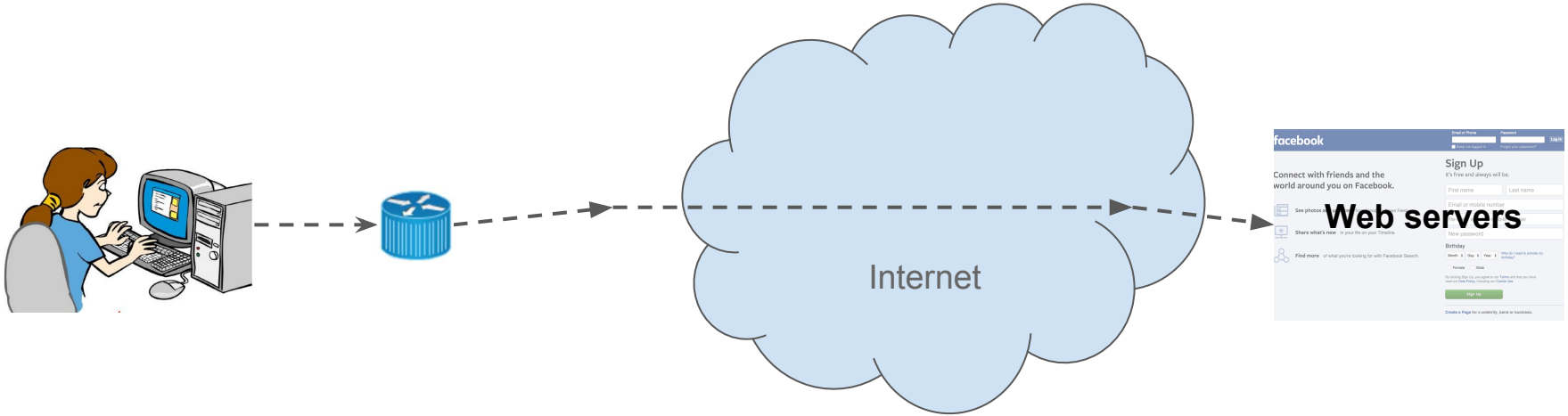
Examining How The Great Firewall Discovers Hidden Circumvention Servers

Roya Ensafi, David Fifield, Philipp Winter, Nick Feamster, Nicholas Weaver, and Vern Paxson

Oct 29, 2015

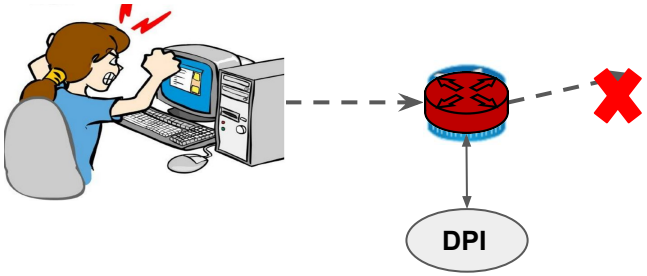


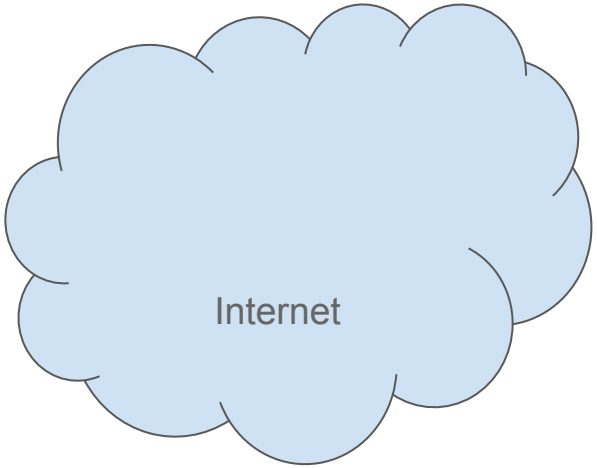
Circumventing Internet Censorship Using Proxies



Circumventing Internet Censorship Using Proxies

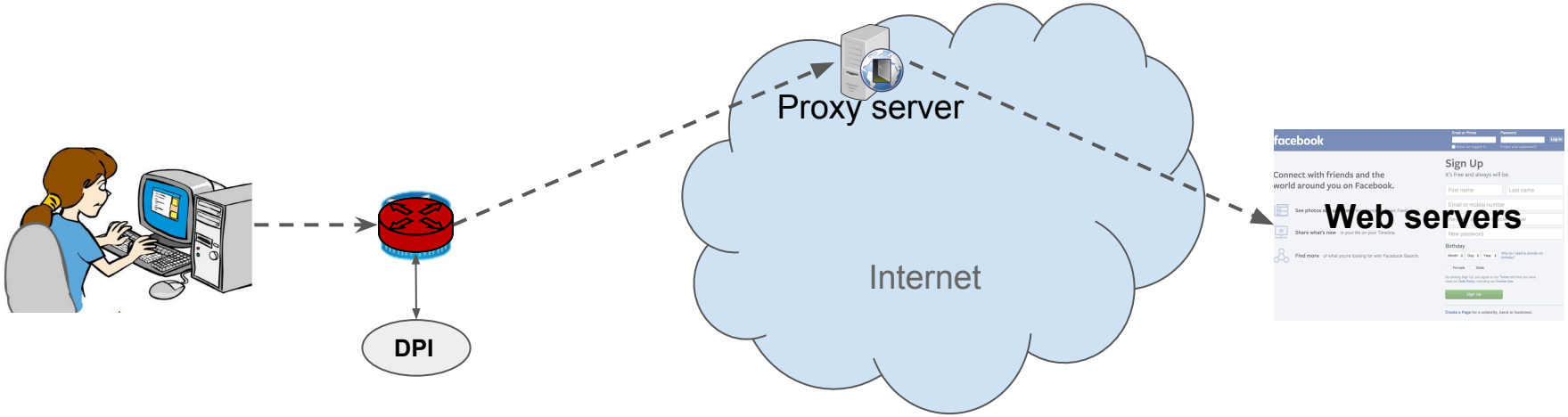
- Not everyone can connect to all web servers





