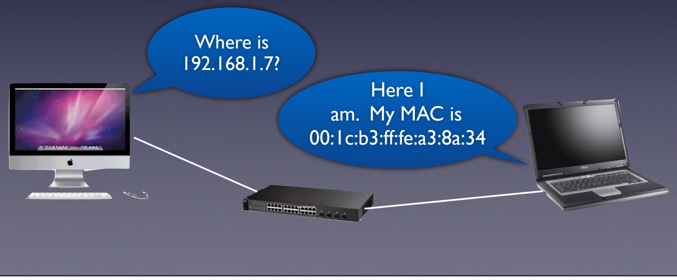
Ethernet Frame

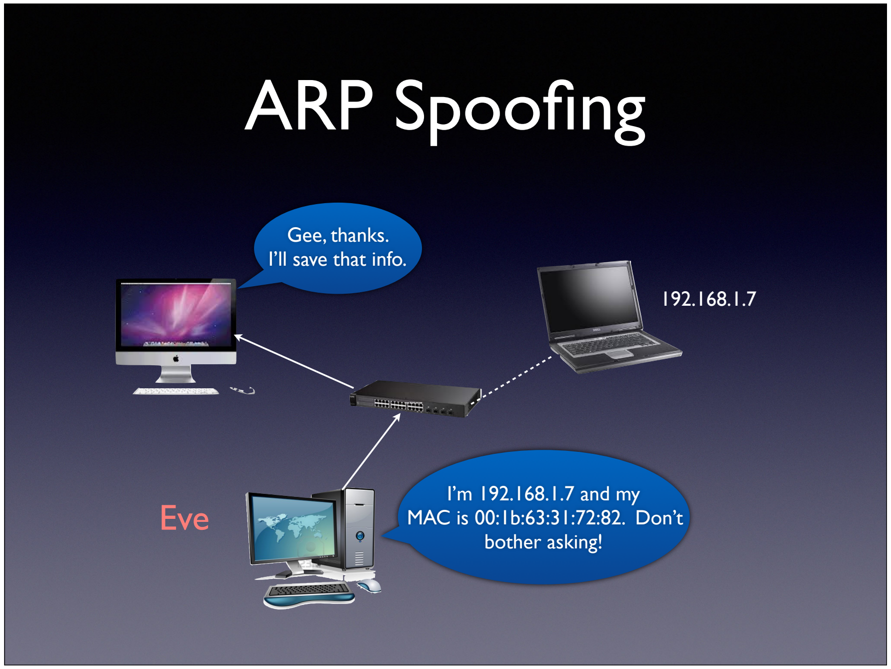
Preamble	SFD	Destination Address	Source Address	Type/ Length	Data	CRC
8 bytes		6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes

- Preamble not used in modern networks; SFD is Starting Frame Delimiter
- Destination & Source addresses are MACs
- Type / Length indicates protocol being carried, e.g. 0x800 for IPv4, 0x0806 for ARP, etc.

Address Resolution

- *Address Resolution Protocol* (ARP) maps IP addresses to MACs on a local network
- Host broadcasts a message requesting MAC for a given IP; machine with the given IP responds with its MAC





