Malicious PhpMyAdmin Served From SourceForge Mirror

Posted by **Unknown Lamer** on Wednesday September 26, @10:40AM
from the tin-foil-hat-activate dept.

An anonymous reader writes with a bit of news about the compromised download of phpMyAdmin discovered on an sf.net mirror yesterday:

> "A malicious version of the open source Web-based MySQL database administration tool phpMyAdmin has been discovered on one of the official mirror sites of SourceForge, the popular online code repository for free and open source software. The file — phpMyAdmin-3.5.2.2-all-languages.zip — was modified to include a backdoor that allowed attackers to remotely execute PHP code on the server running the malicious version of phpMyAdmin."

The Sourceforge weblog has details. Someone compromised a mirror (since removed from rotation of course) around September 22nd. Luckily, only around 400 people grabbed the file before someone caught it.

Link layer security

CS642: Computer Security

Professor Ristenpart

http://www.cs.wisc.edu/~rist/

rist at cs dot wisc dot edu

Announcements

- Projects:
  - See email for instructions
  - Proposal due October 11th
  - Undergrads can do for extra credit

Getting started on network security

Internet protocol stack

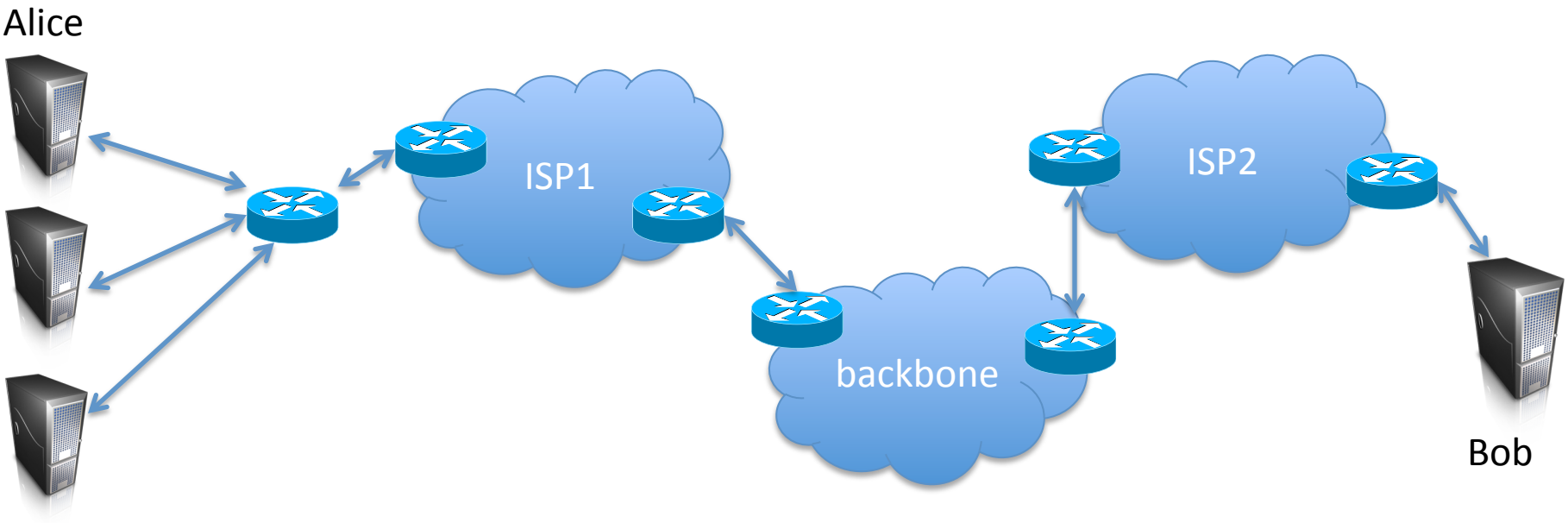
Man-in-the-middle

Address resolution protocol and ARP spoofing

802.11

Jamming and MITM prevention

Internet



Local area network (LAN)

Ethernet

802.11

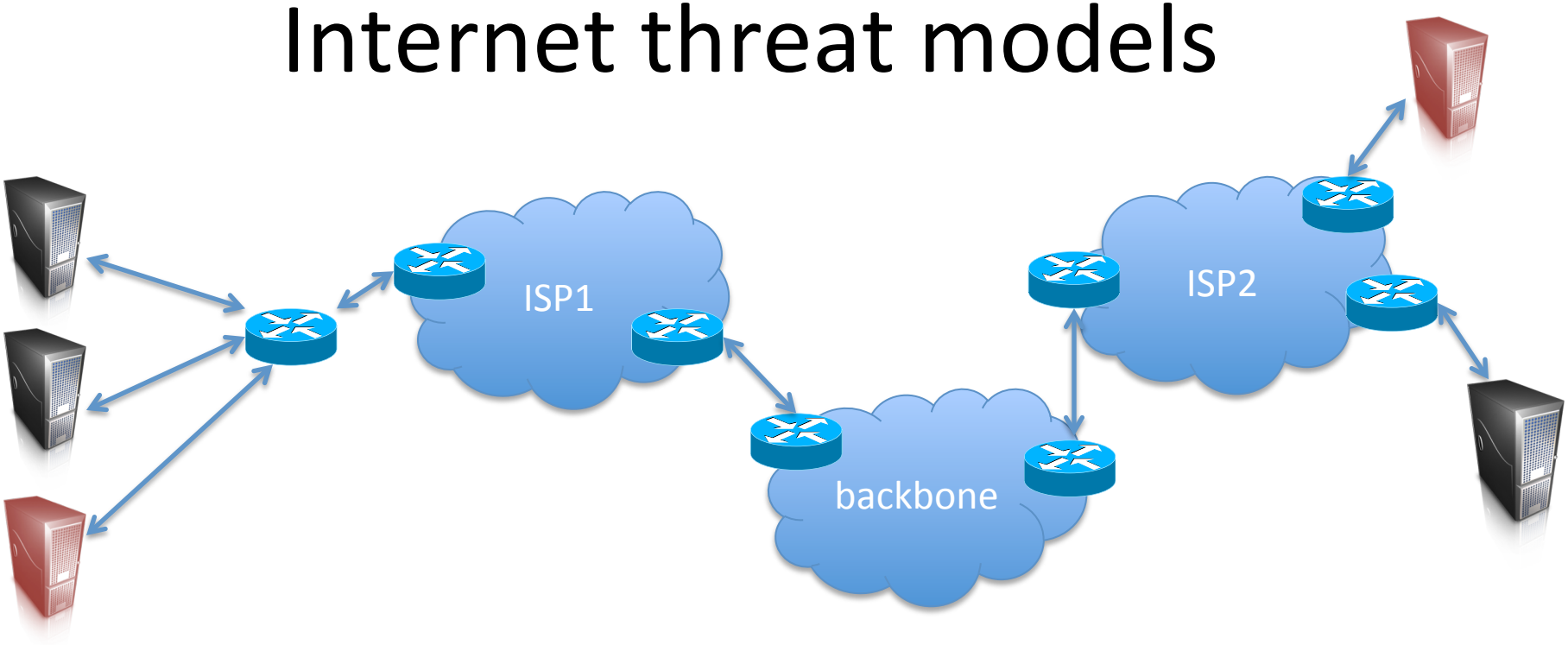
Internet

TCP/IP

BGP (border gateway protocol)

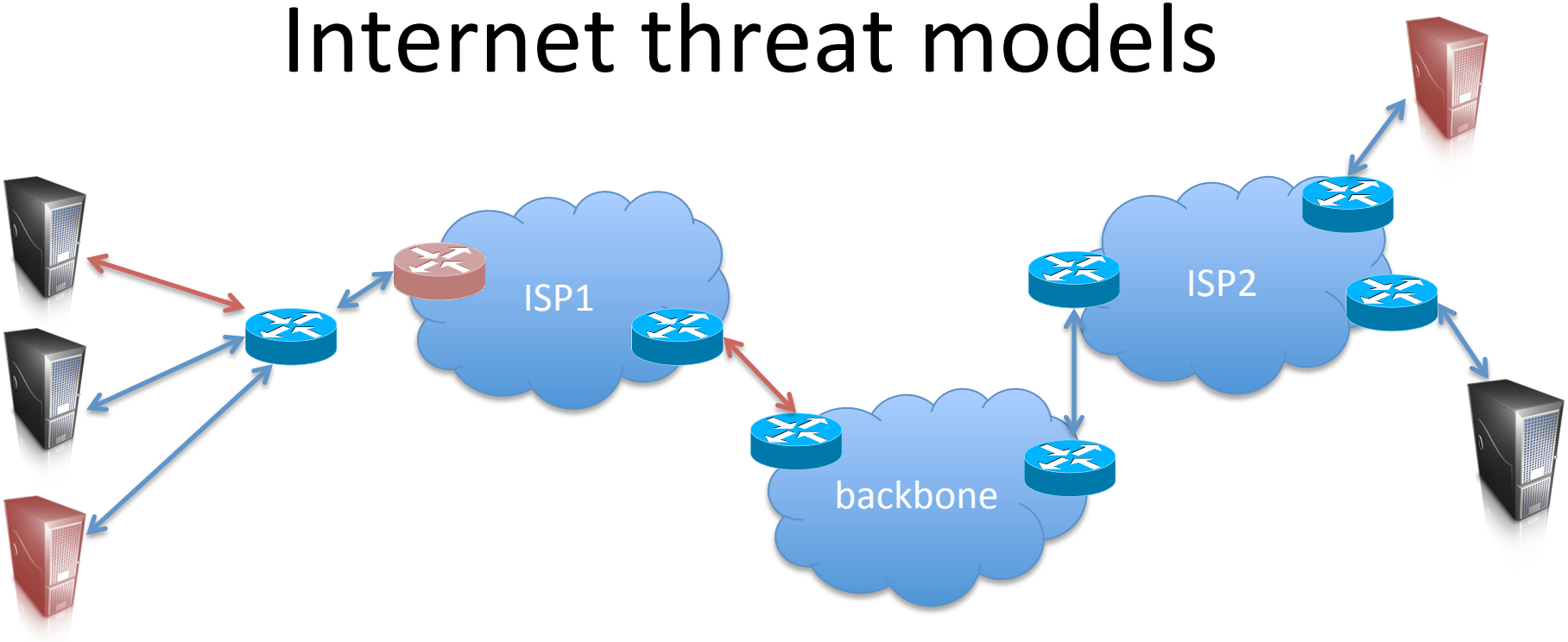
DNS (domain name system)

