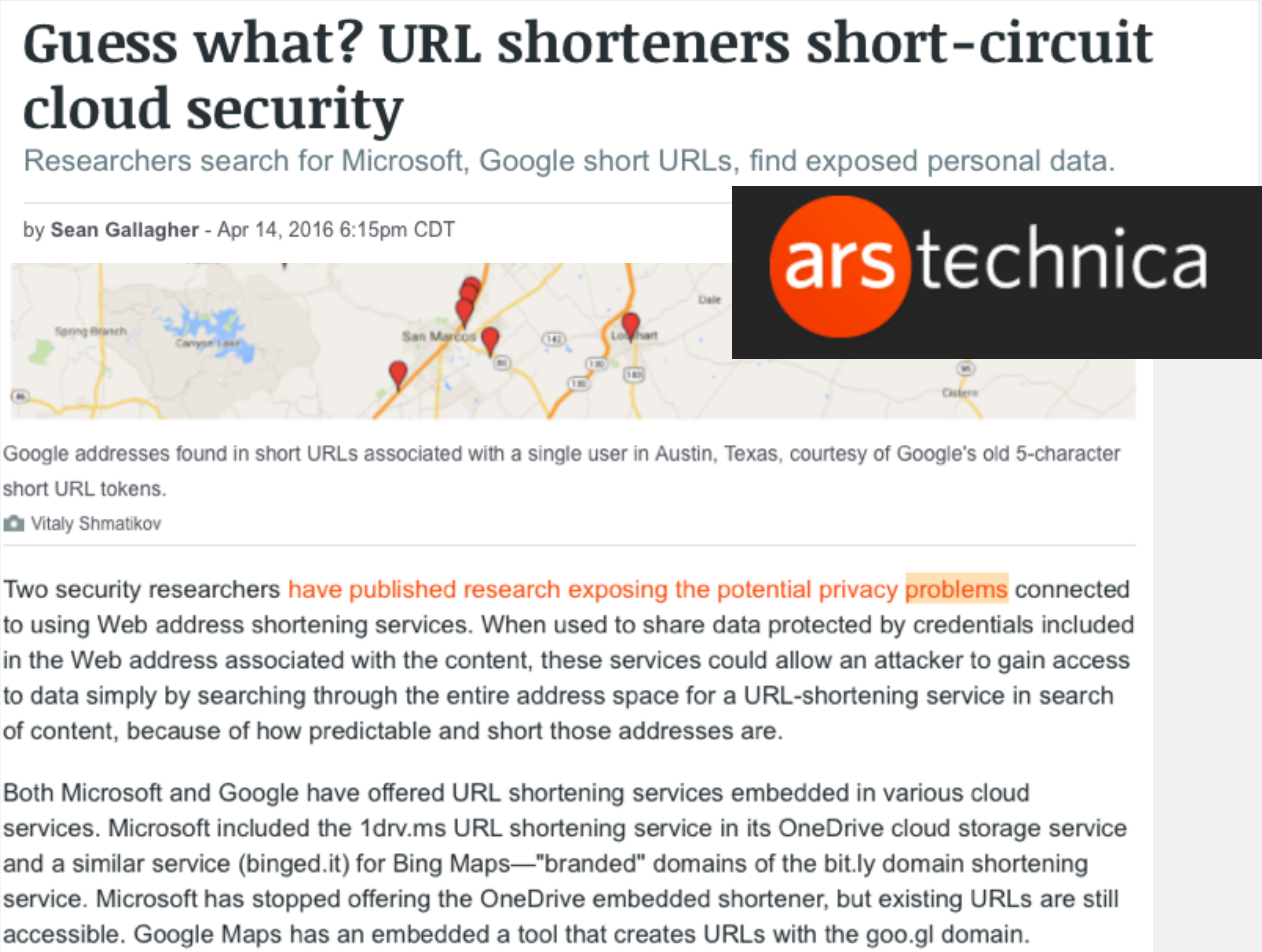


Guess what? URL shorteners short-circuit cloud security

Researchers search for Microsoft, Google short URLs, find exposed personal data.

by **Sean Gallagher** - Apr 14, 2016 6:15pm CDT

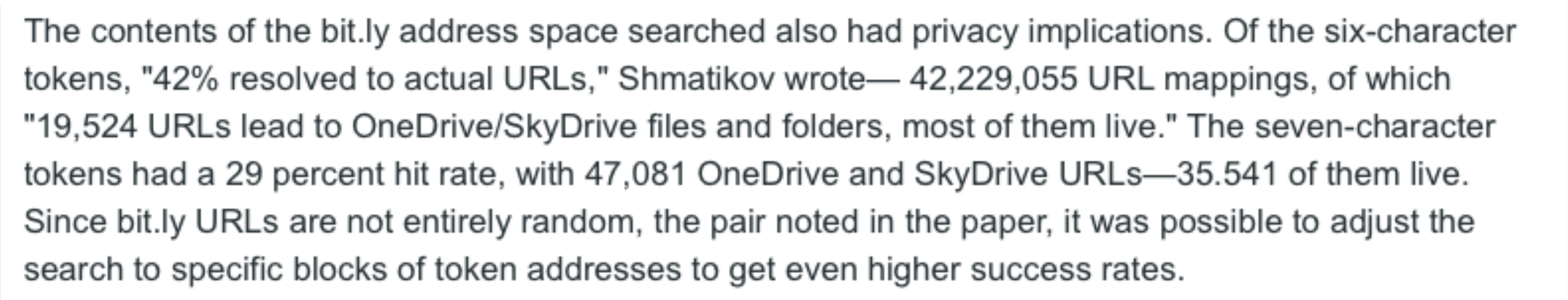
Google addresses found in short URLs associated with a single user in Austin, Texas, courtesy of Google's old 5-character short URL tokens.

