19. Network Attacks I

Blase Ur and David Cash
(some slides borrowed from Ben Zhao, Christo Wilson, & others)
February 23rd, 2022
CMSC 23200 / 33250

THE UNIVERSITY OF CHICAGO

Network threat model

- Network scanning
- Attacks on confidentiality
  (e.g., eavesdropping, side channel information)
- Attacks on integrity
  (e.g., spoofing, packet injection)
- Attacks on availability
  (e.g., denial of service, or **DoS**)

Scanning and observing networks

Network Scanning: Ping

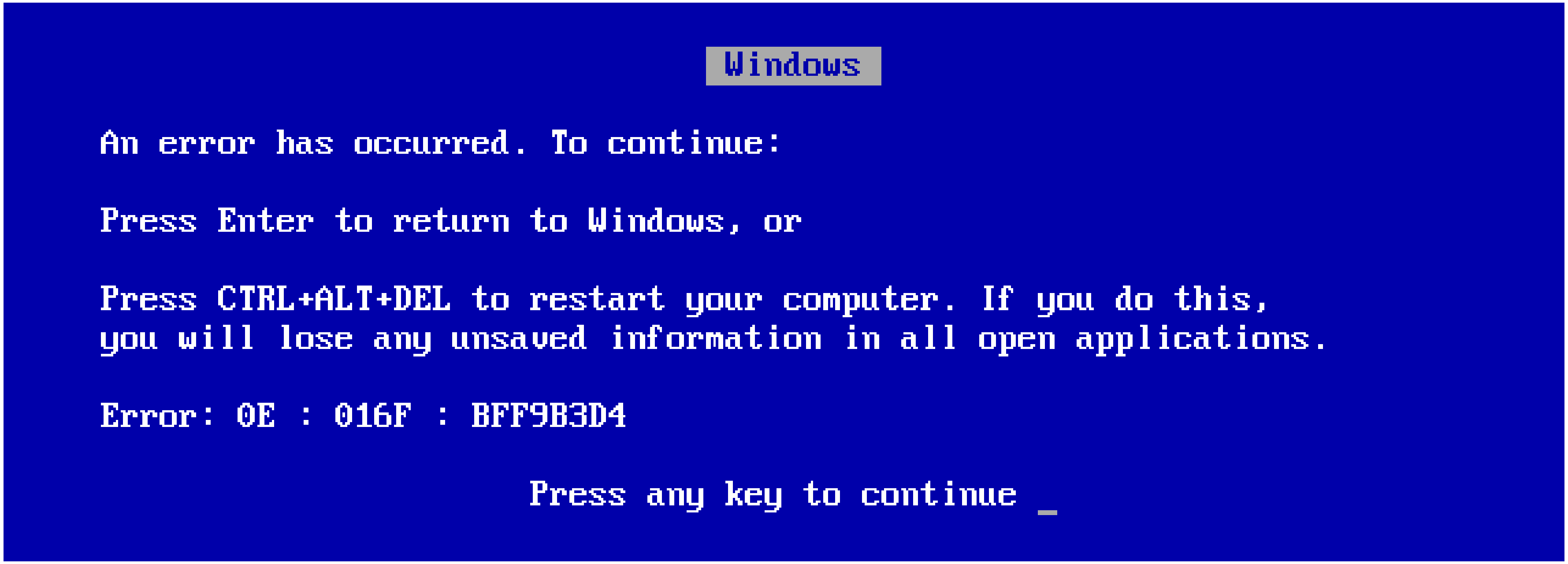
- Essential, low-level network utility
- Sends a “ping” ICMP message to a host on the internet

```
$ ping 66.66.0.255
PING 66.66.0.255 (66.66.0.255) 56(84) bytes of data.
64 bytes from 66.66.0.255: icmp_seq=1 ttl=58 time=41.2 ms
```

- Destination host is supposed to respond with a “pong”
  – Indicating that it can receive packets
- By default, ping messages are 56 bytes long (+ some header bytes)
  – Maximum size 65535 bytes
- What if you send a ping that is >65535 bytes long?

Ping of Death

- $ ping –s 65535 66.66.0.255
  - Attack identified in 1997
  - IPv6 version identified/fixed in 2013



