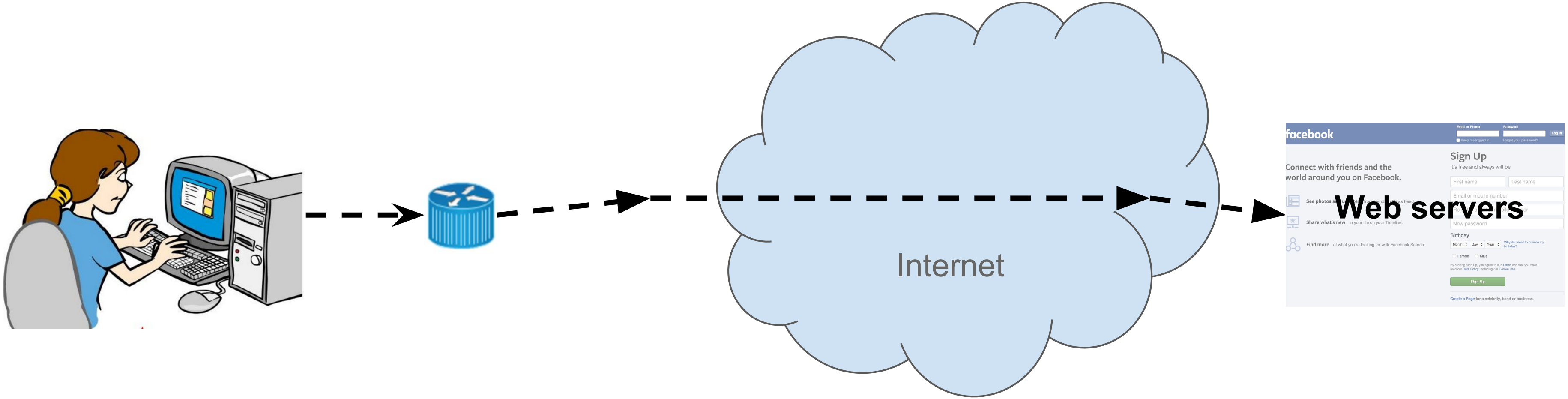
How the Great Firewall discovers hidden circumvention servers

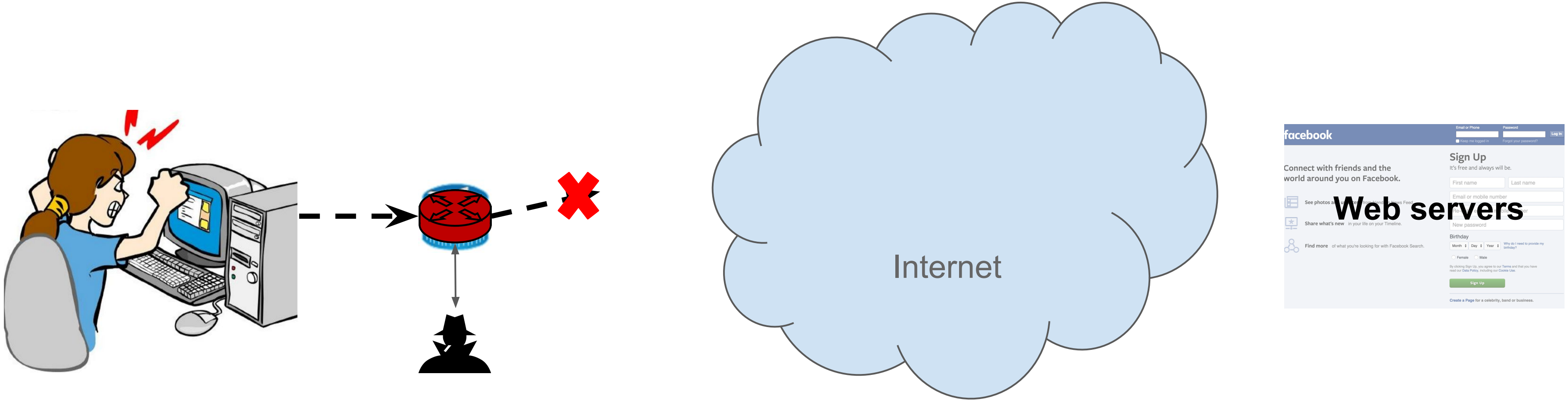
Roya Ensafi, David Fifield, Philipp Winter,
Nick Feamster, Nicholas Weaver, and Vern Paxson

Circumventing Internet Censorship Using Proxies


Internet
facebook
Connect with friends and the world around you on Facebook.
Sign Up
It's free and always will be.
Web servers

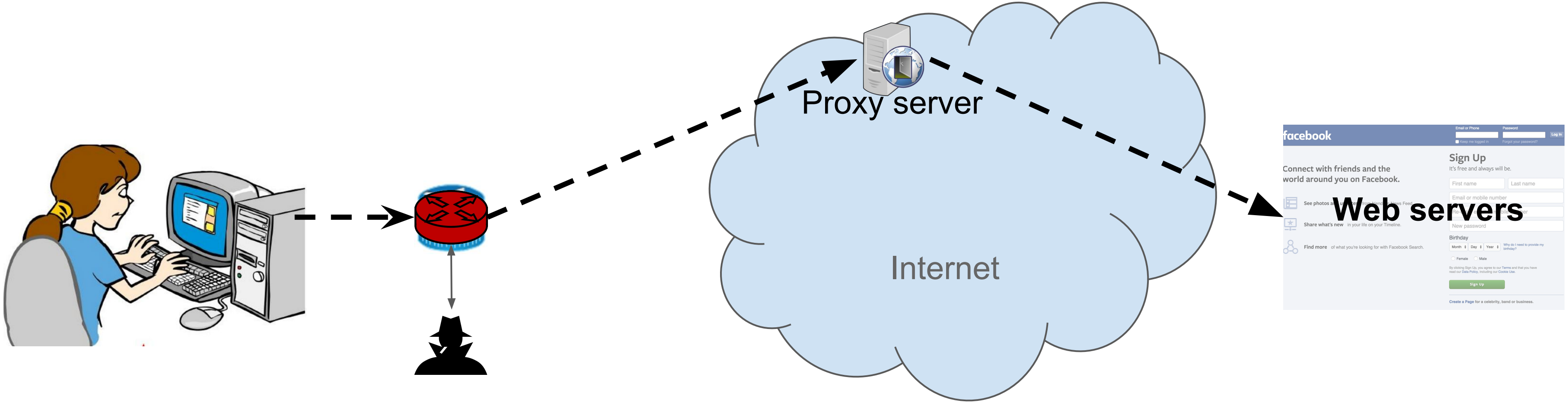
Circumventing Internet Censorship Using Proxies

- Not everyone can connect to all web servers



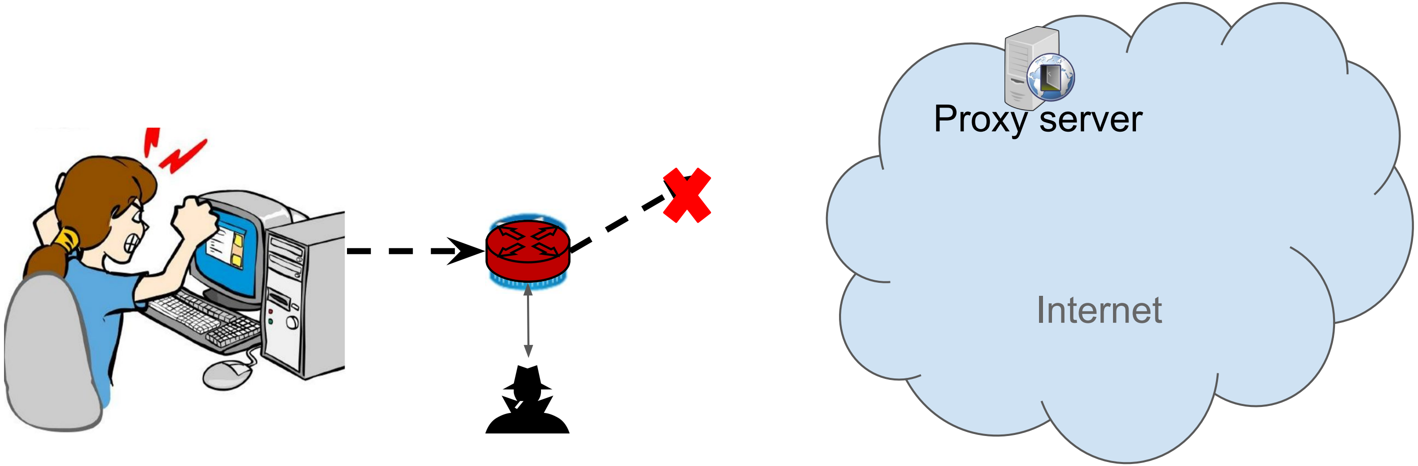
Circumventing Internet Censorship Using Proxies

