Lec #11: Surveillance and Circumvention

Examining How the Great Firewall Discovers Hidden Circumvention

Roya Ensafi
rensafi@cs.princeton.edu

Oct 20, 2016

Roadmap

- Surveillance / Censorship Location Doesn’t Matter!
  - The Great Cannon of China

- Is the GFW a country-wide, distributed NIDS?
  - Analyzing the Great Firewall of China Over Space and Time

- How is the GFW blocking Tor?
  - Examining How the GFW Discovers Hidden Circumvention Servers*

* Applied Networking Research Prize, IRTF 2016

Roadmap

- Surveillance / Censorship Location Doesn’t Matter!
  - The Great Cannon of China

- Is the GFW a country-wide, distributed NIDS?
  - Analyzing the Great Firewall of China Over Space and Time

- How is the GFW blocking Tor?
  - Examining How the GFW Discovers Hidden Circumvention Servers

Surveillance / Censorship Location Doesn’t Matter!

- Why care about surveillance and censorship in other countries?
  - **Advocacy** is important

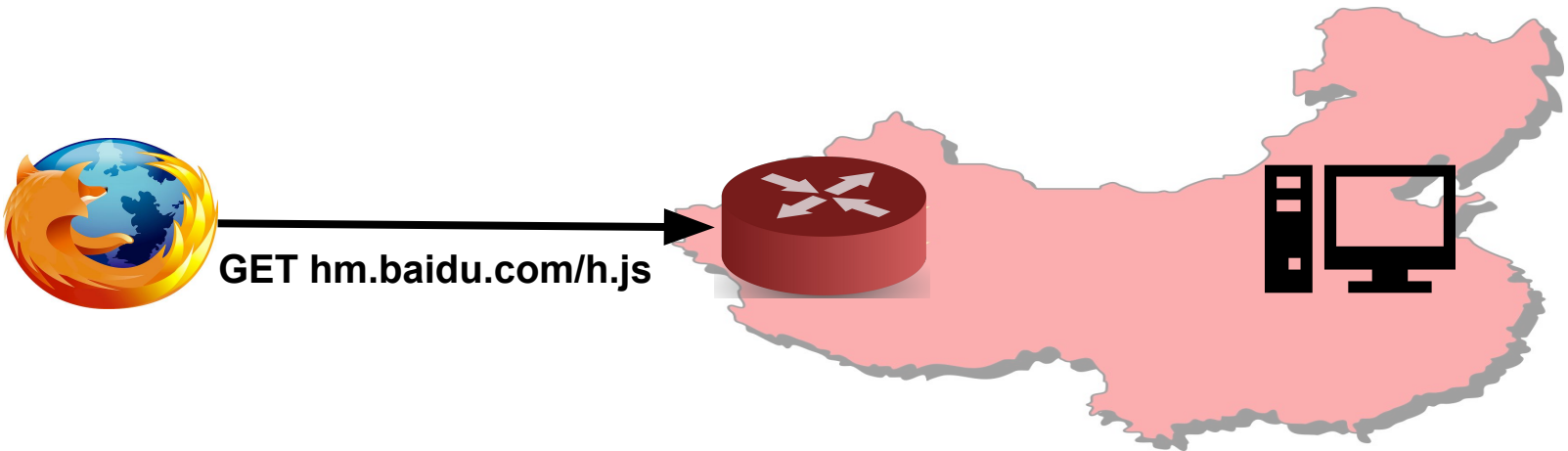
Surveillance / Censorship Location Doesn’t Matter!

- Why care about surveillance and censorship in other countries?
  - **Advocacy** is important
  - National firewalls can **harm international** traffic
  - GFW can cause harm → Great Cannon [1]

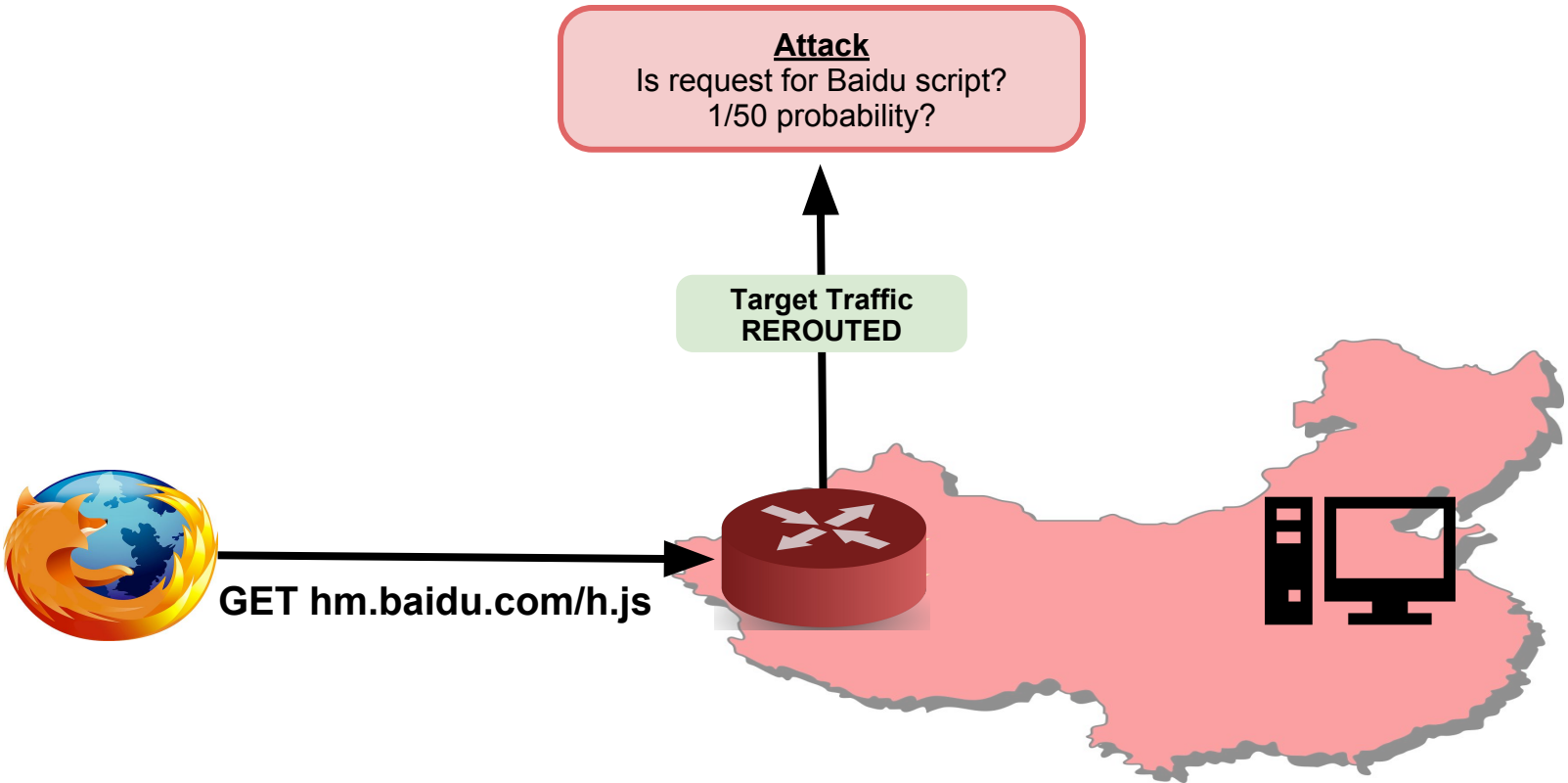
[1] **Analysis of China’s “Great Cannon”**
by B. Marczak, N. Weaver, J. Dalek, **R. Ensafi**, D. Fifield, S. McKune, A. Rey, J. Scott-Railton, R. Deibert and V. Paxson
(In: *USENIX FOCI’15*)

China's Attack Against GitHub

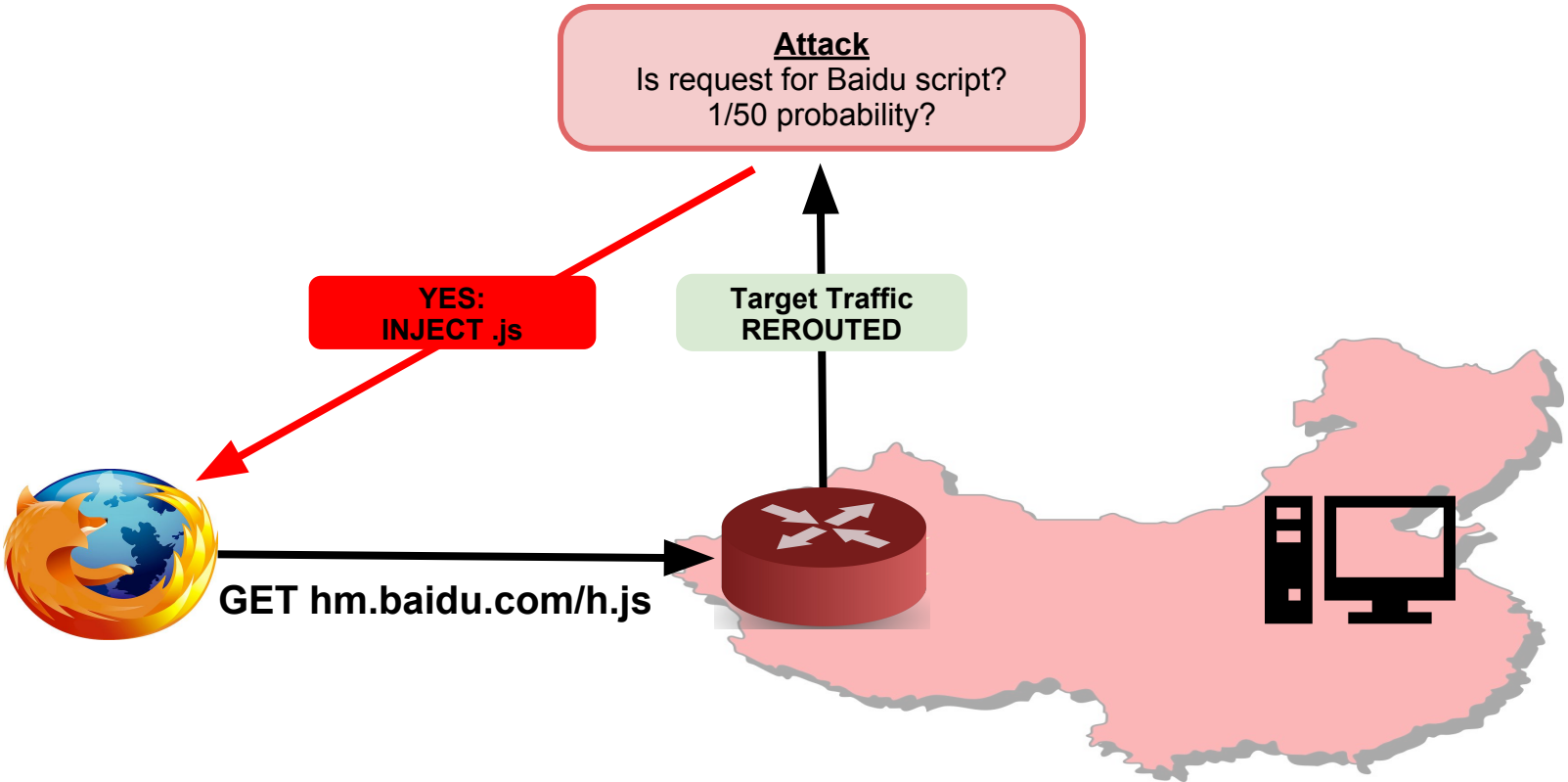
- The **Great Cannon** of China is an attack tool
- Target were hosters of **greatfire.org**
  - **GitHub** and **Cloudflare**



