Chasing Shadows:

A security analysis of the ShadowTLS proxy

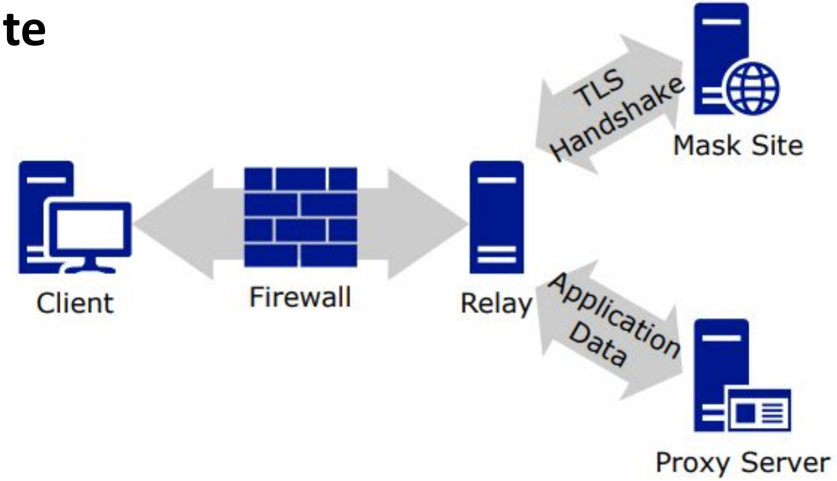
Gaukas Wang, Anonymous, Jackson Sippe, Hai Chi, Eric Wustrow