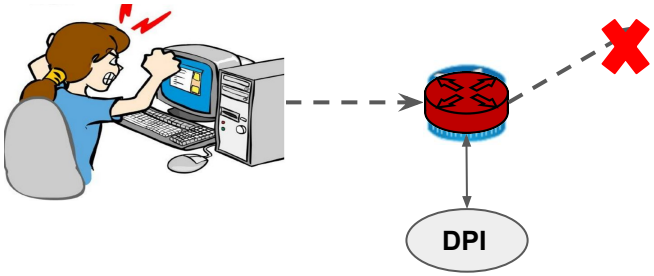
Circumventing Internet Censorship Using Proxies

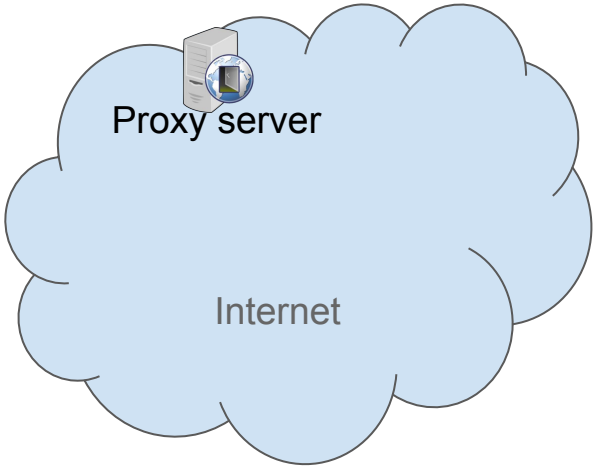
- Not everyone can connect to all web servers
- Many use **proxy servers** to circumvent censorship



Circumventing Internet Censorship Using Proxies

- Not everyone can connect to all web servers
- Many use **proxy servers** to circumvent censorship
- Governments are getting smarter at detecting proxy servers

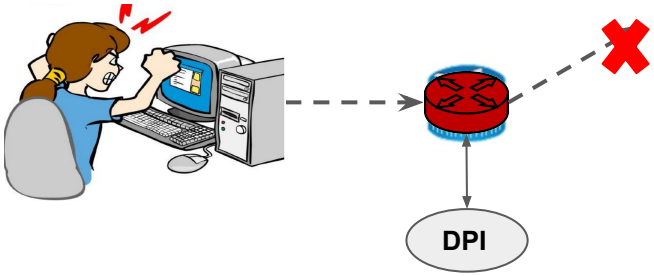


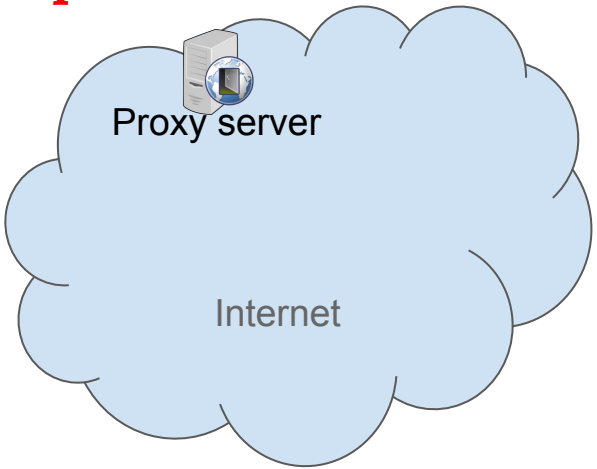


Circumventing Internet Censorship Using Proxies

- Not everyone can connect to all web servers
- Many use **proxy servers** to circumvent censorship
- Governments are getting smarter at detecting proxy servers

How do governments find these proxies?