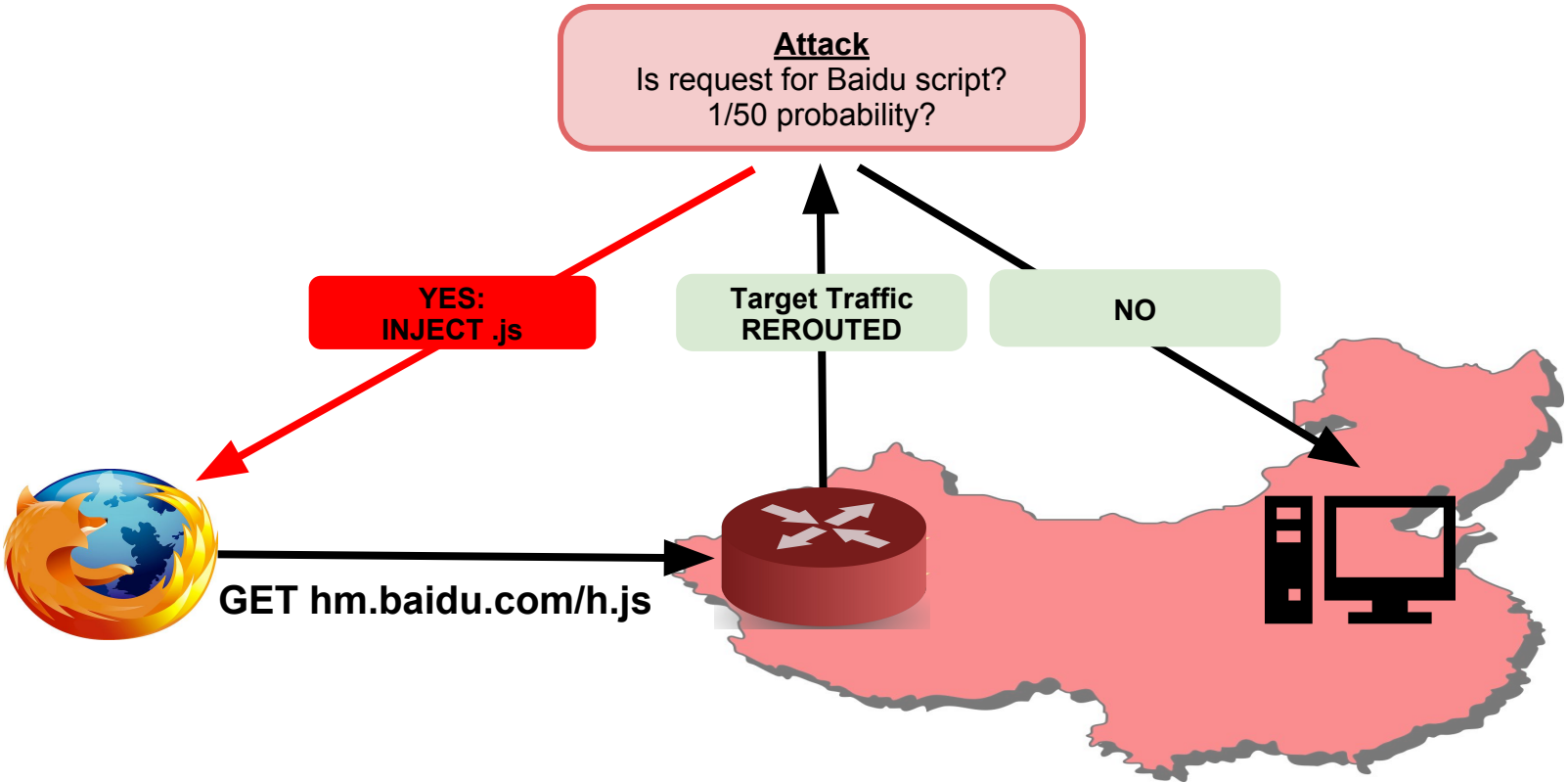
Great Cannon



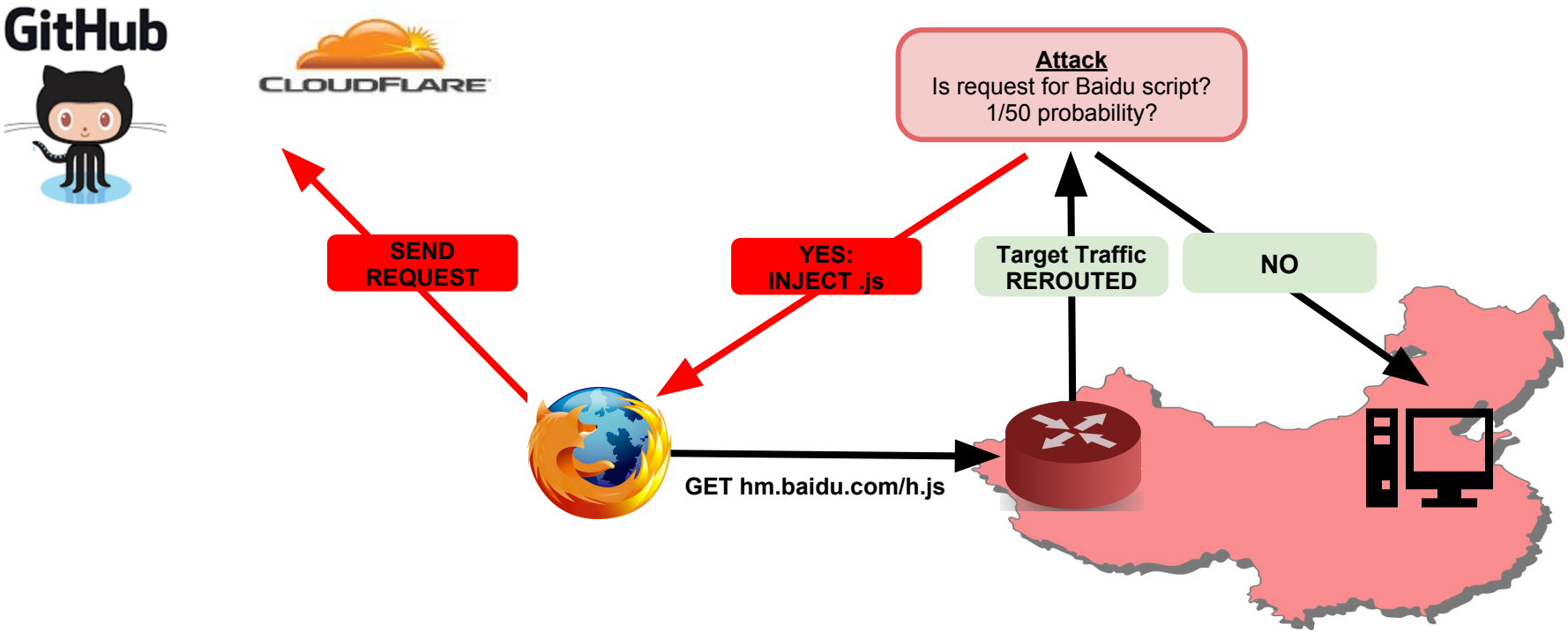
Great Cannon



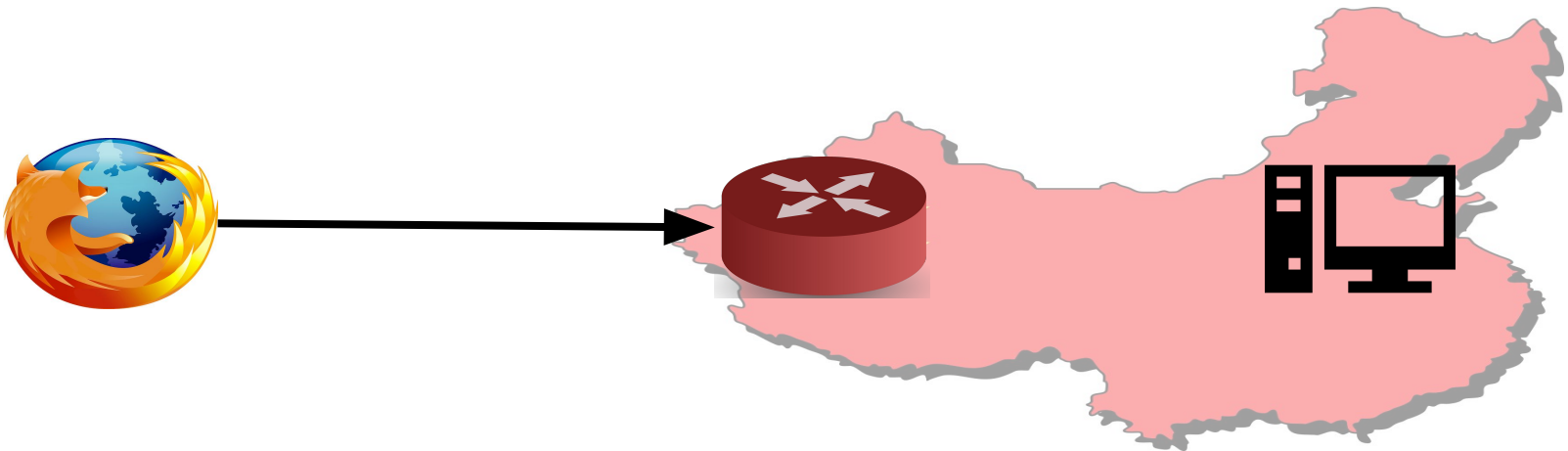
Great Cannon



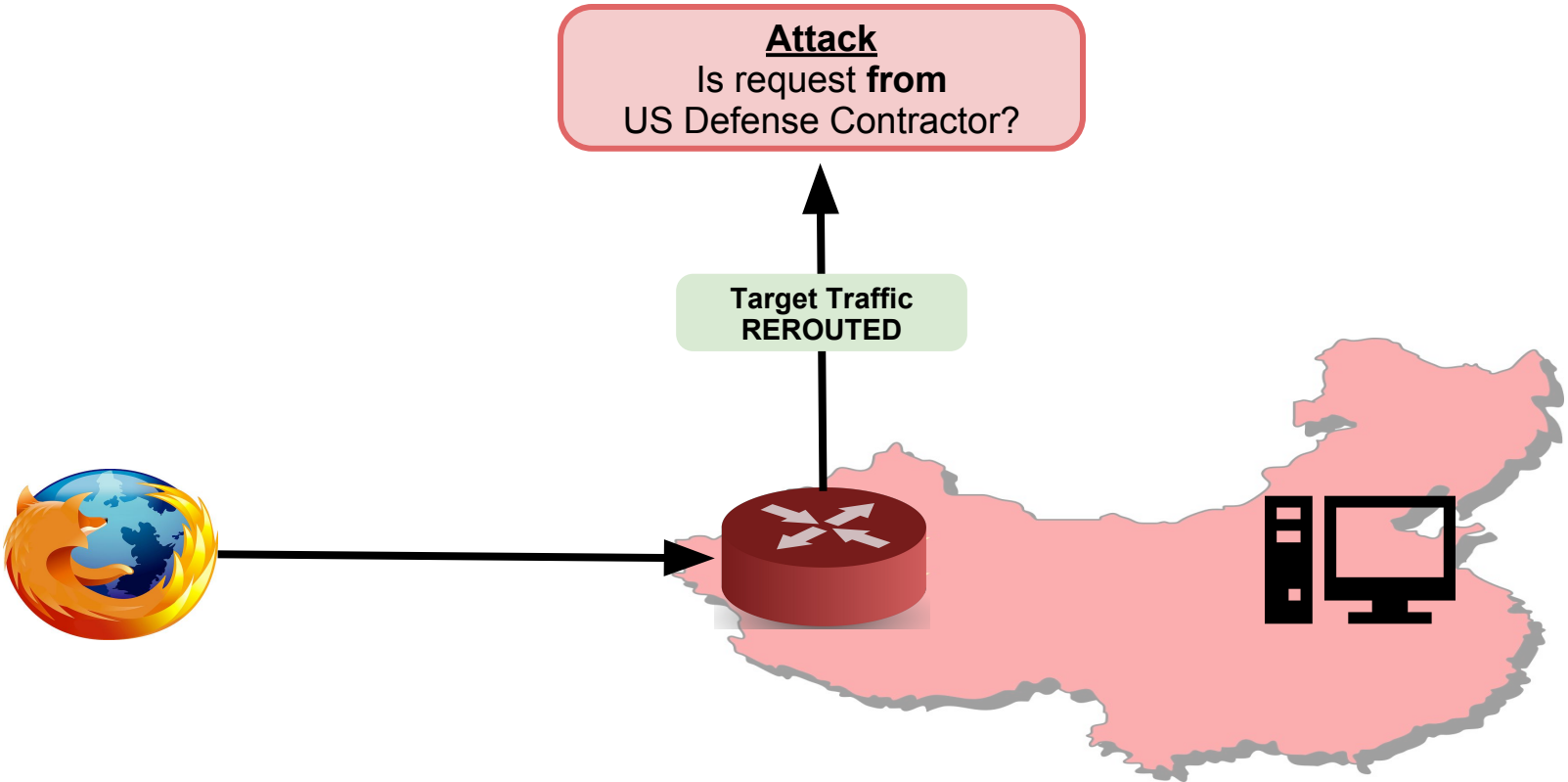
Great Cannon



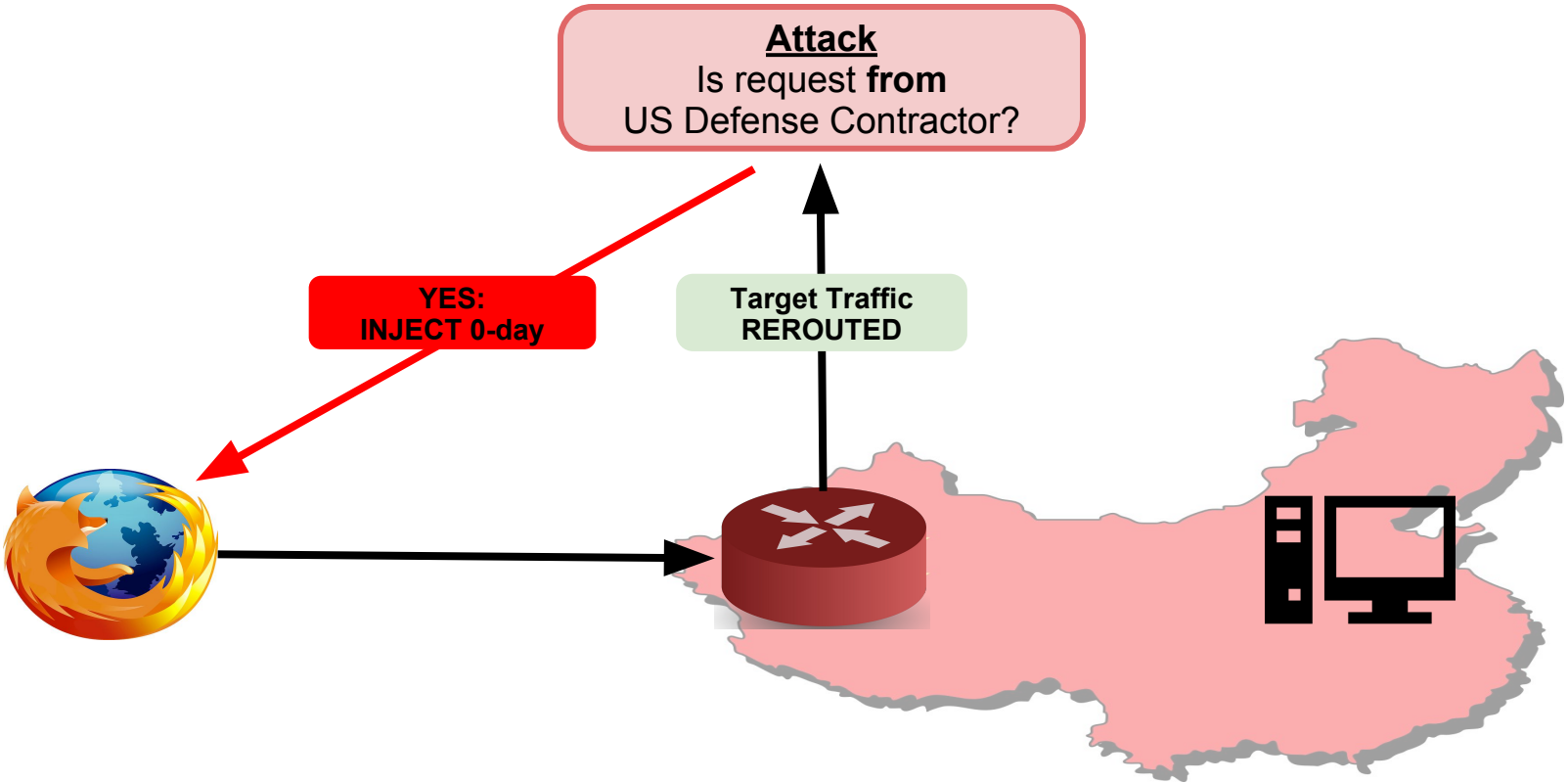
Great Cannon for Targeted Exploitation



Great Cannon for Targeted Exploitation



Great Cannon for Targeted Exploitation



Lessons Learned

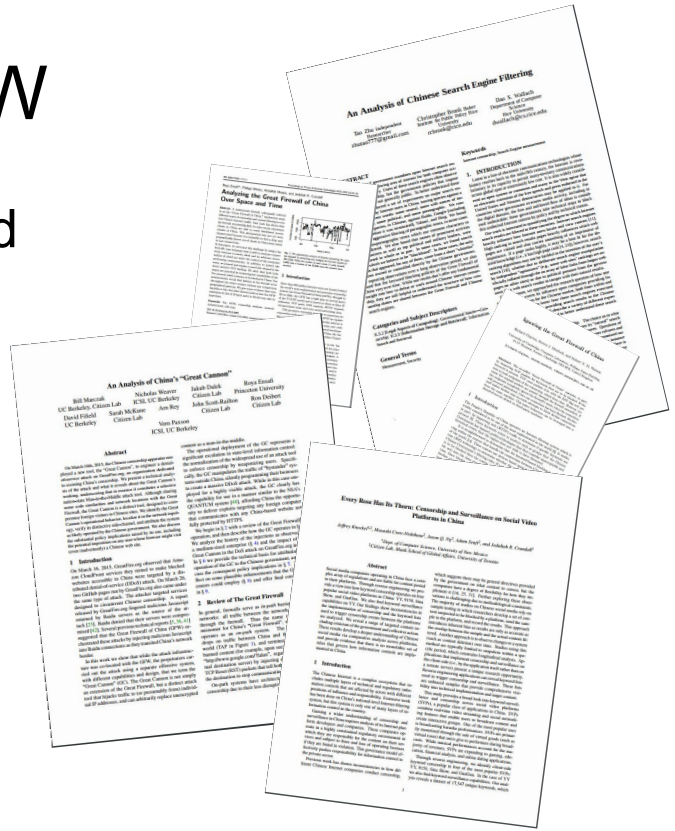
- Widespread deployment of **HTTPS crucial**
  - Particularly, we need HTTPS' **authentication** here, and not **encryption**

- Exploitation of plain text traffic **will not stop**

National Firewalls are powerful

Much Already Known About GFW

- Thanks to numerous research papers and blog posts, we know…