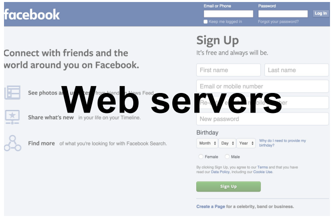
- Not everyone can connect to all web servers
- Many use **proxy servers** to circumvent censorship


Proxy server
Internet
Web servers
facebook
Connect with friends and the world around you on Facebook.
Sign Up
It's free and always will be.

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





facebook

Email or Phone | Password | Log In

□ Keep me logged in | Forgot your password?

Connect with friends and the world around you on Facebook.

See photos and updates from friends in News Feed.

Share what's new in your life on your Timeline.

Find more of what you're looking for with Facebook Search.

Sign Up

It's free and always will be.

First name | Last name

Email or mobile number

Re-enter email or mobile number

New password

Birthday

Month | Day | Year | Why do I need to provide my birthday?

○ Female ○ Male

By clicking Sign Up, you agree to our Terms and that you have read our Data Policy, including our Cookie Use.

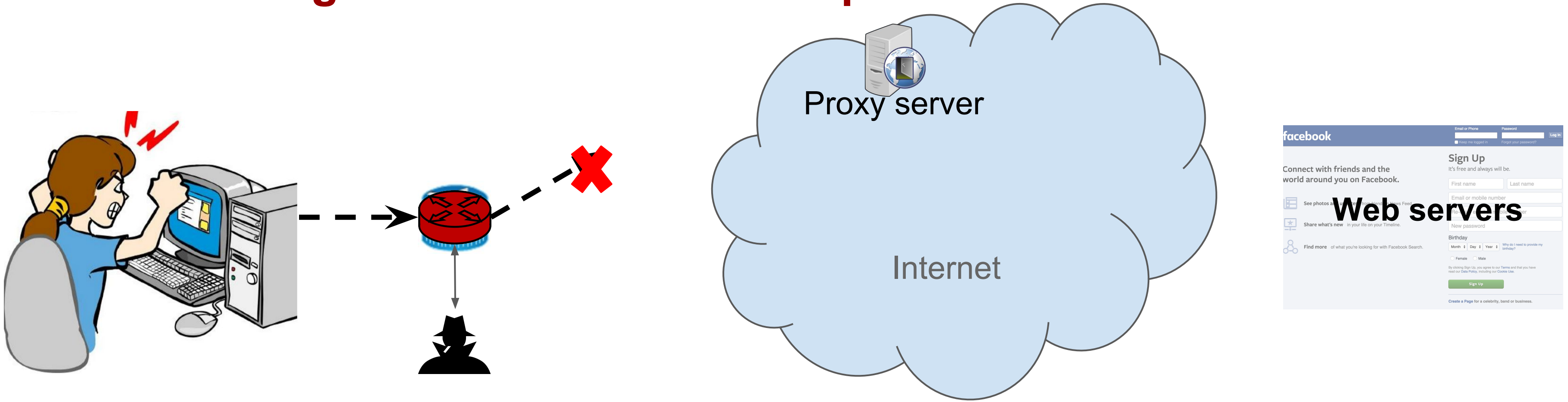
Sign Up

Create a Page for a celebrity, band or business.

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?



Proxy server
Internet
facebook
Sign Up
It's free and always will be.
Connect with friends and the world around you on Facebook.
Web servers