Internet threat models



(1) Malicious hosts

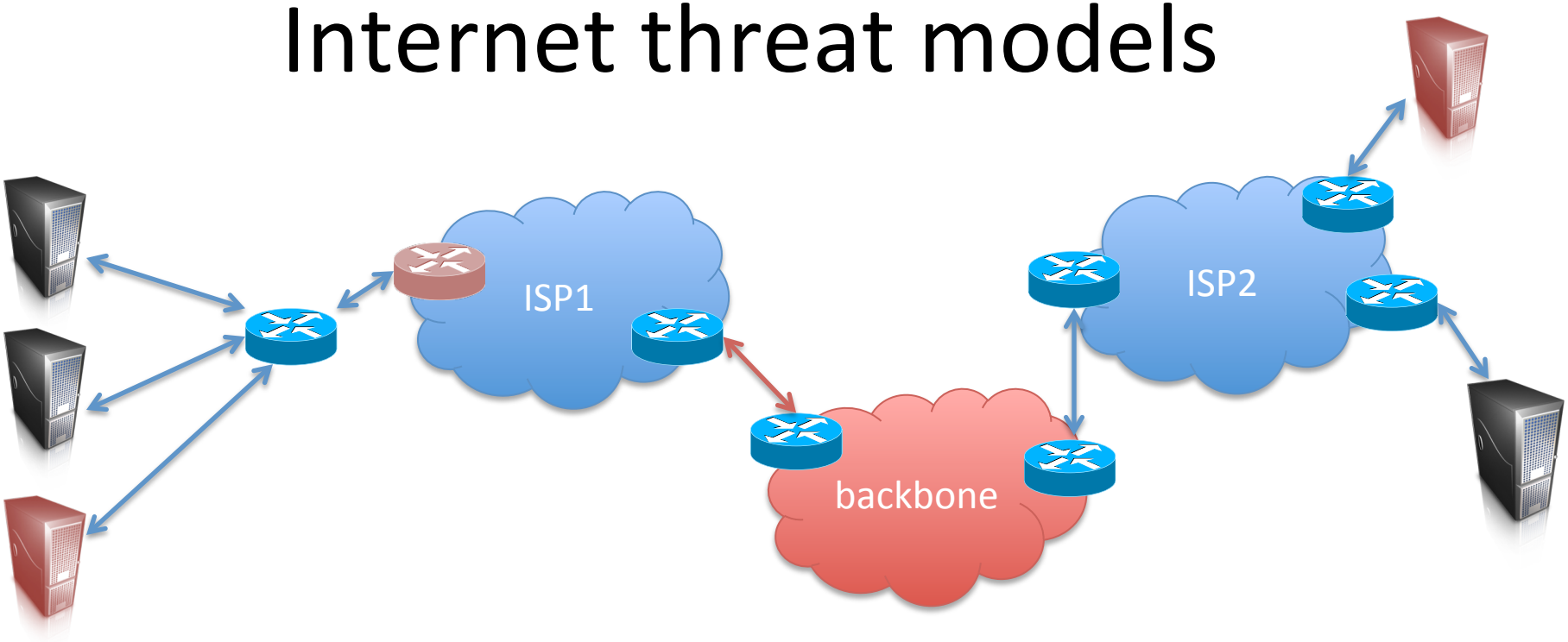
Internet threat models



(1) Malicious hosts

(2) Subverted routers or links

Internet threat models



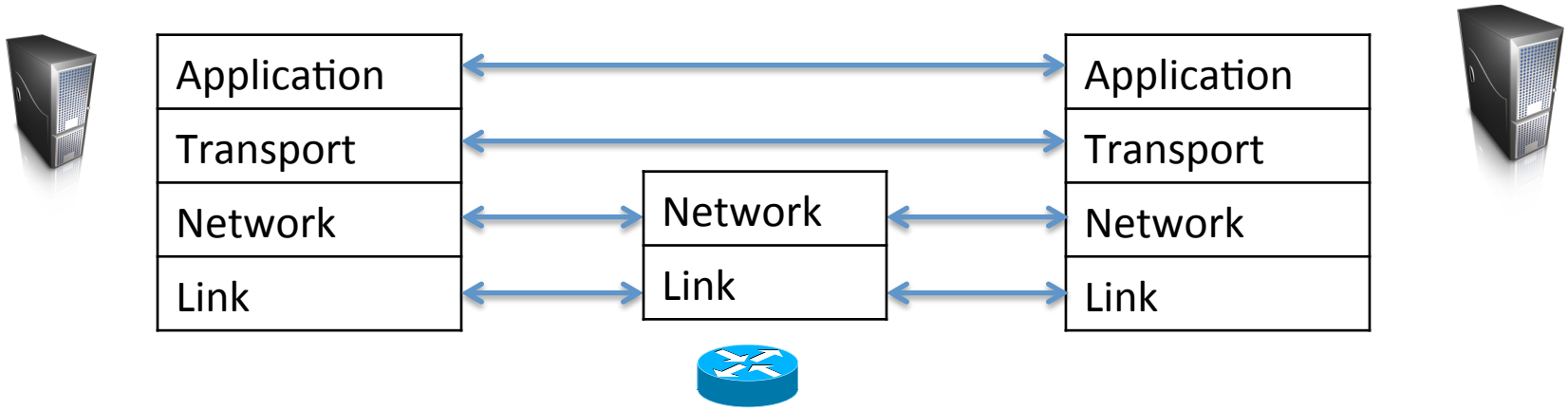
(1) Malicious hosts

(2) Subverted routers or links

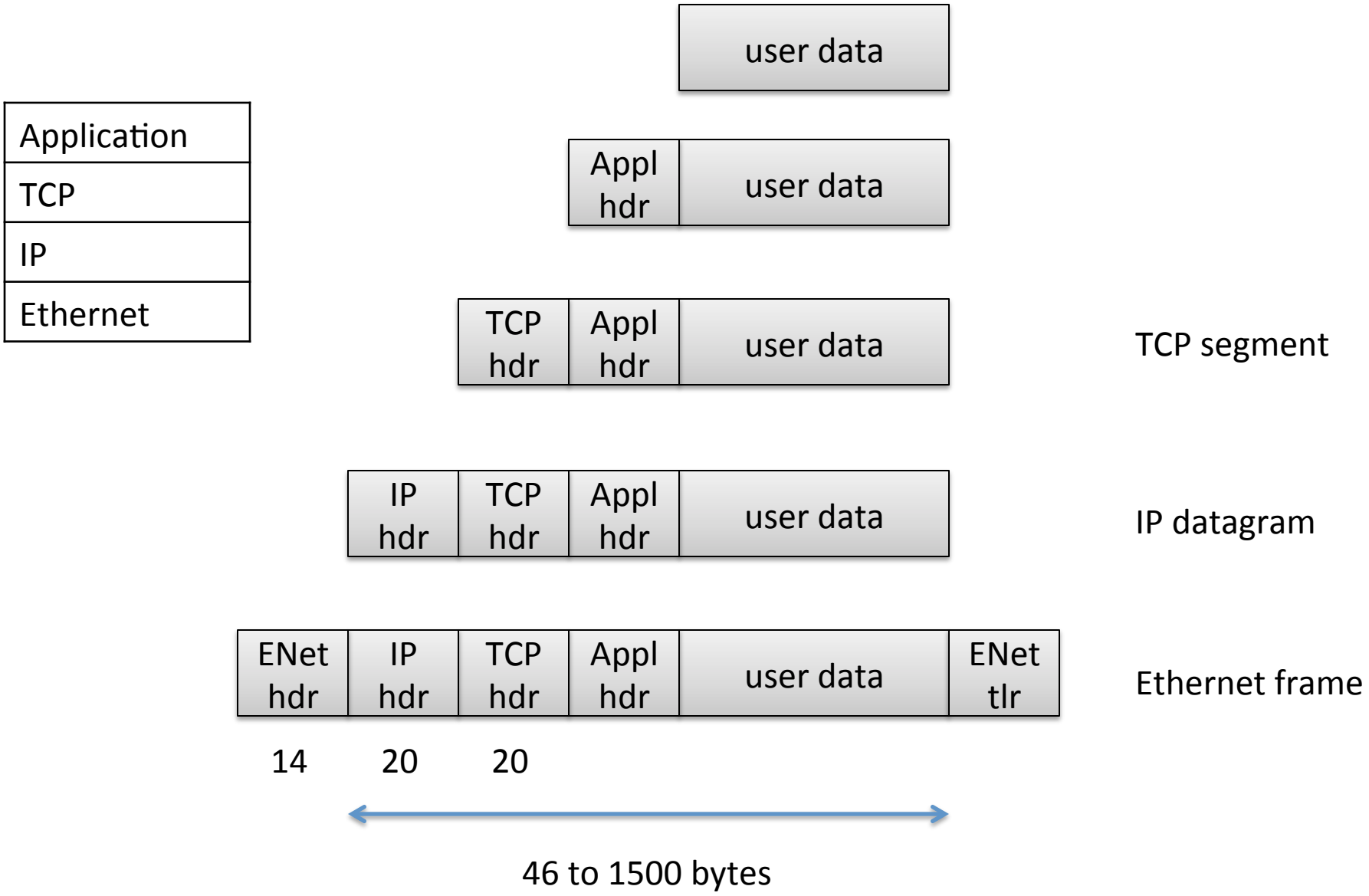
(3) Malicious ISPs or backbone

Internet protocol stack

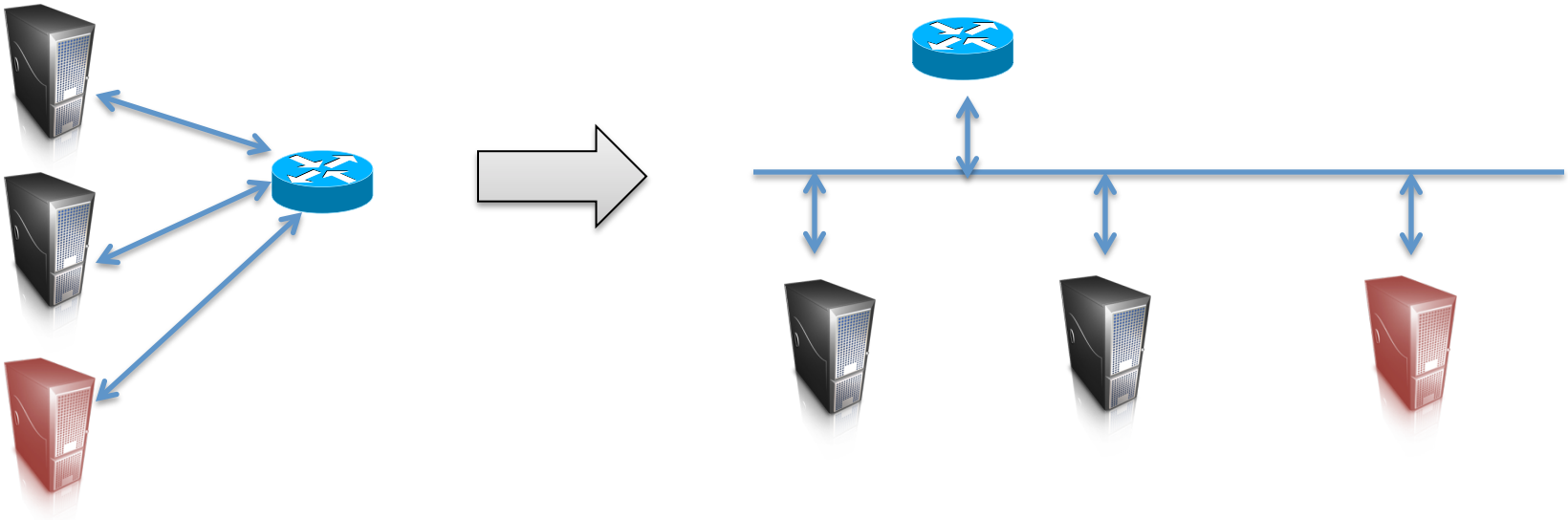
Application	HTTP, FTP, SMTP, SSH, etc.
Transport	TCP, UDP
Network	IP, ICMP, IGMP
Link	802x (802.11, Ethernet)



Internet protocol stack



Ethernet



