Presentation and notes on - How China detects and Blocks ShadowSocks

Presentation and notes on - "How China Detects and Blocks Shadowsocks" by Alice, Bob, Carol, Jan Beznazwy, and Amir Houmansadr, presented at ACM Internet Measurement Conference 2020.

0. Motivation

What is the Great Firewall Of China, why censorship circumvention tools are needed?

The Great Firewall of China is the country's mechanism to control and censor the internet within its borders. It blocks access to selected foreign websites, filters out certain keywords, and monitors individuals' online activity. Key features include:

1. Blocking: The Great Firewall blocks access to many foreign websites, including widely used platforms such as Google, Facebook, Twitter, and Wikipedia. Chinese citizens are directed towards local alternatives that comply with government regulations and censorship.

2. Keyword Filtering: The system scans for sensitive keywords in online text – if it detects anything related to politically sensitive topics, it may block the content.

3. Deep Packet Inspection (DPI): This technology allows the system to intercept and inspect data packets being sent over the internet. If the packets contain information flagged as sensitive or inappropriate, they are blocked.

4. Self-Censorship: Chinese internet companies are legally responsible for the content on their platforms. This forces them to self-censor and prevent any content that could be perceived as challenging the government from being published.

5. Real-Name Registration Laws: These laws require users to identify themselves to service providers, including internet service providers and social media platforms, discouraging anonymous online expression.

Why it's important to study the Great Firewall and it's techniques, and try to be one step ahead of the game?

Law Enforcement: Individuals found disseminating blocked content, or using tools to circumvent the Great Firewall, can face legal repercussions. This suppresses freedom of expression and fuels a climate of fear.

- circumvention tools/ what's so special about shadowsocks/ diff b/w VPN

1.Introduction

Shadowsocks is a protocol for Internet censorship circumvention, especially popular in China. According to a research survey in July 2015, of 371 faculty members and students from Tsinghua University, 21% used Shadowsocks to bypass censorship in China.

Since as early as October 2017, users in China have reported their Shadowsocks servers becoming unreliable or being blocked by the Great Firewall (GFW), especially during politically sensitive times.

2. Background on shadow Socks

2.1 Shadowsocks: A protocol for Internet censorship circumvention

About

Encrypted proxy protocol which works by encrypting traffic so it appears as uniform byte stream

Uses: 1. stream ciphers 2. AEAD ciphers

Shadowsocks is a secure split proxy loosely based on SOCKS5.

```
client <---> ss-local <--[encrypted-rand-looking]--> ss-remote <---> target
```

The Shadowsocks local component (ss-local) acts like a traditional SOCKS5 server and provides proxy service to clients. It encrypts and forwards data streams and packets from the client to the Shadowsocks remote component (ss-remote), which decrypts and forwards to the target. Replies from target are similarly encrypted and relayed by ss-remote back to ss-local, which decrypts and eventually returns to the original client.

Implementations:

1. shadowsocks-libenv: https://github.com/shadowsocks/shadowsocks-libev/blob/master/README.md

2. outlineVPN:

Popularity in China:

1. light weight design

2. easy implementation on various platforms

3. one click installation scripts

VPN vs Shadowsocks

VPN, shadowsocks, proxy tunnels, tor are tools for censorship circumvention and can be used as and when required.

Major differences b/w VPN and shadowshocks.

VPN- 1. Operates at a network level.

2. All traffic is routed through the VPN

Shadowsocks: 1. Operate at the application level

2. You can choose which application's traffic to send over the tunnel

How GFW Detects Shadowsocks:

1. Passive Traffic Analysis

Based on **size** and **entropy** of the first data packet in each connection the GFW takes a list of suspected targets

2. Active Probing: Active probing to confirm suspected targets

The probes are partial replays of past legitimate connections, and random probes of varies length

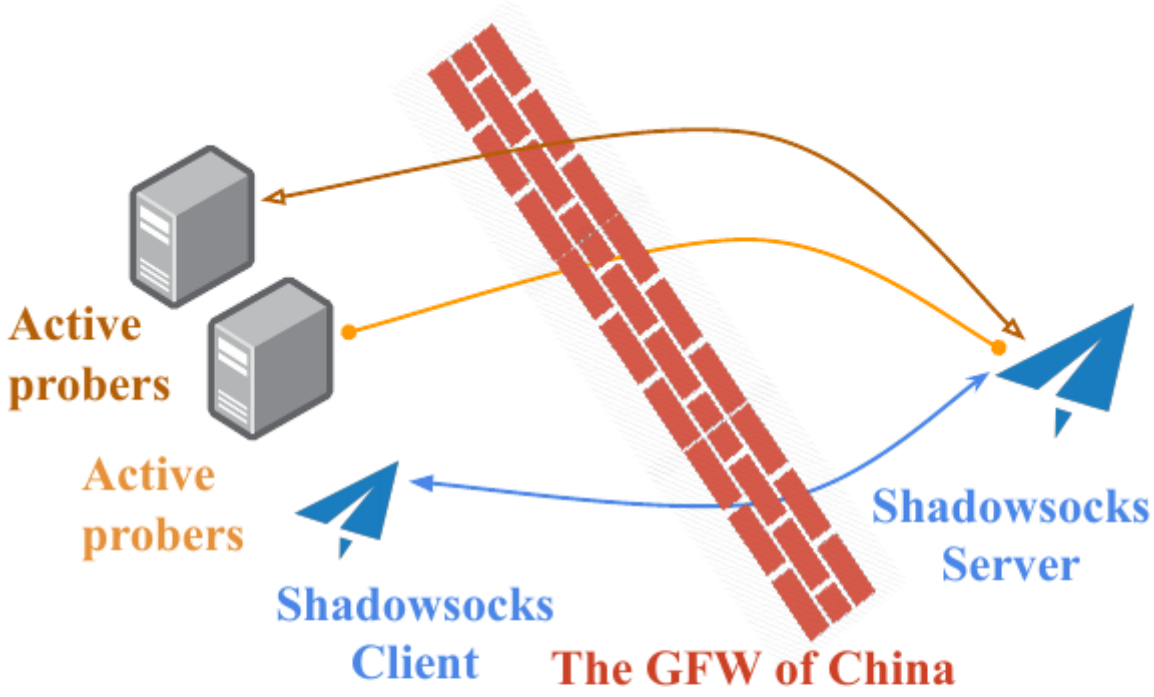


Figure 1: How active probing works. A genuine Shadowsocks client connects to a Shadowsocks server; Once the GFW passively determines that the connection *may* be Shadowsocks, it directs its active probers to confirm this guess.