How GFW Discovers Hidden Circumvention Servers

We focus on the **GFW** and **Tor**

- GFW is a **sophisticated censorship system**
- Tor has a long history of being used for **circumventing** government censorship

Censorship Arms Race: GFW vs. Tor

Use **public Tor network** to circumvent GFW

Time

Censorship Arms Race: GFW vs. Tor

Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

Time

Censorship Arms Race: GFW vs. Tor

Time

Use **public Tor network** to circumvent GFW

Introduce **private bridges**, whose distribution is **rate-limited**

Download consensus and **block relays**

Censorship Arms Race: GFW vs. Tor

Time

Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

Introduce **private bridges**, whose distribution is **rate-limited**

Use **DPI** to detect Tor **TLS handshake**

Fingerprinting the Tor TLS Handshake

- TLS handshake is **unencrypted** and **leaks information**
- Tor's use of TLS has some **peculiarities**
  - X.509 certificate life times
  - Cipher suites
  - Randomly generated server name indication (e.g., www.6qgoz6epdi6im5rvxnlx.com)
- GFW looks (at least) for **cipher suites** in the **TLS client hello**

Censorship Arms Race: GFW vs. Tor

Time

Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

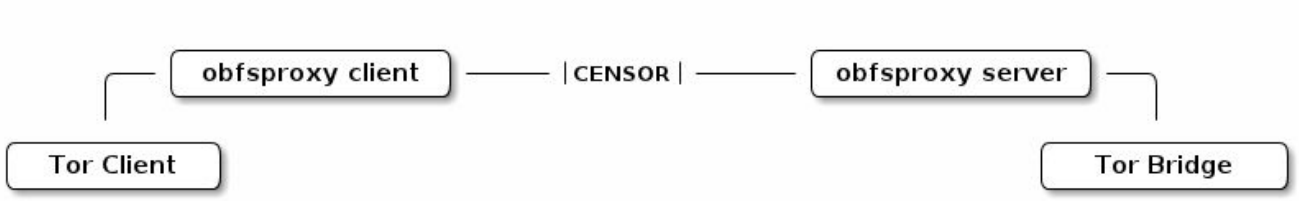
Introduce **private bridges**, whose distribution is **rate-limited**

Use **DPI** to detect Tor **TLS handshake**

Introduce **pluggable transports** to hide the handshake such as obfs2, obfs3

Tor Pluggable Transport

- Pluggable transports are drop-in modules for traffic obfuscation
- Many modules have been written, but we focus on
  - **obfs2** (First deployed module)
    - First 20 bytes can be used to detect Tor traffic with high confidence.
  - **obfs3** (obfs2's successor)
    - Makes Tor traffic look like a uniformly random byte stream



Censorship Arms Race: GFW vs. Tor

- Detection of pluggable transports is **uncertain**
  - Implies false positives → collateral damage

Time

handshake such as obfs2, obfs3

Censorship Arms Race: GFW vs. Tor

- Detection of pluggable transports is **uncertain**
  - Implies false positives → collateral damage

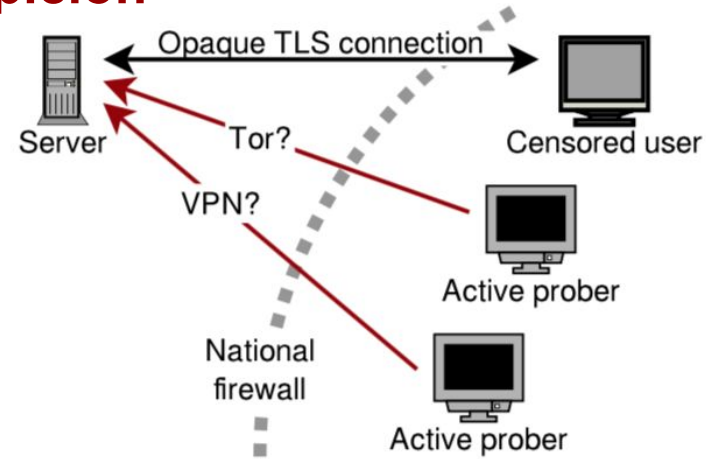
GFW added **active probing** to complement the DPI fingerprinting

Time

handshake such as obfs2, obfs3

How does GFW Block Tor Hidden Circumvention Servers?

1. Network monitoring (e.g., switch mirror port)
2. DPI for suspicious traffic (e.g., cipher suite)
3. **Actively probing server to verify suspicion**
4. Blocking server