Network Scanning: Traceroute

- traceroute — hops between me and host
  - Sends repeated ICMP reqs w/ increasing TTL

```
thor Wed Oct 24(12:51am)[~]:-> traceroute www.slack.com
traceroute to www.slack.com (52.85.115.213), 64 hops max, 52 byte packets
 1  v11router (128.135.11.1)  1.265 ms  0.788 ms  0.778 ms
 2  a06-021-100-to-d19-07-200.p2p.uchicago.net (10.5.1.186)  1.292 ms  0.749 ms  0.833 ms
 3  d19-07-200-to-h01-391-300.p2p.uchicago.net (10.5.1.46)  2.124 ms  2.435 ms  2.072 ms
 4  192.170.192.34 (192.170.192.34)  0.755 ms
    192.170.192.32 (192.170.192.32)  0.810 ms  0.701 ms
 5  192.170.192.36 (192.170.192.36)  0.887 ms  0.918 ms  0.877 ms
 6  r-equinix-isp-ae2-2213.wiscnet.net (216.56.50.45)  1.625 ms  1.803 ms  1.866 ms
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  178.236.3.103 (178.236.3.103)  4.516 ms  4.326 ms  4.320 ms
12  * * *
13  * * *
14  * * *
15  server-52-85-115-213.ind6.r.cloudfront.net (52.85.115.213)  4.554 ms  4.398 ms  4.757 ms
thor Wed Oct 24(12:52am)[~]:->
```

Port Scanning

- What services are running on a server? Nmap

```
linux3 Wed Oct 24(12:54am)[~]:-> nmap www.cs.uchicago.edu

Starting Nmap 7.01 ( https://nmap.org ) at 2018-10-24 00:55 CDT
Nmap scan report for www.cs.uchicago.edu (34.203.108.171)
Host is up (0.019s latency).
Other addresses for www.cs.uchicago.edu (not scanned): 54.164.17.80 54.85.61.218
rDNS record for 34.203.108.171: ec2-34-203-108-171.compute-1.amazonaws.com
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.99 seconds
linux3 Wed Oct 24(12:55am)[~]:->
```

- 5 seconds to scan a single machine!!

SYN scan

Only send SYN

Responses:

- SYN-ACK — port open
- RST — port closed
- Nothing — filtered (e.g., firewall)

Port Scanning on Steroids

zmap

- How do you speed up scans for all IPv4?
  - Don’t wait for responses; pipeline
  - Parallelize: divide & conquer IPv4 ranges
  - Randomize permutations w/o collisions
- Result: the zmap tool
  - Scan all of IPv4 in 45mins (gigabit connection)
  - IPv4 in 5 mins (10 gigabit connection)