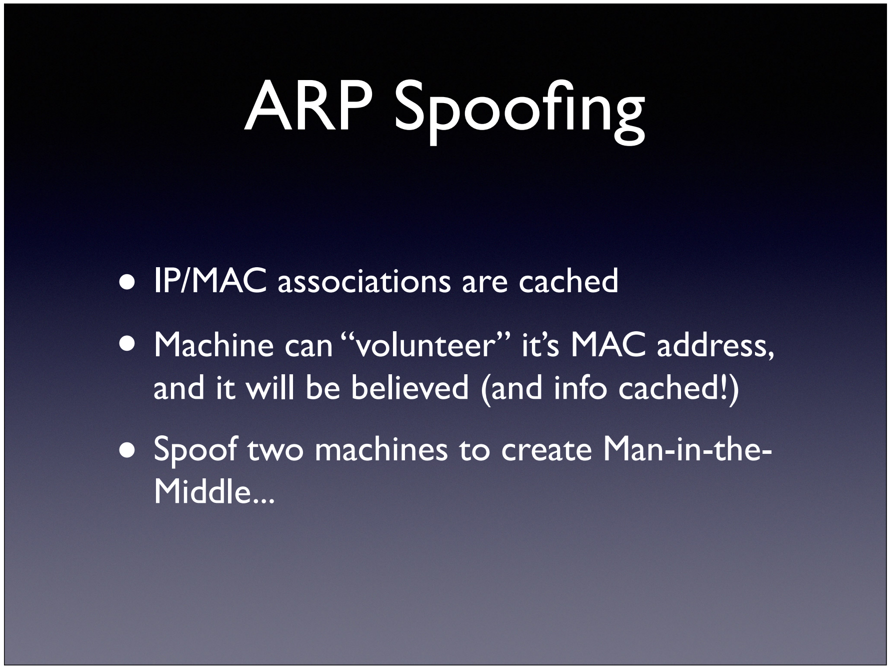
ARP Spoofing

Gee, thanks.
I'll save that info.

192.168.1.7

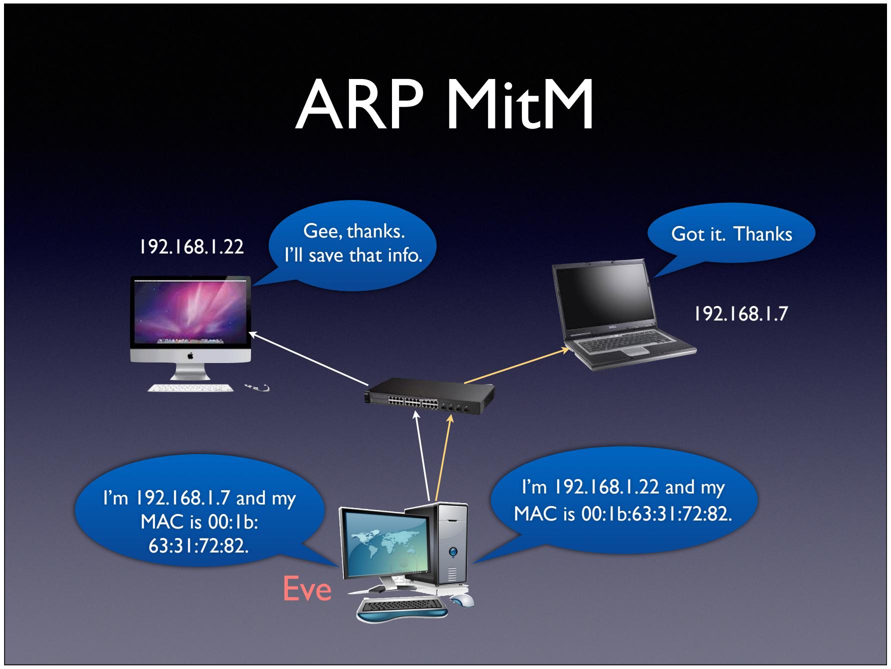
Eve

I'm 192.168.1.7 and my
MAC is 00:1b:63:31:72:82. Don't
bother asking!



ARP Spoofing

- IP/MAC associations are cached
- Machine can "volunteer" it's MAC address, and it will be believed (and info cached!)
- Spoof two machines to create Man-in-the-Middle...



ARP MitM

192.168.1.22

Gee, thanks.
I'll save that info.