Vitaly Shmatikov

ars technica

Two security researchers have published research exposing the potential privacy problems connected to using Web address shortening services. When used to share data protected by credentials included in the Web address associated with the content, these services could allow an attacker to gain access to data simply by searching through the entire address space for a URL-shortening service in search of content, because of how predictable and short those addresses are.

Both Microsoft and Google have offered URL shortening services embedded in various cloud services. Microsoft included the 1drv.ms URL shortening service in its OneDrive cloud storage service and a similar service (binged.it) for Bing Maps—"branded" domains of the bit.ly domain shortening service. Microsoft has stopped offering the OneDrive embedded shortener, but existing URLs are still accessible. Google Maps has an embedded a tool that creates URLs with the goo.gl domain.



The contents of the bit.ly address space searched also had privacy implications. Of the six-character tokens, "42% resolved to actual URLs," Shmatikov wrote— 42,229,055 URL mappings, of which "19,524 URLs lead to OneDrive/SkyDrive files and folders, most of them live." The seven-character tokens had a 29 percent hit rate, with 47,081 OneDrive and SkyDrive URLs—35.541 of them live. Since bit.ly URLs are not entirely random, the pair noted in the paper, it was possible to adjust the search to specific blocks of token addresses to get even higher success rates.

anonymity
& virtualization

cs642
computer security

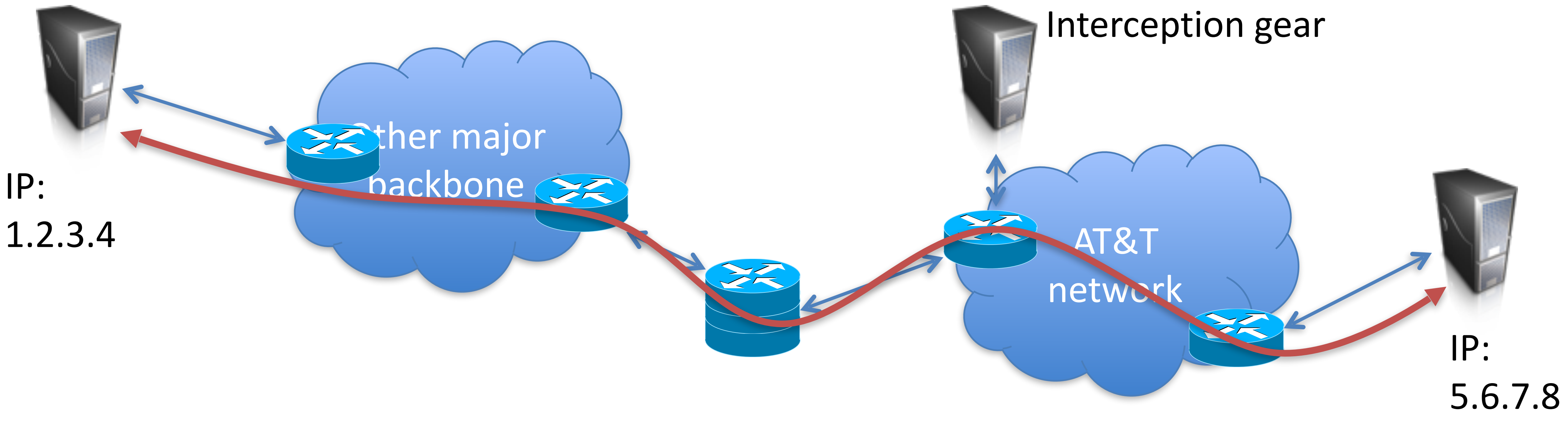
adam everspaugh
ace@cs.wisc.edu

today

- **Announcements:** HW3 due tonight; HW4 posted tomorrow
- Anonymous browsing, TOR
- Virtualization,
- Random number generators and reset vulnerabilities

Preventing intercept

- End-to-end encryption (TLS, SSH)



- What does this protect? What does it leak?
- What can go wrong?