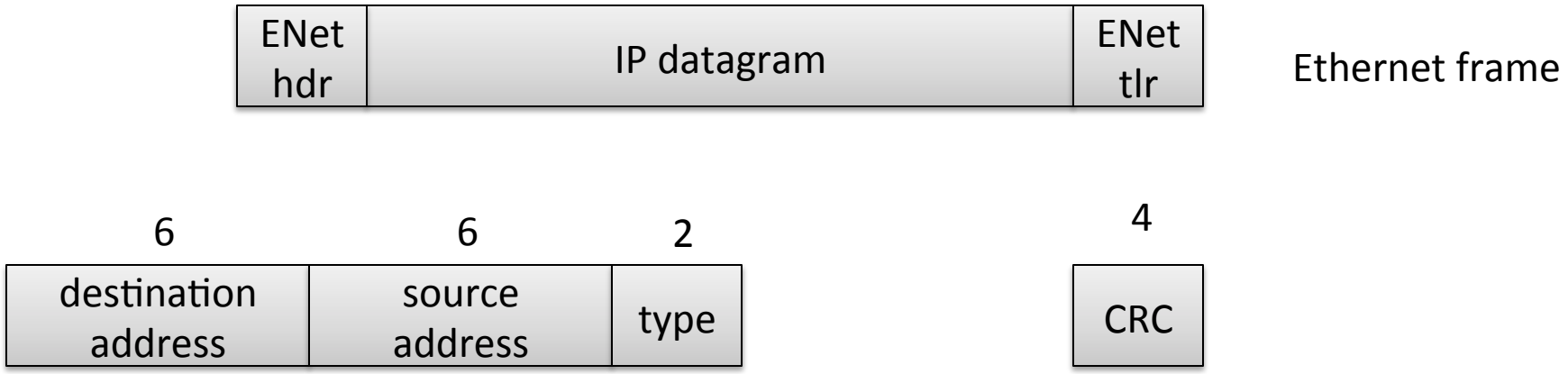
Carrier Sense, Multiple Access with Collision Detection (CSMA/CD)

Take turns using broadcast channel (the wire)

Detect collisions, jam, and random backoff

Security issues?

Ethernet



Media access control (MAC) addresses 48 bits

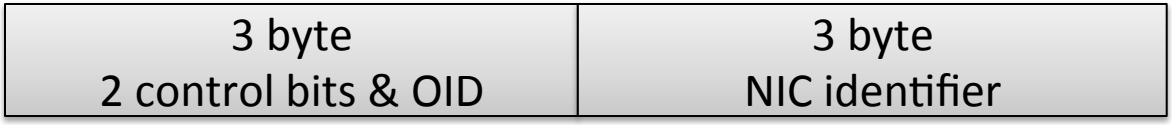
Type = what is data payload (0x0800 = IPv4, 0x0806 = ARP, 0x86DD = IPv6)

32 bit Cyclic Redundancy Check (CRC) checksum

802.2 LLC frame format slightly different, but similar ideas

MAC addresses

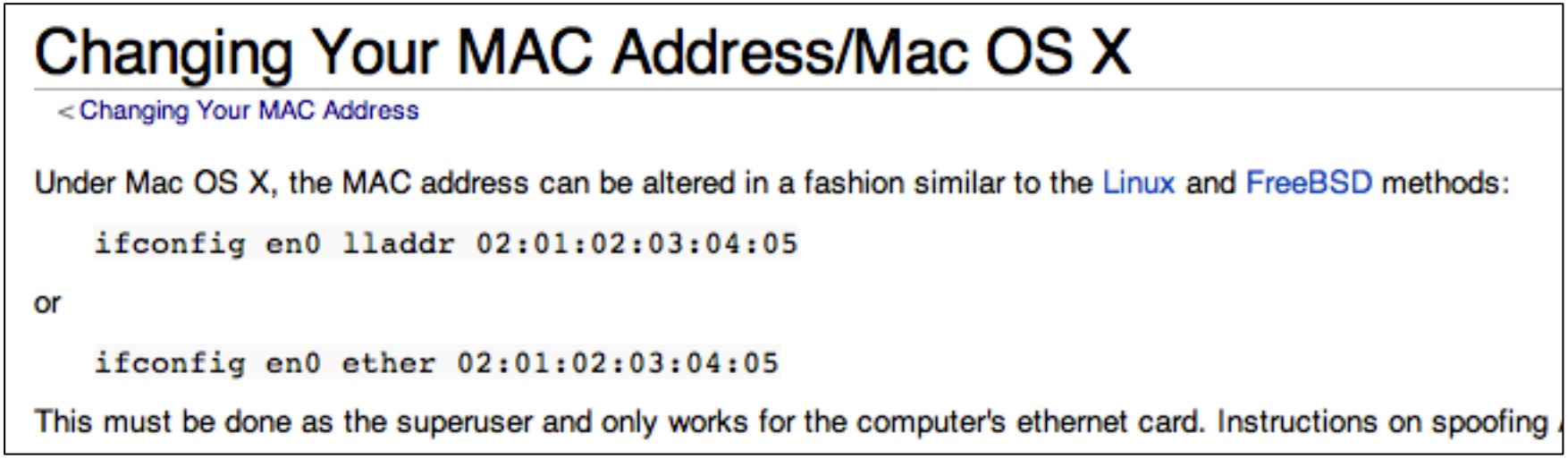
- Two types: universally or locally administered