  - **What** is blocked
  - **How** it is blocked
- Is filtering centralized?



Roadmap

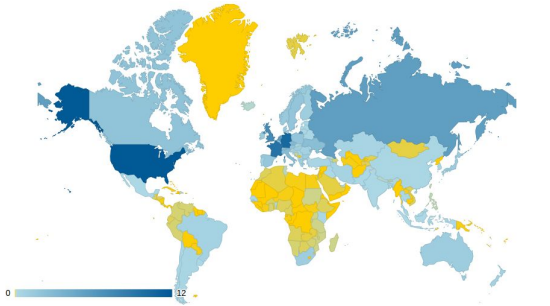
- Surveillance / Censorship Location Doesn’t Matter!
  - The Great Cannon of China

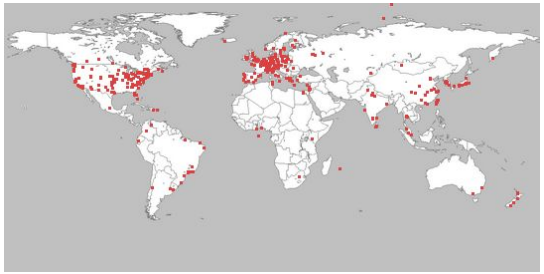
- Is the GFW a country-wide, distributed NIDS?
  - Analyzing the Great Firewall of China Over Space and Time

- How is the GFW blocking Tor?
  - Examining How the GFW Discovers Hidden Circumvention Servers

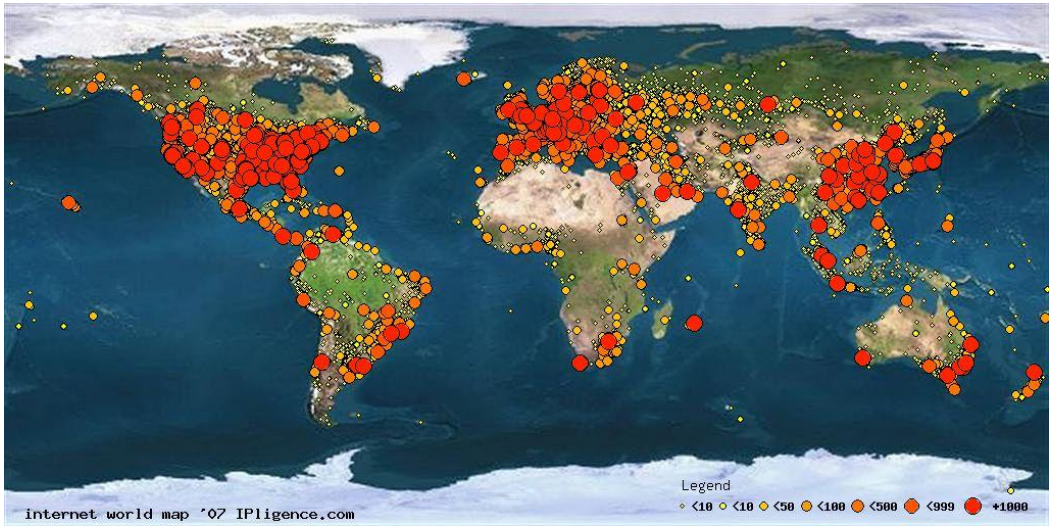
Is the GFW a Country-wide Distributed NIDS?

- Rent a **control machine** (VPS)
- Cooperate with **volunteers**
- **Not** always possible to rent VPS in interesting area



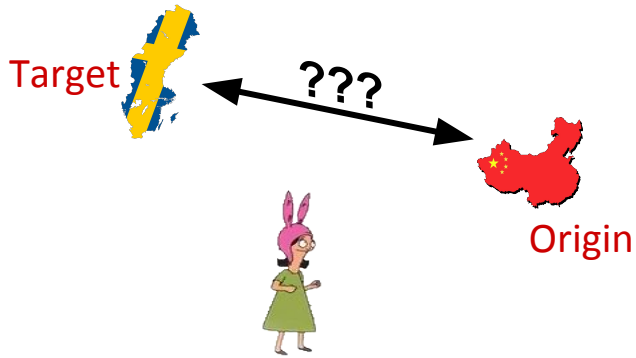


Side Channels to the Rescue!