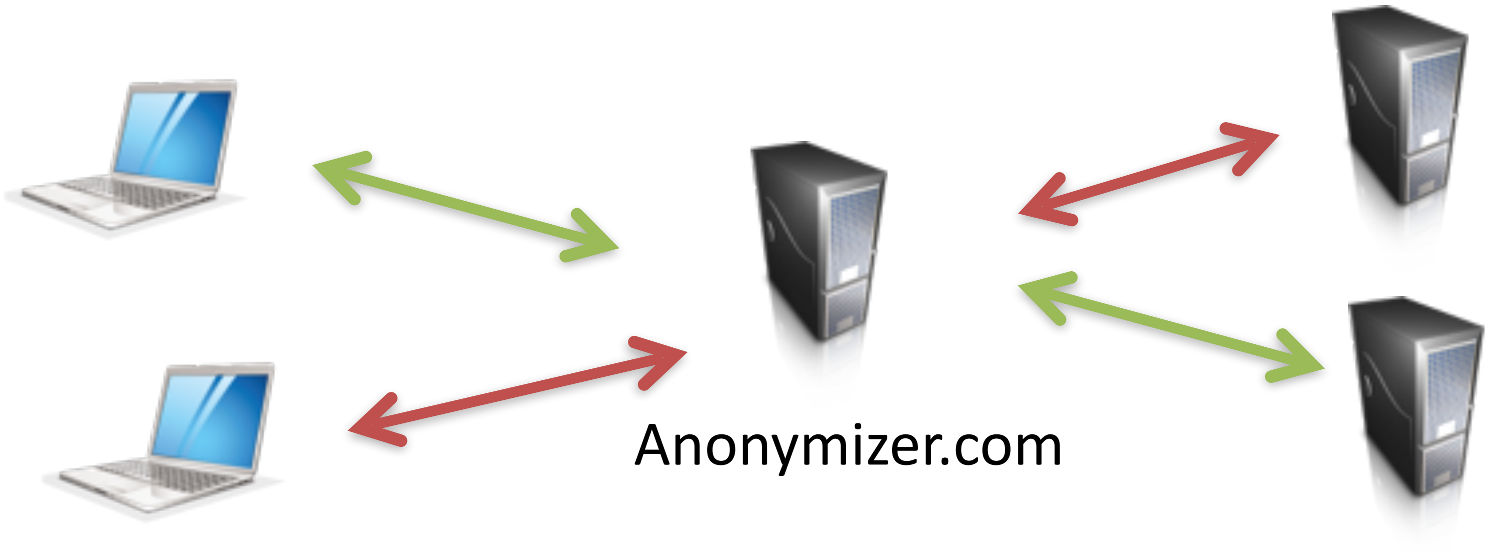
think-*pair*-share

Hiding connectivity is harder

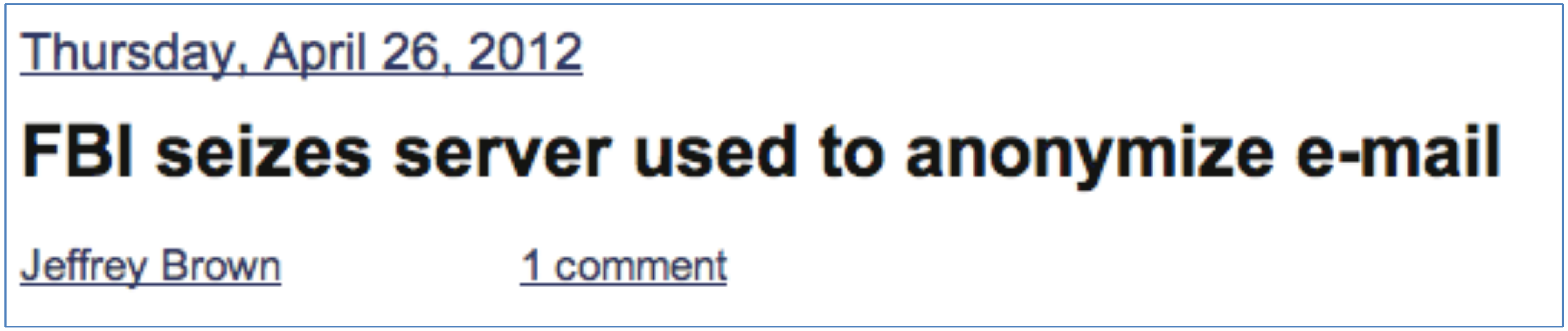
- IP addresses are required to route communication, yet not encrypted by normal end-to-end encryption
  - 1.2.3.4 talked to 5.6.7.8 over HTTPs
- How can we hide connectivity information?