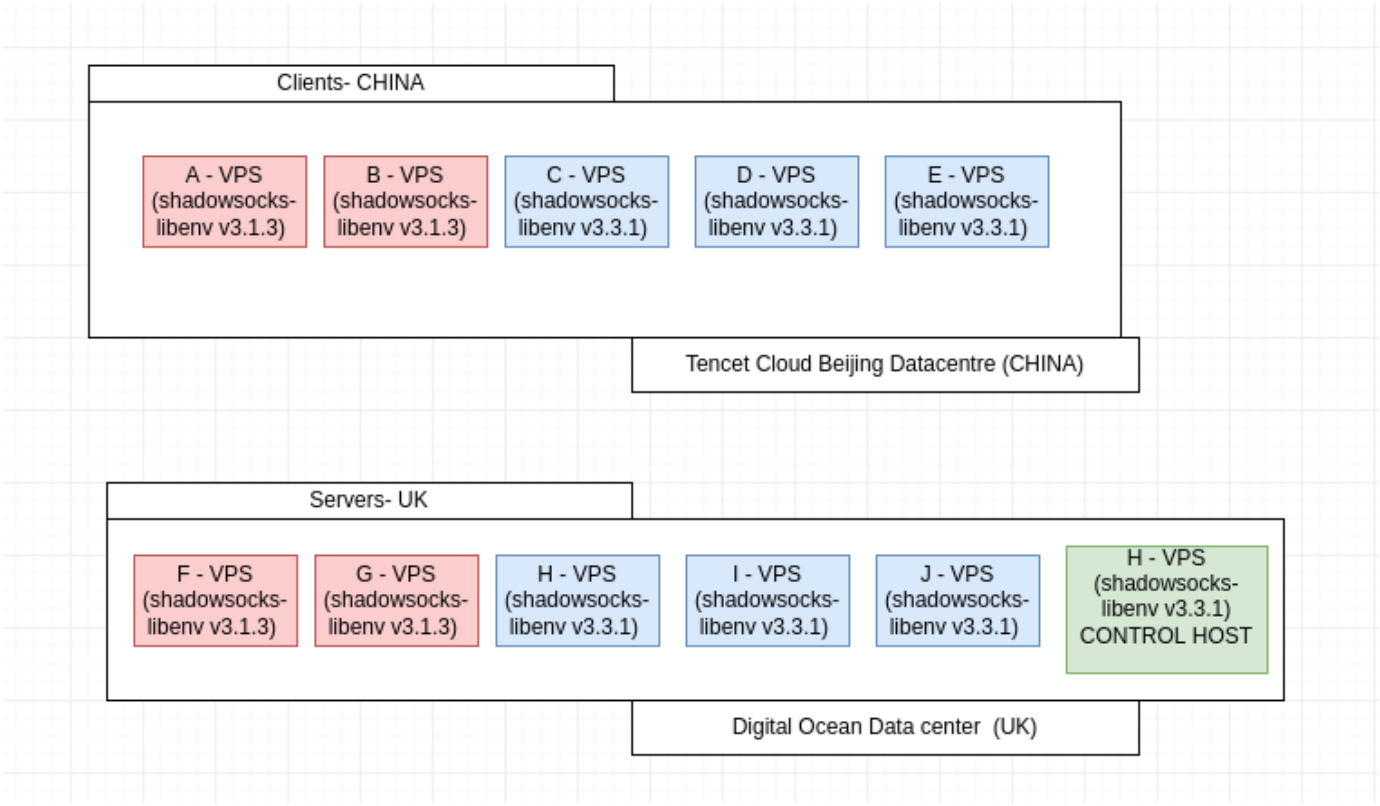
3. Probes

3. 1 Shadowsocks server experiment

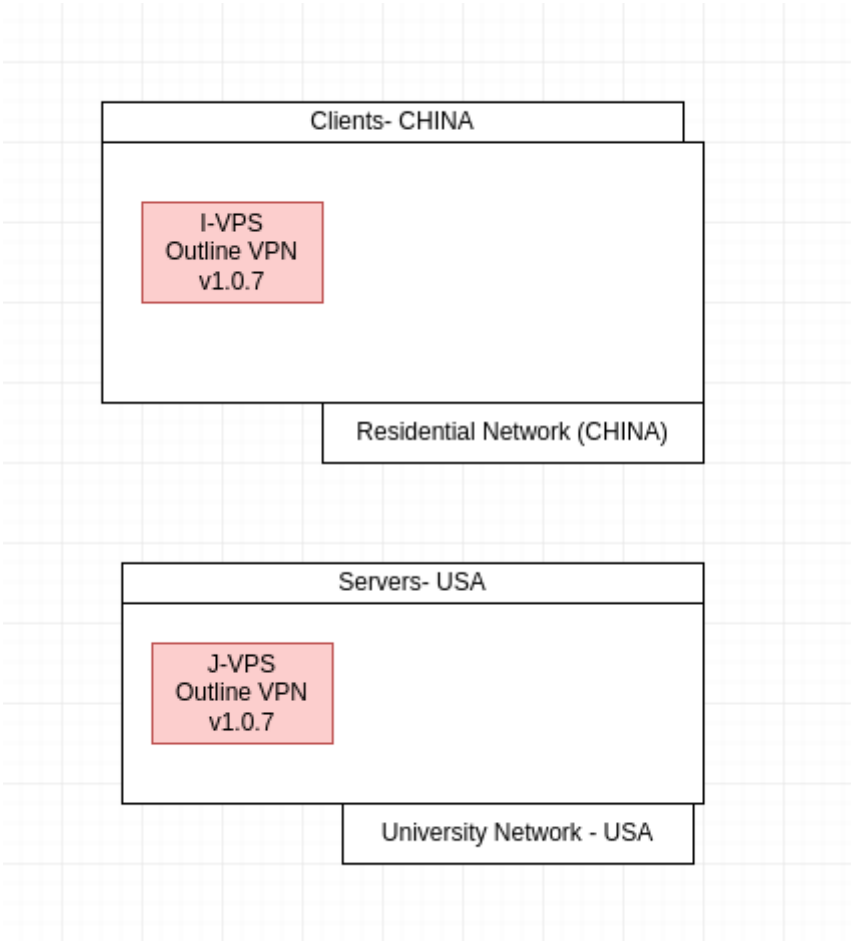
The paper identifies and fingerprints different types of active probes and infers the probable intention behind them.

The experiment is setup as follows:

Shadowsocks-libev. We installed Shadowsocks-libev clients on five VPSes in a Tencent Cloud Beijing datacenter, and Shadowsocks-libev servers on five VPSes in a Digital Ocean UK datacenter. Each client was configured to connect to only one of the servers. Two pairs of the clients and servers used v3.1.3 of Shadowsocks-libev, and the other three pairs used v3.3.1. As a control, we set up an additional VPS within the same UK datacenter and never connected to it, only capturing all incoming traffic. We generated client traffic using curl. Through the Shadowsocks proxy, we constantly fetched one of the websites at a given frequency: **https://www.wikipedia.org, http://example.com, and https://gfw.report.**



OutlineVPN. We installed an OutlineVPN v1.0.7 server in a US university network. The OutlineVPN client we used was the latest as of October 2019. The client was in a residential network in China. **Client traffic was provided by an instance of Firefox, configured to automatically browse a subset of the Alexa top 1 million sites that is censored in China.**



They analyse all connections to the server port running Shadowsocks, and use all the traffic received by the control host to verify that the probes observed were indeed triggered by by their own connections and not by radiant background "internet scans"

3.2 Types of Probes

We observed a total of 51,837 active probes across all experiments

1. Replay Based Probes.

payload derived from the first data-carrying packet of some previously recorded legitimate connection. classified into: **R1, R2, R3, R4, R5 ie GFW may record the first data carrying packet of a genuine connection and replay it later**

Type R1 Identical replay.