Censorship Arms Race: GFW vs. Tor

Time

Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

Introduce **private bridges**, whose distribution is **rate-limited**

Use **DPI** to detect Tor **TLS handshake**

Introduce **pluggable transports** to hide the handshake such as obfs2, obfs3

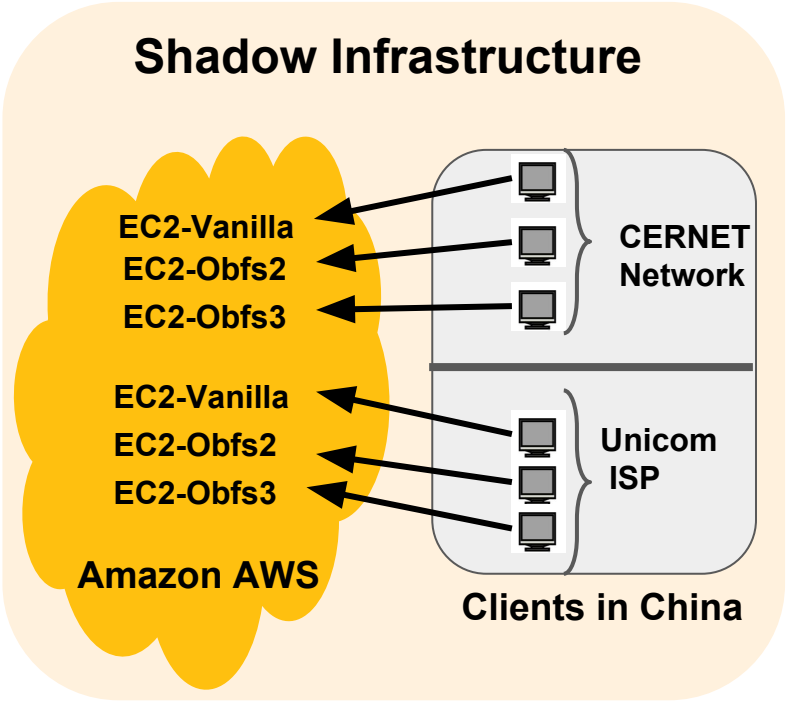
Use **DPI** + **Active probing**

Many Questions about Active Probing are Unanswered!

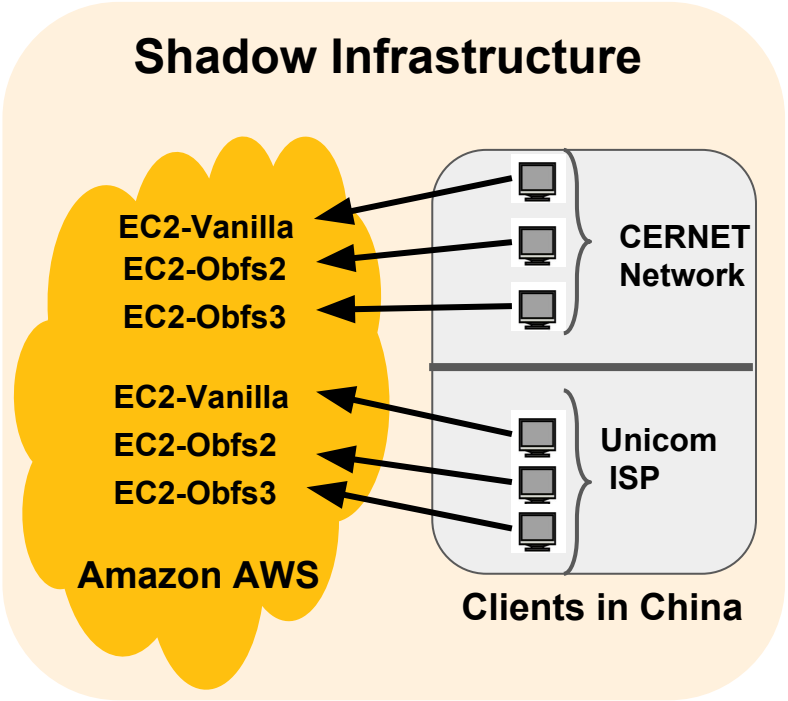
- Only two blog posts and Winter's FOCI'12 paper
- We lack a comprehensive picture of more complicated questions

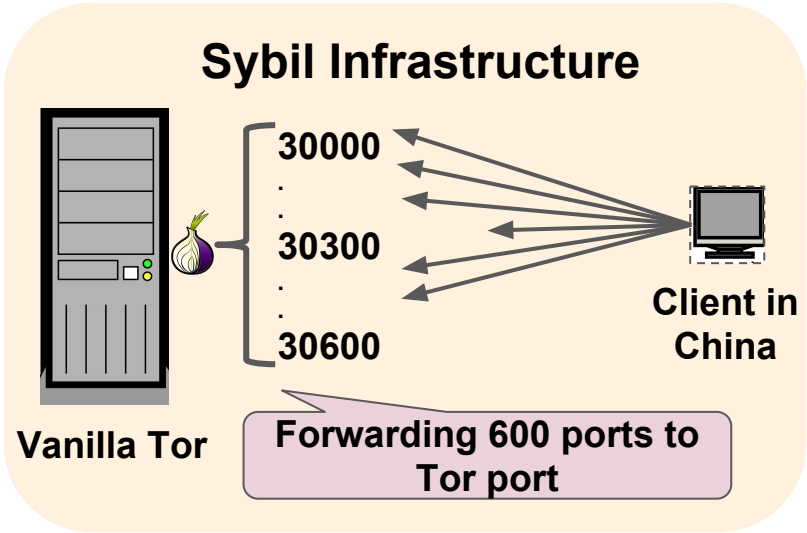
- We want to know:
  - **Implementation**, i.e., how does it block?
  - **Architecture**, i.e., how is a system added to China's backbone?
  - **Policy**, i.e., what kind of protocols does it block?
  - **Effectiveness**, i.e., what's the degree of success at discovering Tor bridges?