Presentation for FOCI'23

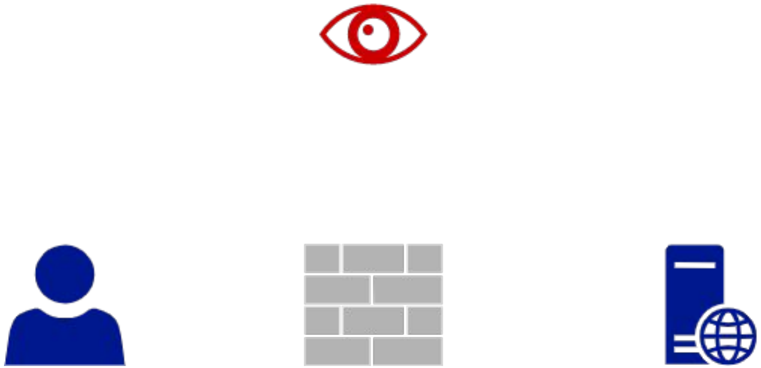
ShadowTLS

- Performs TLS handshake with a **real site**

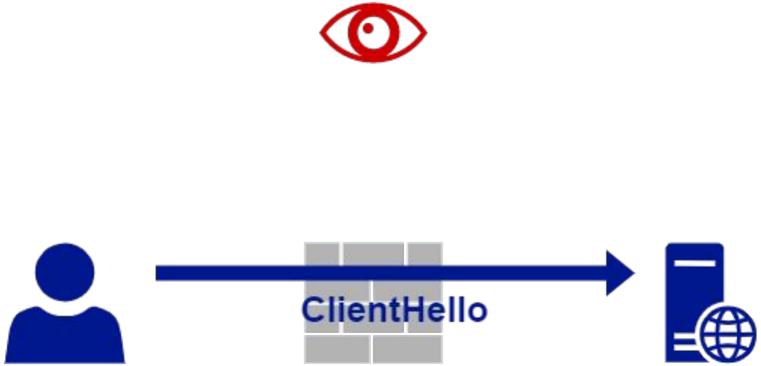
- Evades SNI/certificate blocking



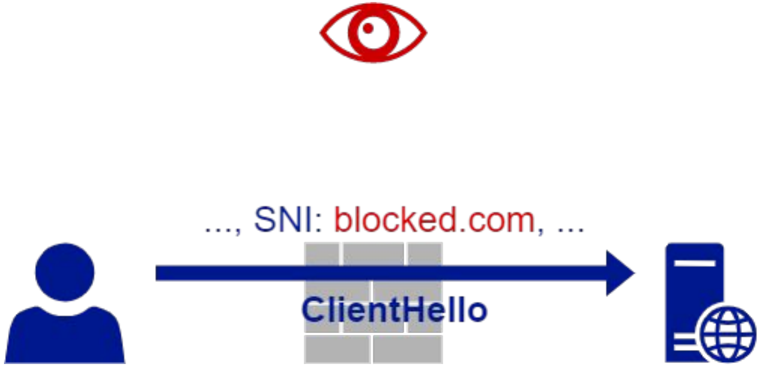
TLS Censorship



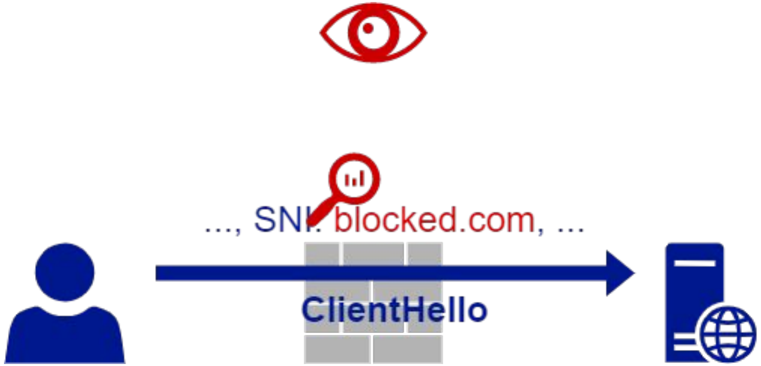
TLS Censorship



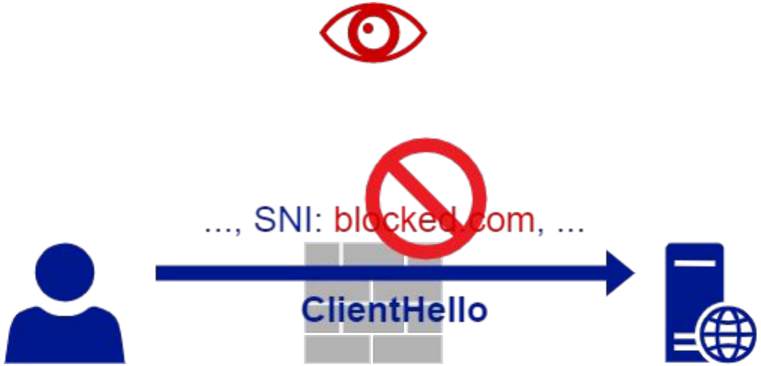
TLS Censorship



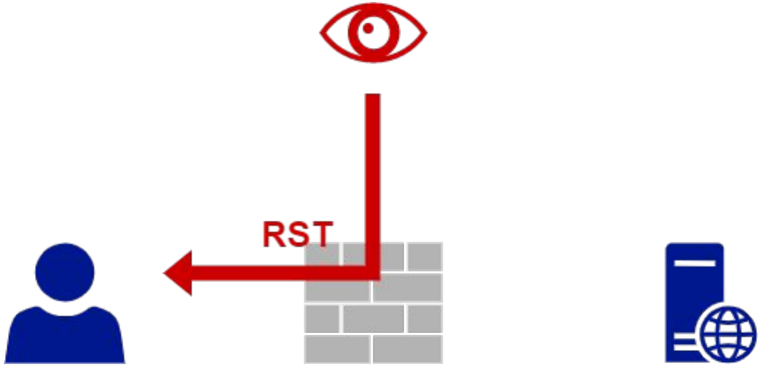
TLS Censorship