Type R2 Replay with byte 0 changed.

Type R3 Replay with bytes 0–7 and 62–63 changed.

Type R4 Replay with byte 16 changed.

Type R5 Replay with bytes 6 and 16 changed.

2. Seemingly Random

these have varying length and contents do NOT resemble prior connections

Type NR1 Probes of length 7–9, 11–13, 15–17, 21–23, 32–24, 40–42, or 48–50 bytes.

Type NR2 Probes of length exactly 221 bytes.

- Probe types R3, R4, and R5 were received only in the OutlineVPN experiment, not in the Shadowsocks-libev one. Only two type R5 probes were received in our experiments.

3.3 Origin Of the Probers

3.3.1 IP addresses

- A simple defence against active probing is to discover IP addressed and ban them.

- However, this is not effective as GFW probes from a large and diverse pool of IP addresses, with high churn .

Observations:

1. The 51,837 active probes were sent from 12,300 unique source IP addressed, all located in China

2. Figure 3 shows the distribution of the number of probes sent per unique IP address. The most common prober IP addresses are summarized Table 2.

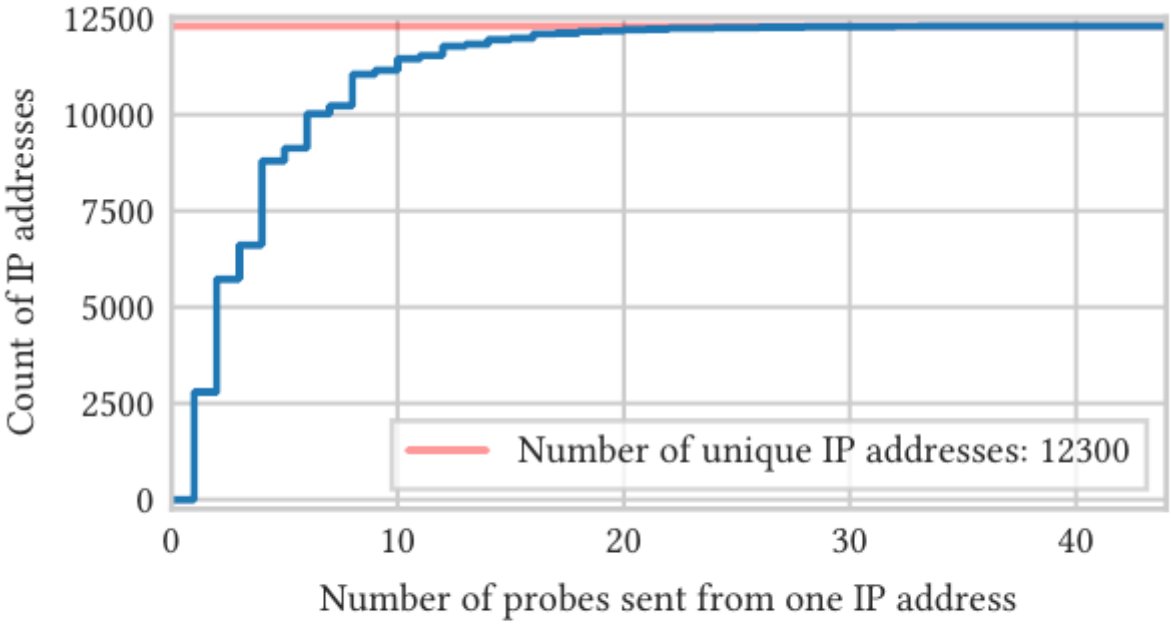


Figure 3: Cumulative number of probes per prober IP address.

Table 2: The most common prober IP addresses and their number of occurrences.

Prober IP address	Count
175.42.1.21	44
223.166.74.207	38
124.235.138.113	36
113.128.105.20	36
221.213.75.88	33
112.80.138.231	32
116.252.2.39	32
124.235.138.231	32
221.213.75.126	32
223.166.74.110	31

3. We compared our list of prober IP addresses against 934 that were observed to send active probes to Tor servers in 2018 by Dunna et al. [13], and 22,000 that were observed to send various types of

active probes between 2010 and 2015 by Ensafi et al. [14]. Figure 4 shows that three sets overlap only slightly. We note the IP address 202.108.181.70, which was responsible for an inordinate number of probes in previous work [14, §5.3], does not appear in our data. The small overlap is not unexpected, given that past work has observed high churn in prober IP addresses.

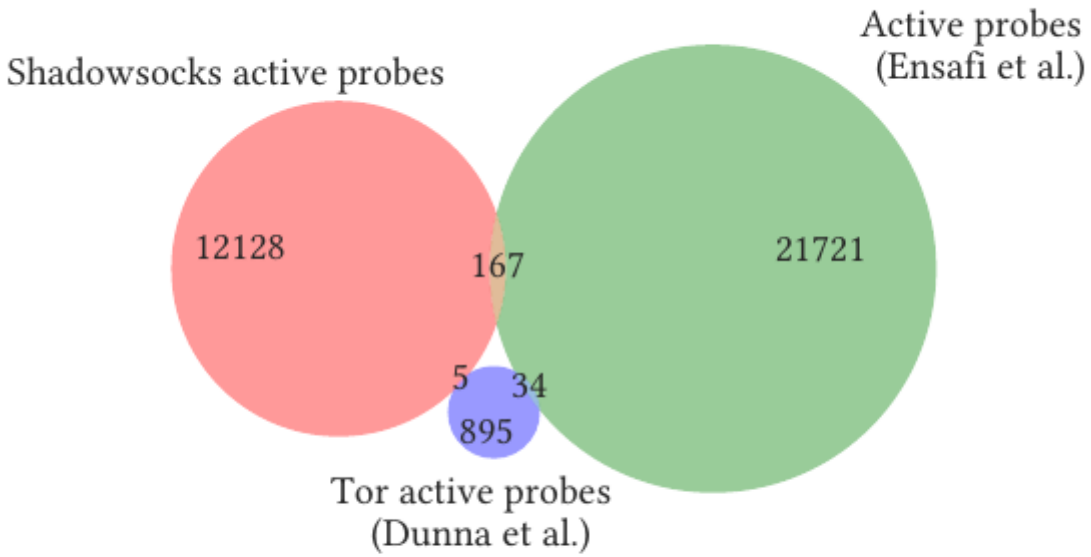


Figure 4: Overlap in prober source IP addresses across independently collected datasets.