Simple Anonymization Services

- Single-hop proxy services



- JonDonym, anonymous remailers (MixMaster, MixMinion), many others

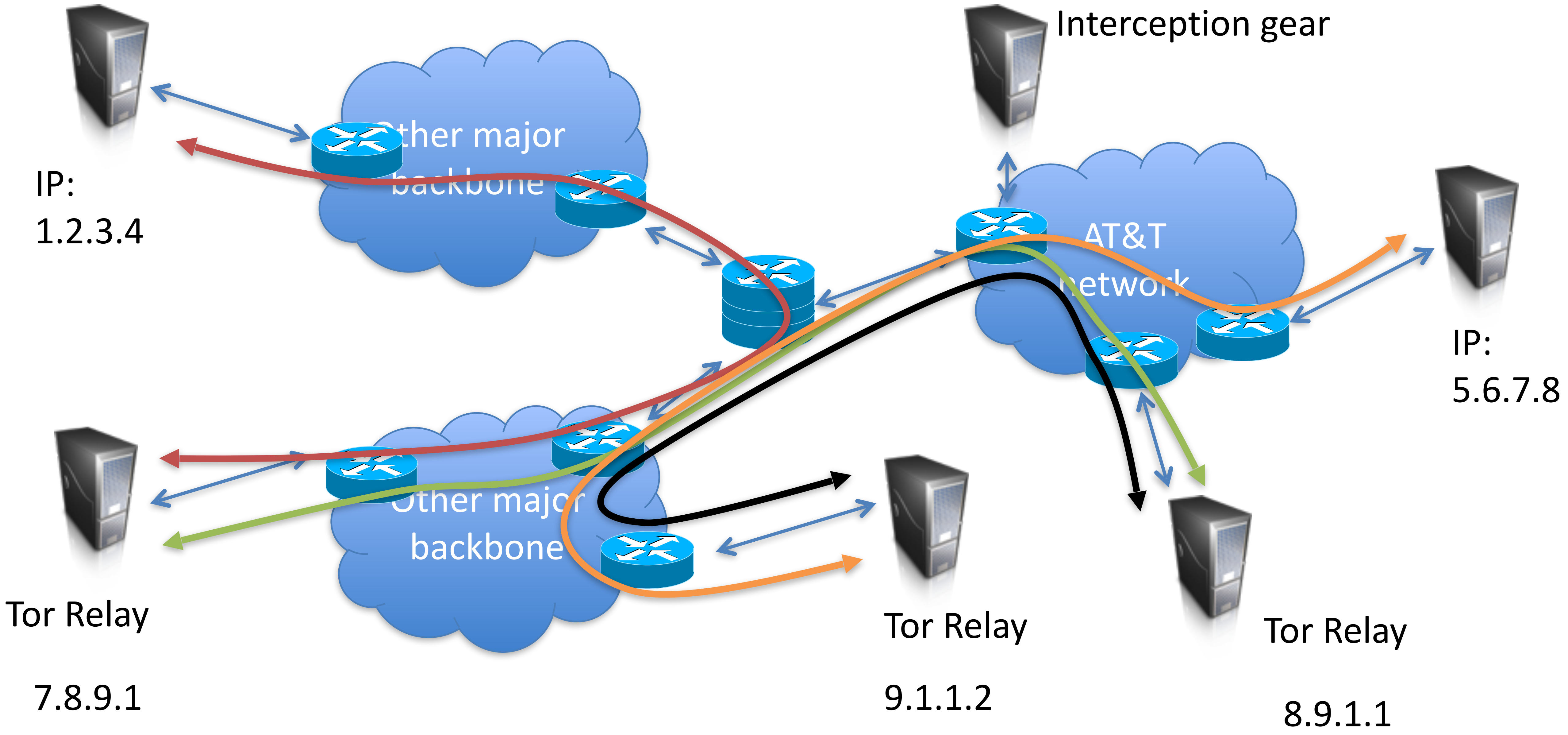


Thursday, April 26, 2012

FBI seizes server used to anonymize e-mail

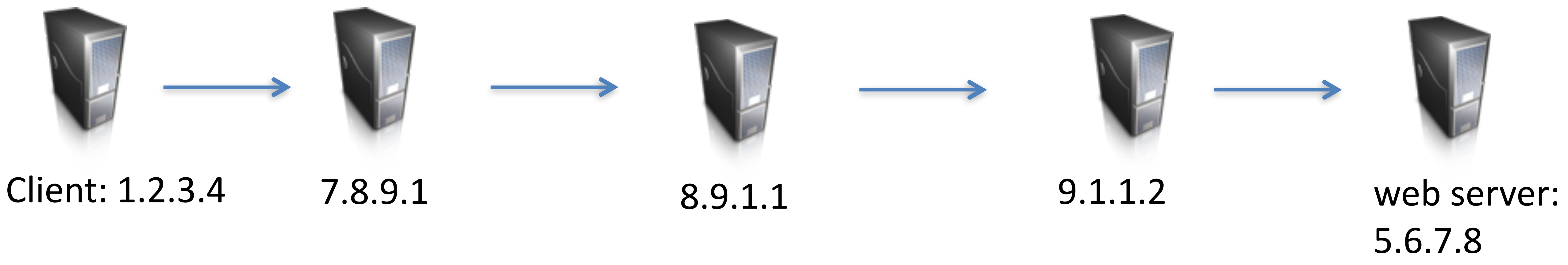
Jeffrey Brown 1 comment

Tor (The Onion Router)