Overview of Our Datasets:

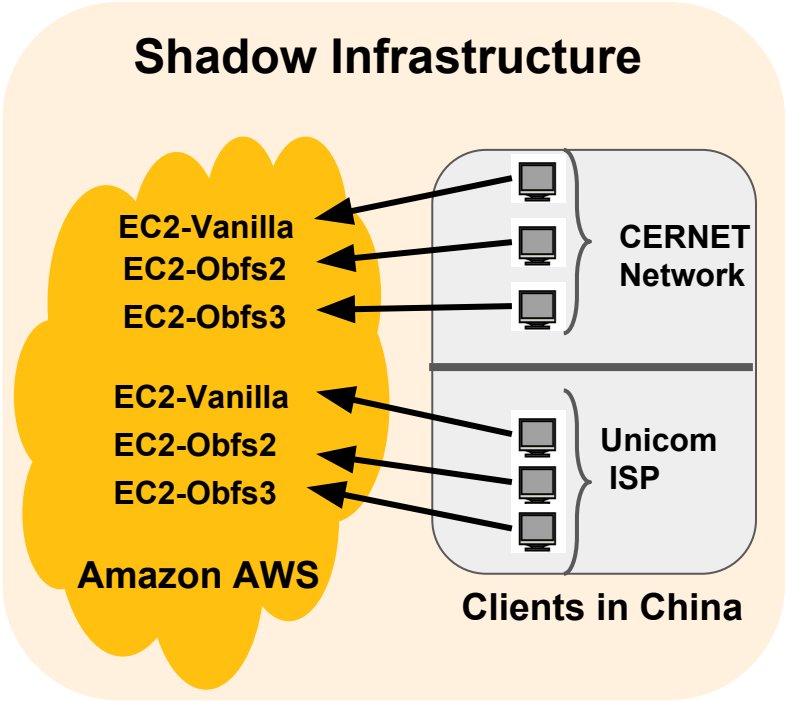


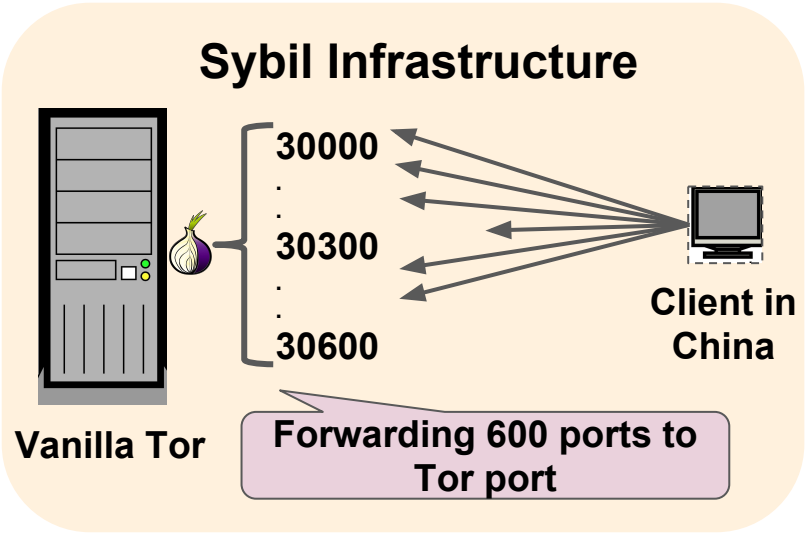
Overview of Our Datasets:

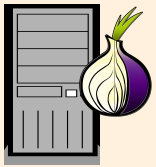




Overview of Our Datasets:







Server Log Analysis
Application logs of a web server that also runs a Tor bridge since 2010.

Overview of Our Datasets:

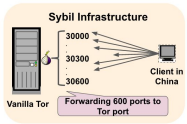
- For the Shadow and the Sybil datasets:
  - We had pcap files of both the clients and the bridges.
- For the Log dataset, we only had application logs.

Dataset	Time span
Shadow	Dec 2014 -- Feb 2015 (3 months)
Sybil	Jan 29, 2015 -- Jan 30, 2015 (20 hours)
Log	Jan 2010 -- Aug 2015 (5 years)

How to Distinguish Probers from Genuine Clients?

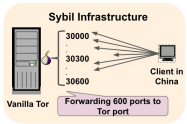
How to Distinguish Probers from Genuine Clients?

- Detecting probers in Sybil dataset is easy, all the probers:
  - Visited our vanilla Tor bridge after our client established connections
  - Originated from China



How to Distinguish Probers from Genuine Clients?