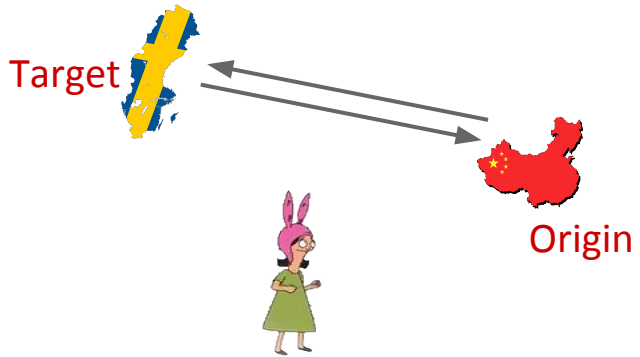
Solving the Vantage Point Problem

- Side channels turn **ordinary** machines into **vantage** points!



Idle Scan

- Idle Scan uses side channel techniques to bounce scans off of a “Target” host to stealthily scan an “Origin” [2]
- Spooky scan a.k.a Hybrid Idle scan can detect the **direction of blocking** between an Origin and a Target [3]

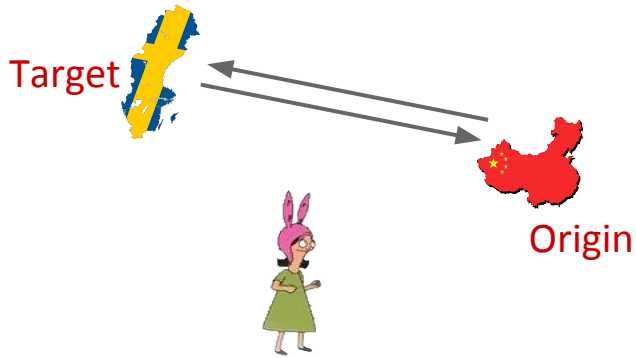


[2] **Idle Port Scanning and Non-interference Analysis of Network Protocol Stacks Using Model Checking**
by **Roya Ensafi**, J.Park, D. Kapur, and J. Crandall (In: *USENIX Sec’ 2010*)

[3] **Detecting Intentional Packet Drops on the Internet via TCP/IP Side Channels**
by **Roya Ensafi**, Jeffrey Knockel, Geoffrey Alexander, and Jedidiah R. Crandall (In: *PAM’14*)