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

Time

Use **public Tor network** to circumvent GFW

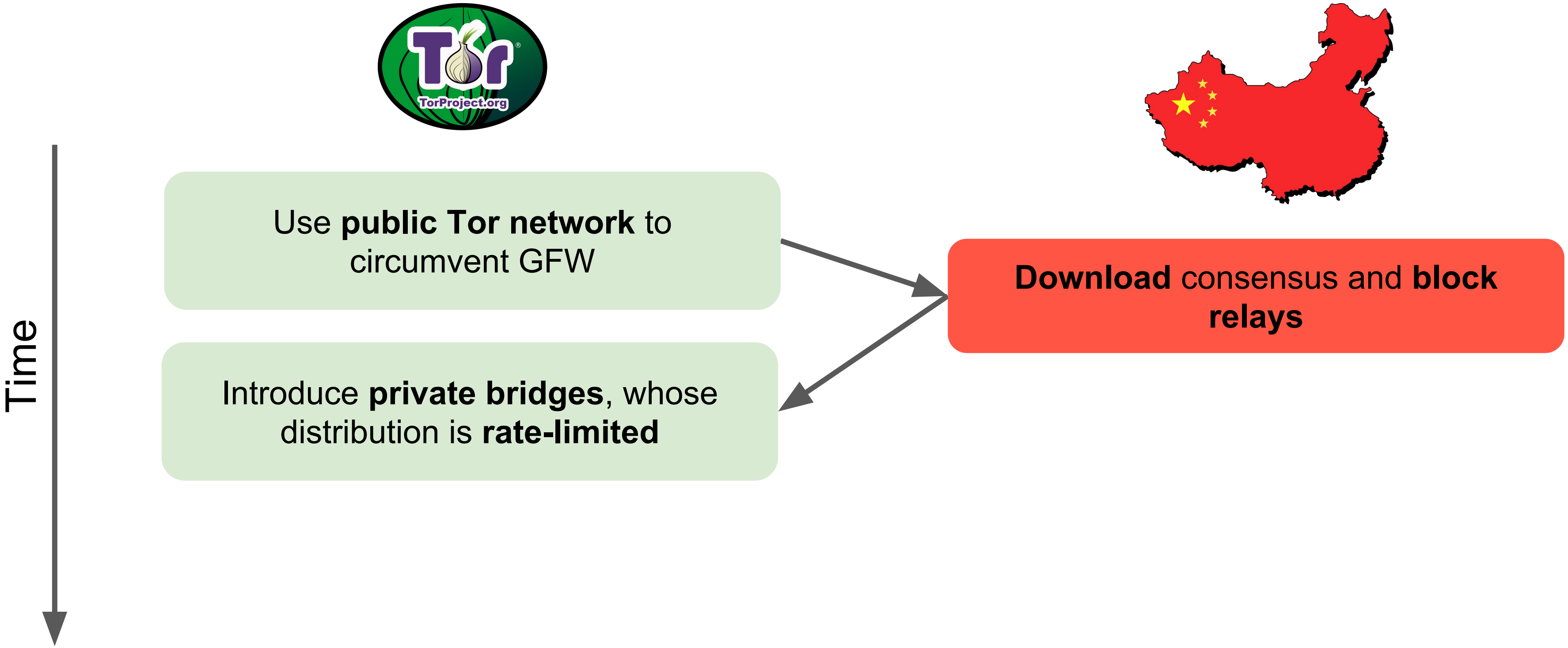
Censorship Arms Race: GFW vs. Tor

Time

Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

Censorship Arms Race: GFW vs. Tor

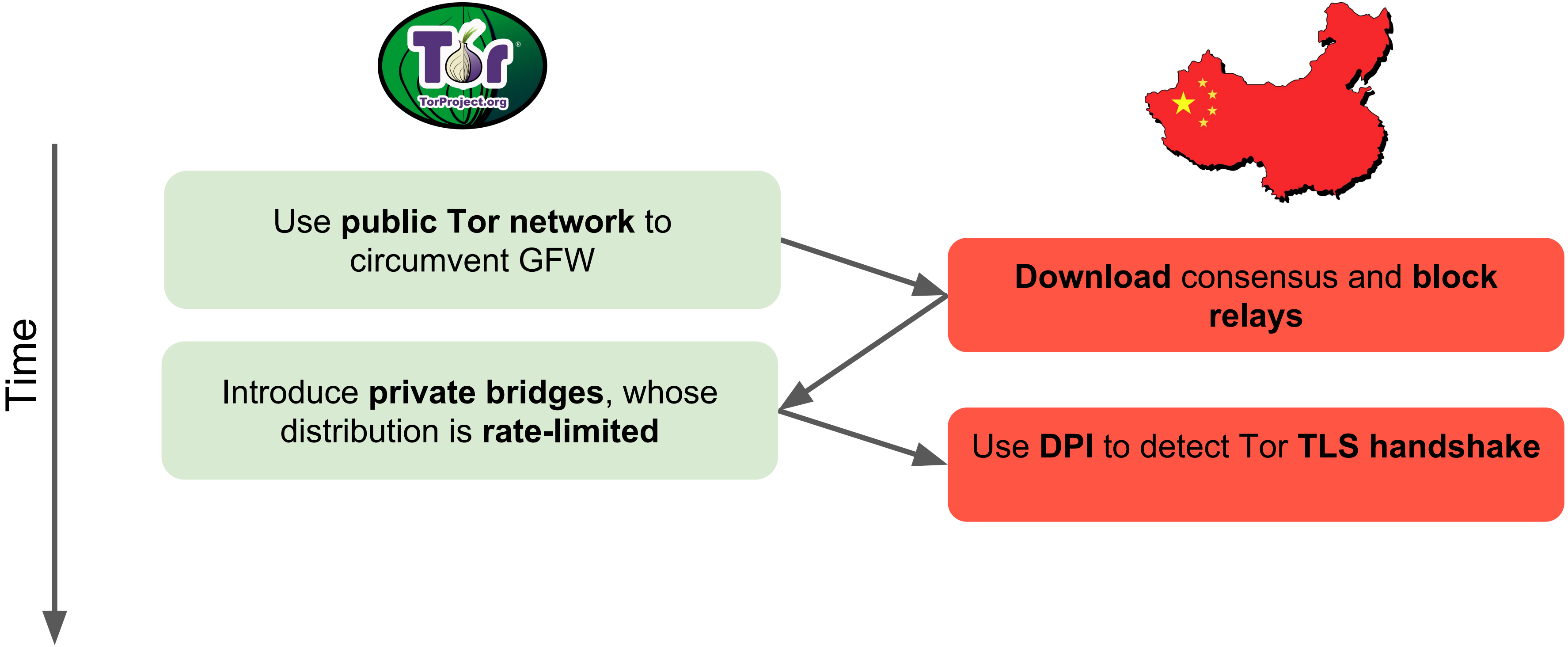


Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

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

Censorship Arms Race: GFW vs. Tor



Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

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

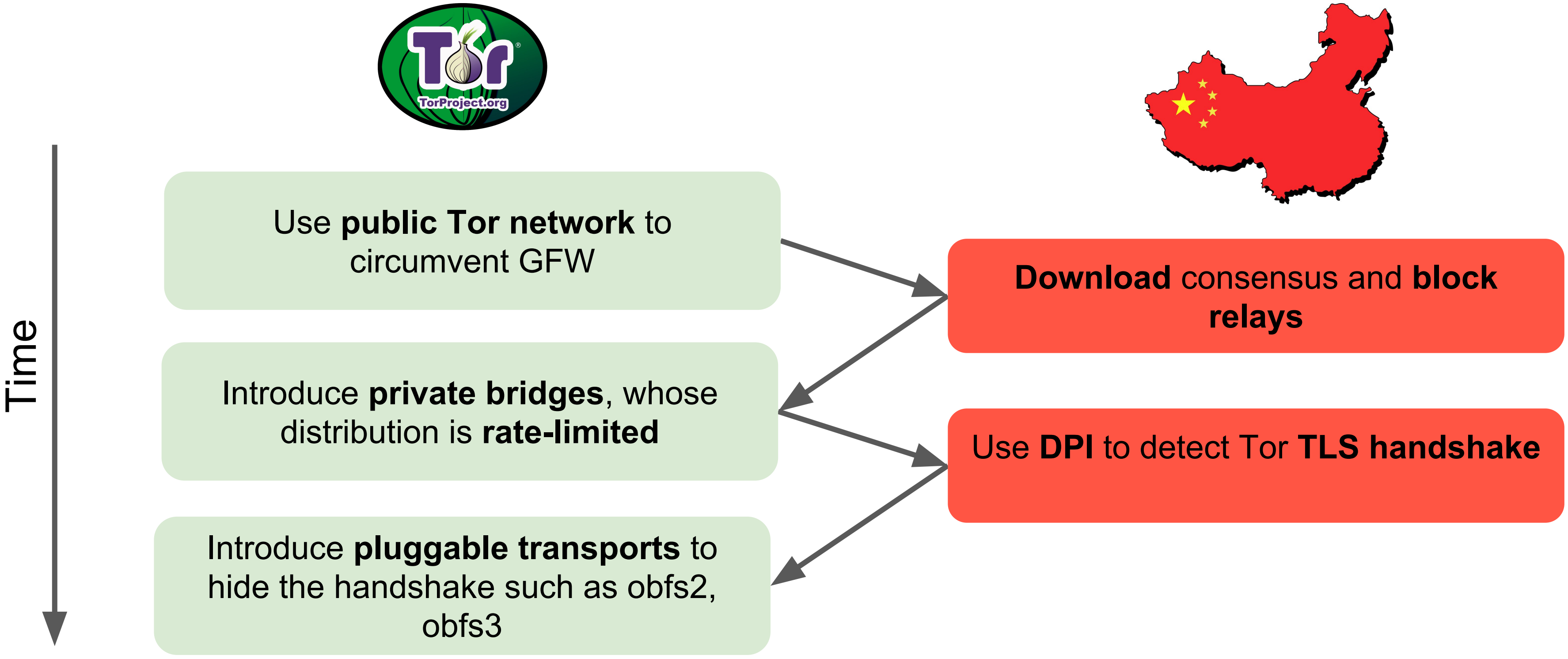
Use **DPI** to detect Tor **TLS handshake**

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



Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

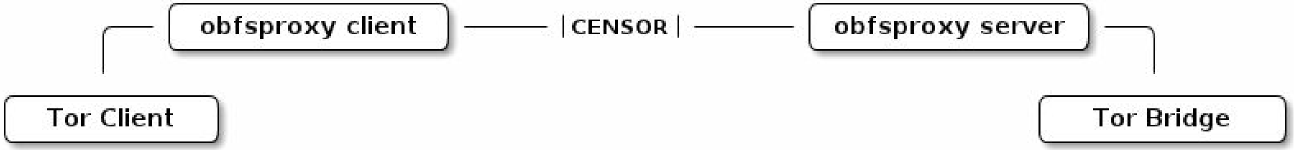
Introduce **private bridges**, whose distribution is **rate-limited**