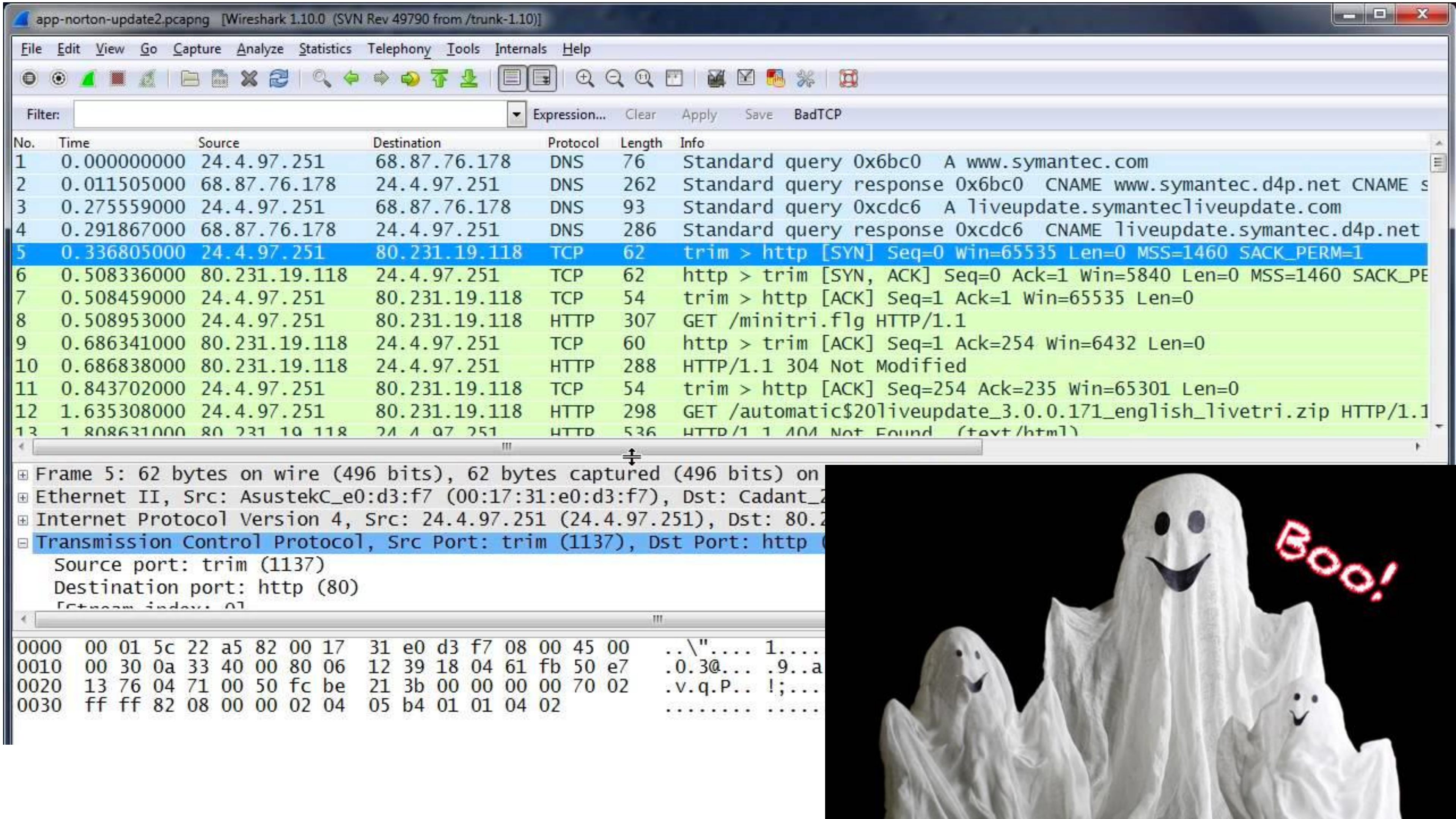
Eavesdropping

Tools: Wireshark, tcpdump, Zeek (Bro), …

Steps:

1. Parse data link layer frames
2. Identify network flows
3. Reconstruct IP packet fragments
4. Reconstruct TCP connections
5. Parse app protocol messages

Wireshark, Detailed Protocol Analyzer



Side channels

Overview

- Transport Layer Security (TLS) enables secure communication
- Frequently encountered with web browsing (HTTPS) and more behind the scenes in app, VOIP, etc.

What Does HTTPS Hide? (Ghost)

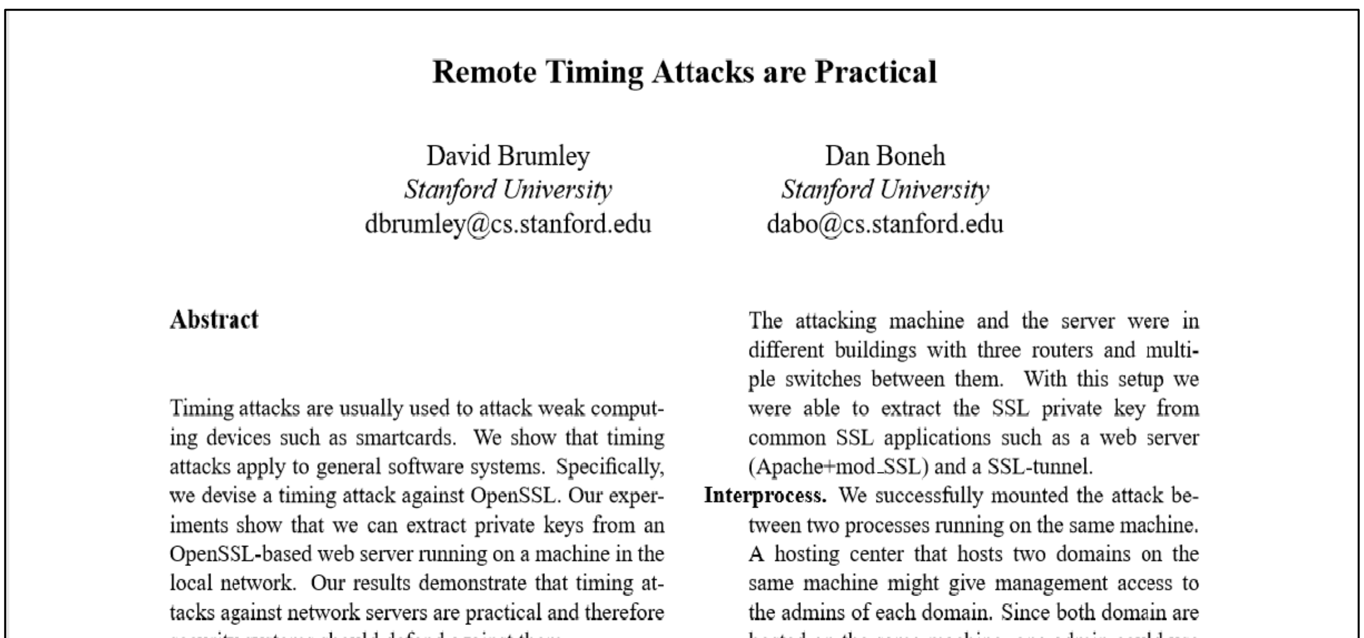
- Body of the HTTP request / response is hidden
- …So what's left to be seen / inferred?