Spooky Scan Requirements

- **Global IPID** machine for the Origin
- Target that has **open port**



Spooky Scan

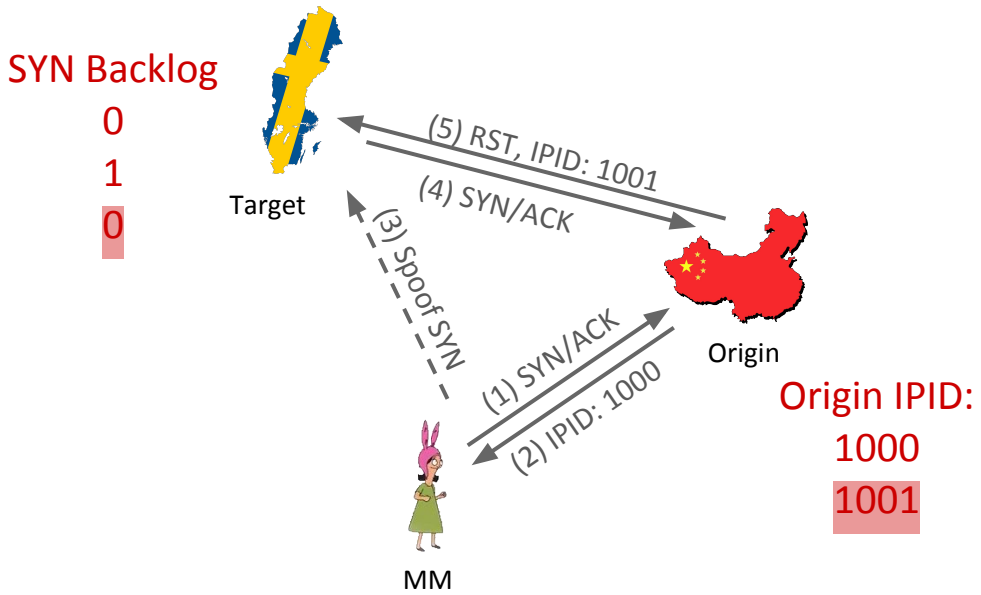
No direction blocked

Spooky Scan

No direction blocked

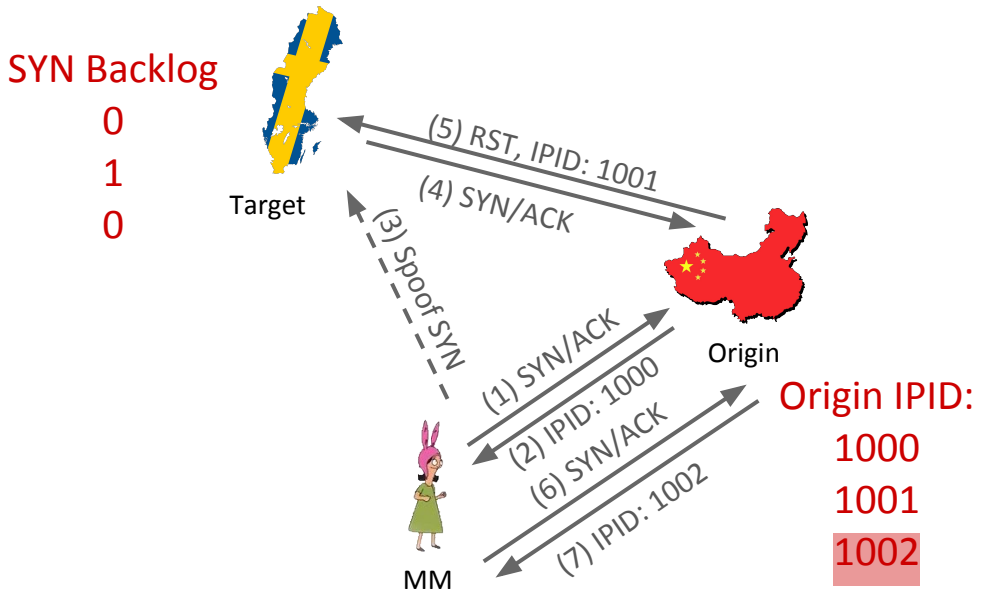
Spooky Scan

No direction blocked



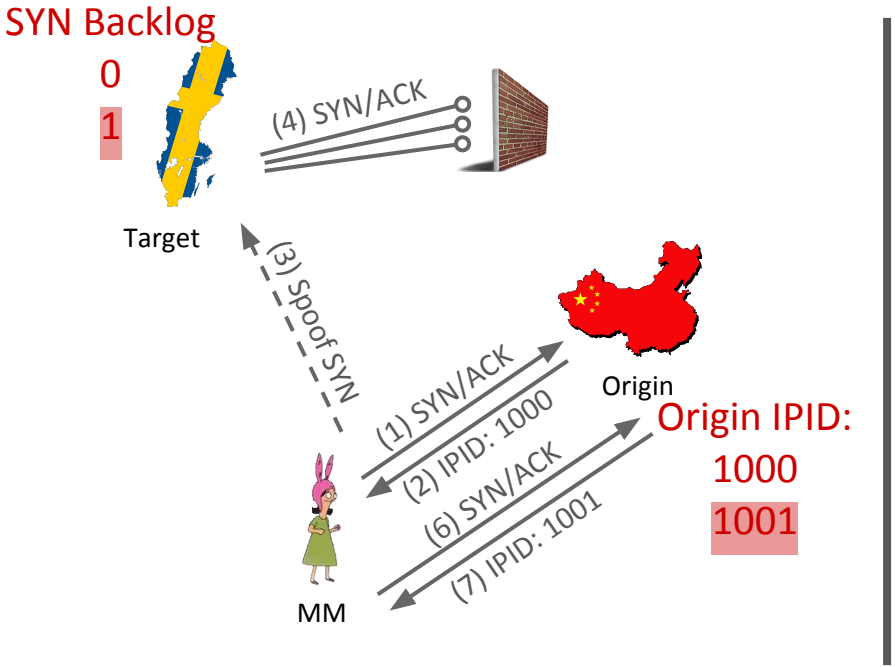
Spooky Scan

No direction blocked



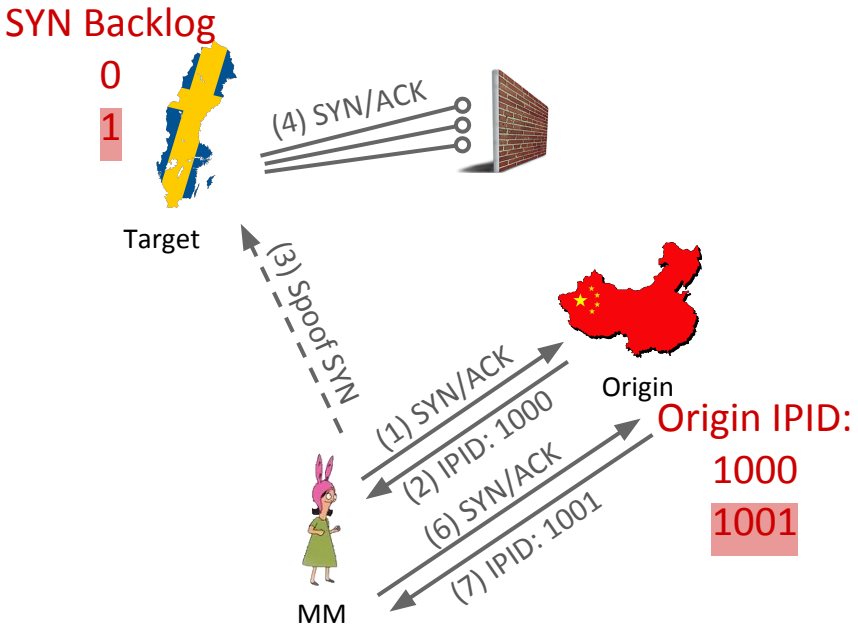
Spooky Scan

Target to Origin Blocked

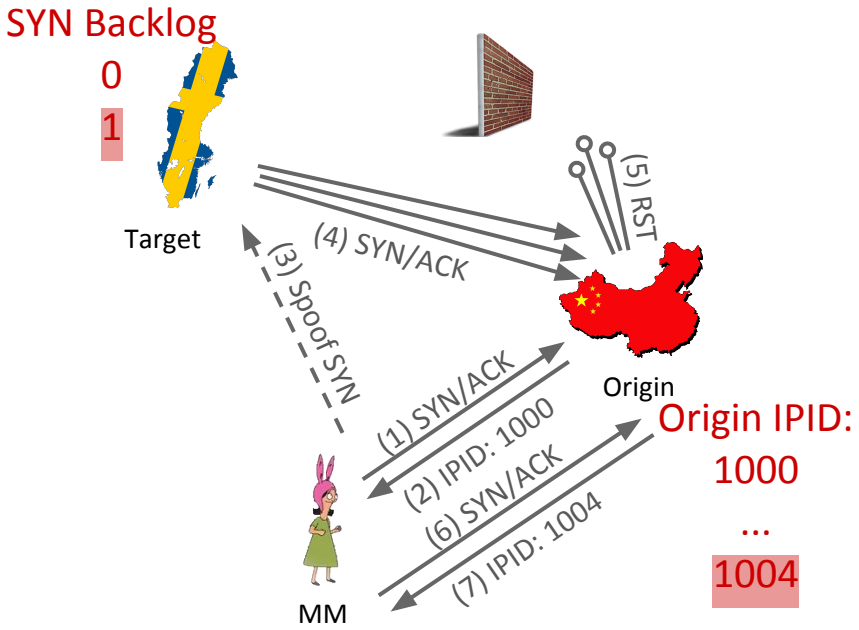


Spooky Scan

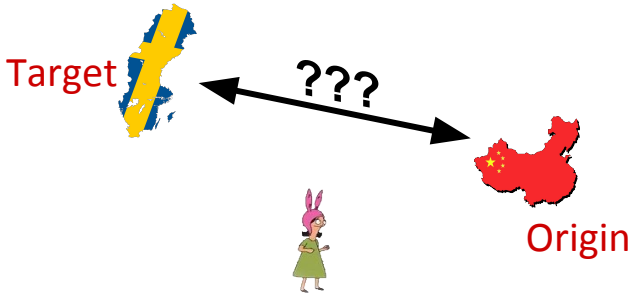
Target to Origin Blocked



Origin to Target Blocked

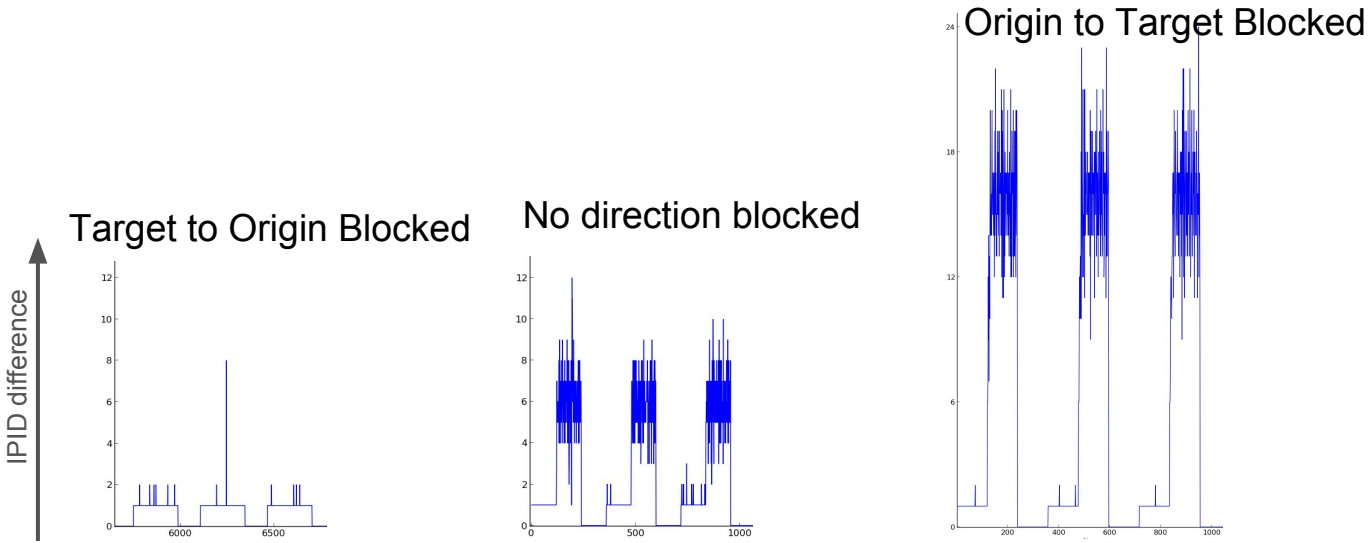


Real Data



Phase 1: just query IPID

Phase 2: send 5 spoofed SYN packets per sec & query IPID for 120 sec



Spooky scan is Used in Practice.

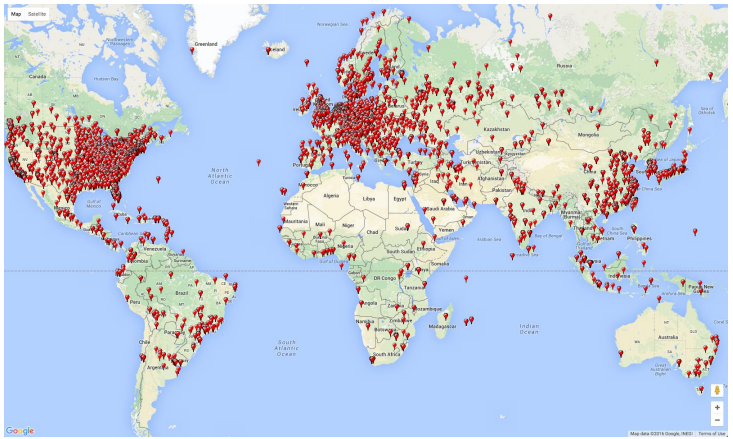
- Find **Global IPID** machines (list of origins)
- Find network you can **spoof** packets from
- Check if your targets satisfy necessary **requirements**
- Then, run the Spooky scan to detect the direction of blocking between origin and target

BOOM! DONE!

Remember the Challenges: Ethics, Scalability

- Side channels address **scalability**

- How can we address the **ethics** problem?
  - **Infrastructure** machines!



Can We Use Spooky Scan to Learn about GFW?

- Many **open questions** about the GFW and Tor
  - Does censorship of Tor differ for users in different regions?
  - Does filtering depend on **when and**

    where you are?
  - Does blocking **change** from one ISP to another?

- Revisit **old beliefs** about the GFW
  - Is the GFW a country-wide distributed NIDS?

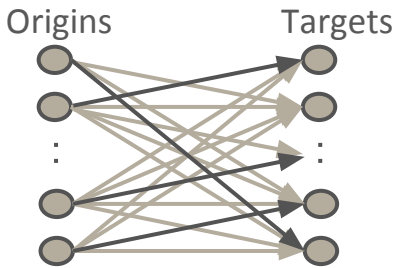
[4] **Analyzing the Great Firewall of China Over Space and Time**
by **Roya Ensafi**, P. Winter, A. Mueen, and J. Crandall (In: *PETS'15*)

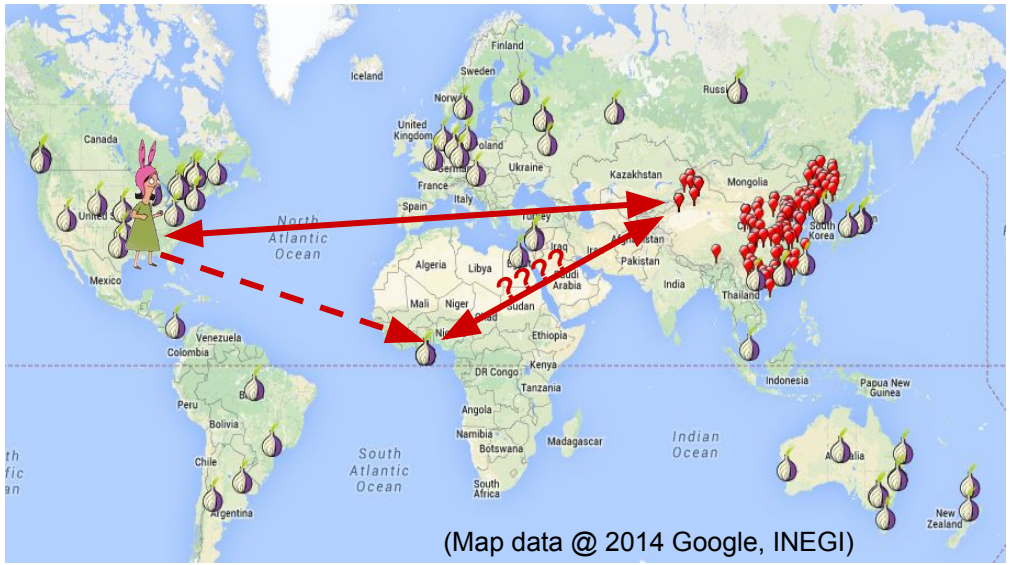
Methodology - Relays and Origins

(Map data @ 2014 Google, INEGI)

Methodology - Machines Under Our Control

- We ran Hybrid Idle Scans for **27 days**
- Each pair of origins and servers were tested **hourly for a day**





Results: No Obvious Geographical Pattern

- No **geographical** or **topological** pattern is visible. Instead, the distribution matches the geographic Internet penetration patterns of China.

- More in Ensafi et.al. PETS ‘15

Roadmap

- Surveillance / Censorship Location Doesn’t Matter!
  - The Great Cannon of China

- Is the GFW a country-wide, distributed NIDS?
  - Analyzing the Great Firewall of China Over Space and Time

- How is the GFW blocking Tor?
  - Examining How the GFW Discovers Hidden Circumvention Servers

How is the GFW blocking Tor?

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

Time

Use **public Tor network** to circumvent GFW

Download consensus and **block relays**

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

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

Download consensus and **block relays**

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

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

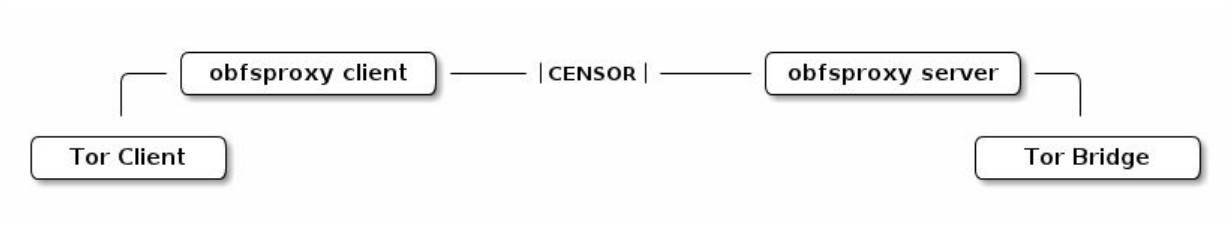
Introduce **pluggable transports** to hide Tor's handshake: obfs2, obfs3

Download consensus and **block relays**

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

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

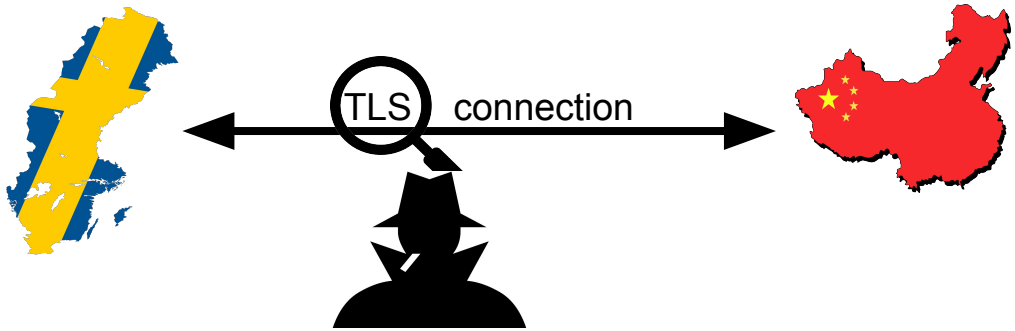
- Detection of pluggable transports is **uncertain**
  - Implies false positives → **collateral damage**

Encryption Reduces Blocking Accuracy

- Detection of pluggable transports is **uncertain**
  - Implies false positives → **collateral damage**
- GFW added **active probing** to complement the DPI fingerprinting

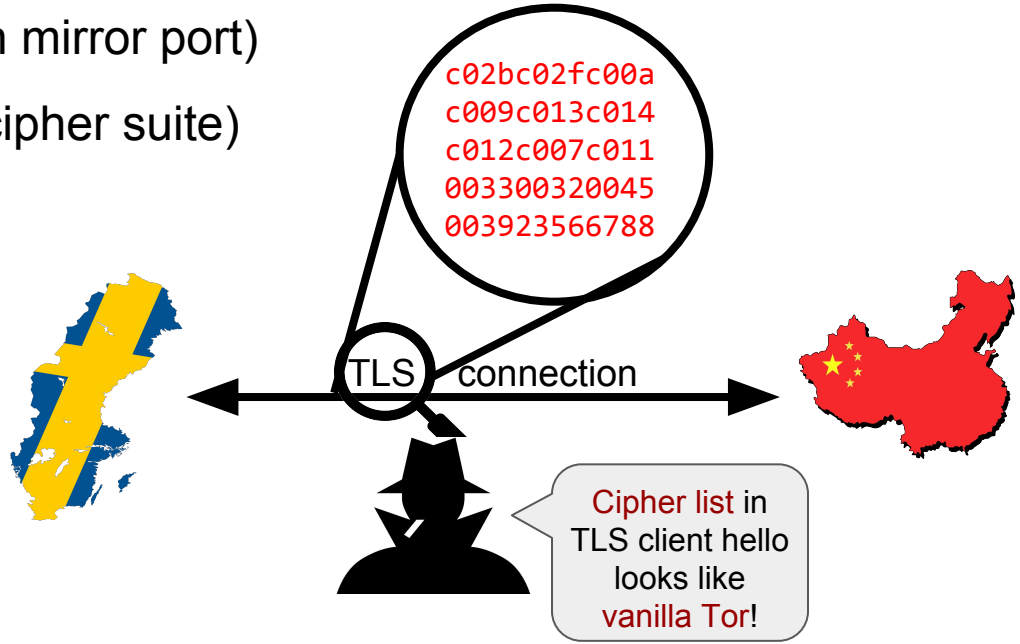
How does GFW Block Tor Hidden Circumvention Servers?