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



Encryption Reduces Blocking Accuracy

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

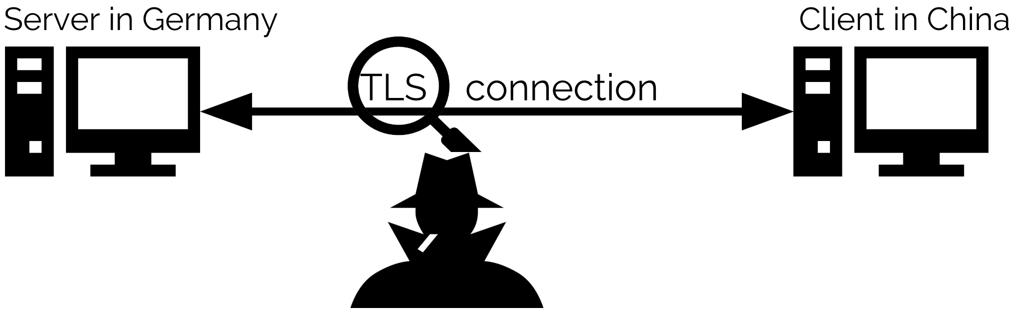
Encryption Reduces Blocking Accuracy

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

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

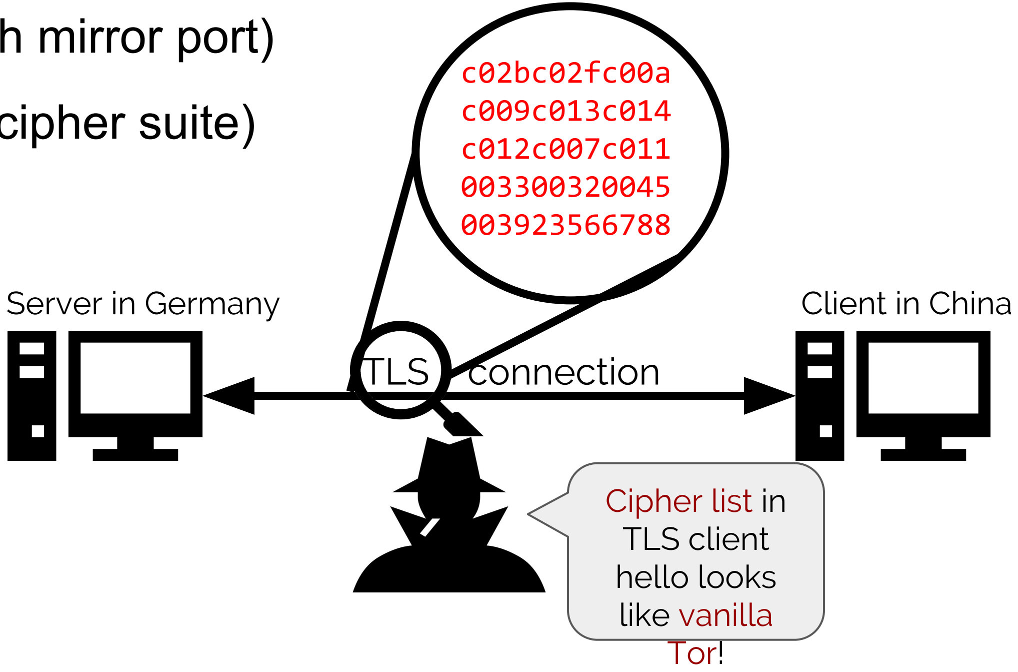
How does GFW Block Tor Hidden Circumvention Servers?

1. Network monitoring (e.g., switch mirror port)



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)

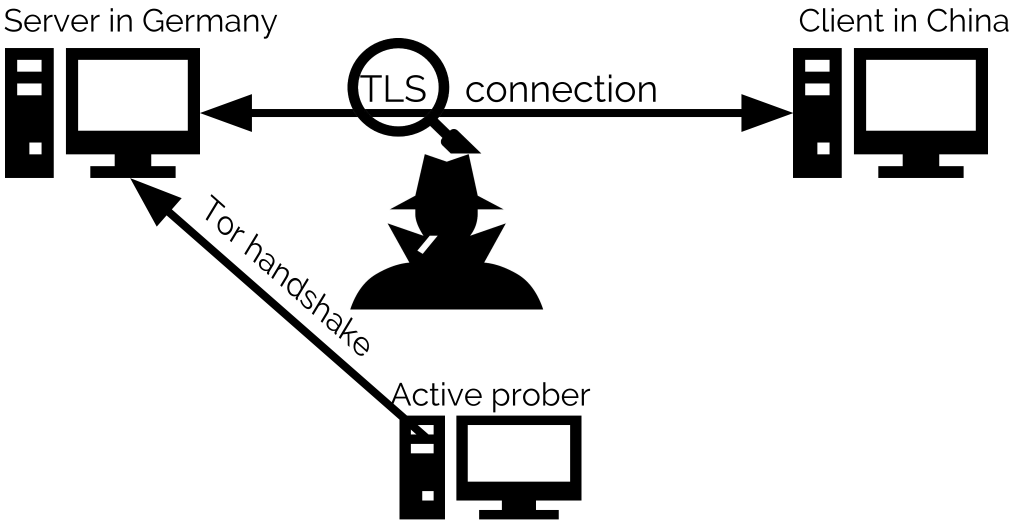


```
c02bc02fc00a
c009c013c014
c012c007c011
003300320045
003923566788
```

Cipher list in TLS client hello looks like vanilla Tor!

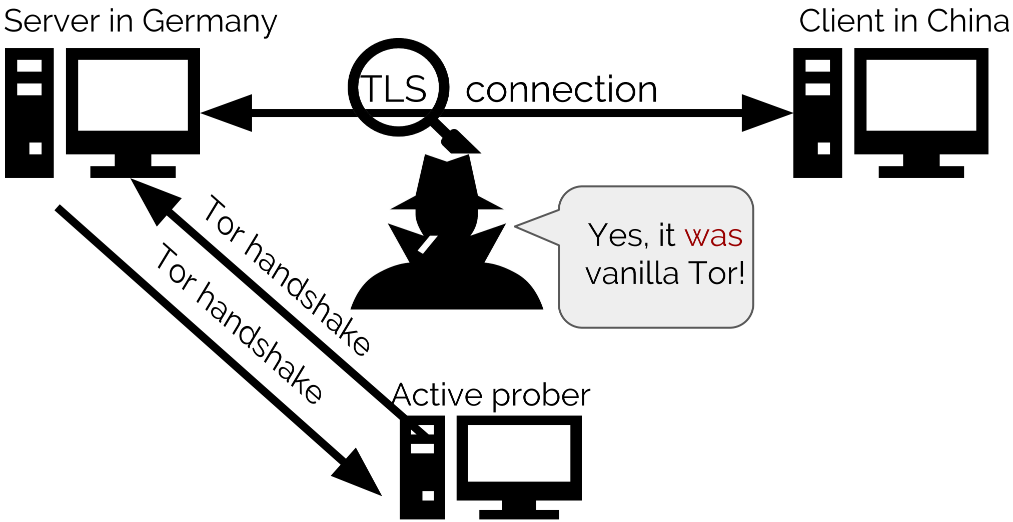
How does GFW Block Tor Hidden Circumvention Servers?

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



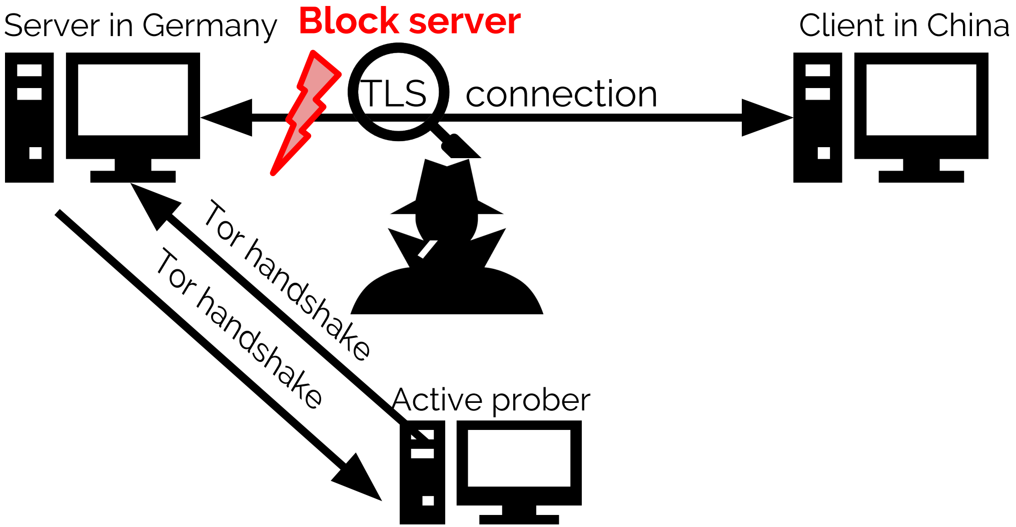
How does GFW Block Tor Hidden Circumvention Servers?