Side Channels

- Using metadata or outside observations to make inferences about the data

Web Side Channels Include:

- Size of packets
  – How can this reveal what pages you are visiting?
- Timing



Remote Timing Attacks are Practical

David Brumley
Stanford University
dbrumley@cs.stanford.edu

Dan Boneh
Stanford University
dabo@cs.stanford.edu

Abstract

Timing attacks are usually used to attack weak computing devices such as smartcards. We show that timing attacks apply to general software systems. Specifically, we devise a timing attack against OpenSSL. Our experiments show that we can extract private keys from an OpenSSL-based web server running on a machine in the local network. Our results demonstrate that timing attacks against network servers are practical and therefore

The attacking machine and the server were in different buildings with three routers and multiple switches between them. With this setup we were able to extract the SSL private key from common SSL applications such as a web server (Apache+mod_SSL) and a SSL-tunnel.

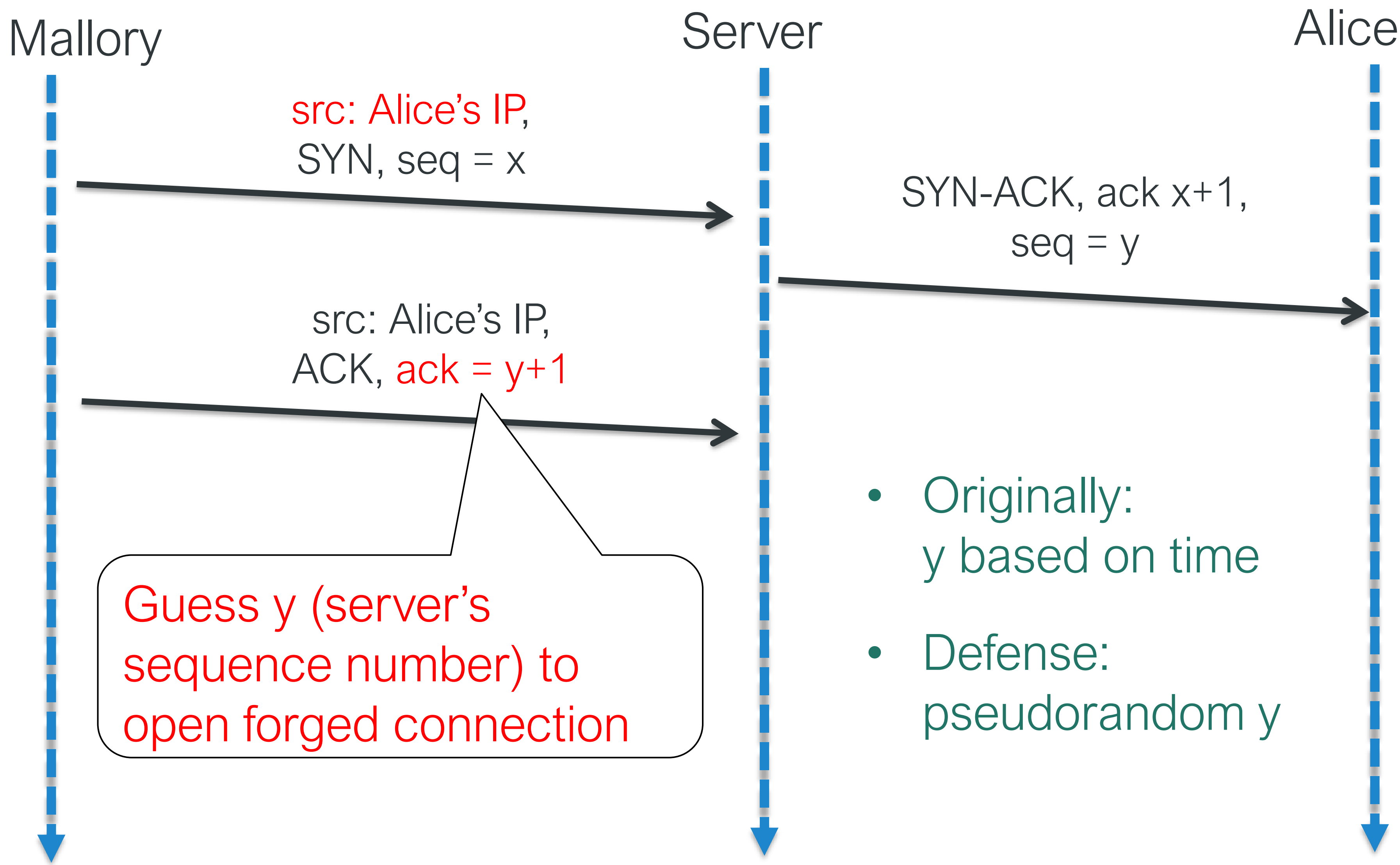
Interprocess. We successfully mounted the attack between two processes running on the same machine. A hosting center that hosts two domains on the same machine might give management access to the admins of each domain. Since both domain are

Web Side Channels Include:

- Color
  - link one
  - second link
  - link three (visited)
  - fourth link

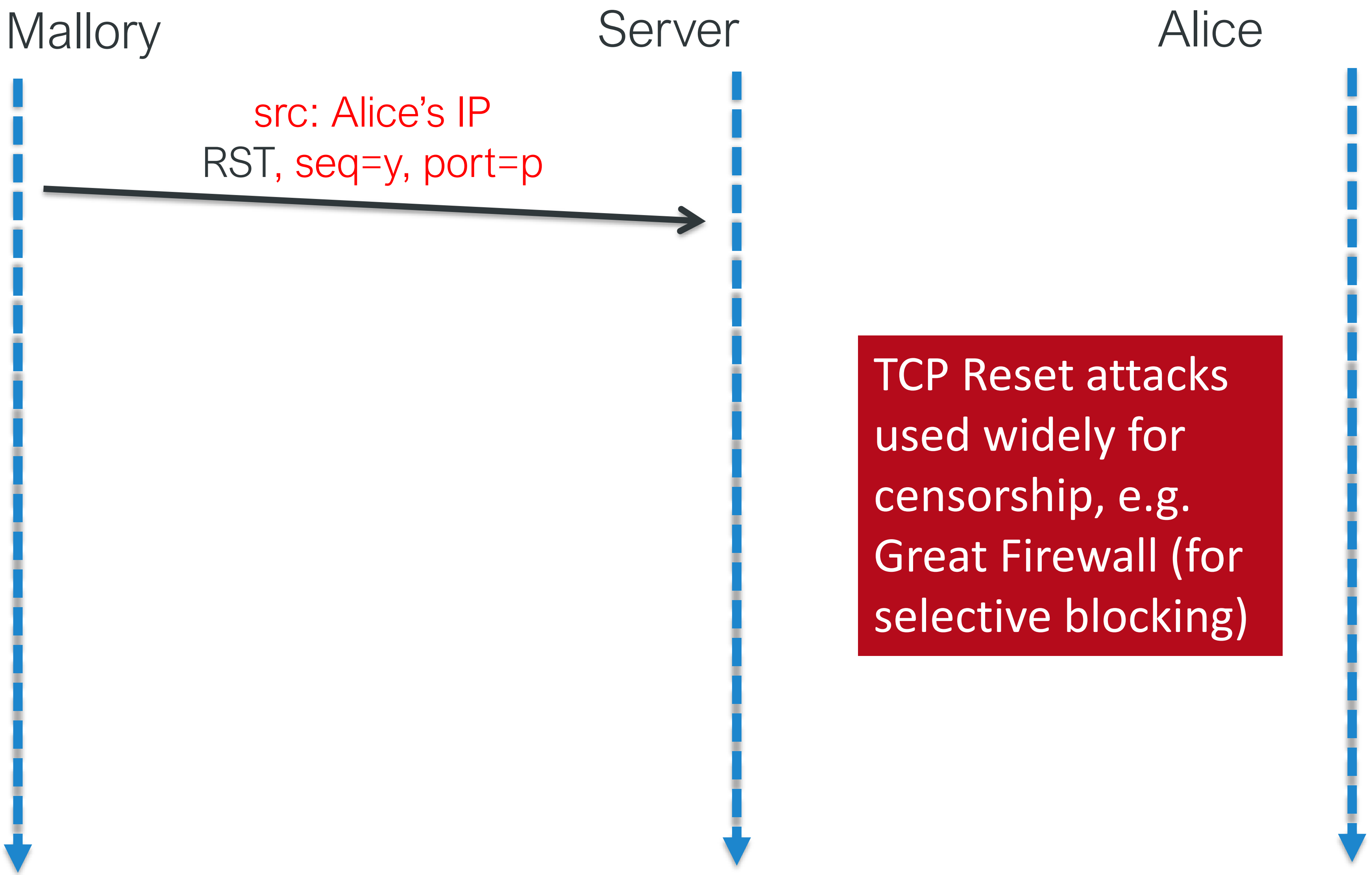
Protocol attacks

Active Attacks: Blind Spoofing



- Originally:
  y based on time
- Defense:
  pseudorandom y

RST Hijacking

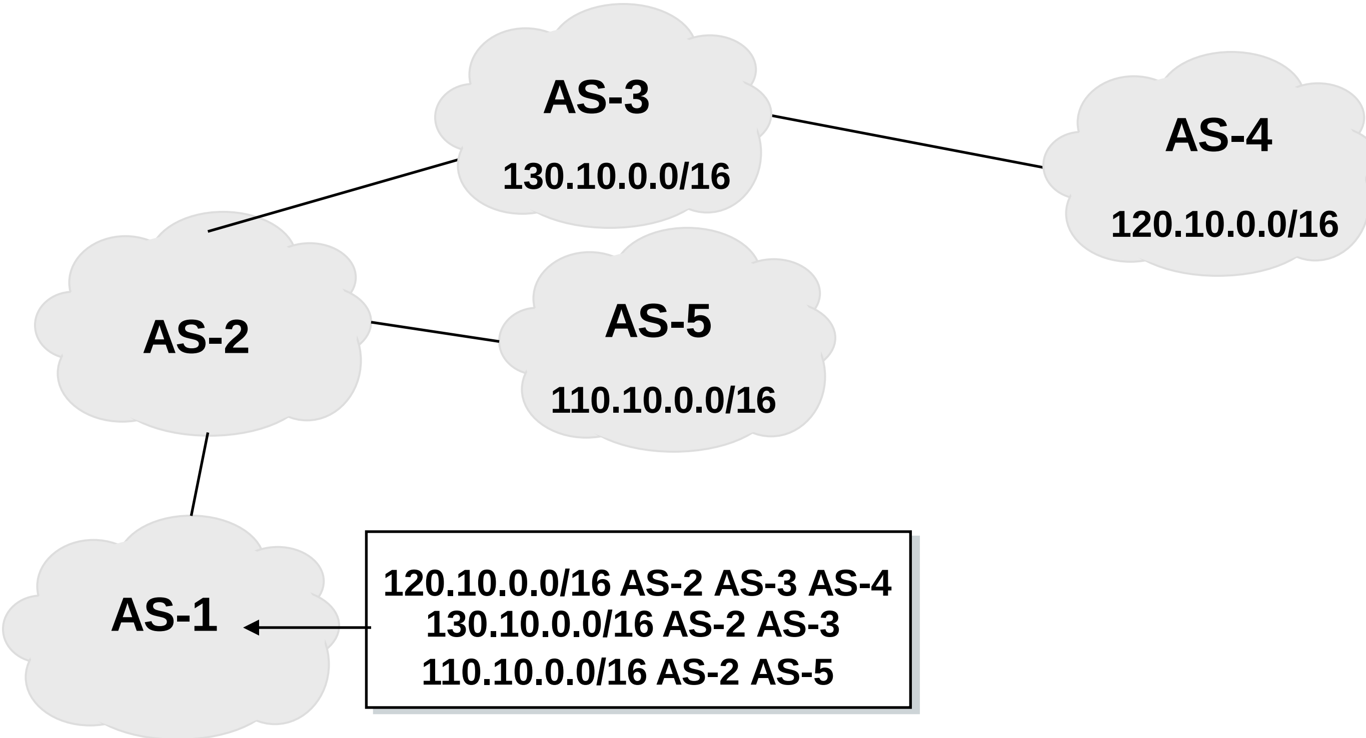


TCP Reset attacks used widely for censorship, e.g. Great Firewall (for selective blocking)

Inter-domain routing (BGP) attacks and large-scale observation

Recall: BGP (Path-Vector Protocol)

- An AS-path: sequence of AS's a route traverses
- Used for loop detection and to apply policy



BGP Prefix Hijacking

- Advertise a more desirable route even if the route isn't actually more desirable, or even real
- Goal 1: Route traffic through networks you control so that you can observe the traffic
- Goal 2: Send lots of traffic to someone you don't like (denial of service)

Corrigendum- Most Urgent

GOVERNMENT OF PAKISTAN
PAKISTAN TELECOMMUNICATION AUTHORITY
ZONAL OFFICE PESHAWAR
Plot-11, Sector A-3, Phase-V, Hayatabad, Peshawar.
Ph: 091-9217279- 5829177 Fax: 091-9217254
www.pta.gov.pk

NWFP-33-16 (BW)/06/PTA February ,2008

Subject: **Blocking of Offensive Website**

Reference: This office letter of even number dated 22.02.2008.

I am directed to request all ISPs to immediately block access to the following website

URL: http://www.youtube.com/watch?v=o3s8jtvvg00

IPs: 208.65.153.238, 208.65.153.253, 208.65.153.251

Compliance report should reach this office through return fax or at email peshawar@pta.gov.pk today please.

Deputy Director
(Enforcement)

To:

1. M/s Comsats, Peshawar.
2. M/s GOL Internet Services, Peshawar.
3. M/s Cyber Internet, Peshawar.
4. M/s Cybersoft Technologies, Islamabad.
5. M/s Paknet, Limited, Islamabad
6. M/s Dancom, Peshawar.
7. M/s Supernet, Peshawar.