Client -> **7.8.9.1 -> 8.9.1.1 -> 9.1.1.2** -> Destination   Called a *circuit*

Onion routing: the basic idea



Client: 1.2.3.4

7.8.9.1

8.9.1.1

9.1.1.2

web server:
5.6.7.8

Src: 9.1.1.2 | Dest: 5.6.7.8 | HTTP packet

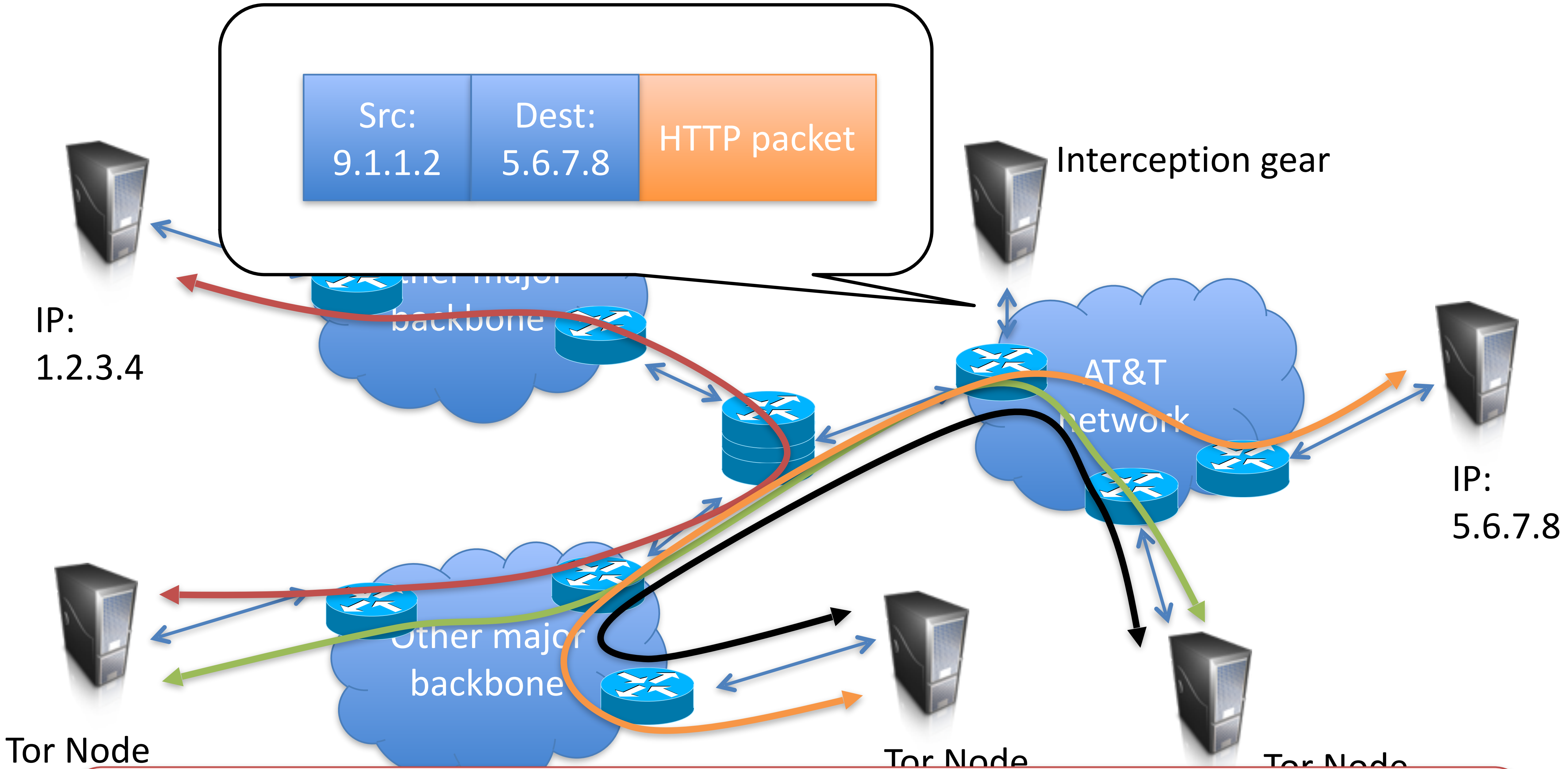
Src: 8.9.1.1 | Dest: 9.1.1.2 | Encrypted to 9.1.1.2

Src: 7.8.9.1 | Dest: 8.9.1.1 | Encrypted to 8.9.1.1

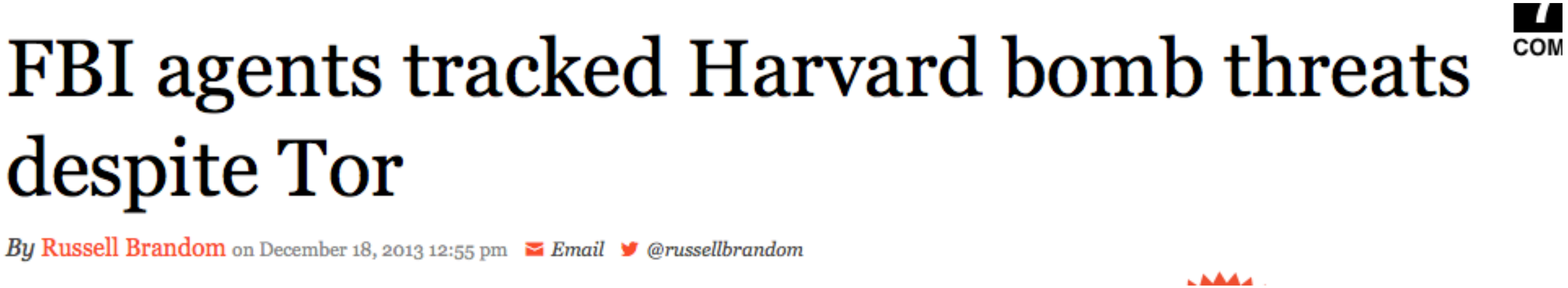
Src: 1.2.3.4 | Dest: 7.8.9.1 | Encrypted to 7.8.9.1

Tor implements more complex version of this basic idea

What does adversary see?