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

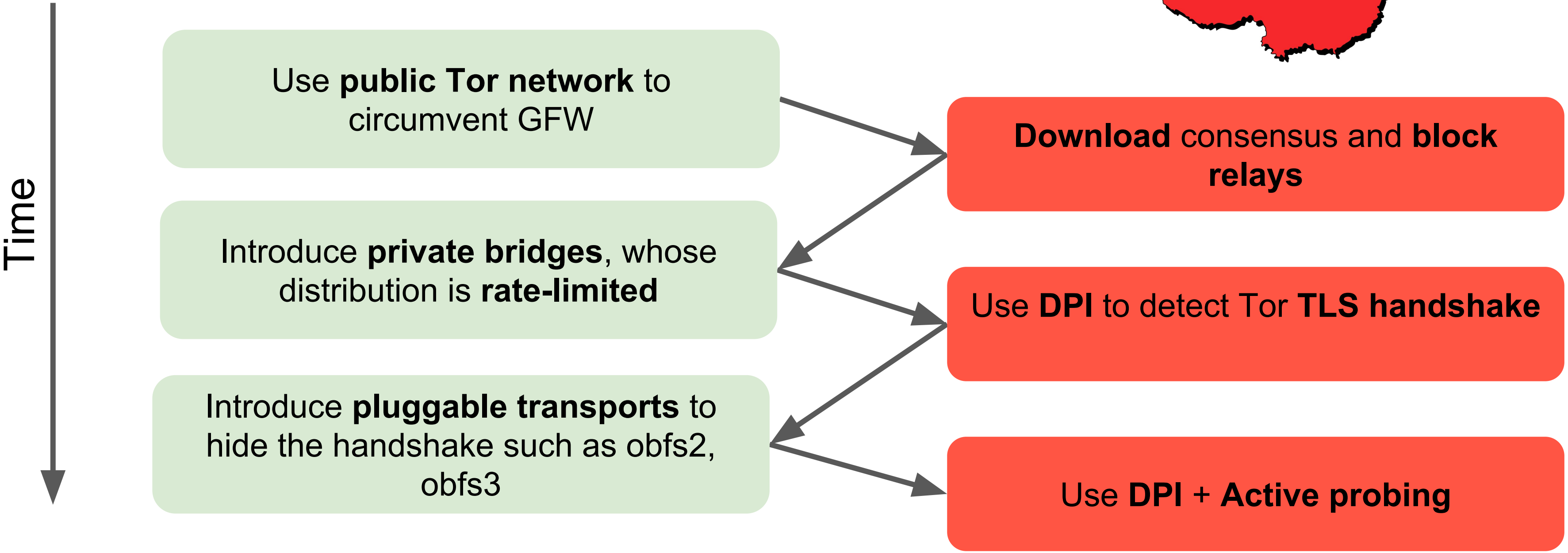


How does GFW Block Tor Hidden Circumvention Servers?

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



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**

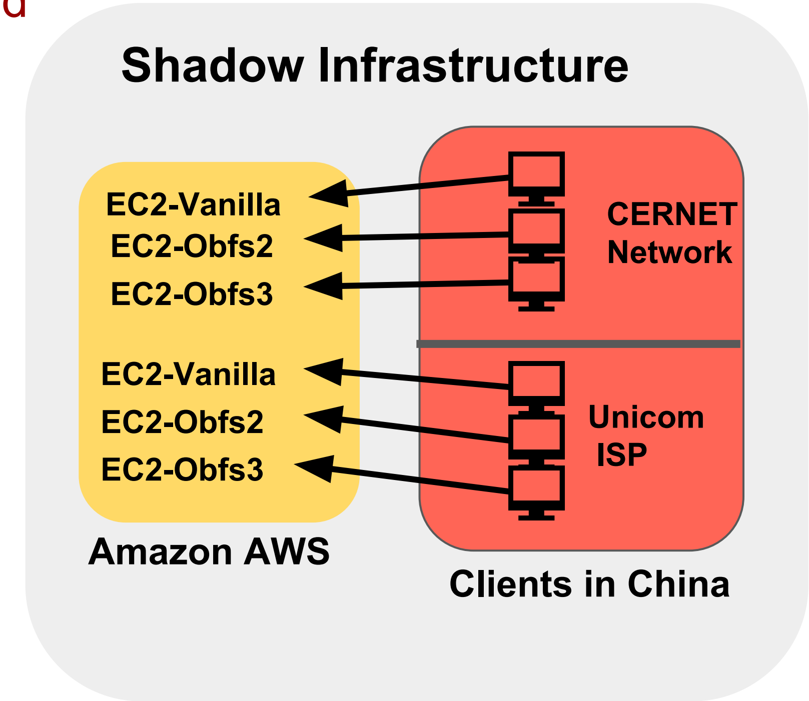
Many Questions about Active Probing are Unanswered!

- 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?

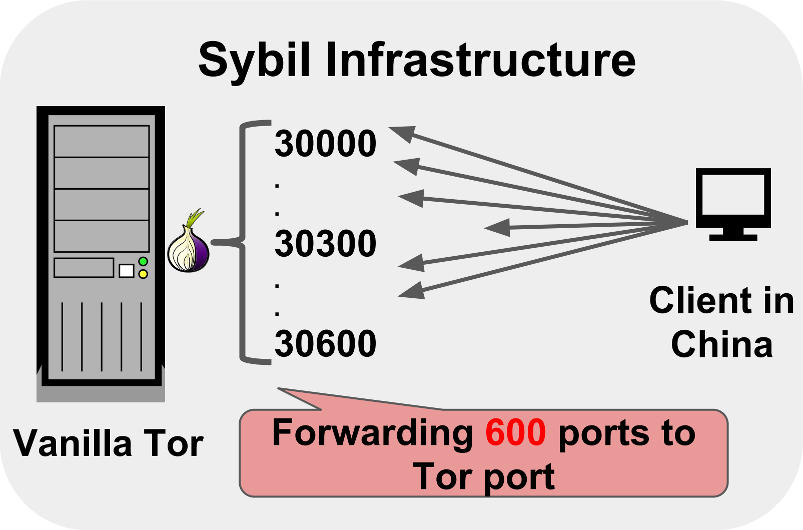
Overview of Our Datasets:

- Clients in China repeatedly connected to bridges under our control
- 3 months
- pcap files of both the clients and the bridges



Overview of Our Datasets:

- Redirected 600 ports to Tor port
- Client in China connects to 600 ports
- 22 hours
- pcap files of both the client and the relay