BGP Prefix Hijacking

4/25/2019
02:30 PM

Marc Laliberte
Commentary

Connect Directly

0 COMMENTS
COMMENT NOW

Login

100% 0%

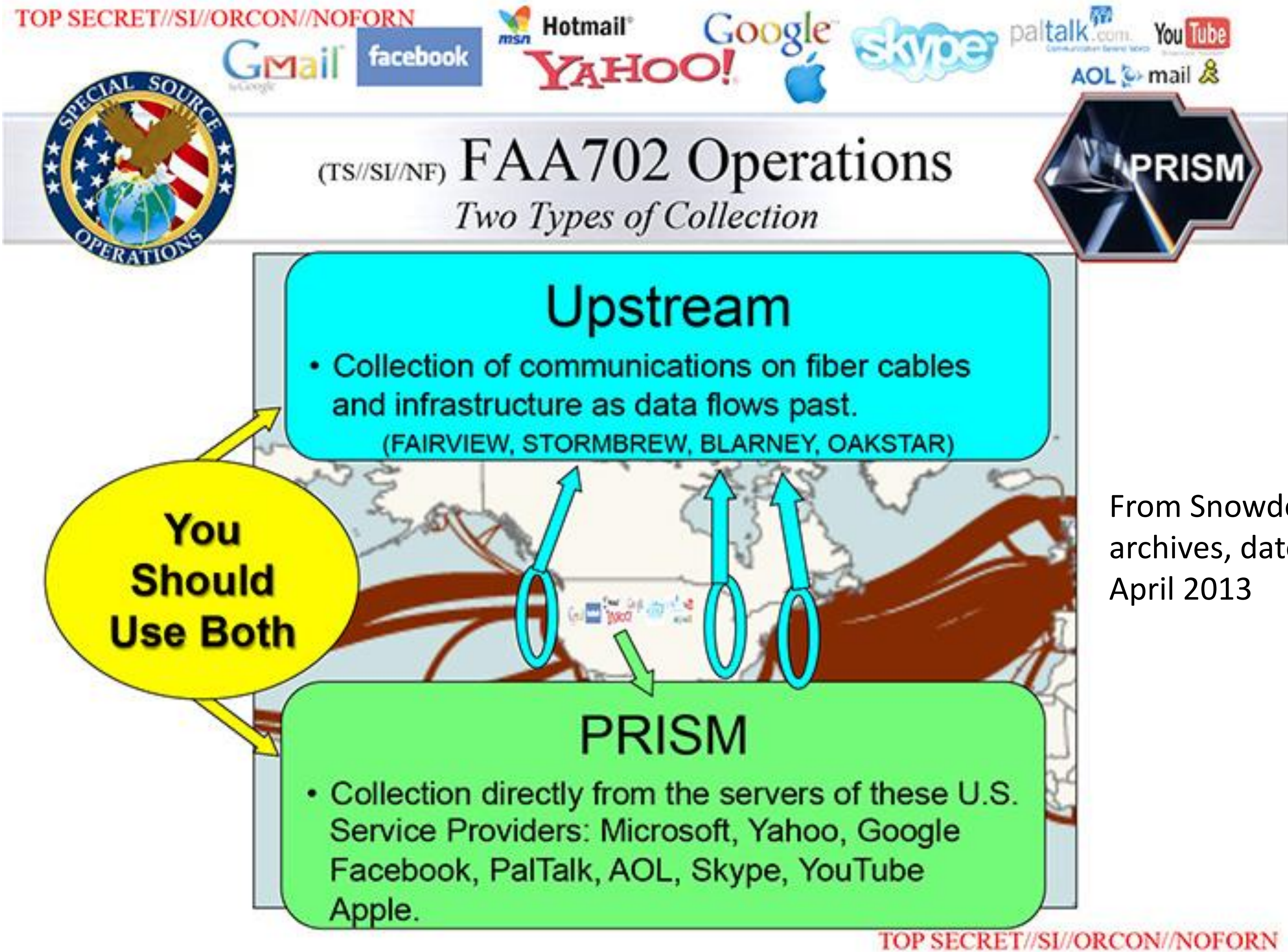
How a Nigerian ISP Accidentally Hijacked the Internet

For 74 minutes, traffic destined for Google and Cloudflare services was routed through Russia and into the largest system of censorship in the world, China's Great Firewall.

On November 12, 2018, a small ISP in Nigeria made a mistake while updating its network infrastructure that highlights a critical flaw in the fabric of the Internet. The mistake effectively brought down Google — one of the largest tech companies in the world — for 74 minutes.

To understand what happened, we need to cover the basics of how Internet routing works. When I type, for example, HypotheticalDomain.com into my browser and hit enter, my computer creates a web request and sends it to Hypothtetical.Domain.com servers. These servers likely reside in a different state or country than I do. Therefore, my Internet service provider (ISP) must determine how to route my web browser's request to the server across the Internet. To maintain their routing tables, ISPs and Internet backbone companies use a protocol called Border Gateway Protocol (BGP).

https://www.darkreading.com/cloud/how-a-nigerian-isp-accidentally-hijacked-the-internet/a/d-id/1334482