  - 2 LSBs of first byte are control bits:
    - 1st LSB: multicast/unicast
    - 2nd LSB: universal/local flag
- Hardware (ethernet card/WiFi card) initialized with MAC address
- But:
  - Most ethernet cards allow one to change address

MAC spoofing

- Many LANs, WiFis use MAC-based access controls



Changing Your MAC Address/Mac OS X

< Changing Your MAC Address

Under Mac OS X, the MAC address can be altered in a fashion similar to the Linux and FreeBSD methods:

```
ifconfig en0 lladdr 02:01:02:03:04:05
```

or

```
ifconfig en0 ether 02:01:02:03:04:05
```

This must be done as the superuser and only works for the computer's ethernet card. Instructions on spoofing

Courtesy of wikibooks
http://en.wikibooks.org/wiki/Changing_Your_MAC_Address/Mac_OS_X

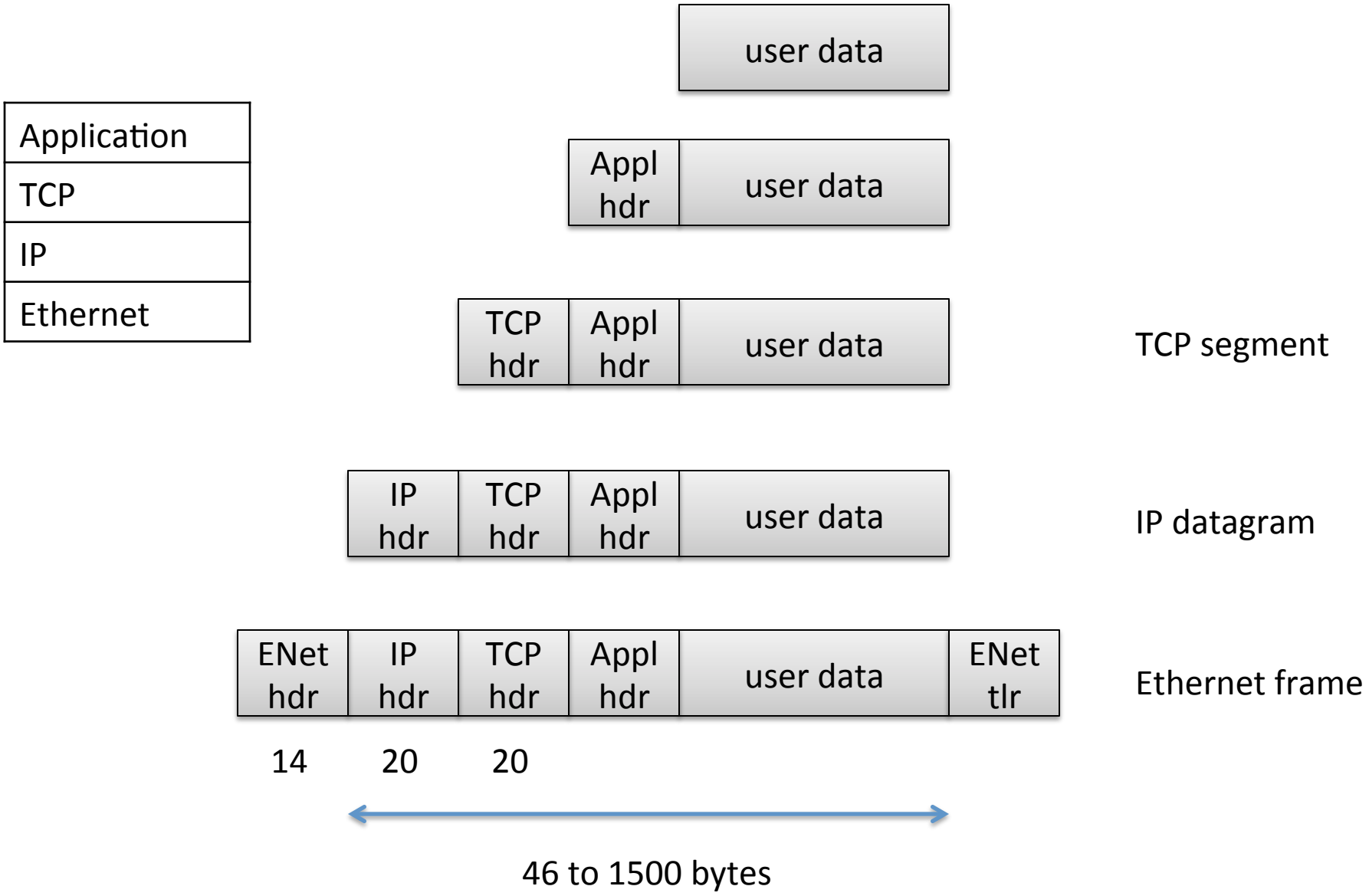
MAC spoofing

Aaron Swartz, a fellow at Harvard University's Center for Ethics and an open source programmer involved with creating the RSS 1.0 specification and more generally in the open culture movement, has been arrested and charged with wire fraud, computer fraud, unlawfully obtaining information from a protected computer, and recklessly damaging a protected computer after he entered a computer lab at MIT in Cambridge, Massachusetts and downloaded two-thirds of the material on JSTOR, an academic journal repository.

http://en.wikinews.org/wiki/Aaron_Swartz_arrested_and_charged_for_downloading_JSTOR_articles

Supposedly used MAC spoofing to get onto MIT network

Internet protocol stack

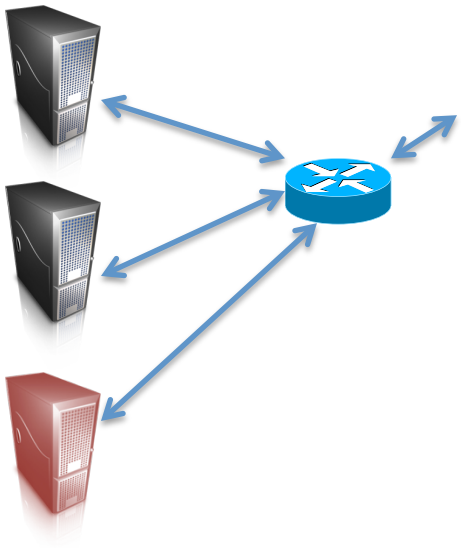


IPv4

Ethernet frame containing IP datagram

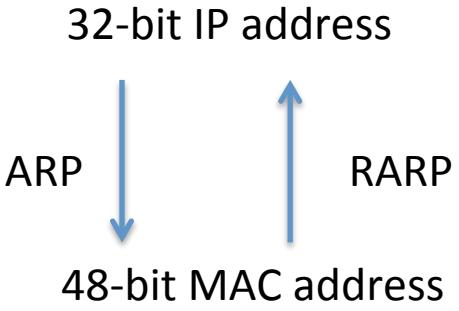
4-bit version	4-bit hdr len	8-bit type of service	16-bit total length (in bytes)	
16-bit identification			3-bit flags	13-bit fragmentation offset
8-bit time to live (TTL)		8-bit protocol	16-bit header checksum	
32-bit source IP address				
32-bit destination IP address				
options (optional)				

Address resolution protocol



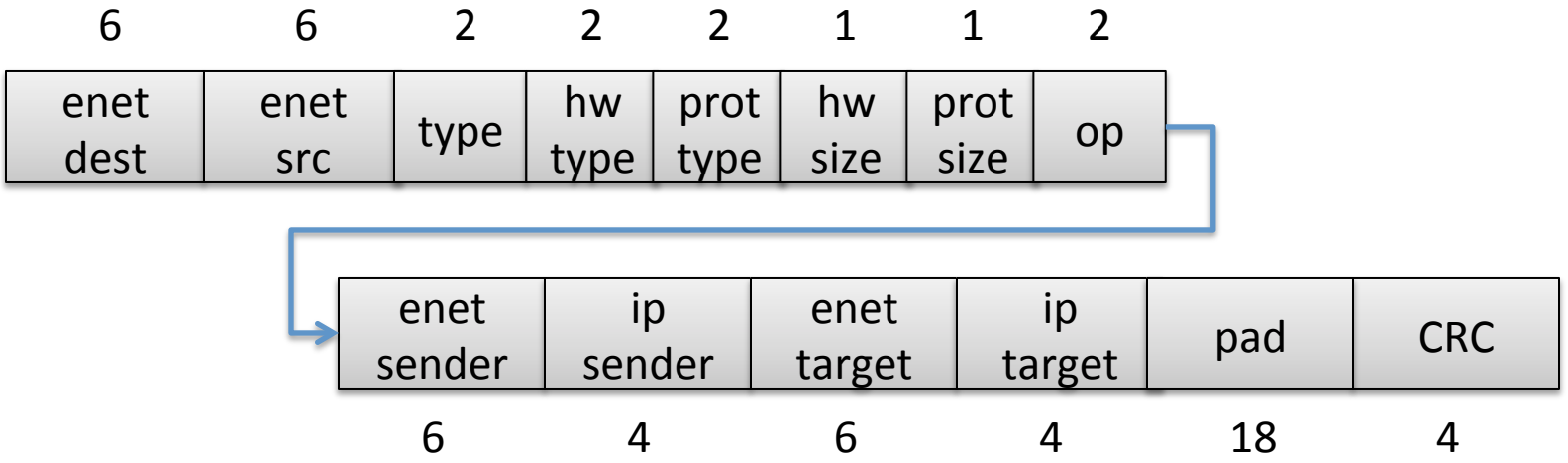
IP routing:
Figure out where to send
an IP packet based on destination address.