Tor obfuscates who talked to whom, need end-to-end encryption (e.g., HTTPS) to protect payload



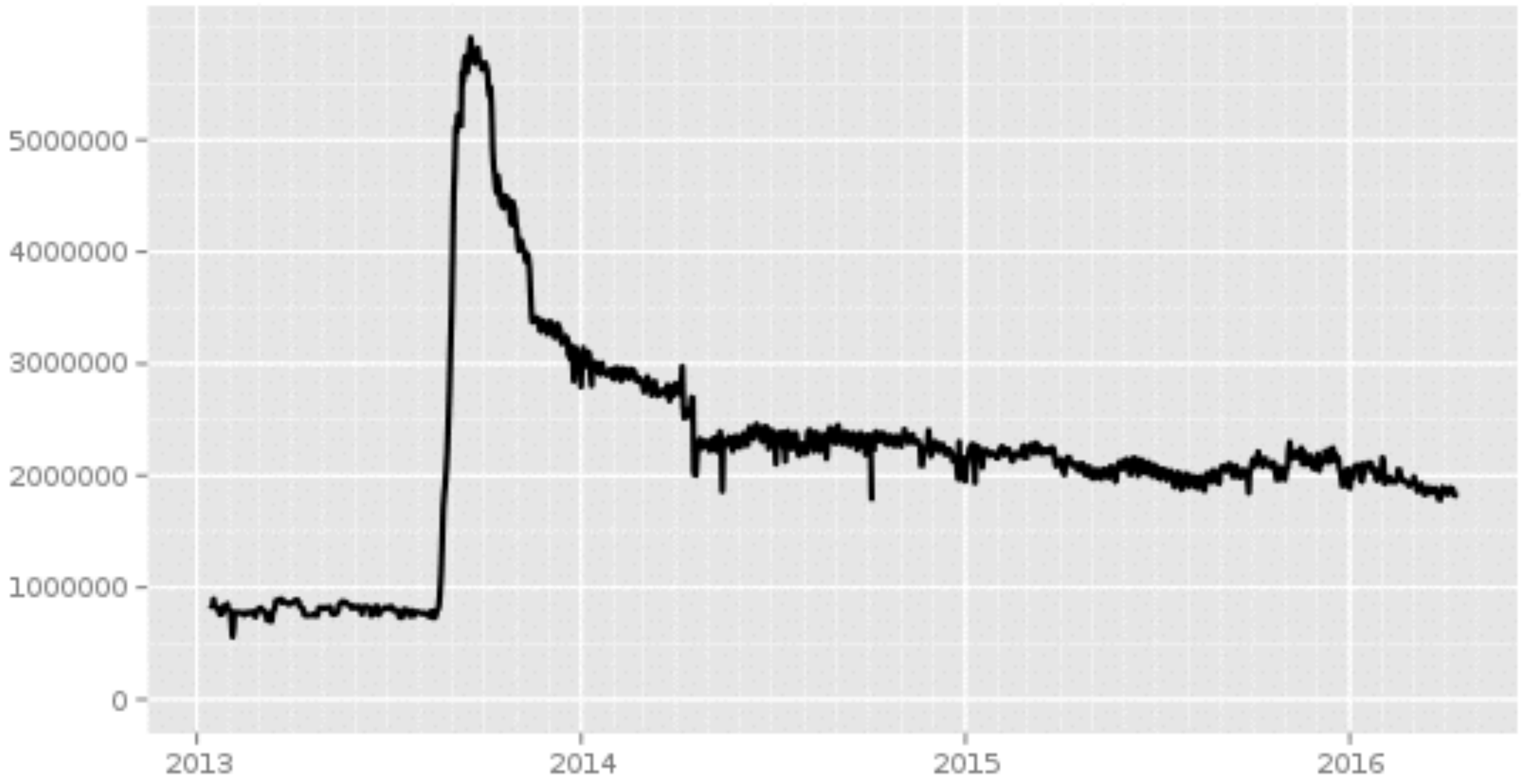
FBI agents tracked Harvard bomb threats despite Tor

By Russell Brandom on December 18, 2013 12:55 pm ✉ *Email* *@russellbrandom*

- Dec 2016: Eldo Kim, Harvard sophomore, sent bomb threats using Guerilla Mail (anonymous email service)
- Used ToR to connect to Guerilla Mail (from his dorm room)
- Caught within 2 days

- How did he get caught?
  - Guerilla Mail indicated user connected via ToR node
  - FBI compared timestamp on email to Harvard network logs,
  - He was the only one using ToR at that time (on the local network), confessed when confronted