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



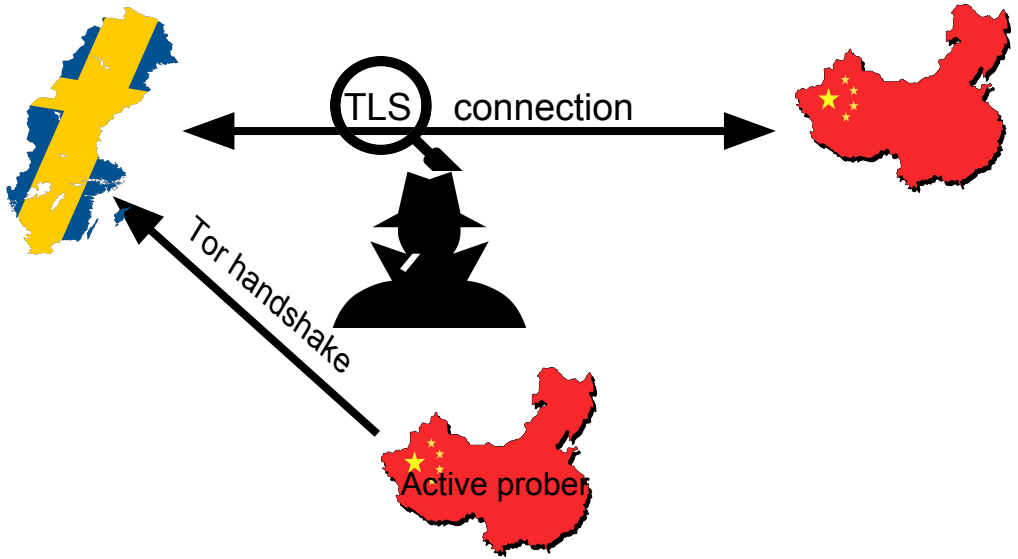
How does GFW Block Tor Hidden Circumvention Servers?

1. Network monitoring (e.g., switch mirror port)
2. DPI for suspicious traffic (e.g., cipher suite)



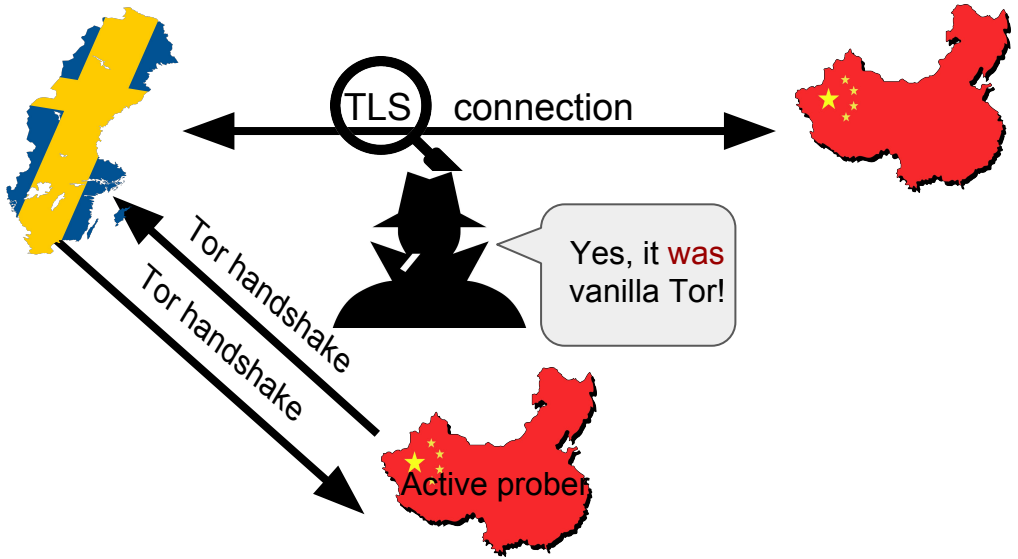
How does GFW Block Tor Hidden Circumvention Servers?

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



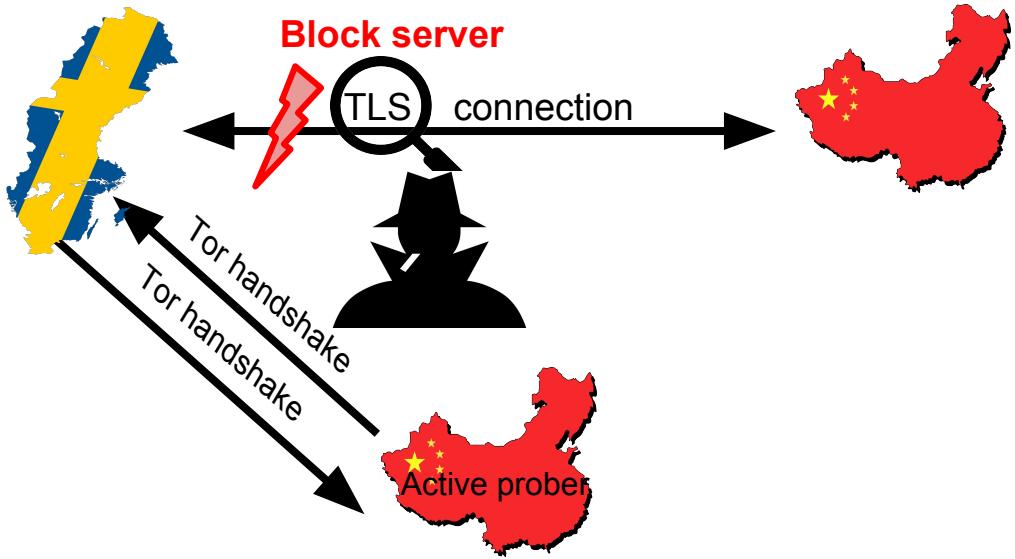
How does GFW Block Tor Hidden Circumvention Servers?

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

Time

Use **public Tor network** to circumvent GFW

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

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

Download consensus and **block relays**

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

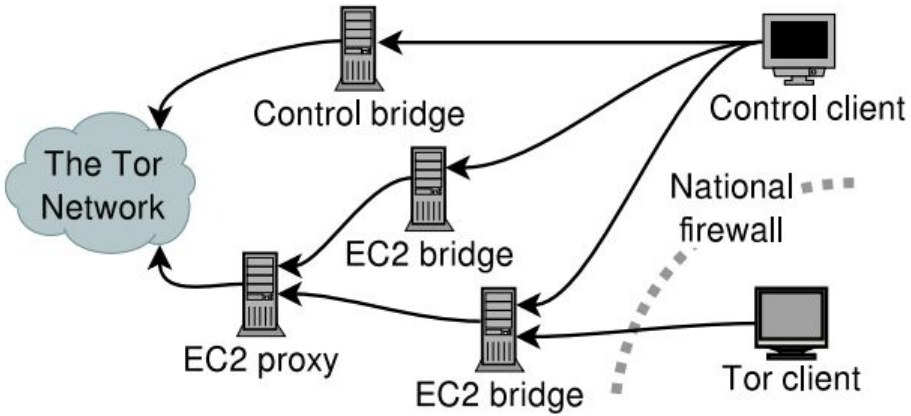
Use **DPI** + **Active probing**

Many Questions about Active Probing are Unanswered!

- We set out to answer: [5]
  - **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?

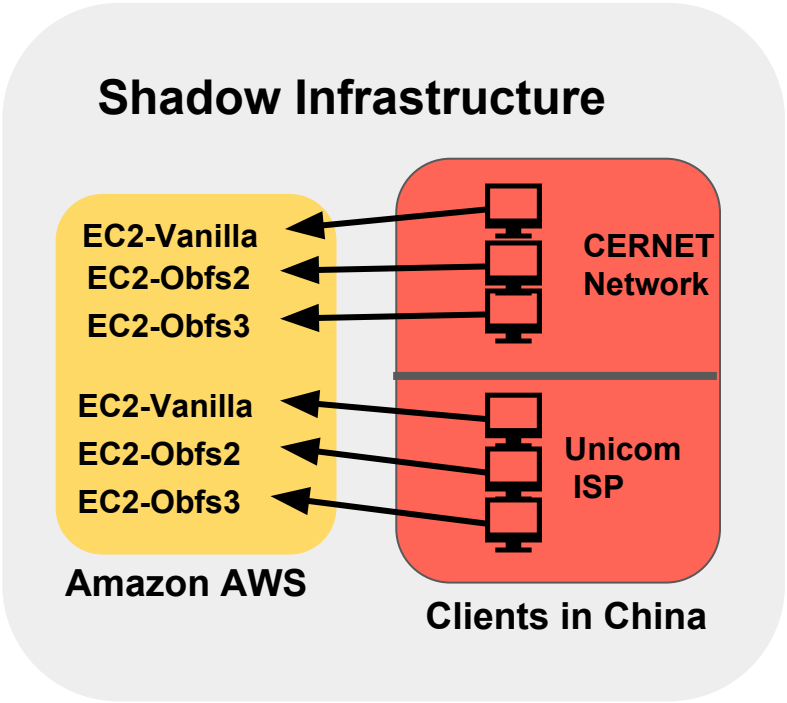
[5] **How the Great Firewall discovers hidden circumvention servers**
By **Roya Ensafi,** D. Fifield, P. Winter, N. Feamster, N. Weaver, and V. Paxson (In: *IMC'15*)

How Can We Design Measurements?