Link layer and IP must cooperate to get things sent



ARP/RARP enables this cooperation by mapping IPs to MACs

Address resolution protocol



frame type = 0x0806 (ARP) or 0x8035 (RARP)

enet dest is all 1's, 0xFFFFFFFFFFFF for broadcast

hw type, prot(ocol) type specify what types of addresses we're looking up

op specifies whether this is an ARP request, ARP reply, RARP request, RARP reply

ARP caches

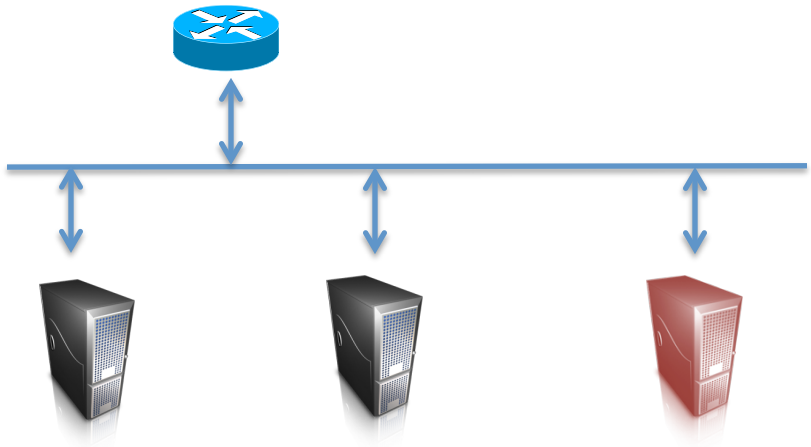
- Hosts maintain cache of ARP data
  - just a table mapping between IPs and MACs

```
rist@wifi-212:~/work/teaching/642-fall-2011/slides$ arp
usage: arp [-n] [-i interface] hostname
       arp [-n] [-i interface] -a
       arp -d hostname [pub] [ifscope interface]
       arp -d [-i interface] -a
       arp -s hostname ether_addr [temp] [reject] [blackhole] [pub [only]] [ifsc
ope interface]
       arp -S hostname ether_addr [temp] [reject] [blackhole] [pub [only]] [ifsc
ope interface]
       arp -f filename
rist@wifi-212:~/work/teaching/642-fall-2011/slides$ arp -a
? (172.16.219.1) at 0:50:56:c0:0:1 on vmnet1 ifscope permanent [ethernet]
? (172.16.219.255) at (incomplete) on vmnet1 ifscope [ethernet]
? (192.168.1.1) at 98:fc:11:91:73:92 on en1 ifscope [ethernet]
? (192.168.1.255) at (incomplete) on en1 ifscope [ethernet]
? (192.168.38.255) at (incomplete) on vmnet8 ifscope [ethernet]
rist@wifi-212:~/work/teaching/642-fall-2011/slides$
```

ARP has no authentication

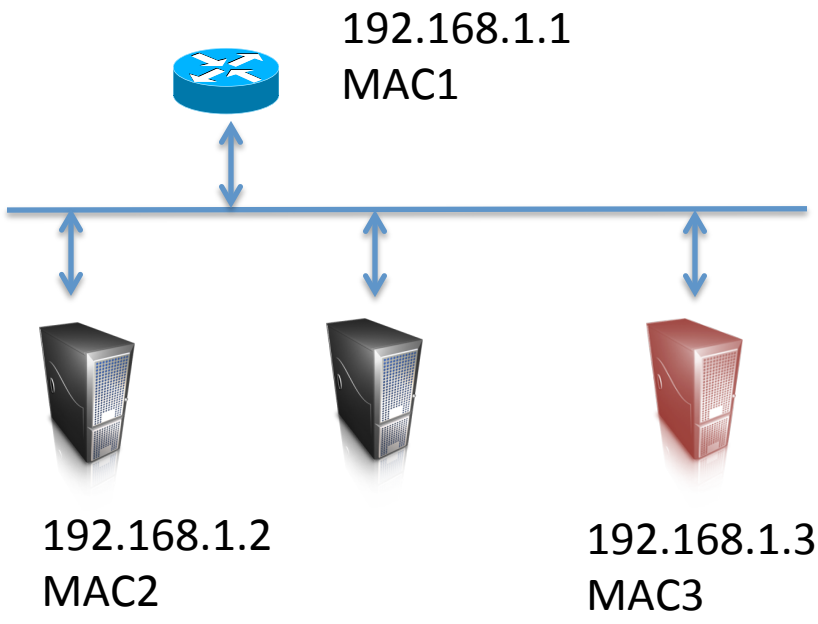
- Easy to sniff packets on (non-switched) ethernet
- What else can we do?

<u>Easy Denial of Service (DoS):</u>
Send ARP reply associating gateway 192.168.1.1 with a non-used MAC address



ARP has no authentication

- Easy to sniff packets on (non-switched) ethernet
- What else can we do?



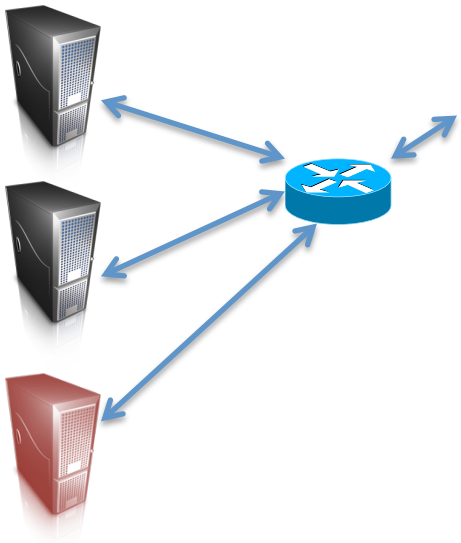
Active Man-in-the-Middle:

ARP reply to MAC2
192.168.1.1 -> MAC3

ARP reply to MAC1
192.168.1.2 -> MAC3

Now traffic “routed” through malicious box

ARP and switched networks

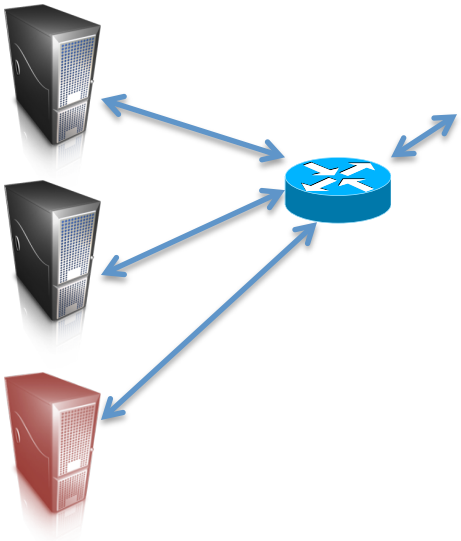


Switches do not broadcast, but transfer traffic through appropriate ports

Inhibits traffic sniffing

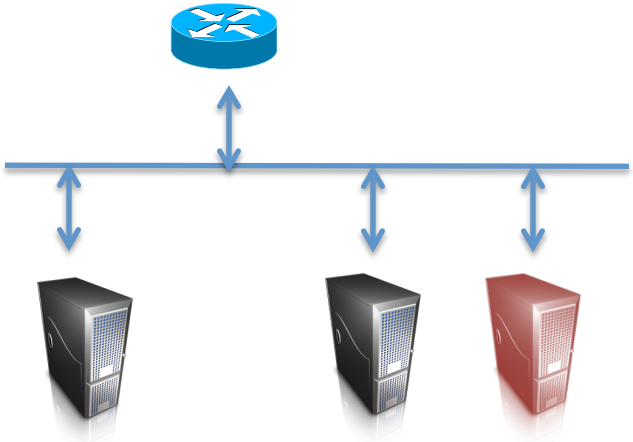
ARP poisoning MitM inhibited (one MAC address per port)