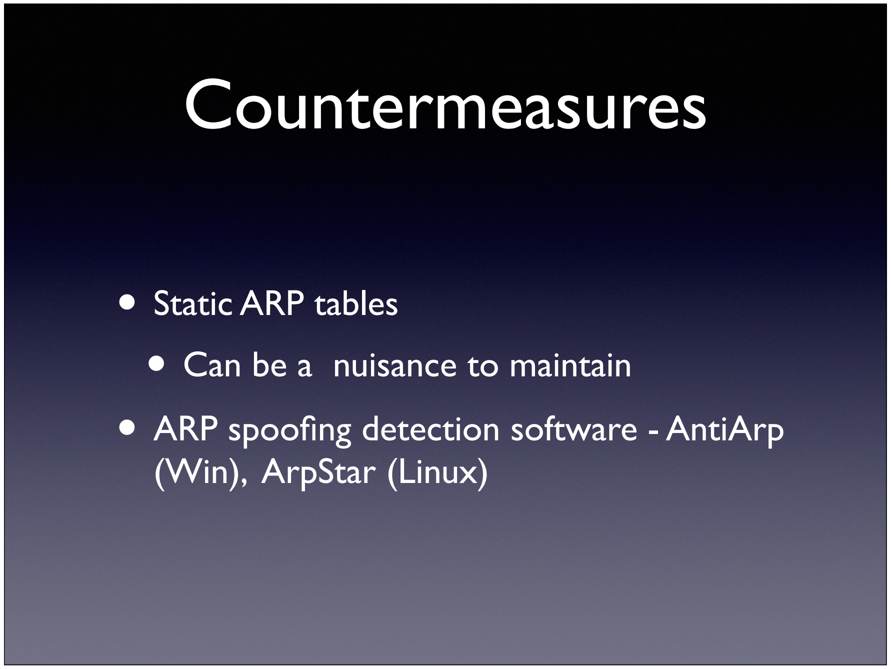
Got it. Thanks

192.168.1.7

I'm 192.168.1.7 and my
MAC is 00:1b:
63:31:72:82.

I'm 192.168.1.22 and my
MAC is 00:1b:63:31:72:82.

Eve



Countermeasures

- Static ARP tables
  - Can be a nuisance to maintain
- ARP spoofing detection software - AntiArp (Win), ArpStar (Linux)

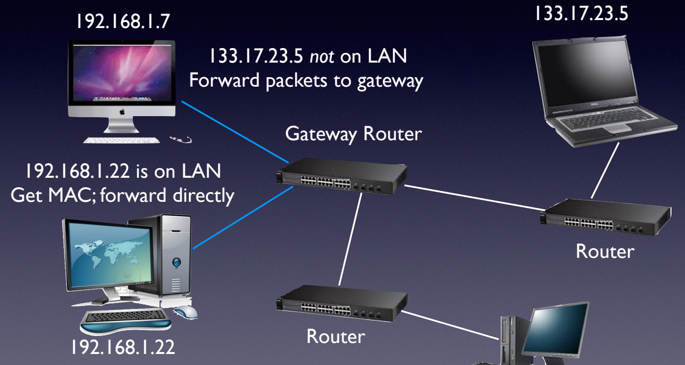
Internet Layer

- Transports packets from one host to another, across network boundaries if necessary
- *Internet Protocol* (IP) - best effort routing of data packets
- IP addresses - IPv4 (32 bits), IPv6 (128 bits)

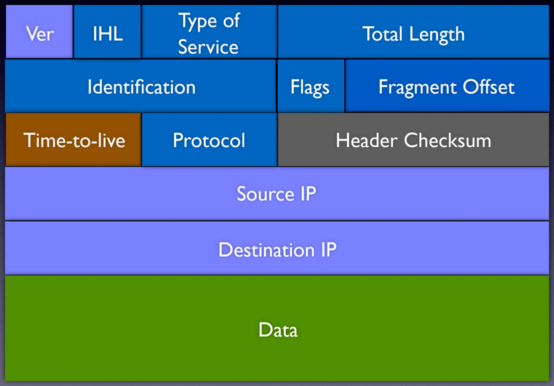
IP Routing

- Destination IP on same LAN?
  - Get MAC via ARP and forward packets directly
- Destination IP on different LAN?
  - Forward packets to *gateway router*
- Gateway is responsible for further routing
- *Routing tables* indicate which router packets should be sent to next

IP Routing



IPv4 Packet



Ver	IHL	Type of Service	Total Length
Identification		Flags	Fragment Offset
Time-to-live	Protocol	Header Checksum	
Source IP			
Destination IP			
Data			

Time-to-live

- Don’t want packets to bounce around the network forever
- *Time-to-live* (TTL) is the maximum number of router visits (*hops*) that a packet is allowed before it is dropped
- TTL is decremented by each router that handles a packet
- When TTL goes to zero, packet is dropped and an error packet is returned to source host

17

ICMP

- *Internet Control Message Protocol* (ICMP) - Internet layer protocol for testing and error notification.
- ICMP packet types include
  - *Echo Request* - asks destination to acknowledge
  - *Echo Response* - acknowledges an Echo Request
  - *Time Exceeded* - notification that packet expired
  - *Destination Unreachable* - packet could not be delivered