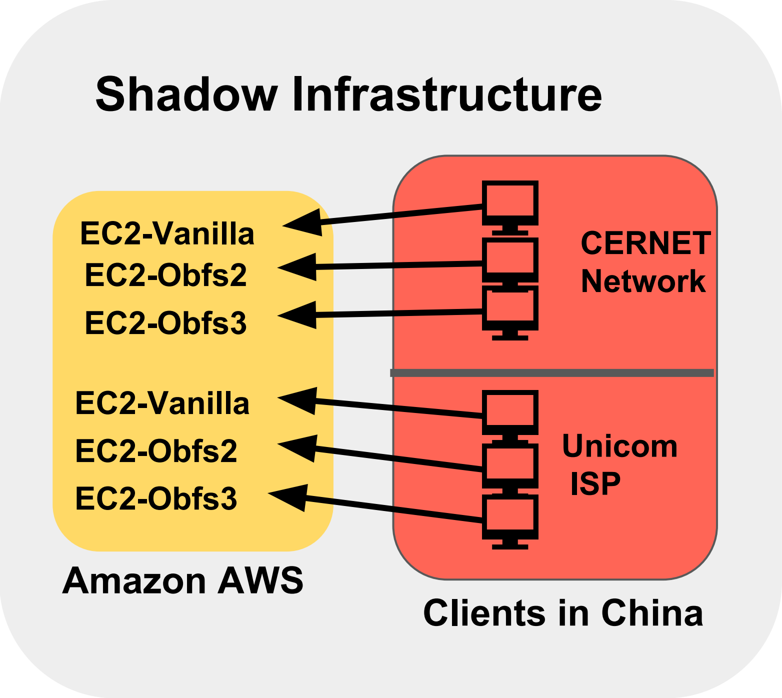


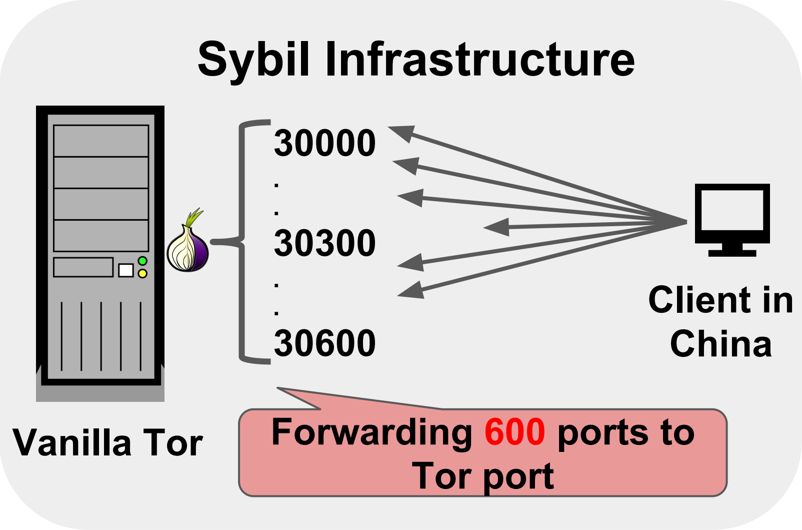
Overview of Our Datasets:

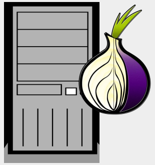
- Web server that also runs a Tor bridge located in US
- Web server logs dating back to Jan 2010

Server Log Analysis

Overview of Our Datasets:





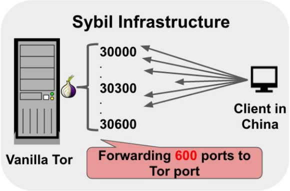


Server Log Analysis

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

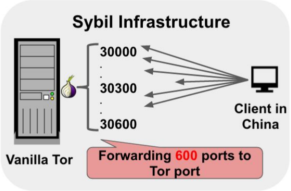
How to Distinguish Probers from Genuine Clients?

How to Distinguish Probers from Genuine Clients?



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

How to Distinguish Probers from Genuine Clients?



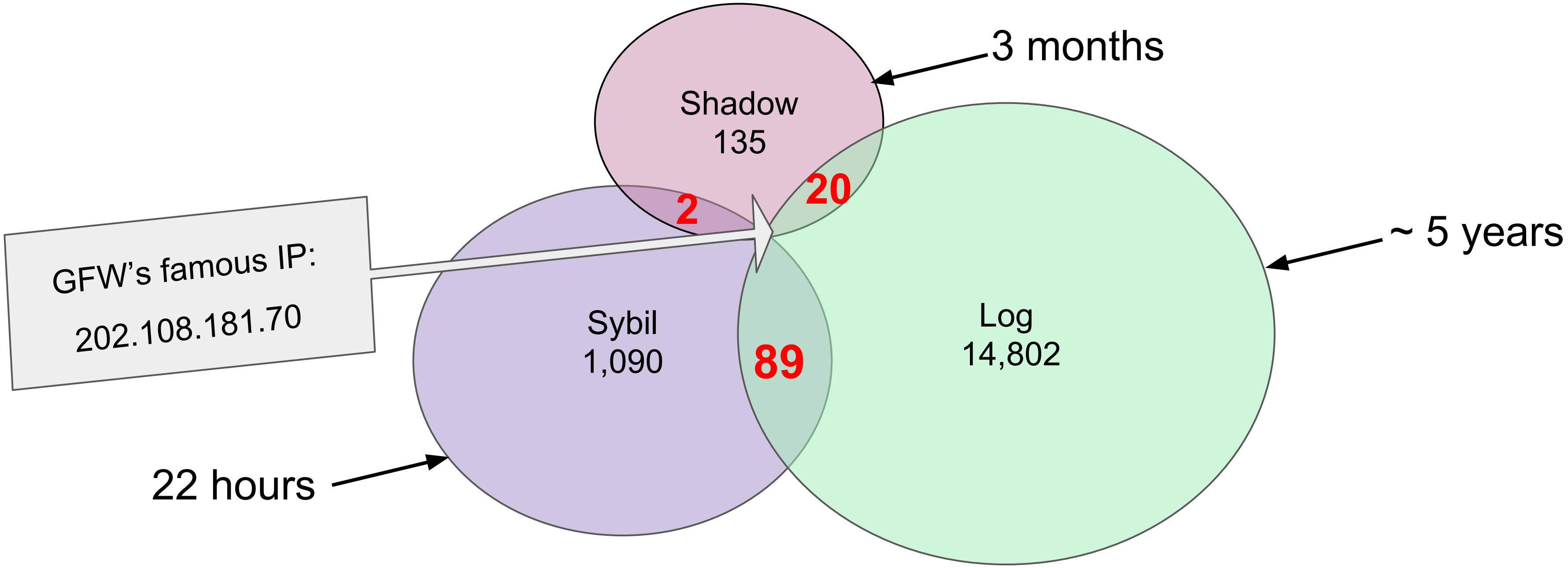
- Detecting probers in Sybil dataset is easy,
  - Probers:
    - Visited our vanilla Tor bridge after our client established connections
    - Originated from China

- 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
  - ...

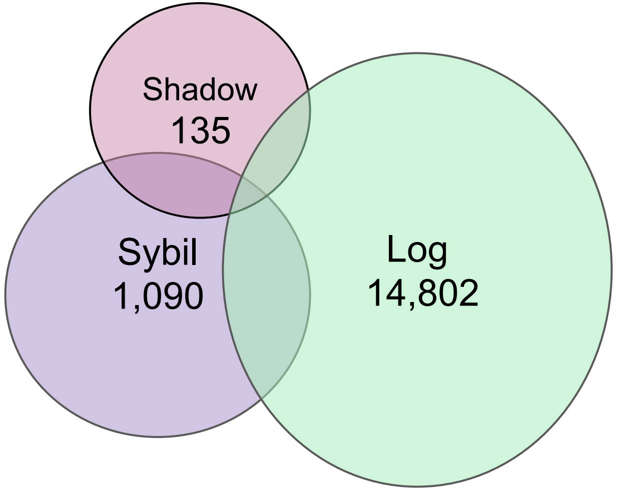
How Many Unique Probers did We Find?

How Many Unique Probers did We Find?