**3.3.2 Autonomous Systems: **

- The autonomous system (AS) distribution of probers is shown in Table 3.

- The two ASes that account for the most Shadowsocks probes are

1. AS4837 (CHINA169-BACK-BONECNCGROUP China169 Backbone)

2. AS4134 (CHINANET-BACK-BONE No.31, Jin-rong Street).

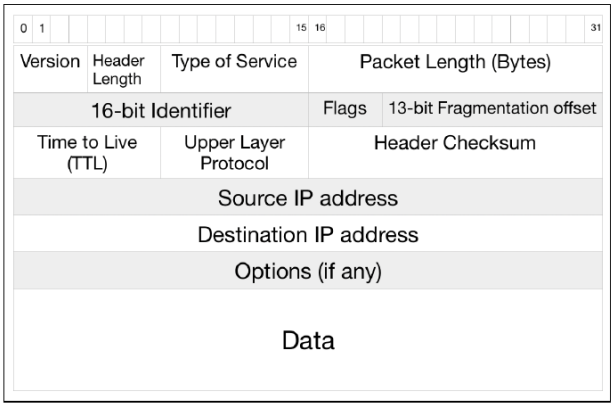
- These two were the most common in previous work [14, 56] as well.

3.3 Fingerprinting the probes

Fingerprint packet level features of active probes.

IP Layer fields: **1. ID 2.TTL**

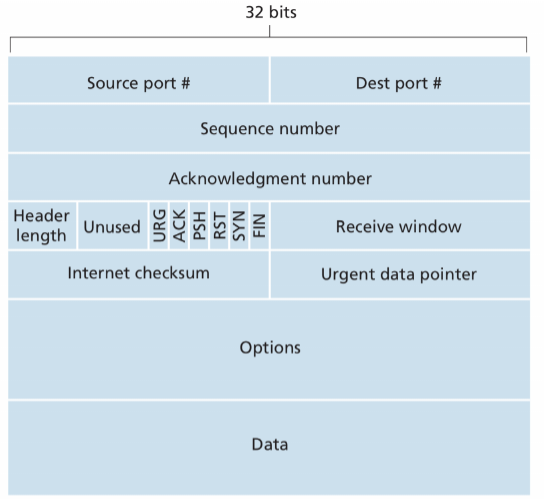
- IPv4 Packet Format



- **Version:** These 4 bits specify the IP protocol version of the packet
- **Header Length:** Since header length is variable for IP packets, these 4 bits are needed to determine where in the IP packets the data actually begins.
  - Measured in words (32-bits)
  - 20 bytes $\leq$ Header Length $\leq$ 60 bytes
- **Type of Service:** Used to distinguish IP packets based on service
- **Packet Length:** It measures the total size of the packet - header + data.
  - Measured in bytes.
  - Packet length $\leq$ 65535 bytes
  - Though, they are rarely larger than 1500 bytes.

- **Identifier, flags, fragmentation offset:** These three fields have to do with so-called IP fragmentation. An in depth discussion to follow shortly.
- **Time-to-live**
  - The TTL field is included to ensure that ip packets do not circulate forever in the network.
  - This field is decremented by 1 each time the datagram is processed by a router!
  - If the TTL field reaches 0, the ip packet must be dropped.
- **Protocol** field is used only when an ip packet reaches its final destination.
  - The value of this field indicates the specific upper layer protocol to which the data portion of this ip packet should be passed.

TCP Layer fields: **1. Source Posts 2. Timestamps(TSval)**



- Sequence and Acknowledgement Number fields are used by TCP sender and receiver to implement reliable data transfer protocol
- Receive Window is used for flow control
- Header Length The 4-bit field specifies the length of the TCP header in 32-bit words.

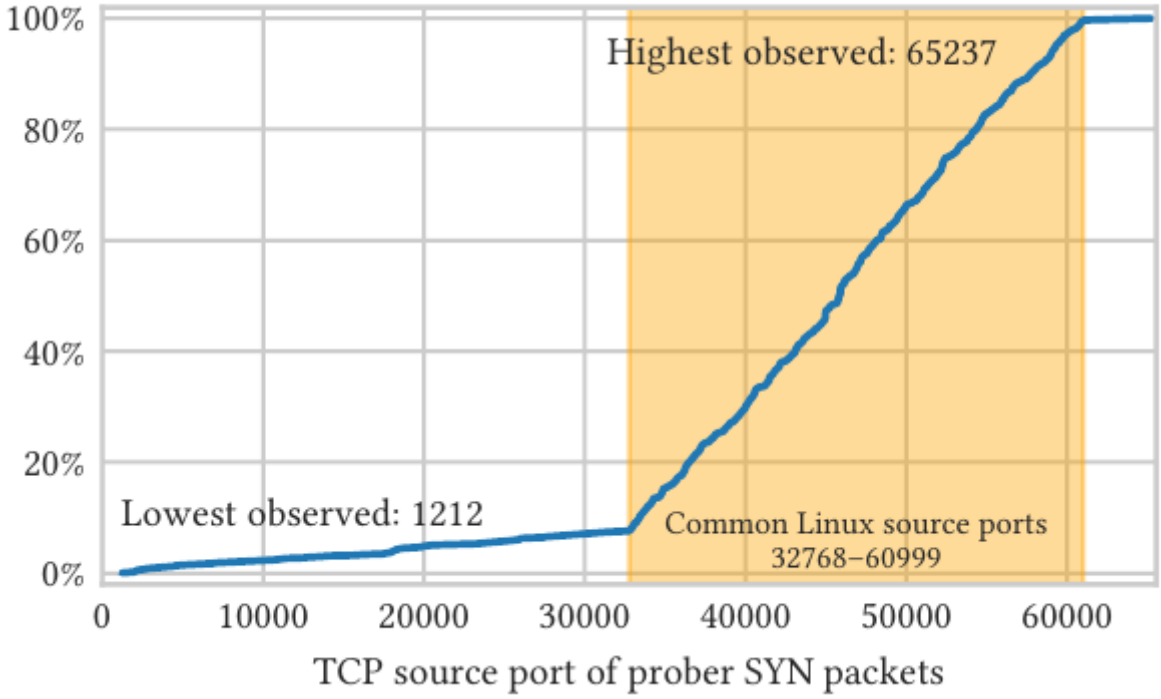
- Flag bits
  - URG set to "1" indicates presence of some urgent data. The location is given by 16-bit urgent data pointer field.
  - ACK - used for reliable data transfer protocols.
  - PSH set to "1" indicates that the receiver should pass the data to the upper layer immediately
  - RST, SYN, and FIN are used for TCP connection establishment and tear down

---

1. IP ID and TTL:

We fingerprint the IP ID and TTL of PSH/ACK packets sent by the probers. As in Ensafi et al. [14, §5.5], we find no clear pattern in the IP ID sequences, and that TTLs remain within the range 46–50.