18

ICMP Applications

- *Ping* - Echo Request / Echo Response to determine if a host is operating
- *Traceroute* - determine path to a host; clever use of TTL field

19

Sample Traceroute

```
traceroute to www.google.com (74.125.225.51), 64 hops max, 52 byte packets
 1  wireless_broadband_router (192.168.1.1)  3.451 ms  1.110 ms  1.155 ms
 2  l100.bltmmd-vfttp-45.verizon-gni.net (98.117.58.1)  6.712 ms  8.411 ms  8.739 ms
 3  g0-5-1-5.bltmmd-lcr-21.verizon-gni.net (130.81.109.194)  16.032 ms  17.611 ms  14.511 ms
 4  ae20-0.res-bb-rtr1.verizon-gni.net (130.81.151.112)  41.737 ms  87.260 ms  22.076 ms
 5  0.ae5.xl1.iad8.alter.net (152.63.8.121)  14.508 ms
    0.ae4.xl2.iad8.alter.net (152.63.8.125)  12.286 ms
    0.ae5.xl1.iad8.alter.net (152.63.8.121)  11.854 ms
 6  0.xe-10-3-1.gw9.iad8.alter.net (152.63.41.250)  13.974 ms
    0.xe-11-0-0.gw9.iad8.alter.net (152.63.33.165)  41.517 ms
    0.xe-10-3-0.gw9.iad8.alter.net (152.63.41.246)  14.323 ms
 7  pool-96-236-104-66.burl.east.verizon.net (96.236.104.66)  16.263 ms  10.645 ms  11.618 ms
 8  216.239.46.248 (216.239.46.248)  14.432 ms  14.701 ms  12.373 ms
 9  72.14.236.148 (72.14.236.148)  13.907 ms
    209.85.243.175 (209.85.243.175)  19.339 ms
    216.239.48.163 (216.239.48.163)  22.302 ms
10  209.85.246.83 (209.85.246.83)  32.797 ms
    209.85.246.37 (209.85.246.37)  31.901 ms
    72.14.232.73 (72.14.232.73)  31.142 ms
11  216.239.50.235 (216.239.50.235)  33.255 ms
    72.14.237.132 (72.14.237.132)  34.054 ms
    216.239.50.235 (216.239.50.235)  33.166 ms
12  209.85.250.28 (209.85.250.28)  30.287 ms  34.383 ms  29.685 ms
13  ord08s06-in-f19.1e100.net (74.125.225.51)  47.738 ms  34.682 ms  32.038 ms
```

20

IP Spoofing

- There is no authentication of the *Source Address* in an IP packet - can be spoofed
- Valid use of IP Spoofing in e.g. server testing
- Attacker may spoof the source address, but he will not see responses
  - May not care about response, e.g. in Denial of Service attacks
  - May have other way to collect response

Preventing Spoofing

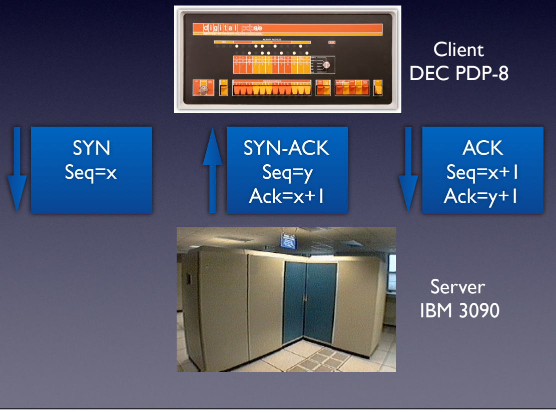
- Filtering at the network border
  - block incoming packets with source address that is inside the administrative domain
  - block outgoing packets with source address that is outside the domain
- *IP traceback* - techniques for determining a packets source and path thru the network