Directly connecting users

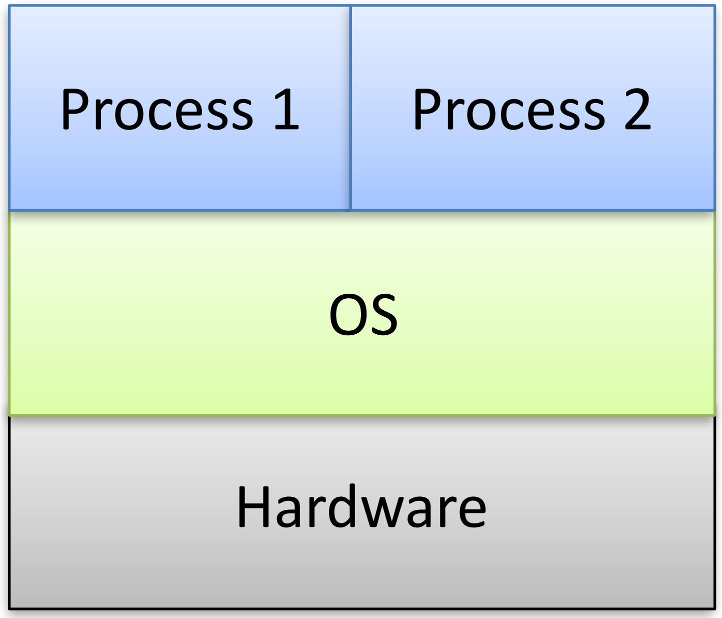


The Tor Project - https://metrics.torproject.org/

[As of: April 13, 2016]

virtualization

Virtualization



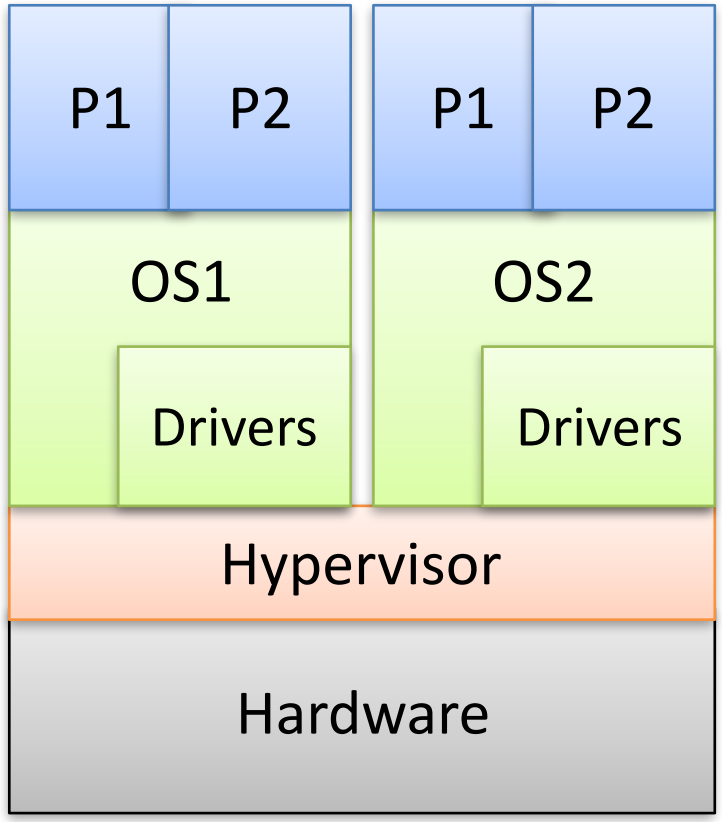
Process 1

Process 2

OS

Hardware

No virtualization



P1

P2

P1

P2

OS1

OS2

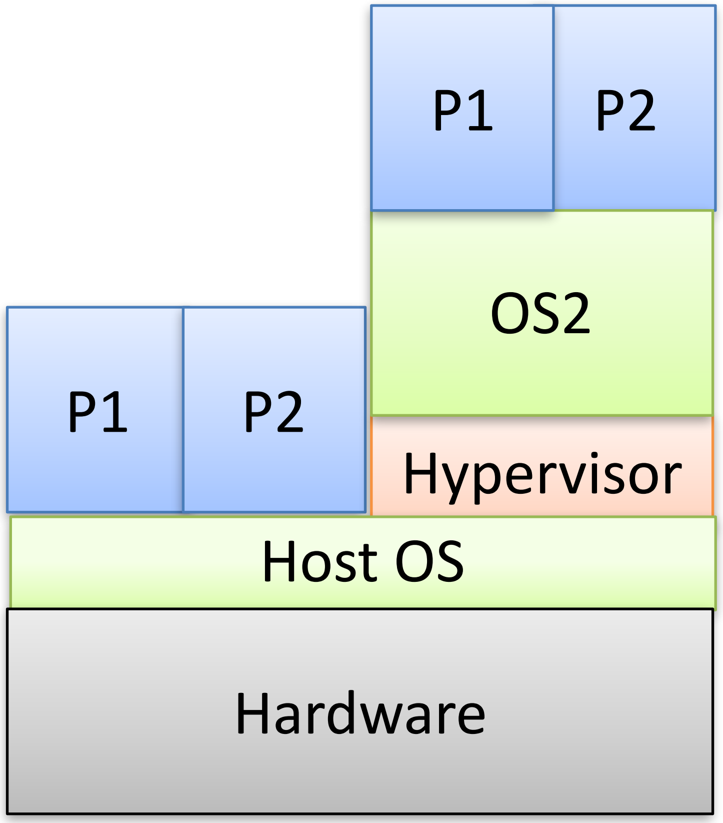
Drivers

Drivers

Hypervisor

Hardware

Type-1 Virtualization
(Xen, VMware ESX)



P1

P2

OS2

P1

P2

Hypervisor

Host OS

Hardware

Type-2 Virtualization
(VMware Workstation,
Virtual Box)