Some switches allow MAC flooding attacks



Flood ARP replies to switch

Switch can't store all values, fails to broadcast



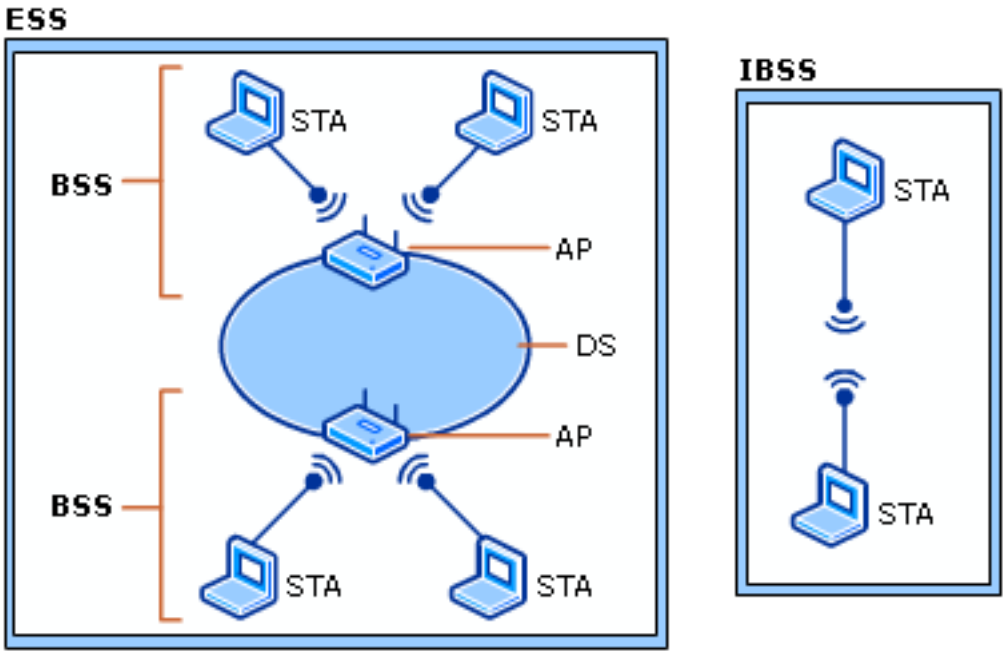
Detection and prevention

- ARPWATCH
  - logs ARP mapping changes
  - emails admin if something suspicious comes up
- Switched networks with real authentication

802.11

STA = station
BSS = basic service set
DS = distribution service
ESS = extended service set

SSID (service set identifier) identifies the 802.11 network

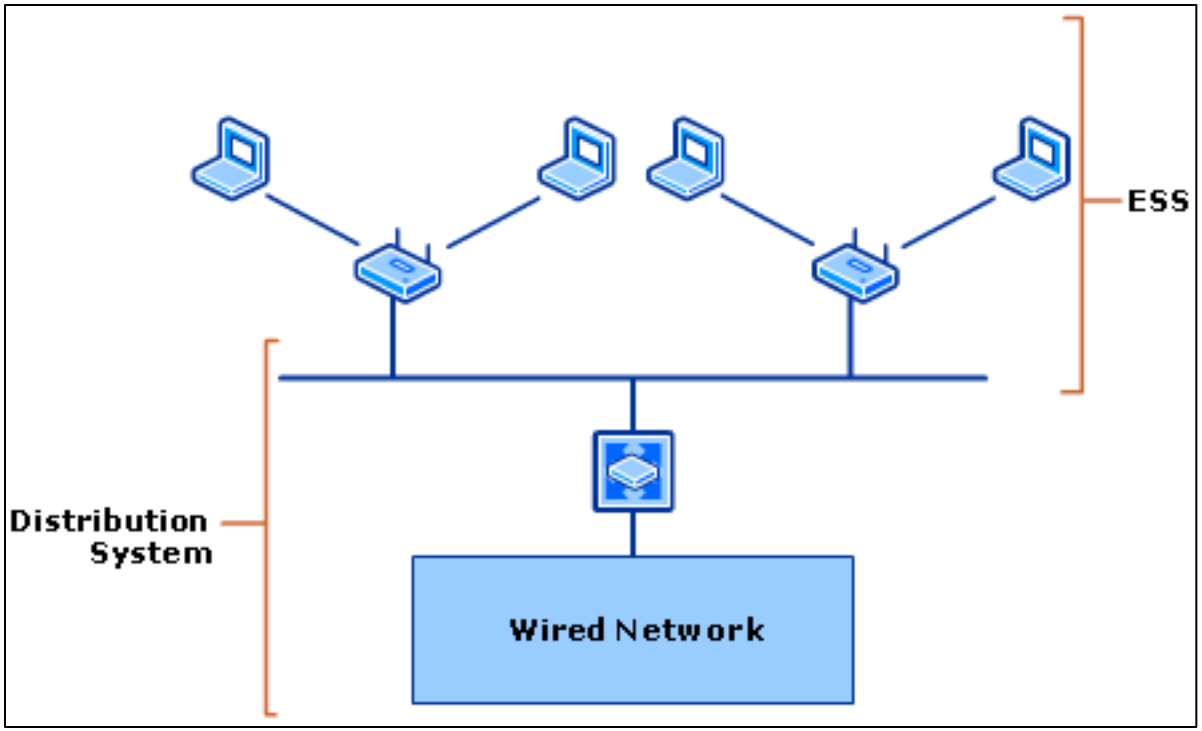


http://technet.microsoft.com/en-us/library/cc757419(WS.10).aspx

802.11

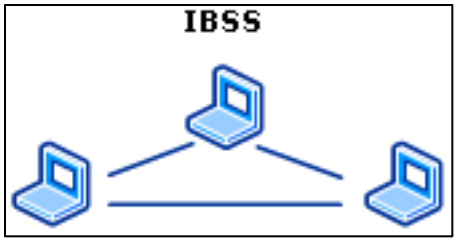
STA = station
BSS = basic service set
DS = distribution service
ESS = extended service set

SSID (service set identifier) identifies the 802.11 network

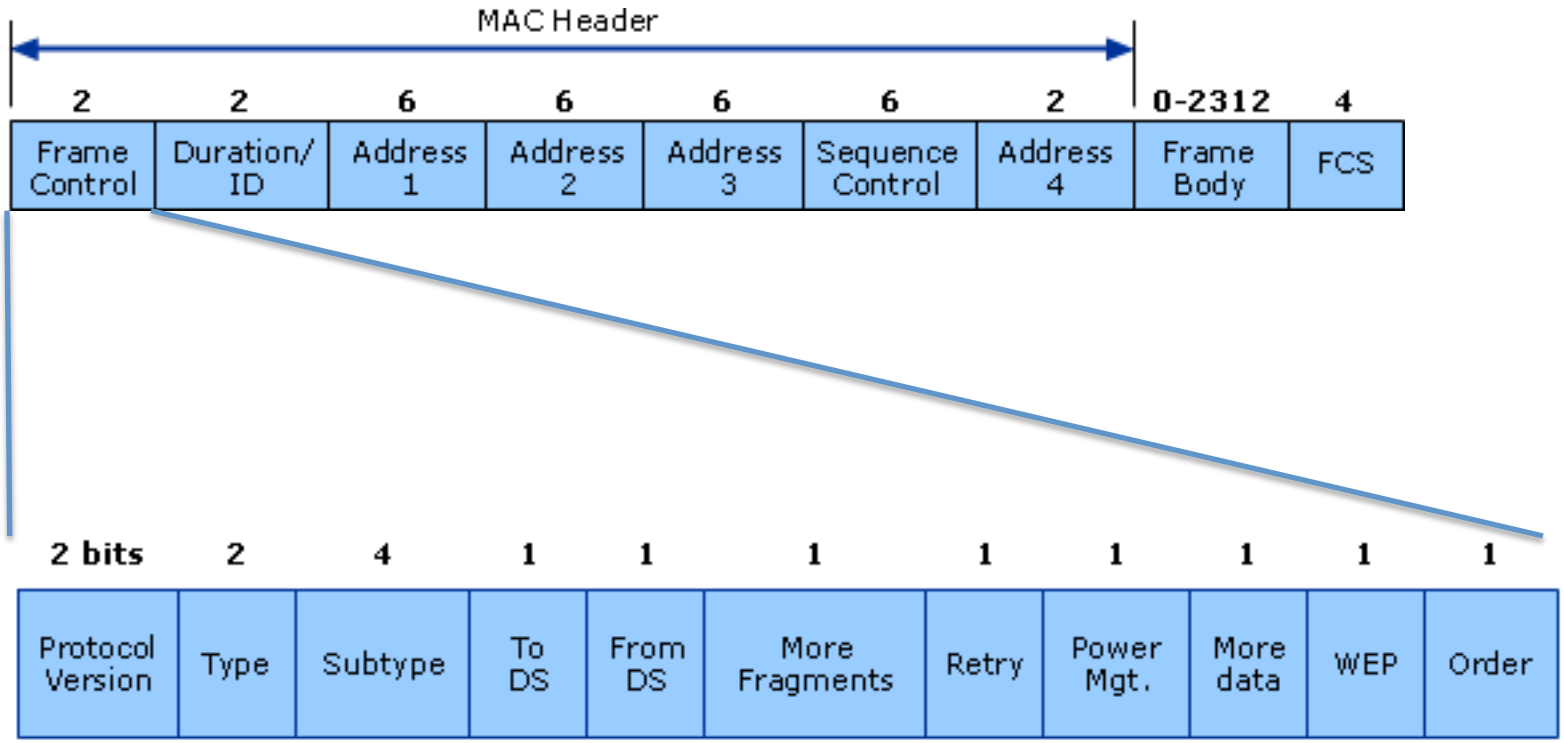


http://technet.microsoft.com/en-us/library/cc757419(WS.10).aspx

Infrastructure mode (top)
versus
Ad-hoc (bottom)