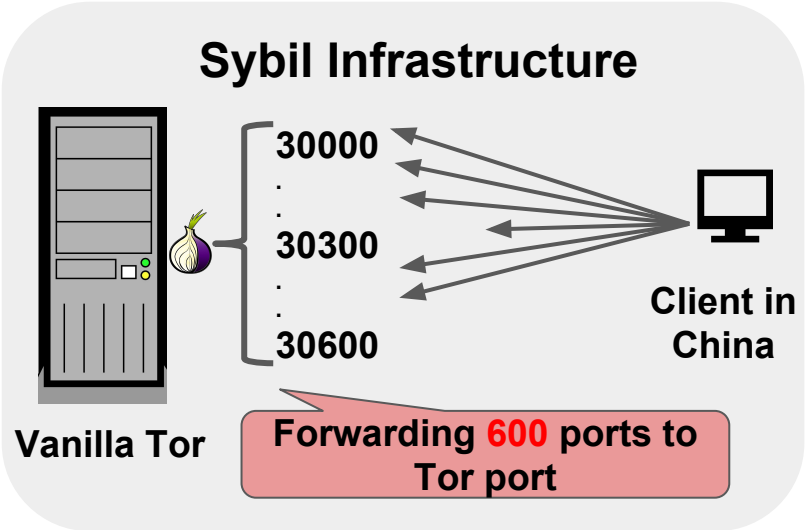
Overview of Our Datasets:

- Clients in China **repeatedly connected** to bridges under our control
- Pcap files of both the clients and the bridges



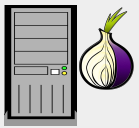
Overview of Our Datasets:

- **Redirected 600 ports to Tor port**
- Client in China connects to 600 ports
- Pcap files of both the client and the relay



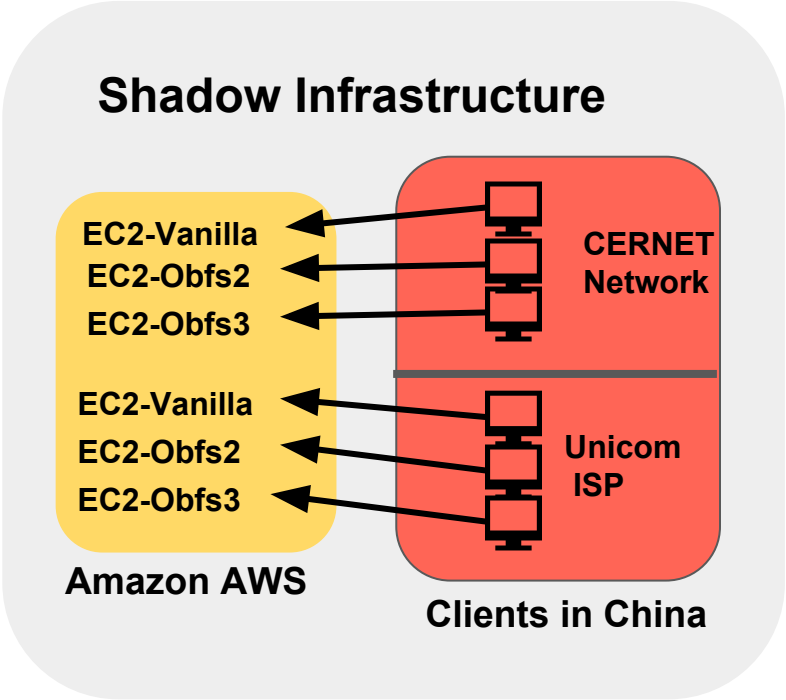
Overview of Our Datasets:

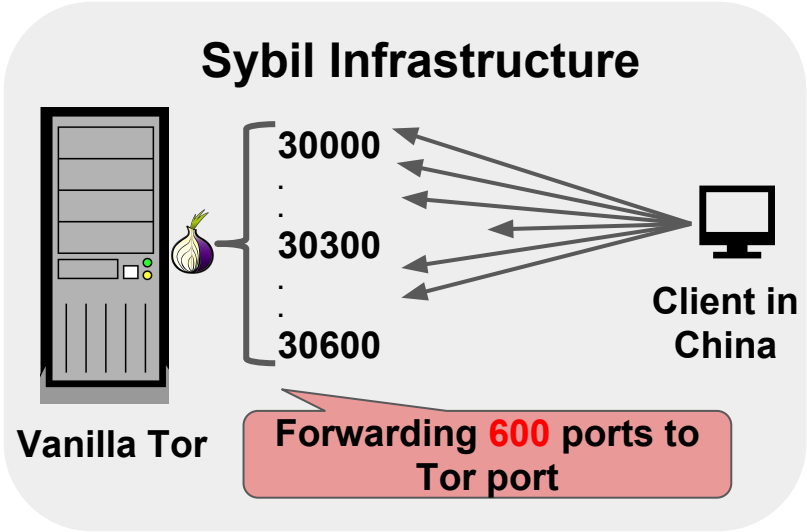
- Web server that also runs a Tor bridge located in US
- Web server logs dating back to Jan 2010

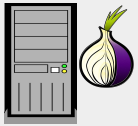


Server Log Analysis

Overview of Our Datasets:





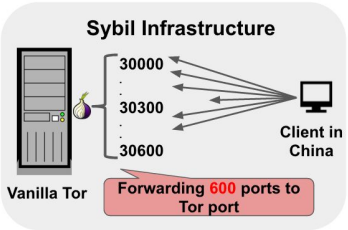


Server Log Analysis

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

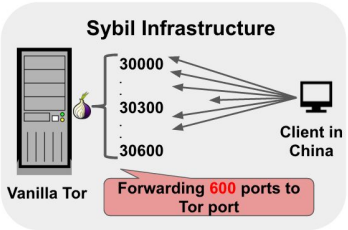
How to Distinguish Probers from Genuine Clients?

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?



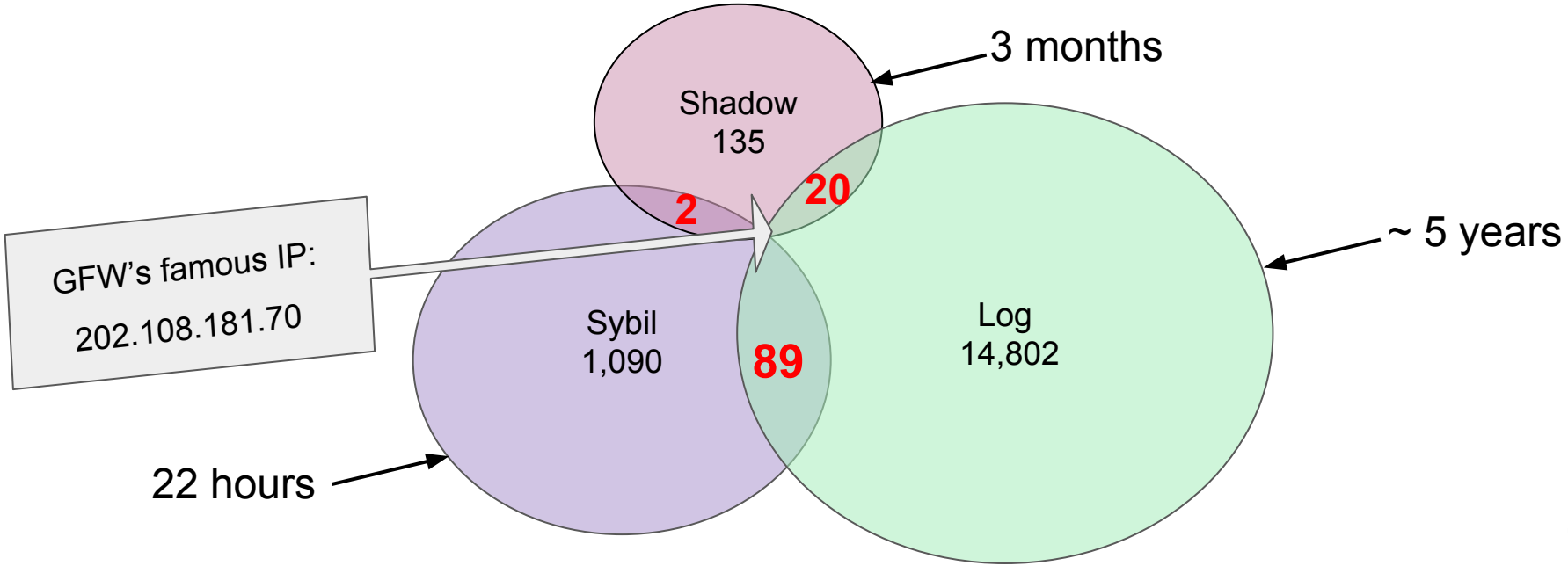
- Detecting probers in Sybil dataset is easy,
  - Probers:
    - Visited our vanilla Tor bridge after our client established connections
    - Originated from China

- 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



Can We Fingerprint Active Probers?

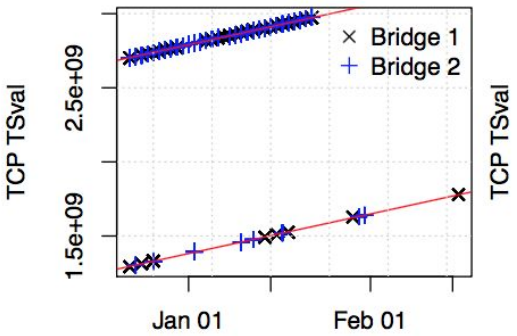
Can We Fingerprint Active Probers?

- TCP layer → TSval
  - TSval intercept: (rough) system uptime

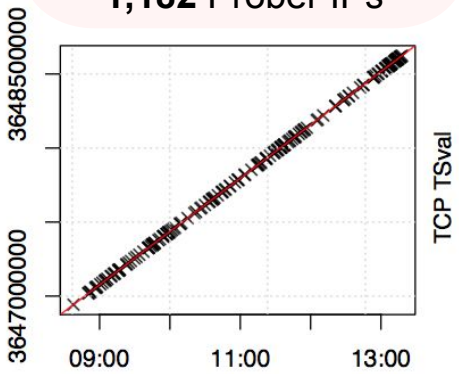
Can We Fingerprint Active Probers?

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

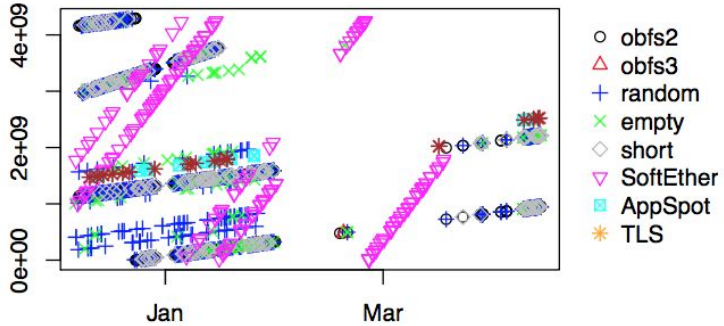
Shadow exp. with **158** Prober IPs



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

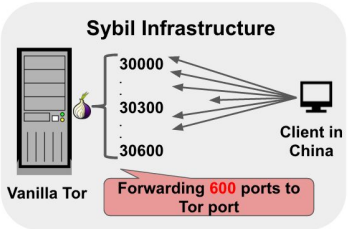


Log dataset with **14,912** Prober IPs



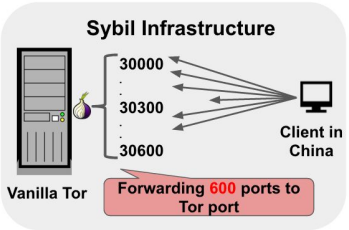
Can We Fingerprint Active Probers?

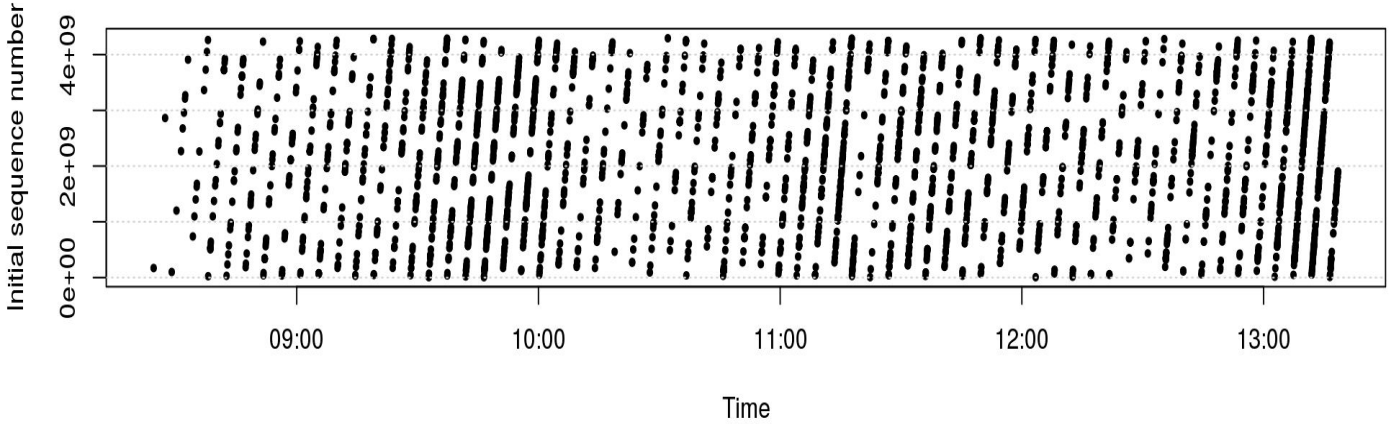
- TCP layer → ISN
  - TCP uses 32-bit initial sequence numbers (**ISNs**)
  - Protects against **off-path attackers**
  - Attacker must guess correct ISN range to inject segments
  - Every SYN segment should have **random** ISN



Can We Fingerprint Active Probers?