From Snowden archives, dated April 2013

(TS//SI//NF) PRISM Collection Details

Current Providers

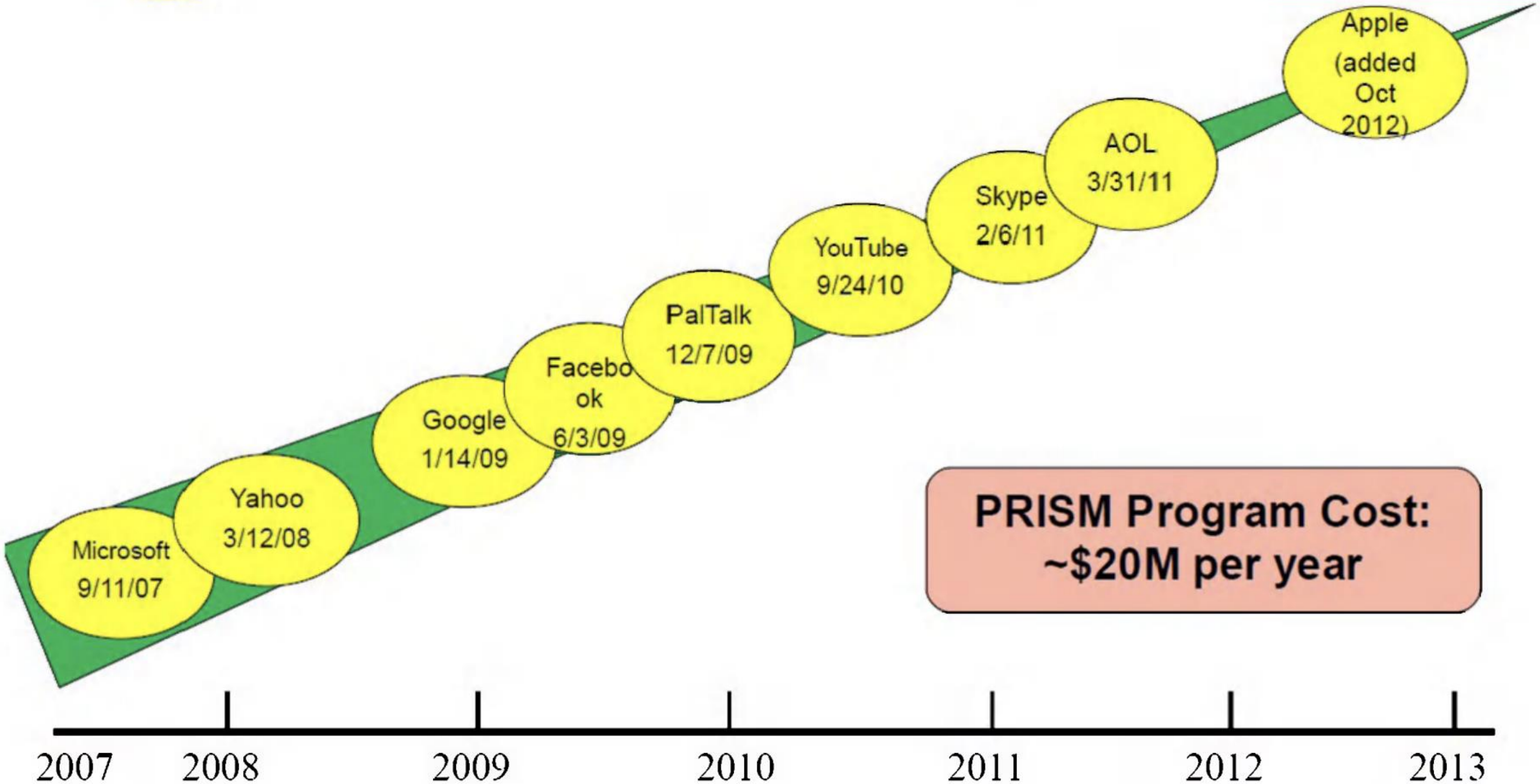
- Microsoft (Hotmail, etc.)
- Google
- Yahoo!
- Facebook
- PalTalk
- YouTube
- Skype
- AOL
- Apple

What Will You Receive in Collection (Surveillance and Stored Comms)?
It varies by provider. In general:

- E-mail
- Chat – video, voice
- Videos
- Photos
- Stored data
- VoIP
- File transfers
- Video Conferencing
- Notifications of target activity – logins, etc.
- Online Social Networking details
- **Special Requests**

Complete list and details on PRISM web page:
Go PRISMFAA

(TS//SI//NF) Dates When PRISM Collection Began For Each Provider



PRISM Program Cost: ~$20M per year

S-BGP / BGPsec

IP prefix announcements signed

Routes signed
— previous hop authorizes next hop

Higher levels vouch for lower levels
— e.g., ICANN vouches for ARIN, ARIN vouches for AT&T, …

Problem?
Costly and slow adoption

HTTP Session Hijacking

Firesheep (now discontinued)

- On shared networks (e.g., wifi), the Firesheep browser extension would sniff session cookies sent unencrypted (over HTTP)