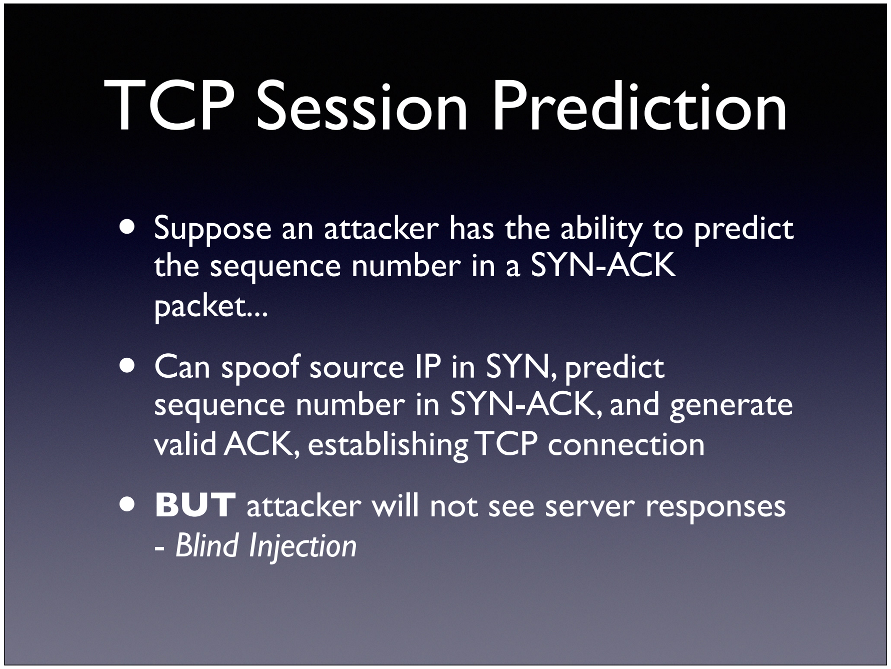
Transport Layer

- Provide communications between processes / services on networked hosts
- Processes / services associated with *ports*; there are 2^{16} different port numbers
- *Transmission Control Protocol* (TCP) - reliable, connection-oriented protocol
- *User Datagram Protocol* (UDP) - “best effort” communications