2. TCP source ports. Around 90% of probes came from source ports in the range **32768–60999**. This range, highlighted in Figure 5, happens to be the default source port range of many Linux kernels. No probes came from ports below 1024



**3. tcp Timestamp(TSval): **

The TCP timestamp is a 32-bit counter that increases at a fixed rate, attached to every non-RST TCP segment [7, §3]. It is not an absolute timestamp, but is relative to how and when the counter was initialized, and its rate of increase varies across operating systems. Figure 6 shows the timestamp value attached to the SYN segment of each probe. The figure shows that **although the probers use thousands of source IP addresses, they cannot be fully independent, because they share a small number of TCP timestamp sequences.** In this case, there are at least seven different physical systems or processes, with one of the seven accounting for the great majority of probes.

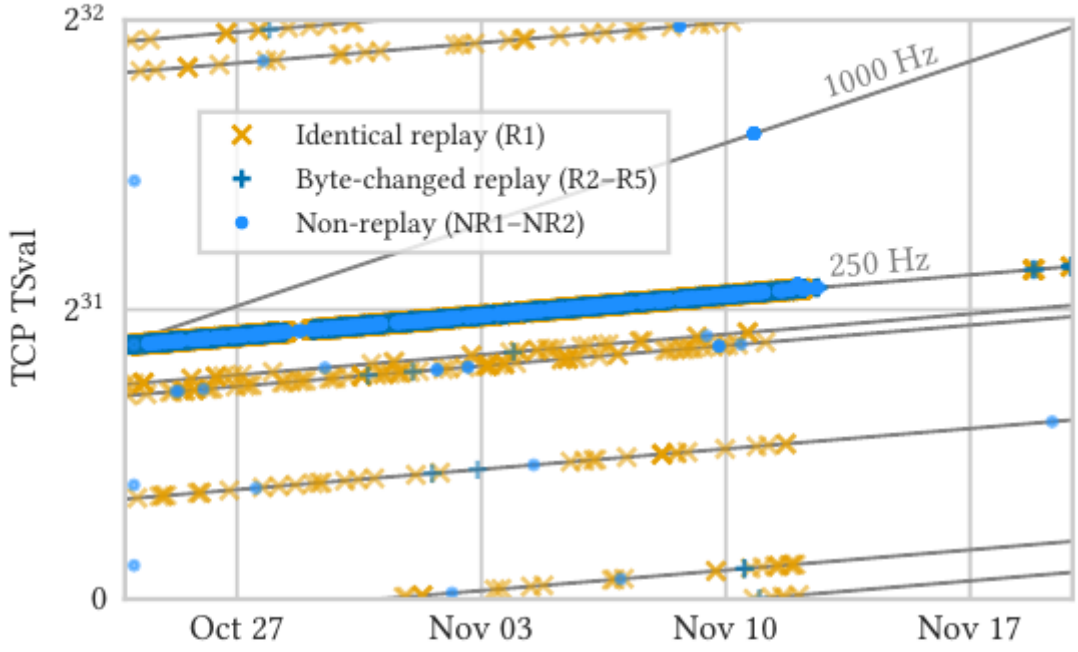


Figure 6: Non-independent processes revealed by common TCP timestamp sequences. The labeled marker lines have slopes of precisely 250 Hz and 1000 Hz. The small cluster of 22 non-replay probes on the 1000 Hz line locally have a slope of 1009 Hz, but here the measurement is less certain because they span only about 3.5 s. The 1000 Hz line does not become 250 Hz, even if connected to one of the sparse non-replay data points at the left edge of the figure.

3.5 Delay of Replay Attacks

The GFW may record the first data-carrying packet of a genuine client connection and replay it later, possibly with modifications, as an active probe. Figure 7 shows the variability in delay between when a legitimate connection is made and when the GFW sends replay-based probes derived from that connection

- More than 20% of first replays arrived within one second; more than 50% within one minute; and more than 75% within 15 minutes.

- The shortest delay we observed was 0.28 seconds and the longest was 570 hours.

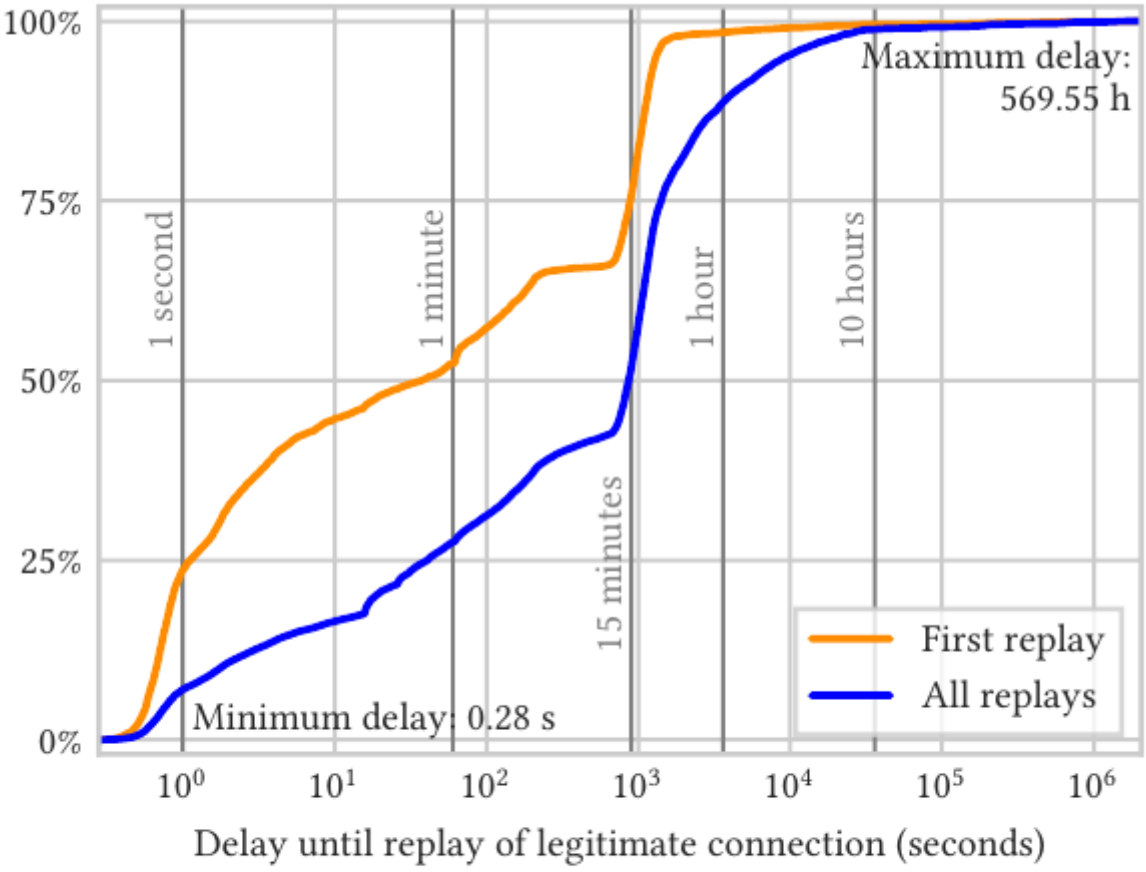


Figure 7: CDF of the delay of replay-based probes. Note the logarithmic x-axis.