- Detecting probers in Sybil dataset is easy, all the probers:
  - Visited our vanilla Tor bridge after our client established connections
  - Originated from China

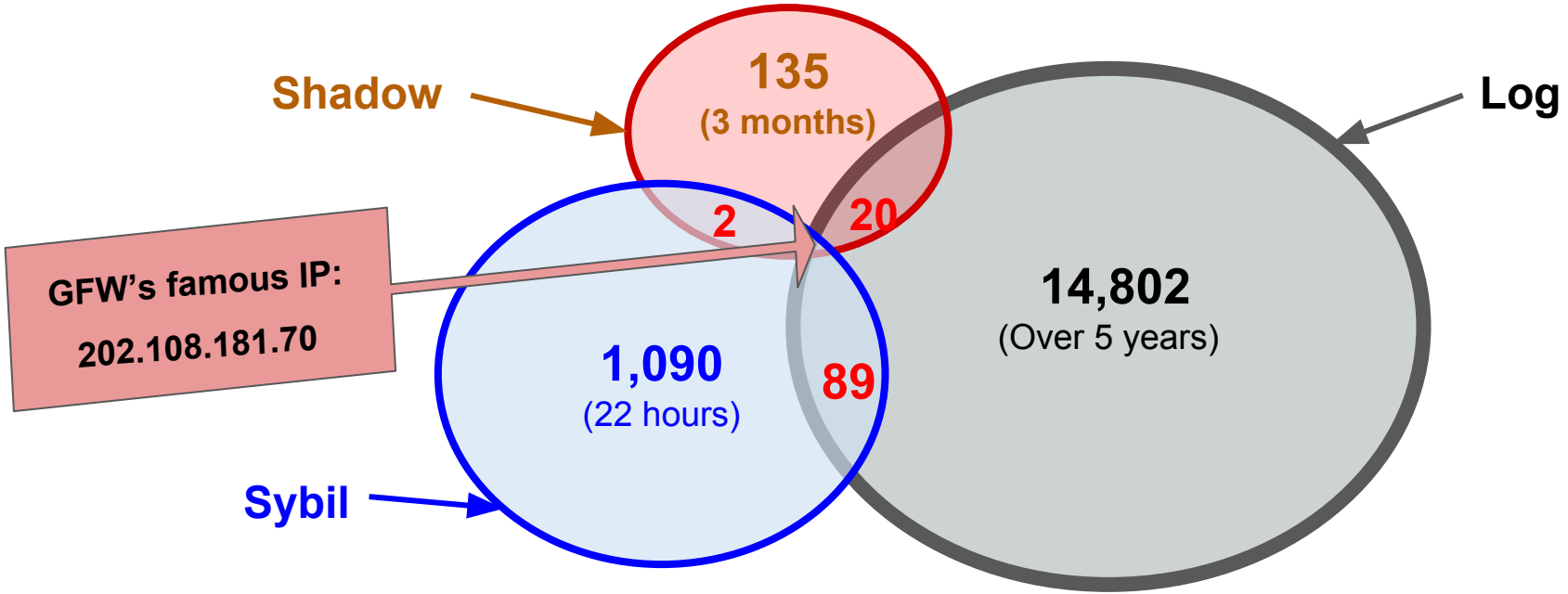


- For the other datasets, we adopt an algorithm:
  - If the cipher suites is in the TLS client hello => Vanilla bridge probes
  - If the first 20 bytes can reveal Obfs2 => Obfs2 bridges probers
  - ...

How Many Unique Probers did We Find?

How Many Unique Probers did We Find?

- Using **Sybil**, **Shadow** and **Log** dataset
  - In total, we collected **16,083** unique prober IP addresses

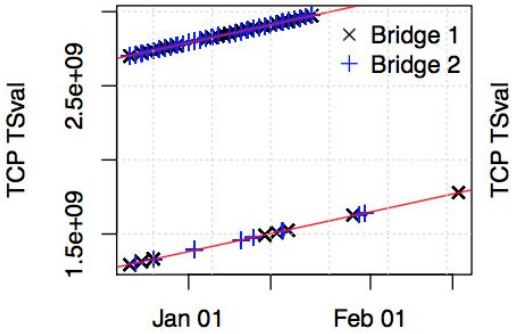


Can We Fingerprint Active Probers?

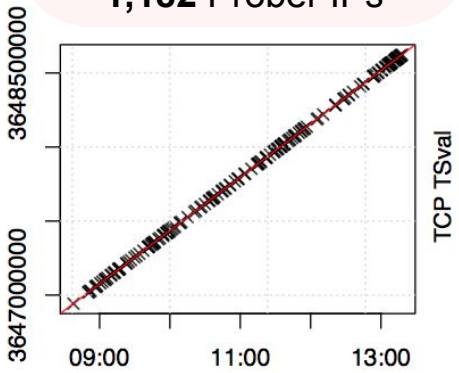
Can We Fingerprint Active Probers?

- TCP layer
  - TSval slope: timestamp clock rate
  - TSval intercept: (rough) system uptime
  - GFW likely operate a handful of physical probing systems

