The use of TLS in Censorship Circumvention
INTRODUCTION

Motivations

Internet Censorship is extremely widespread

- Only 20% of the world has unrestricted access to the internet [2]
- Approximately 1.72 billion people are affected by internet censorship every day [2]

Censorship Circumvention tools

- meek, Psiphon, Signal, VPNs
- But, they all have the same problem of blending in with legitimate traffic [1]
INTRODUCTION

Background Knowledge

Internet Censorship

- Many techniques including
  - IP whitelisting/blacklisting
  - Packet Inspection
  - Fingerprinting

- This paper focuses on fingerprinting.

Censorship Circumvention Tools

- Different techniques for evading censorship
  - Web based proxies
  - Encrypted proxies (e.g., Psiphon)
  - Onion routing (Tor)

- Involves routing traffic through an intermediary system.
- All ultimately hide the true destination of traffic from the censor.
Transport Layer Security (TLS)

- Most popular protocol on the internet, account for 70% of FireFox page loads. [1]
- Provides security for the transport layer (authentication, confidentiality, integrity). [1]
- Commonly used by censorship circumvention tools for masking the true intent of their traffic. [1]

TLS Handshake

- Occurs after TCP connection is established.
- Authenticates parties, establishes keys, ciphers and connection parameters. [1]
- Initiated with a clear-text Client Hello with a range of parameters including:
  - Supported features and parameters
  - Supported versions of TLS
  - Supported cipher suites and compression methods
  - Random nonce
  - List of optional extensions [1]
INTRODUCTION

Problem

Fingerprinting

- Using TLS is insufficient to evade censors since the Client Hello is clear text.
- Parameters of the TLS Client Hello are used by censors to generate a fingerprint of the traffic, allowing censors to differentiate censorship circumvention tools. [1]
- If censors can differentiate the traffic of censorship circumvention tools from legitimate traffic, then it can (and will) block it.

TLS Implementation Mimicking

- Tools can mimic commonly used TLS implementations in order to blend in and increase collateral damage. [1]
- But, this is a difficult task, the developer of the tool needs to:
  - Find an implementation to mimic. [1]
  - Keep up with that implementations updates. [1]
  - Ideally support the extensions, ciphers, and compression methods of that implementation. [1]
Mitigating of Difficulties

- Studied popular implementations and developed a tool to automatically mimic of popular TLS implementations.
- Ensures tools fingerprints are common and up to date, increasing collateral damage if censors use blacklisting.

Fingerprint Study

- 11 billion fingerprints collected. [1]
- 152,000 unique Client Hello fingerprints. [1]
- Top 3000 fingerprints make up 99.9% of connections. [1]
- >50% of connections produce one of top 12 Client Hello fingerprints. [1]
- 50% of all connections would be blocked after 6-months with non-adaptive whitelist. [1]
SOLUTION

uTLS

Automatic Mimicry
- Provides automatically generated code for a preset collection of popular implementations.
- Facilitates automatic updating of censorship circumvention tools through scripts.
- Uses data from study to generate implementations matching the fingerprints of popular implementations.
- Evades whitelist censors.

Fingerprint Randomisation
- Provides implementations to generate randomised fingerprints.
- Ensures randomised fingerprint uses only common combinations of ciphers, compressions, features.
- Evades blacklist censors.

Implementation Sets
- Cycles through a preset collection of TLS implementations until finding a successful one.
- Allows tools to work even when a subset of fingerprints are blocked or unsupported.
CRITICISM

Limited Study

Limitations

- TLS fingerprints collected from a single US university network in an uncensored region.
- Censored regions unlikely to have the same internet usage and traffic.
- Possible limited effectiveness in censored regions.

Suggested Improvements

- Most accurate data would come from the internet traffic of censored regions.
- But, this is unlikely to be possible due to government regulations.
- Data could still be improved by collecting TLS fingerprints from multiple regions and network environments.
CRITICISM

Dependency on Continuous Collection

Limitations

- In order to keep up with rapidly changing TLS implementations, the data collection must continue indefinitely.
- This is costly and poses concerns around network privacy at the university.

Suggested Improvements

- Any approach involving mimicking of popular TLS implementations would require continuous TLS fingerprint collection.
- Randomisation techniques require a much smaller maintenance overhead, only requiring intermittent verification.
THANK YOU

Questions?
REFERENCES