Now that we have observed the probes, their patterns, we can begin to guess what exactly triggers active probing

4. What triggers active Probing?

2 types of probing: 1. Large scale proactive port scanning 2. reactive probing triggered by legitimate connections.

The fact that the unusued control host in the previous section did not receive any active probes leads us to discard the proactive scaning hypothesis. Instead we can safely assume that the probes are sent only when the probing system sees a suspected Shadowsocks connection.

What, then, constitutes a suspected Shadowsocks connection, from the GFW's point of view? In this section, we deal with the following questions:

• How much traffic is required to trigger active probes?

• Why were type R3, type R4 and type R5 probes sent only to the OutlineVPN server, not the Shadowsocks-libev server?

• Does the GFW consider the length of packets?

• Does the GFW consider the entropy of packet payloads?

• Do outside-to-inside connections (with the client outside China and the server inside) result in as much active probing as inside-to-outside connections?

4.1 Experiments

A convincing way to show what features the GFW uses for traffic analysis is to outline a minimal, reproducible set of conditions that trigger active probing. We are aided by two observations.

First, the byte streams sent between Shadowsocks clients and servers are, by design, indistinguishable from random. This means that it may not be necessary to use a real client Shadowsocks implementation; we may be able trigger active probes by sending random data.

Second, as described in Section 3.5, replay probes may be sent as soon as 0.28 seconds after a legitimate data packet. The GFW could have seen only the very beginning of a client-to-server flow, before deciding that the traffic was suspicious.

EXPERIMENT: Guided by these two observations, we implemented a TCP client that connects to a TCP server and sends one data packet, with a specified length and Shannon entropy. We implemented a server with two operating modes: sink mode and responding mode. In sink mode, the server accepts TCP connections, but does not respond with any data, and closes connections after 30 seconds. In responding mode, the server responds to probers—but not our own clients—with between 1 and 1000 bytes of random data.

Table 4 summarizes the design of the random-data experiments.

Table 4: Summary of random-data experiments. $[x, y]$ means the value is uniformly and randomly sampled from a range, independently for each connection. In Exp 1, the server was switched from sink mode to responding mode after 310 hours; we label the two subexperiments 1.a and 1.b.

	Client		Server
Exp #	Length (bytes)	Entropy	Mode
1.a	$[1, 1000]$	> 7	sink
1.b	$[1, 1000]$	> 7	responding
2	$[1, 1000]$	< 2	sink
3	$[1, 2000]$	$[0, 8]$	sink

Table 1 shows the time span of the experiment. Clients ran on different VPSes within the same Tencent datacenter in Beijing. All servers ran in the same Digital Ocean datacenter in the US. Client and server IP addresses were not reused across experiments.

4.2 Experiment Results and Analysis

1. ****Little traffic is required to trigger active probes. Our sink server, despite not being a real Shadowsocks server and never sending data, received many of the same types of probes as in the Shadowsocks server experiment of Section 3.1. After a TCP handshake, a single data packet from client to server suffices to trigger active probes.

2 .**Only certain lengths are replayed.** Although our clients sent data packets with lengths of between 1 and 2000 bytes, virtually all probes that were determined to be replays had a payload length of between 160 and 700 bytes, with the maximum length being 999 bytes. Figure 8 shows the distribution of probe lengths in Exp 1.a. The distribution of lengths exhibits a stair-step pattern, reflecting the fact that certain lengths are more likely to be replayed.

Namely, the lengths of replay probes tend to have certain remainders when divided by 16. Considering type R1 probes (type R2 is similar), of the 376 probes whose length is in the interval 168–263 bytes, 72% have a length whose remainder when divided by 16 is 9; of 1,558 in the interval 384–687, 96% have a length whose remainder is 2; and of 749 in the middle interval 264–383, there is a mix of remainders 9 (37%) and 2 (32%).