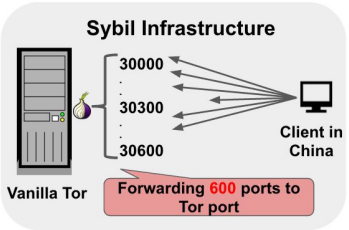
- TCP layer → ISN
  - TCP uses 32-bit initial sequence numbers (ISNs)
  - Protects against off-path attackers
  - Attacker must guess correct ISN range to inject segments
  - Every SYN segment should have random ISN

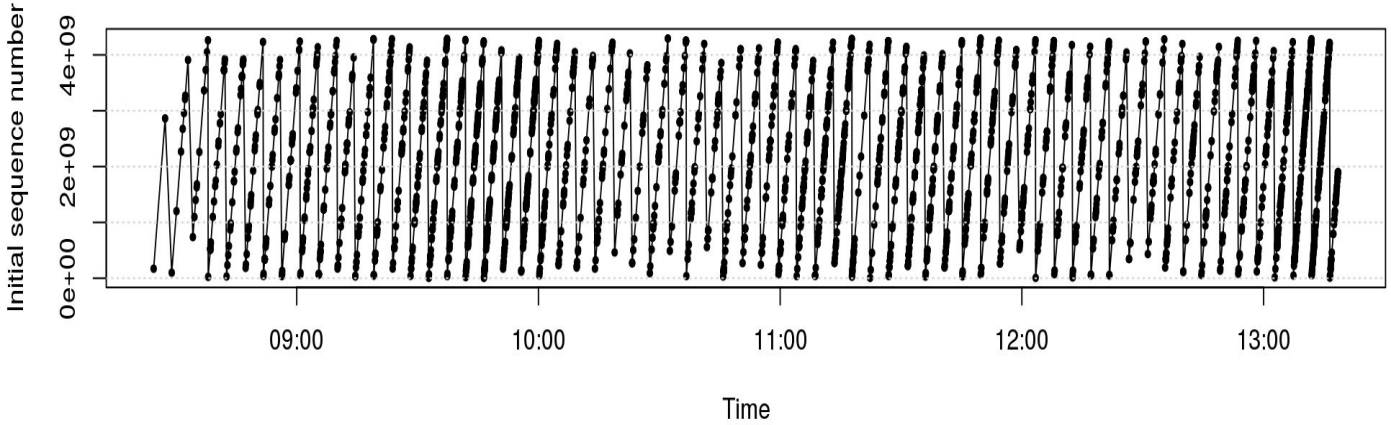




Can We Fingerprint Active Probers?

- TCP layer → ISN
  - TCP uses 32-bit initial sequence numbers (ISNs)
  - Protects against off-path attackers
  - Attacker must guess correct ISN range to inject segments
  - Every SYN segment should have random ISN



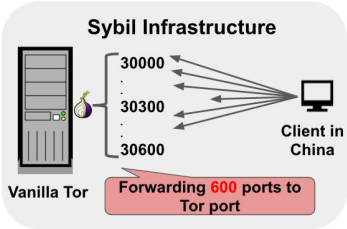


What do These Patterns Mean?

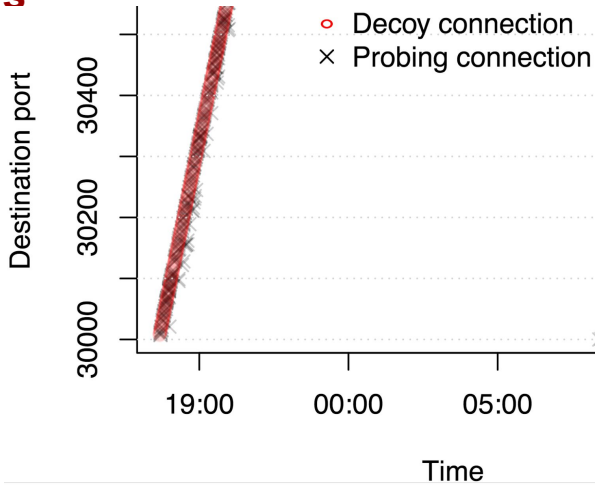
- Active probing connections **leak shared state**
  - ISNs, TSval, source ports, ...
- GFW likely operates only **few physical systems**
- Thousands of IP addresses are controlled by **central source**
- Does does not seem like off-the-shelf networking stack
- User space TCP stack?

How Quickly do Active Probes Show Up?

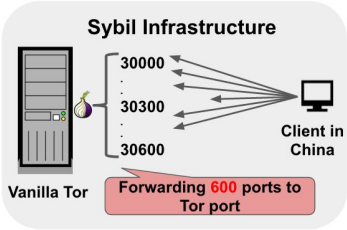
How Quickly do Active Probes Show Up?



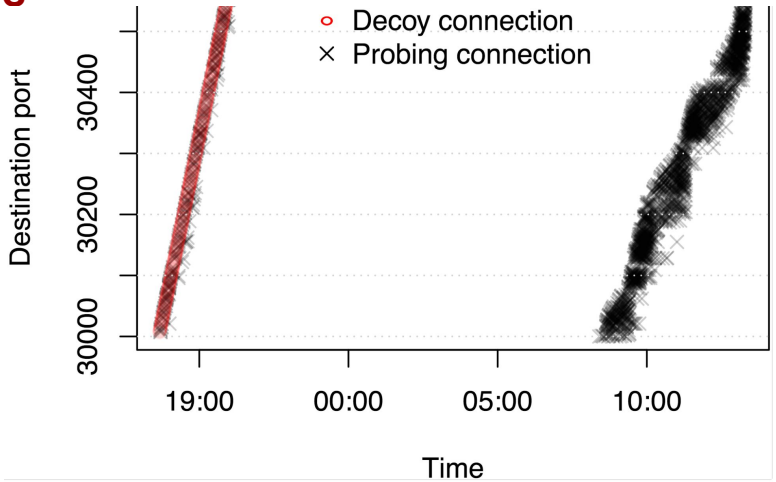
- Sybil dataset shows that system now works in **real time**
  - Median delay between Tor connection and subsequent probing connection is **~500ms**
  - **1,182** distinct probes just in 22 hs



How Quickly do Active Probes Show Up?

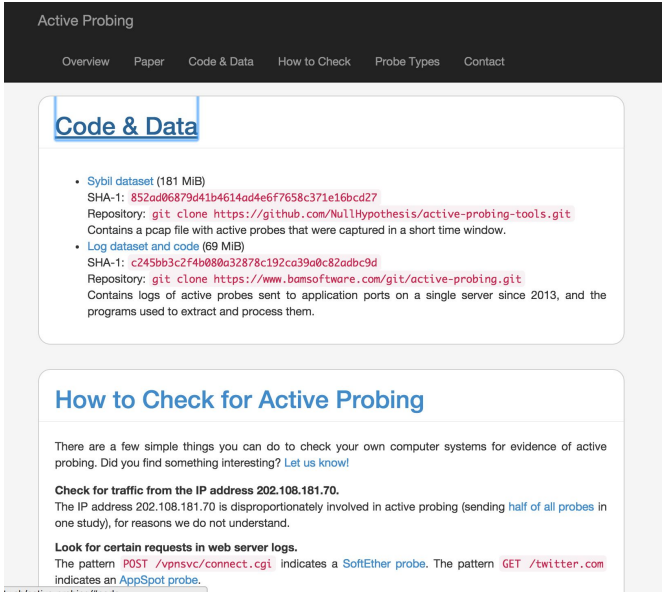


- Sybil dataset shows that system now works in **real time**
  - Median delay between Tor connection and subsequent probing connection is **~500ms**
  - **1,182** distinct probes just in 22 hs



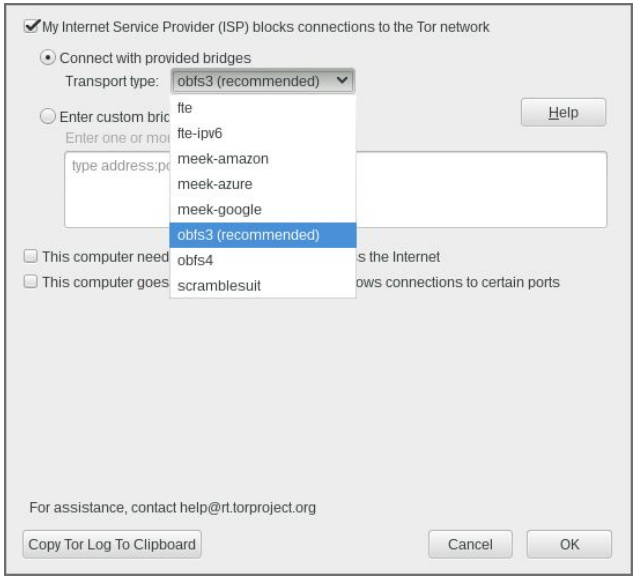
What Other Protocols are Probed?

- **SSH**: In 2011, not anymore?
- **VPN**: OpenVPN occasionally, SoftEther
- **AppSpot**: To find GoAgent?
- Anything else?



Pluggable Transports that Work in China

- **ScrambleSuit and Obfs4**
  - Clients must prove knowledge of **shared secret**
- **meek**
  - Tunnels traffic over **CDNs** (Amazon, Azure, Google)
- **FTE**
  - Shapes ciphertext based on **regular expressions**
- More is in the making!
  - WebRTC-based transport



Summary of Key Findings

- GFW “**hijacks**” many IP addresses, controls them centrally
  - Important for future threat models
- Censorship system is **modular**
  - Tor’s **pluggable transports** caused GFW to build **pluggable censorship**
- System now works in **real-time**

Trolling the GFW: Block List Exhaustion

```
for ip_addr in “$ip_addrs”; do
  for port in $(seq 1 65535); do
    timeout 5 tor --usebridges 1 --bridge “$ip_addr:$port”
  done
done
```

One /24 network can add 16 million blocklist entries

Lessons Learned (1/2)

- GFW **not perfect**... and **does not have** to be!
  - Good enough if 99% of users are affected
  - 1% will **always** be able to circumvent anyway

- **Information asymmetry** makes our job hard
  - Circumvention code and designs often **fully transparent**
  - GFW, however, **fully opaque**

Lessons Learned (2/2)

- Biggest **challenges** often **not technical**
  - Usability of Tor Browser
  - How do we **broadcast** what pluggable transports work?

- We don’t always get our threat models right
  - Our adversary isn’t **omniscient computer scientist**
  - Our adversary is **underpaid** and just wants to **get job done**

- Project page: https://www.cs.princeton.edu/~rensafi/projects/active-probing/
- **Log** and **Sybil** data sets are available online
- Contact: rensafi@cs.princeton.edu
- Twitter: @_ _royaen_ _