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

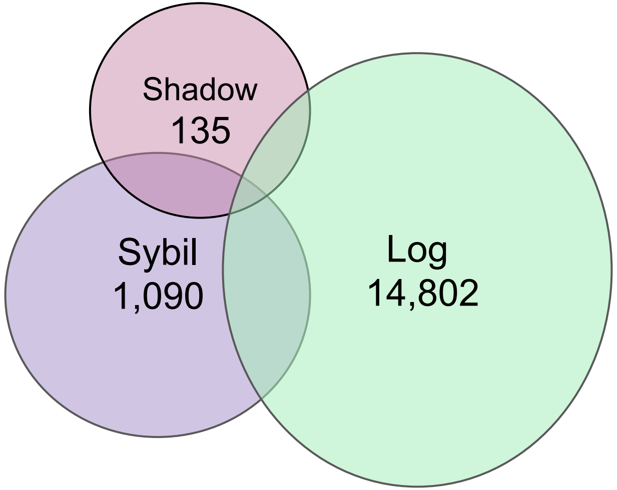


Where Are the Probers Coming from?



Where Are the Probes Coming from?

- Reverse DNS suggests **ISP pools**
  - adsl-pool.sx.cn
  - kd.ny.adsl
  - online.tj.cn
- Majority of probes come from **three** autonomous systems
  - ASN 4837, 4134, and 17622

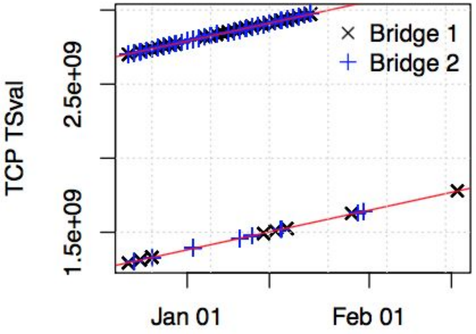


Can We Fingerprint Active Probers?

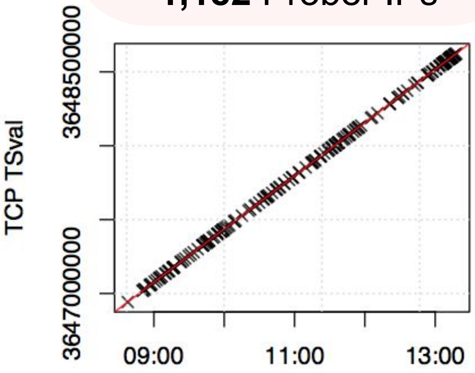
Can We Fingerprint Active Probers?

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

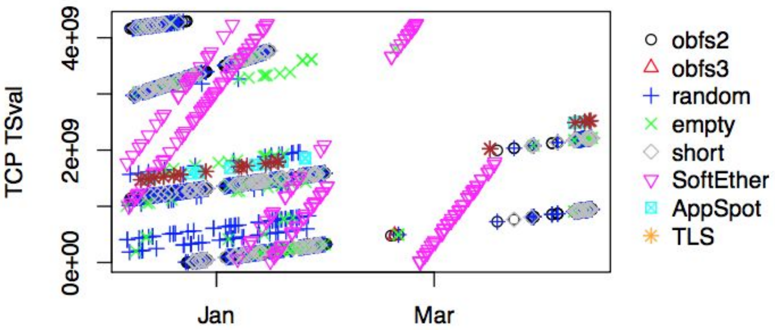
Shadow exp. with
158 Prober IPs



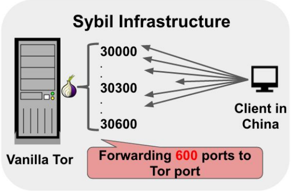
Sybil exp. with
1,182 Prober IPs



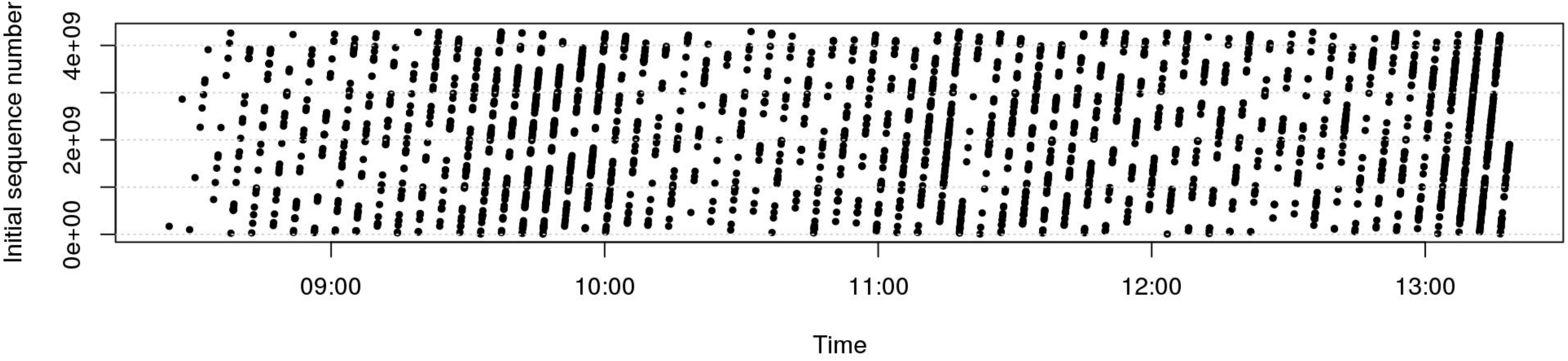
Log dataset with
14,912 Prober IPs



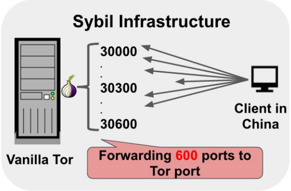
Can We Fingerprint Active Probers?



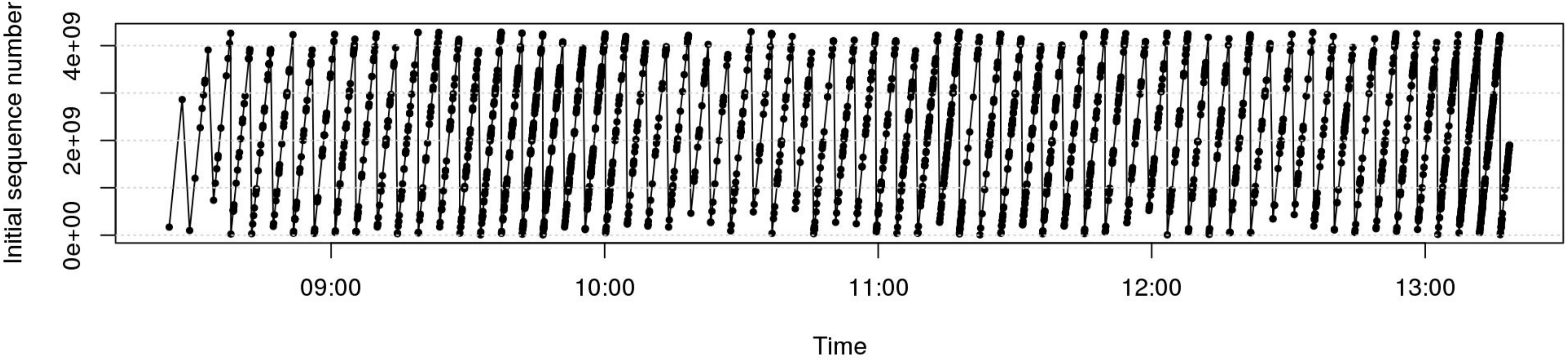
- TCP layer
  - Striking pattern in initial sequence numbers (derived from time) of 1,182 probes
  - Shared pattern in TSval for all three datasets



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

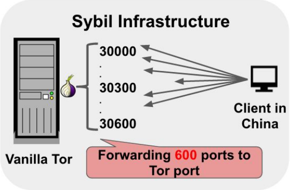


What do These Patterns Mean?