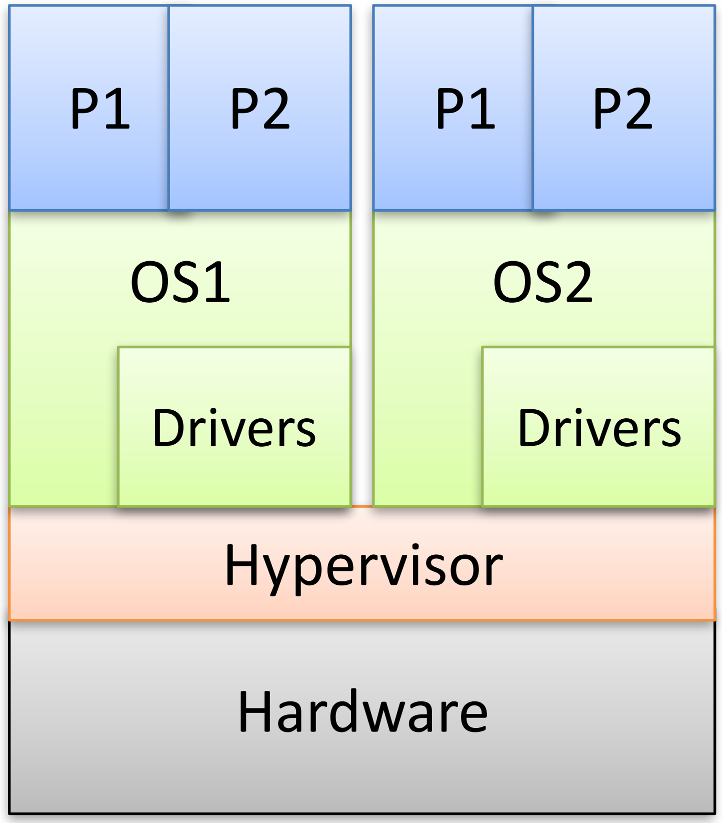
Type-1: Hypervisor runs directly on hardware

Type-2: Hypervisor runs on host OS

VM Use Cases

- Development and testing (especially when we need different OSs)
- Server consolidation
  - Run multiple servers on same hardware: web server, file server, email servers, …
- Cloud computing: Infrastructure-as-a-Service
- Sandboxing / containment

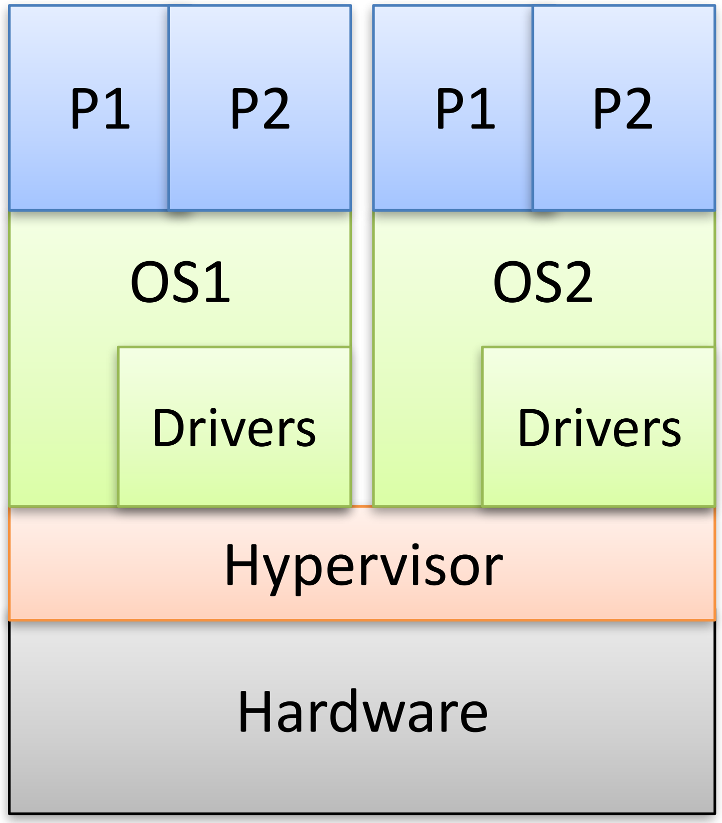
Security Model



Type-1 Virtualization
(VMware Workstation,
Virtual Box)

- What's the desired security model?
- Isolation between OS1/OS2 (and processes)
  - No access to file system, memory pages
- No "escape" from process/OS to hypervisor
- What can go wrong?

Isolation Problems



Type-1 Virtualization (VMware Workstation, Virtual Box)

- Information leakage
  - side-channel attacks using shared resources (instruction/memory caches)
- Degradation of service
  - Violate performance isolation, OS1 degrades OS2 to get more CPU time or network bandwidth
- Other problems?

Virtual Machine Management

- Snapshots
  - Volume snapshot / checkpoint
    - persistent storage of VM
    - must boot from storage when resuming snapshot
  - Full snapshot
    - persistent storage and ephemeral storage (memory, register states, caches, etc.)
    - start/resume in between (essentially) arbitrary instructions
- VM image is a file that stores a snapshot

recap

- Anonymous browsing, TOR
- Virtualization types, use cases
- Virtualization containment problems
- Linux RNG and reset vulnerabilities