TCP Connections

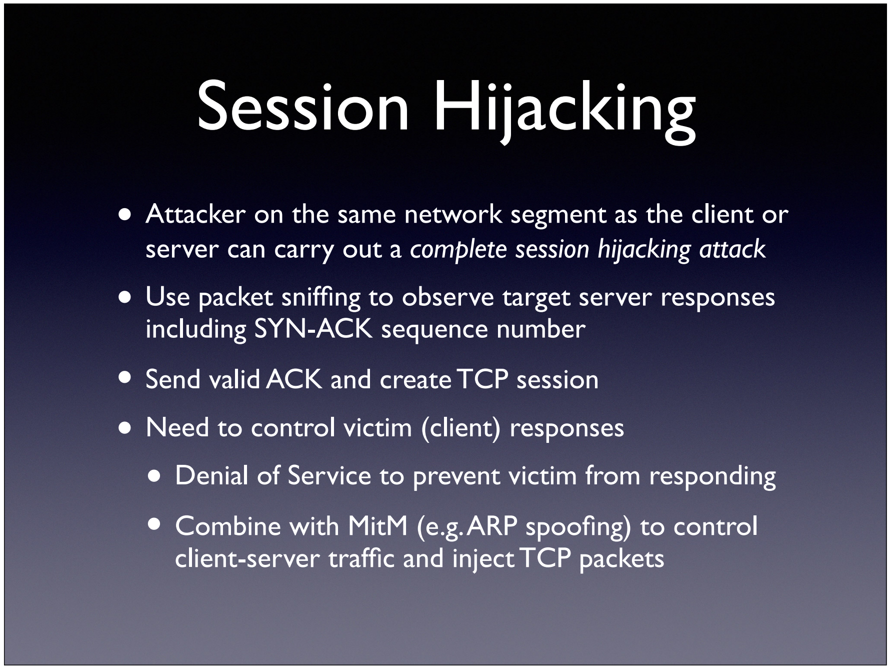
- The *Three-way Handshake*





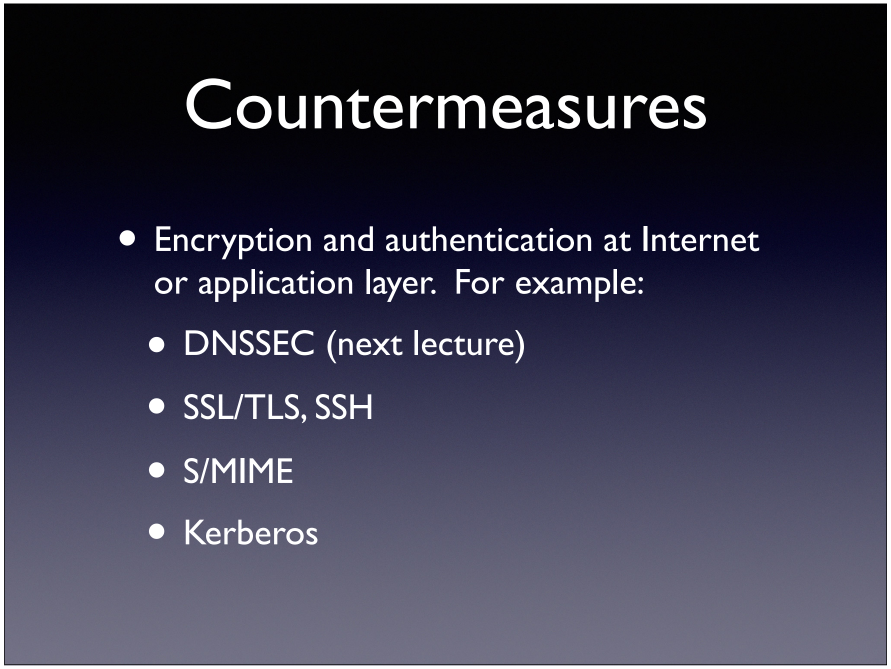
TCP Session Prediction

- Suppose an attacker has the ability to predict the sequence number in a SYN-ACK packet...
- Can spoof source IP in SYN, predict sequence number in SYN-ACK, and generate valid ACK, establishing TCP connection
- **BUT** attacker will not see server responses - *Blind Injection*



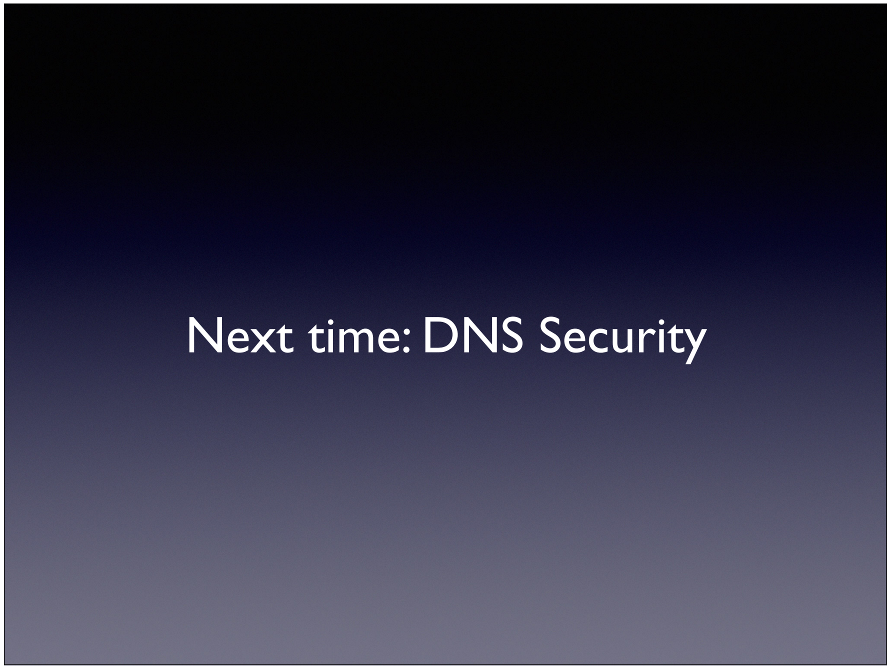
Session Hijacking

- Attacker on the same network segment as the client or server can carry out a *complete session hijacking attack*
- Use packet sniffing to observe target server responses including SYN-ACK sequence number
- Send valid ACK and create TCP session
- Need to control victim (client) responses
  - Denial of Service to prevent victim from responding
  - Combine with MitM (e.g. ARP spoofing) to control client-server traffic and inject TCP packets



Countermeasures

- Encryption and authentication at Internet or application layer. For example:
  - DNSSEC (next lecture)
  - SSL/TLS, SSH
  - S/MIME
  - Kerberos



Next time: DNS Security