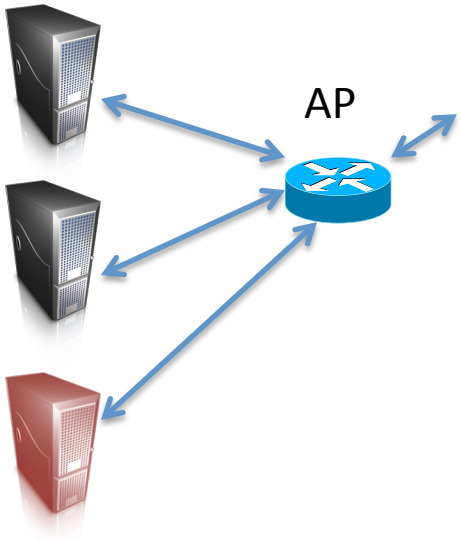


802.11



Images from http://technet.microsoft.com/en-us/library/cc757419(WS.10).aspx

802.11 security issues

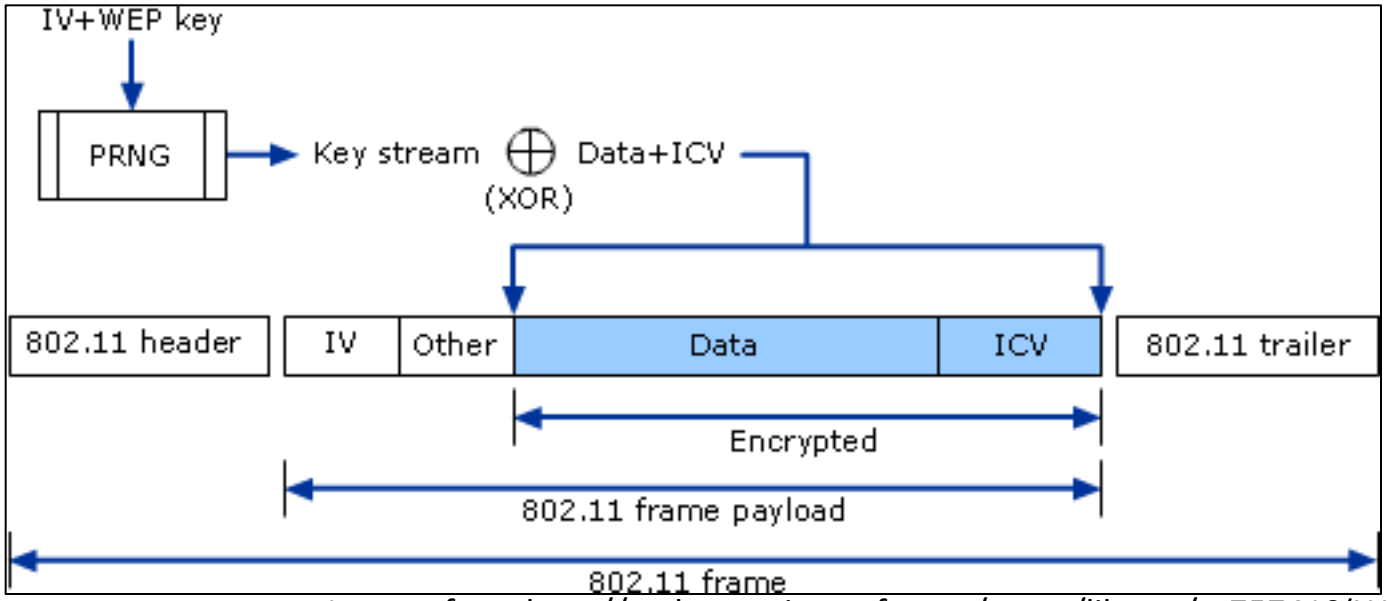


Wired versus wireless

Wireless can (try to) compensate via cryptography

- WEP = epic failure
- WPA = better, but not great
- WPA2 = better yet, but not perfect

We'll see more on this in crypto section



Images from http://technet.microsoft.com/en-us/library/cc757419(WS.10).aspx

December 17, 2009

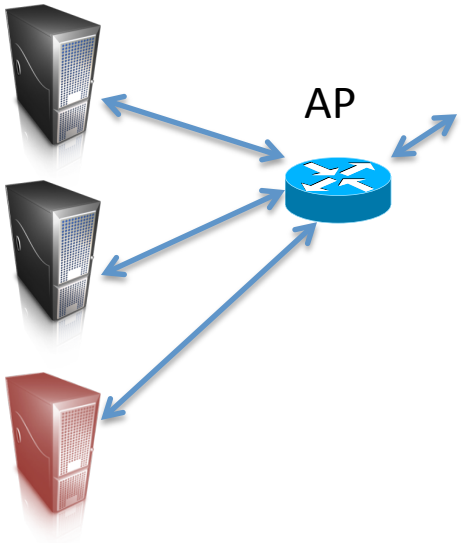
Insurgents Hack U.S. Drones

$26 Software Is Used to Breach Key Weapons in Iraq; Iranian Backing Suspected

http://online.wsj.com/article/SB126102247889095011.html?mod=googlenews_wsj

Interesting report on drone usage by US: http://livingunderdrones.org/

802.11 security issues



WPA-personal

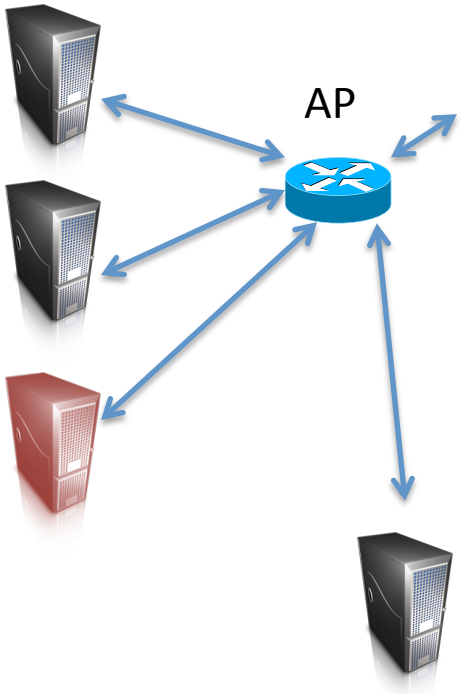
- Pre-shared key mode
- User types in a password to gain access

Default settings

- IP address: 192.168.1.1 (WRT54G-TM and WRT54G-RG: 192.168.0.1)
- Web interface username: "admin" for most routers, no user name or "root" on some
- Password: "admin"

http://en.wikipedia.org/wiki/Linksys_WRT54G_series

802.11 security issues



WPA-personal

- Pre-shared key mode
- User types in a password to gain access

WPA-enterprise

- Extended Authentication Protocol (EAP)
- Centralized Authentication, Authorization, and Accounting (AAA)

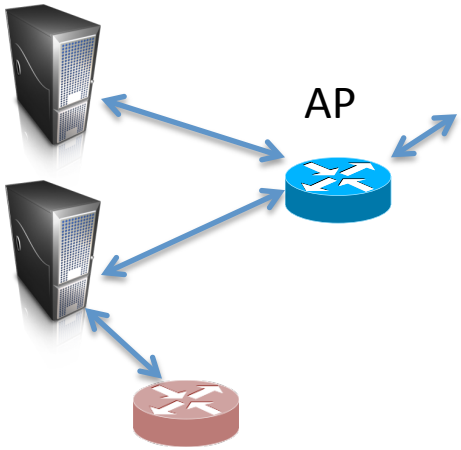
RADIUS (Remote Authentication Dial In User Service) authentication server

Client-server protocol over UDP

1) Authenticate users/devices before granting access to network
2) Authorize users/devices to access certain network services
3) Account for usage of services

Many security issues identified

802.11 evil twins



Basic idea:

- Attacker pretends to be an AP to intercept traffic or collect data

802.11 association

Probe request

SSID: “linksys”, BSSID: MAC1

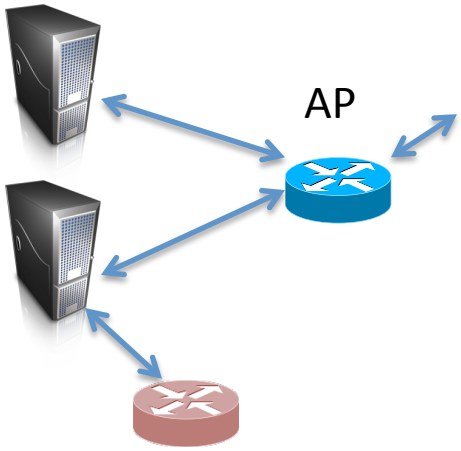
Auth request MAC1

Auth response

Associate request MAC1

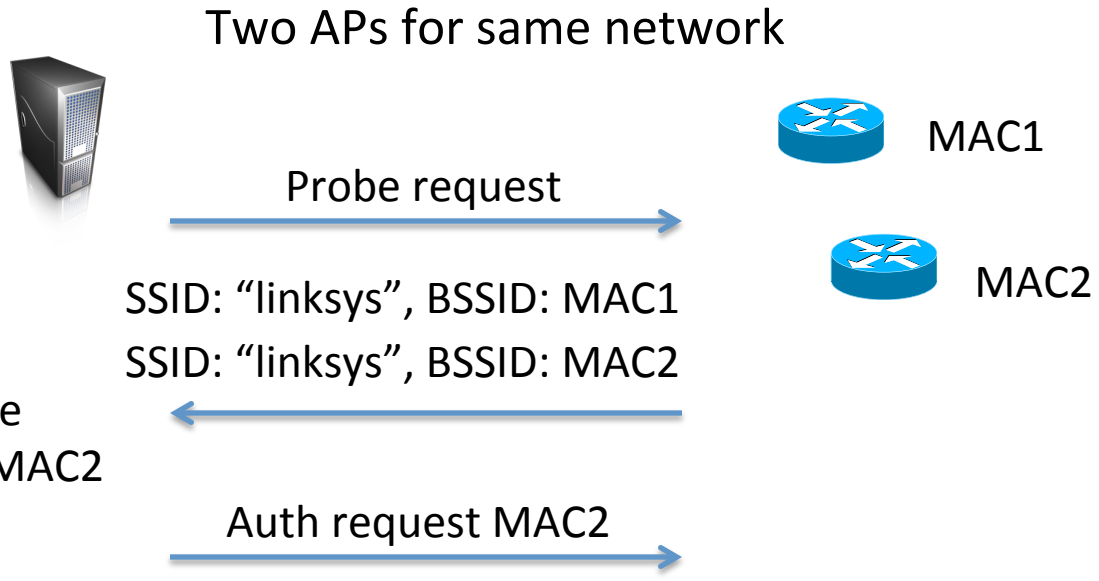
Associate response

802.11 evil twins

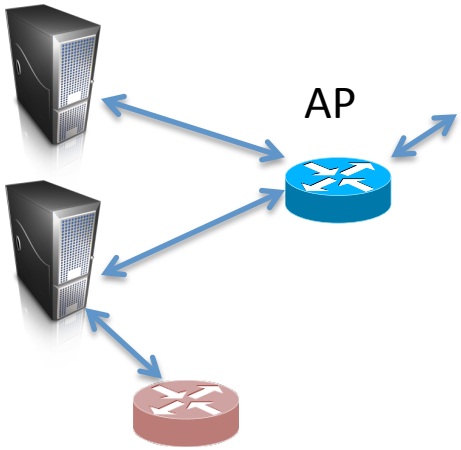


Basic idea:

- Attacker pretends to be an AP to intercept traffic or collect data



802.11 evil twins



Basic idea:

- Attacker pretends to be an AP to intercept traffic or collect data

Basic attack: rogue AP

MAC1

Probe request

MAC2

SSID: “linksys”, BSSID: MAC1

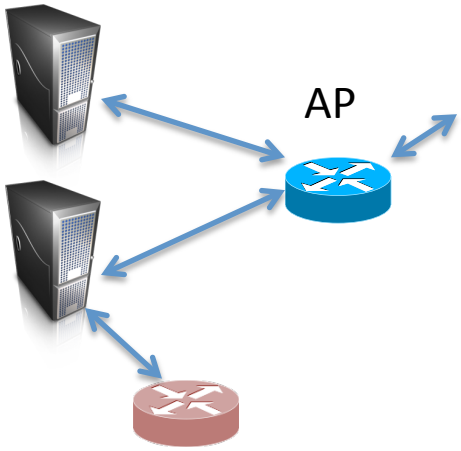
SSID: “linksys”, BSSID: MAC2

Choose one of MAC1, MAC2

Auth request MAC2

...

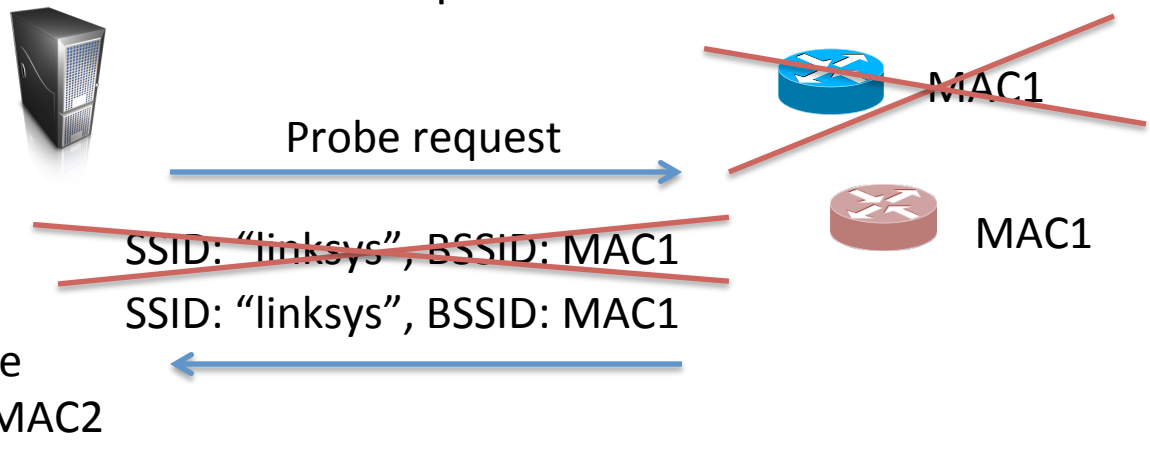
802.11 evil twins



Basic idea:

- Attacker pretends to be an AP to intercept traffic or collect data

Evil twin: spoof MAC1



Choose one of MAC1, MAC2

Auth request MAC2

Attacker can send forged disassociate message to victim to get it to look for new connection

...

Victim might send out probe requests for particular SSIDs, giving attacker info

Conceptually similar to ARP poisoning

Push-button configuration (PBC)

AP

Problems with WPA-personal:

- Users are scared of passwords
- Passwords usually weak
- New devices lack keypads

Push button

PBC probe

PBC probe

PBC probe

Push button

PBC response

Diffie-Hellman Key exchange

shared secret

shared secret

Push-button configuration (PBC)

Push button

PBC probe

PBC probe

PBC response

PBC response

Push button

Diffie-Hellman
Key exchange

Diffie-Hellman
Key exchange

shared secret 1

shared secret 2

shared
secret 1

shared
secret 2

But this is on wireless, so all messages are seen by all parties

Attacker can jam messages, overpower legitimate messages

Can we prevent MitM?

Gollakata et al., Secure In-Band Wireless Pairing, Security 2011

Basic observations:

- Assume all parties in range of each other (all honest broadcasts seen)
- Signals cannot be negated
- Jamming can be made detectable

Can we prevent MitM?

Gollakata et al., Secure In-Band Wireless Pairing, Security 2011

Tamper-evident Announcement:

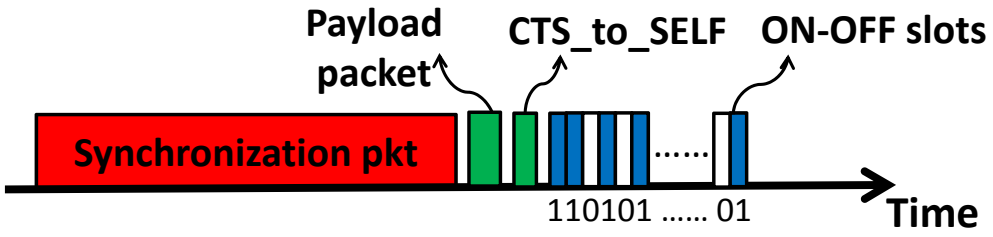


Figure 1: **The format of a tamper-evident announcement (TEA).**

Synchronization: long random data to make overpowering detectable

Payload: key exchange data (public key, etc.)

On-Off slots:

Encode cryptographic hash of payload in a manipulation-detectable way

Can we prevent MitM?

Gollakata et al., Secure In-Band Wireless Pairing, Security 2011

On-Off slots:

Encode cryptographic hash of payload in a manipulation-detectable way

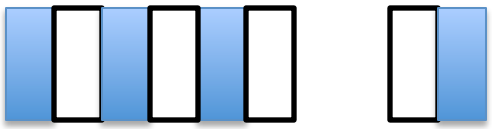
b1 b2 h1 h2 … h128

1 0 1 1 … 0

1 0 10 10 … 01

Encode in a way that balances number of 0's and 1's

Attacker can only turn 0's to 1's



Transmitting a 1:
send packet with random data
Transmitting a 0:
send nothing

Receiver detects if channel in use, concludes a 1
Otherwise concludes a 0
Checks that # of 1's = # of 0's
Checks hash of payload

To change payload, attacker must change hash value, but can't

Discussion

- What attacks aren't prevented?
- PBC relies on what physical assumptions?
- How easy are such jamming based attacks?