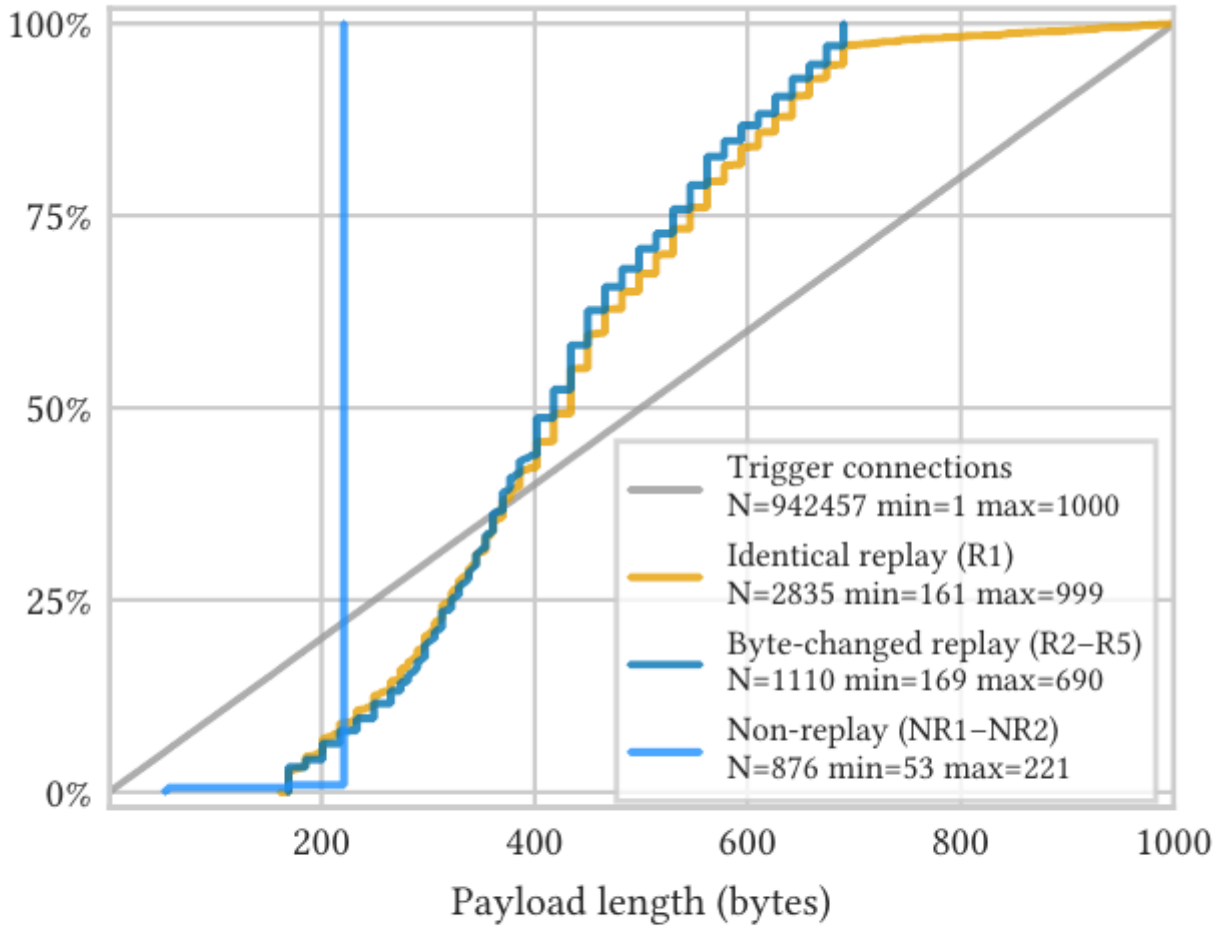


Figure 8: CDF of the payload lengths of replay-based probes over the 310 hours of Exp 1.a. The lengths of replay probes exhibit a stair-step pattern.

conc:

The results suggest that the GFW considers packet lengths in classifying Shadowsocks traffic. Packet length is a reasonable feature to use, because Shadowsocks does not pad the contents of the tunnel, only incidentally changing the underlying packet length distribution by adding an address header prefix (see Section 2) and, with AEAD ciphers, length prefixes and tags. The payload length distribution of the Shadowsocks traffic therefore resembles that of the underlying traffic, which is often HTTP or TLS.

3. High-entropy packets are more likely to be replayed. Two pieces of evidence support this conclusion:

First, Figure 9 shows that while packets of all entropies may be replayed, one with a high per-byte entropy of 7.2 is almost four times as likely to be replayed as one with a low entropy of 3.0.

Second, Exp 1.a and Exp 2 differ only in the entropy of packets, and over the same period of time, the server in Exp 1.a received significantly more probes than the one in Exp 2.

4. Probes of type R3 and R4 are not sent unless the server has pre**viously responded to probes of type R1 and R2.** The thousands of probes received in Exp 1.a, Exp 2, and Exp 3 could all be classified as type R1, R2, or NR2. In other words, we were not able to trigger probes of types R3, R4, R5 or NR1 in these experiments.

This result reminded us of the fact that in the experiment of Section 3.1, type R3, R4 and R5 probes were only ever received by OutlineVPN servers, and not by Shadowsocks-libev servers. As will be expanded on in Section 5.3, one major difference between Shadowsocks-libev and the version of OutlineVPN we used is that Shadowsocks-libev has a filter to defend against replay attacks, and OutlineVPN does not. (At least in the version we use OutlineVPN has since added replay protection [26].) For this reason, Shadowsocks-libev servers does not respond to exact replays of earlier connections, while OutlineVPN servers do.

We therefore hypothesize that the GFW does not send probes of type R3, R4, and R5 unless the server has already responded to probes of type R1 and R2. We switched the server in Exp 1.a to responding mode after 310 hours of operating in sink mode. Soon after the server started responding to type R1 and type R2 probes, it began to receive a large number of type R3 and type R4 probes. The server continued to receive type R1 and R2 probes as well.

conc: These results suggest that the active probing system operates in stages. It does not move on to the next stage until a certain condition is observed. This implementation detail suggests that the censor may have designed its active probing system with not only Shadowsocks in mind. Other, similarly behaving protocols may also be targeted.\

5. New probe types observed. The sink/responding servers received probes that did not match the probe types seen in our earlier ex- periment with Shadowsocks-libev and OutlineVPN. In Exp 1.b, we saw 11 replay-based probes that had bytes from 16 to 32 changed. We additionally saw many non-replay probes across all four experiments. In total, there were 9 probes of 53 bytes, 5 probes of 56 bytes, 3 probes of 169 bytes, 1 probe of 180 bytes, and 1 probe of 402 bytes.

6. The GFW does not distinguish traffic directionality. We set up a Shadowsocks server inside China and made connections to it from outside. The traffic proxied was generated by automatically browsing a subset of Alexa top 1 million websites. The server received a large amount of active probing. This result indicates that the GFW probes suspected servers regardless of whether the server is inside or outside China. This bidirectional triggering behavior differs from Winter and Lindskog's [56, §4.4] observation that outside-to-inside Tor connections did not trigger active probing. On the other hand, the GFW is known not to distinguish traffic directionality for many protocols, including DNS [1, §2], HTTP [11, §3] and TLS [9, §3.1]. The GFW's sensitivity to directionality has even been known to change over time, as in the case of TLS ESNI blocking, which was bidirectional for two weeks before becoming unidirectional [6].

5. INTENTION BEHIND THE PROBES

We conjecture that if the probes elicit reactions from a Shadowsocks server that differ from the reactions of non-Shadowsocks servers, the GFW can be confident in classifying the server as Shadowsocks.

We developed our own prober simulator to observe how Shadowsocks servers react to probes like those sent by the GFW. We further checked the source code of Shadowsocks implementations to understand their internal logic. **Based on this analysis, we formed conjectures regarding what distinguishable server reactions may be exploited for classification.**

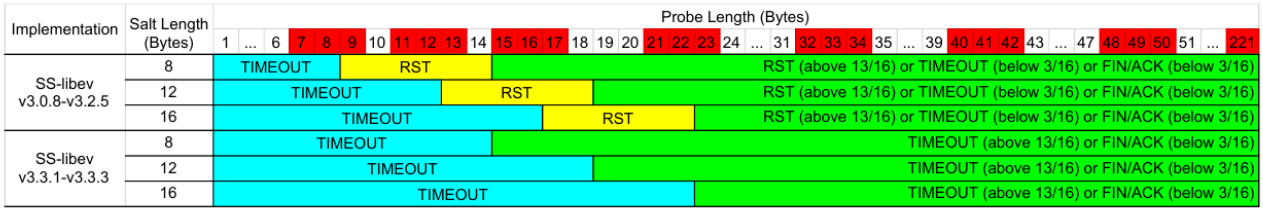
5.1 PROBER SIMULATOR EXPERIMENT

Replay-based probes. To simulate replay-based probes, the simulator records the first data-carrying packet in a connection between a Shadowsocks client and server, then sends the data to the server in a separate connection. To send byte-changed probes, the simulator randomly changes certain bytes of the payload to different values.