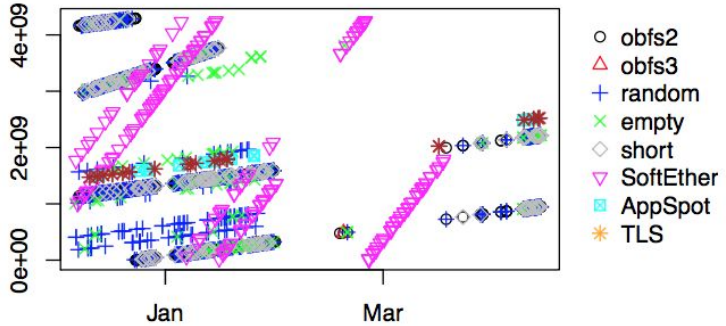
Shadow exp. with **158** Prober IPs



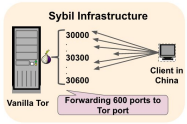
Sybil exp. with **1,182** Prober IPs



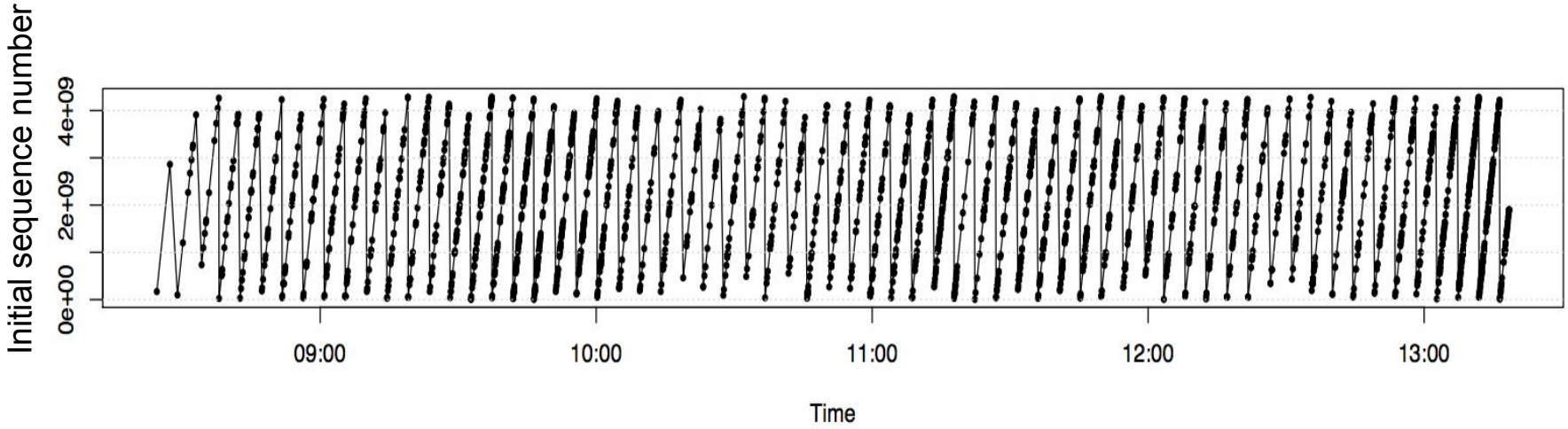
Log dataset with **14,912** Prober IPs



Can We Fingerprint Active Probers?



- TCP layer
  - Striking pattern in initial sequence numbers (derived from time) of 1,182 probes
  - Shared pattern in TSval for all three datasets

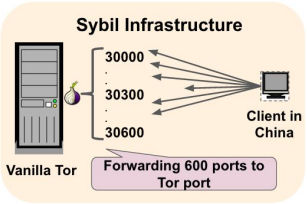


What do These Patterns Mean?

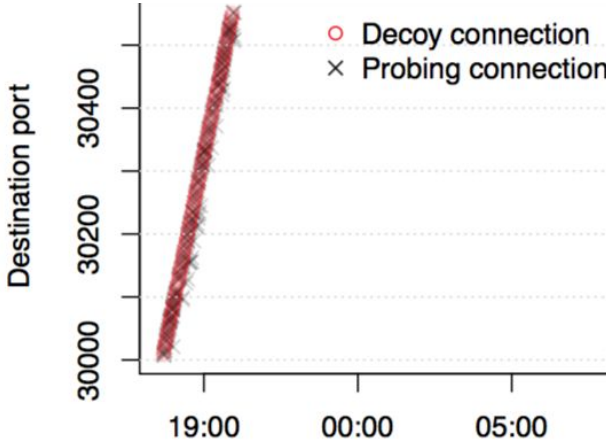
- Active probing connections **leak shared state**
  - ISNs, TSval, source ports, ...
- GFW likely operates only **few physical systems**
- Thousands of IP addresses are controlled by **central source**

How Quickly do Active Probes Show Up?

How Quickly do Active Probes Show Up?

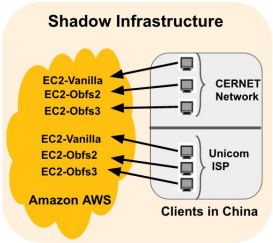


- Sybil dataset shows that system now works in **real time**
  - Median delay between Tor connection and subsequent probing connection is ~500ms
  - **1,182** distinct probes showed up in 22 hours

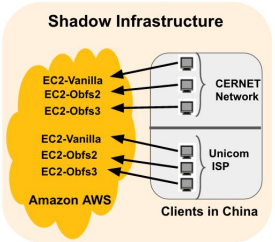


Is Active Probing Successful?