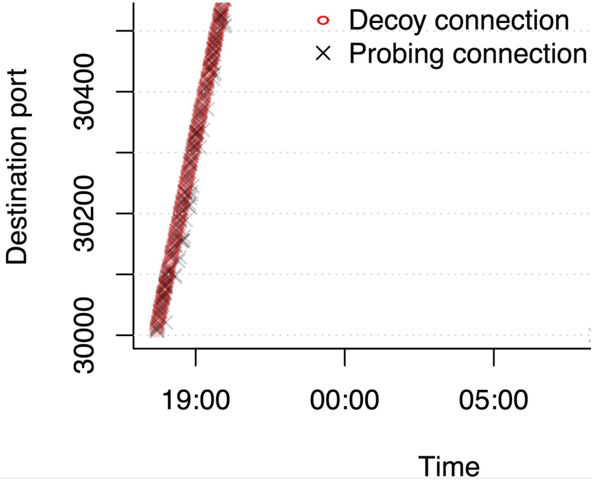
- Active probing connections **leak shared state**
    - ISNs, TSval, source ports, ...
- GFW likely operates only **few physical systems**
- Thousands of IP addresses are controlled by **central source**

How Quickly do Active Probes Show Up?

How Quickly do Active Probes Show Up?



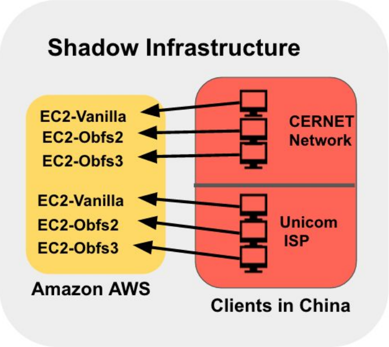
- Sybil dataset shows that system now works in **real time**
  - Median delay between Tor connection & subsequent probing connection is **~500ms**
  - **1,182** distinct probes showed up in 22 hs

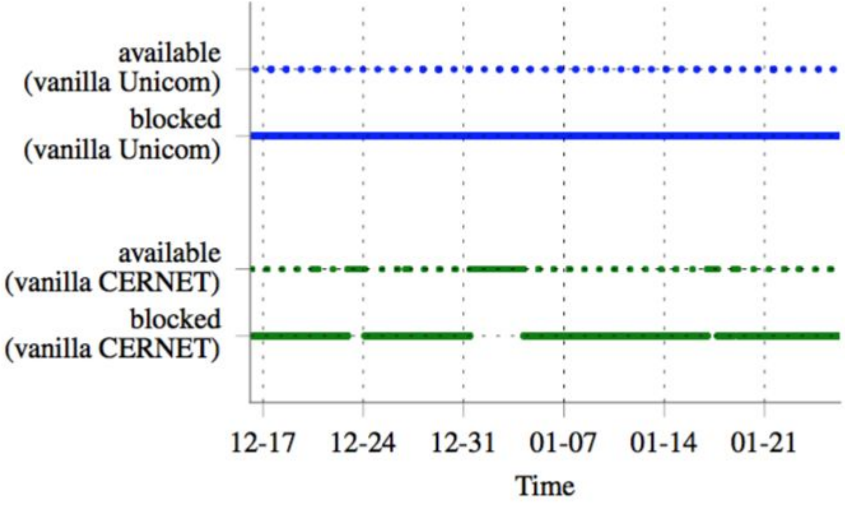


Is Active Probing Successful?

Is Active Probing Successful?

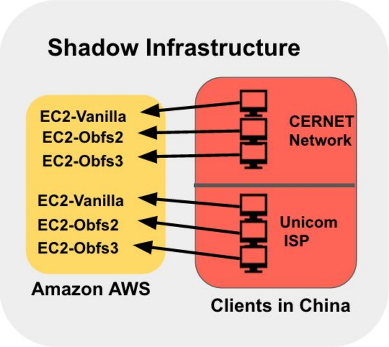
- Tor clients succeed in connecting **roughly every 25 hs**
  - Might reflect **implementation artifact** of GFW

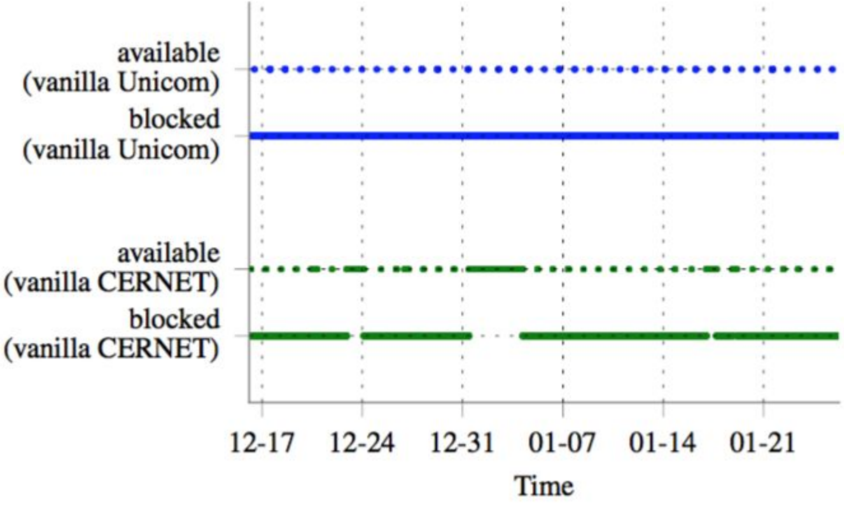




Is Active Probing Successful?

- Tor clients succeed in connecting **roughly every 25 hs**
  - Might reflect **implementation artifact** of GFW
- obfs2 and obfs3 (~98%) were almos always reachable for clients





Takeaway messages

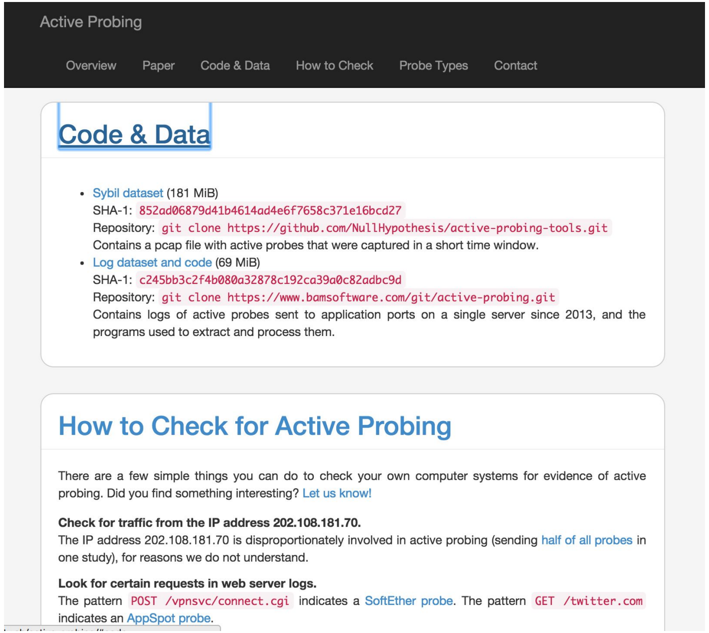
Our results show that the active probing system

- 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
- Twitter: @_ _royaen_ _



Code & Data

- Sybil dataset (181 MiB)
  SHA-1: `852ad06879d41b4614ad4e6f7658c371e16bcd27`
  Repository: `git clone https://github.com/NullHypothesis/active-probing-tools.git`
  Contains a pcap file with active probes that were captured in a short time window.
- Log dataset and code (69 MiB)
  SHA-1: `c245bb3c2f4b080a32878c192ca39a0c82adbc9d`
  Repository: `git clone https://www.bamsoftware.com/git/active-probing.git`
  Contains logs of active probes sent to application ports on a single server since 2013, and the programs used to extract and process them.

How to Check for Active Probing

There are a few simple things you can do to check your own computer systems for evidence of active probing. Did you find something interesting? Let us know!

Check for traffic from the IP address 202.108.181.70.
The IP address 202.108.181.70 is disproportionately involved in active probing (sending half of all probes in one study), for reasons we do not understand.

Look for certain requests in web server logs.
The pattern `POST /vpnsvc/connect.cgi` indicates a SoftEther probe. The pattern `GET /twitter.com` indicates an AppSpot probe.