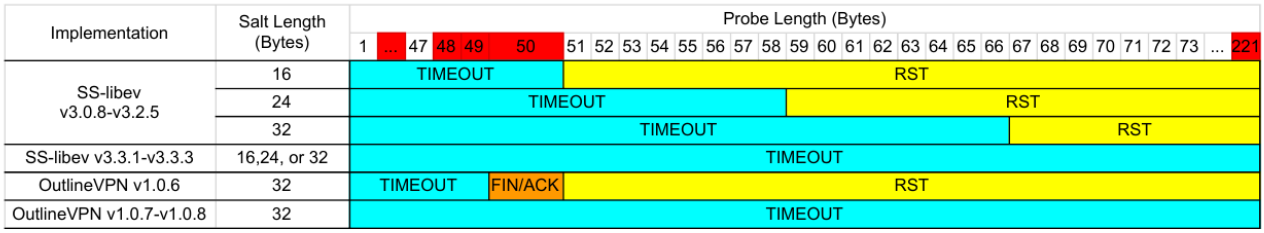
Non-replay probes. To simulate non-replay probes, the simulator simply sends a specific number of random bytes. The justification here is that the servers' reactions to the GFW's non-replay probes are no different from their reaction to random probes

Choice of servers. We chose a set of Shadowsocks implementations that has significant coverage over the Shadowsocks circumvention ecosystem. Specifically, we tested the Shadowsocks implementations that met any of the following conditions: 1) is available in a repository of a major Linux distribution; 2) is available in the pip repository; 3) is the latest version; 4) is widely used by any popular one-click script; 5) has a recent fix to any distinguishable reactions as the result of a preliminary report on these attacks; or 6) was recommended to us by developers. Using this selection process, we chose Shadowsocks-libev (v3.0.8, v3.1.3, v3.2.5,v3.3.1, and v3.3.3) and OutlineVPN (v1.0.6, v1.0.7, and v1.0.8).

When you send random unauthenticated data to a server, the sever reacts differently depending on how much data you send. If you send too little data, the serve is gonna wait for the rest of the data and eventually timeout. If you send beyond that threshold the server is gonna attemp to authenticate the data you sent, be unable to authenticate and close the connection



(a) Stream ciphers



(b) AEAD ciphers

Figure 10: Reactions of Shadowsocks servers to synthetic random probes with different lengths. Figure 10a is for servers using the stream ciphers construction and Figure 10b is for the AEAD ciphers construction. Payload lengths that the GFW has been observed to send are marked in red. "TIMEOUT" means the server waits until the prober or itself times out. "RST" means the server sends an immediate TCP RST. "FIN/ACK" means the server will be the first to send a FIN/ACK to close the connection.

5.2.2 How an attacker/GFW might use the information from randomprobes . An attacker can identify a Shadowsocks server with high confidence using statistical analysis of its reactions to random probes.

Specifically, an attacker can send a set of probes with varying lengths to the server and record its reactions. After collecting enough reactions, the attacker may perform a statistical analysis. A server whose reactions fit into one row of Figure 10 is then very likely to be a Shadowsocks server. The fact that the GFW needs only a single probe to detect and block a Tor server [56], but a set of several probes before blocking a Shadowsocks server, suggests that the GFW does some statistical analysis of this kind to detect Shadowsocks.

Intention behind the probes/what information gets eaked:

1. protocol being used

2. shadowsocks version and implementation - from reaction of the server if there is no defence against replay

3. length of IV- cipher suite

7 CIRCUMVENTION

The detection of Shadowsocks happens in two stages: 1) passive identification of suspected Shadowsocks connections, then 2) active probing of the server. Therefore, to avoid blocking, one can 1) evade the passive detector, or 2) respond to active probes in a way that does not result in blocking

7.1 DEFENCE AGAINST TRAFFIC ANALYSIS

Brdgrd [54] (bridge guard) is software that can be run on a Shadowsocks server that causes the client to break its Shadowsocks handshake into several smaller packets. Brdgrd was originally intended to disrupt the detection of Tor bridges by forcing the GFW to do complicated TCP reassembly [56], but here we take advantage of its ability to shape client packet sizes.

As a test, we set up a Shadowsocks server and let a Shadowsocks client make 16 connections to it every 5 minutes. We enabled and disabled brdgrd at random times, and measured the rate of active probing under both conditions. Table 1 summarizes the time span of the experiment.

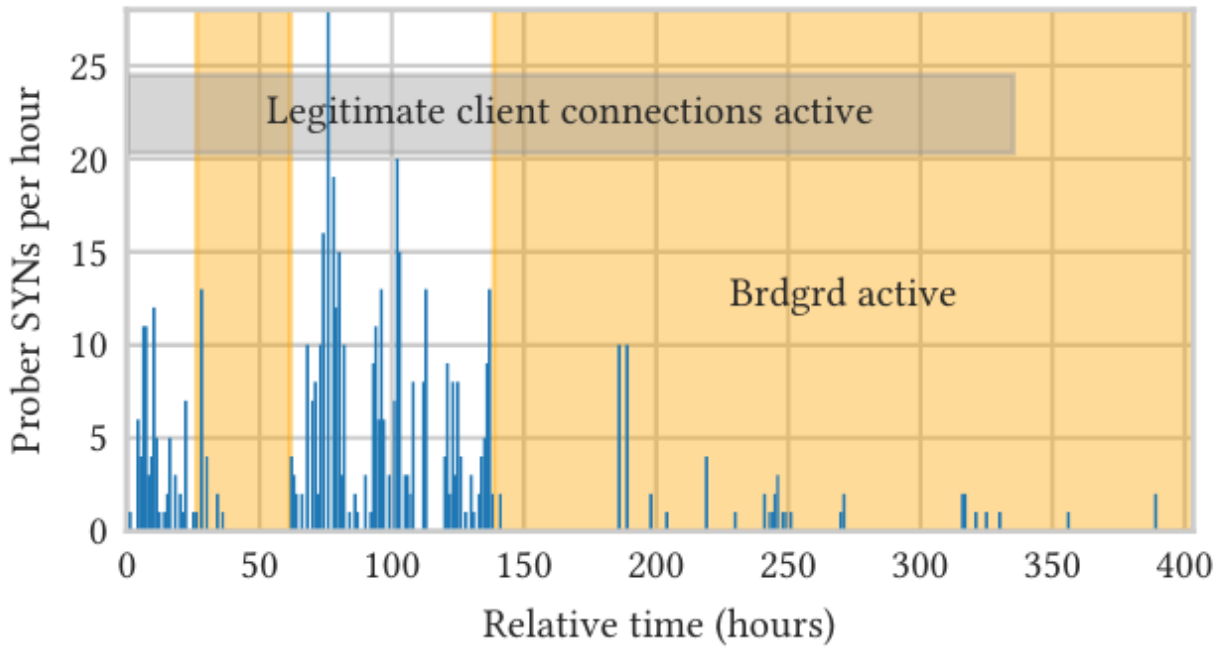


Figure 11: The intensity of active probing diminishes when brdgrd is active.

7.2 DEFENSE AGAINST ACTIVE PROBING

1. PROPER AUTHENTICATION

2. REPLAY FILTERING

3. BEING CONSISTENT WITH SERVER'S REACTIONS

Some useful links:

For network servers and TCP/IP implementations:

- http://www.kegel.com/c10k.html

- https://lwn.net/

- https://github.com/net4people/bbs

References