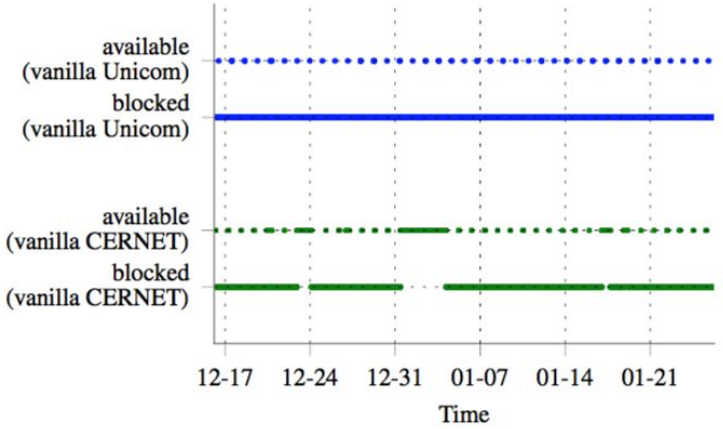
Is Active Probing Successful?



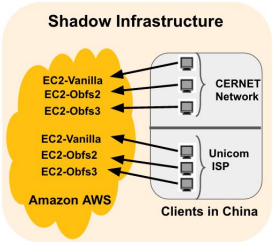
Is Active Probing Successful?



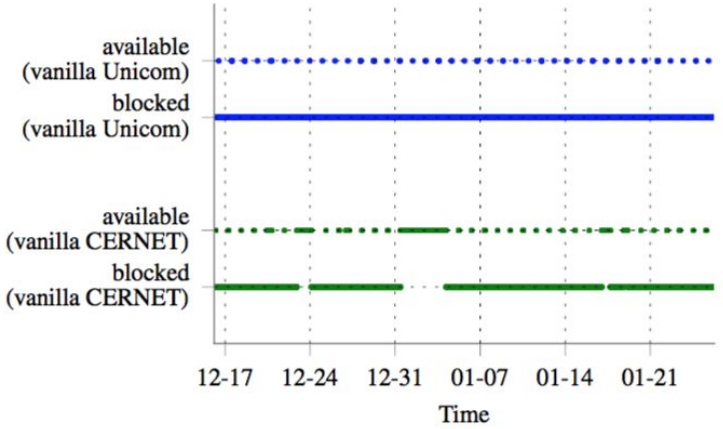
- Tor clients succeed in connecting **roughly every 25 hours**
  - Might reflect **implementation artifact** of GFW



Is Active Probing Successful?



- Tor clients succeed in connecting **roughly every 25 hours**
  - Might reflect **implementation artifact** of GFW



- **obfs2** and **obfs3** (~98%) were almost always reachable for clients
  - Surprising because GFW **can probe and block** obfs2 and obfs3

Takeaway messages

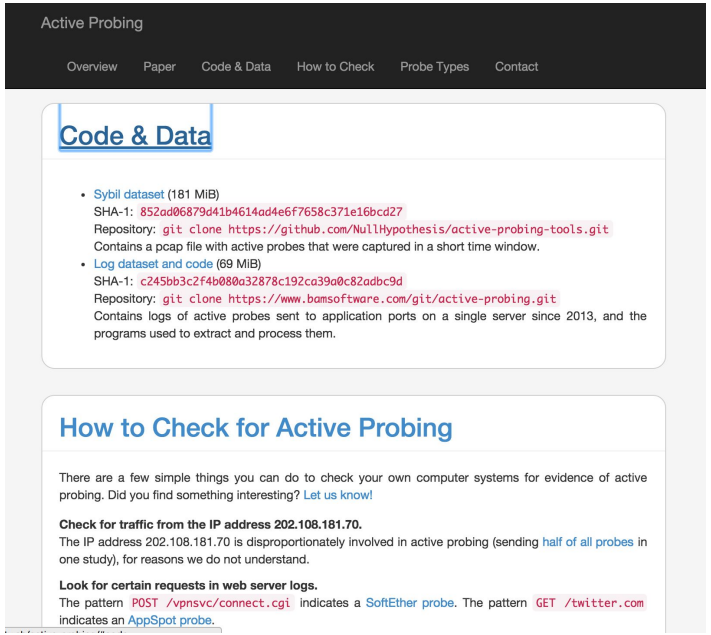
Our results show that the active probing system

- Makes use of a **large amount of IP addresses**, clearly **centrally controlled**
  - We can not just blacklist probers' IP addresses
- Operates in **real time**
- **Probes** Vanilla, Obfs2, and Obfs3 Bridge

Tor's pluggable transports led to GFW's "pluggable censorship"

Q&A

- Project page: https://nymity.ch/active-probing/
- **Log** and **Sybil** data sets are available online
- Contact: rensafi@cs.princeton.edu



What Is the Characteristic of the Probing System?

- Sensor responsible for triggering probes operates **single-sidedly**:
  - SYN, followed by ACK, then Tor's TLS client hello) => trigger probe.
- The sensor does **not** seem to **robustly reassemble** TCP:
  - The fragmented data did not trigger an active probe, which differs from the GFW
- **Traceroute** to the sensors suggested:
  - Unicom's sensor appears to operate on the same link as the GFW
  - CERNET sensor appears one hop closer to our server