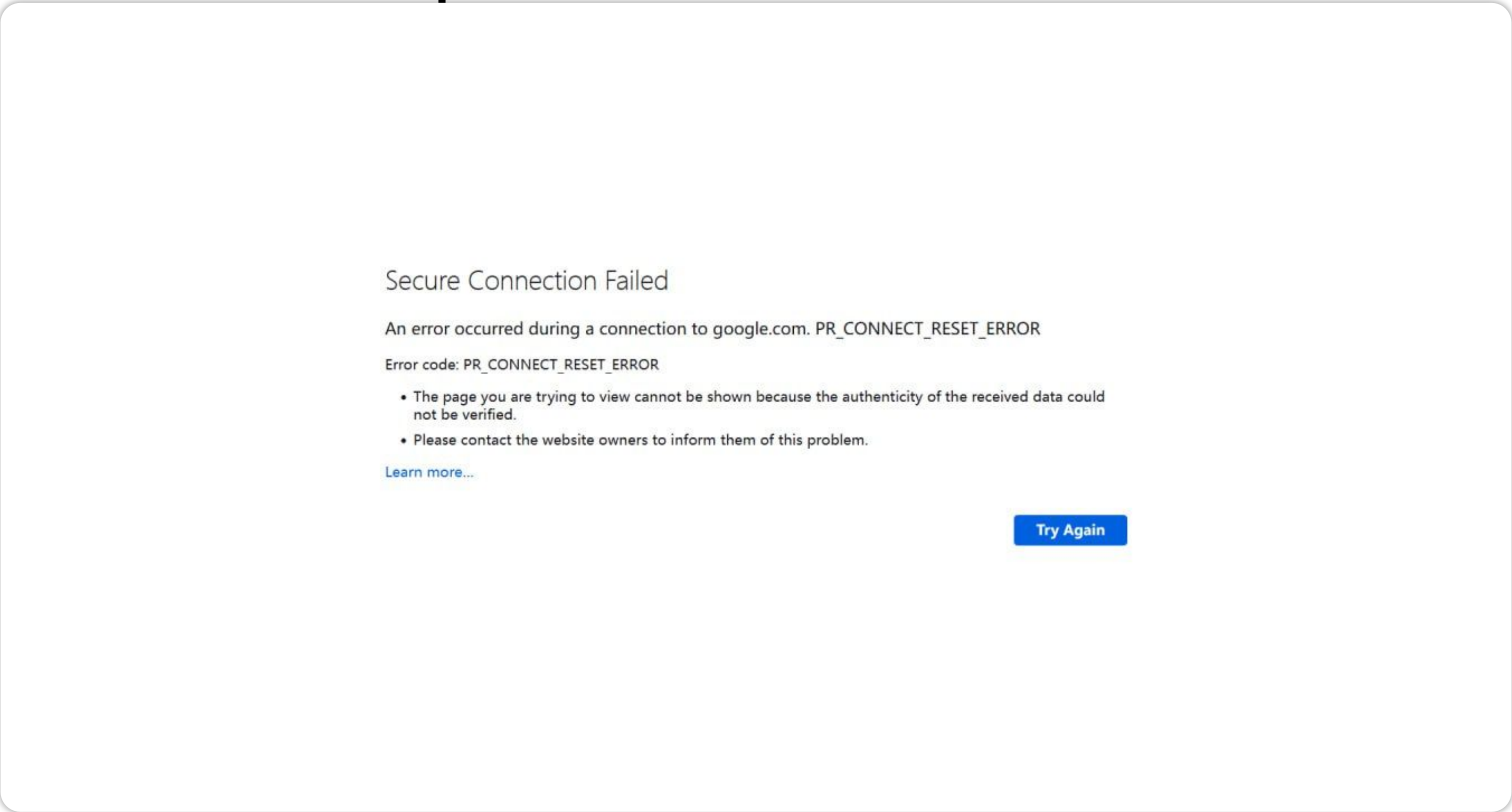
TLS Censorship



TLS Censorship



TLS Censorship



Secure Connection Failed

An error occurred during a connection to google.com. PR_CONNECT_RESET_ERROR

Error code: PR_CONNECT_RESET_ERROR

- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

TLS Censorship

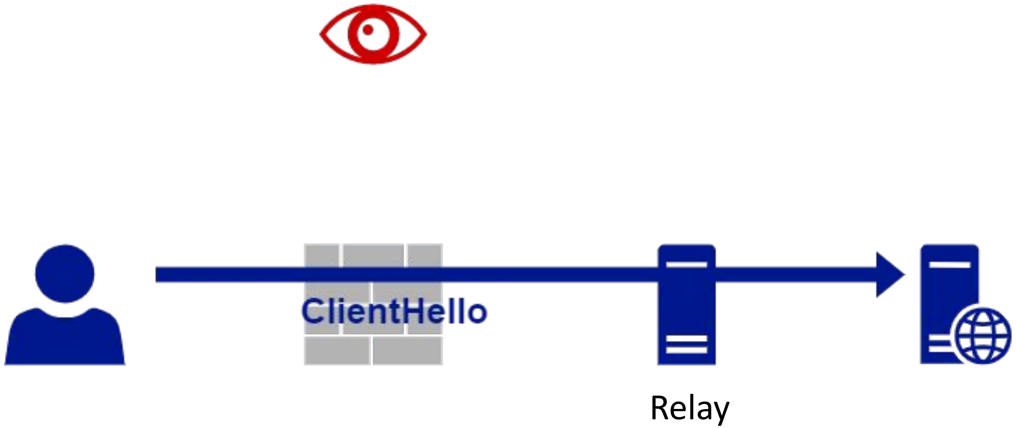
- TLS handshake reveals critical information

- Server Name Indication (SNI)
  - Included in ClientHello
  - Sent in cleartext

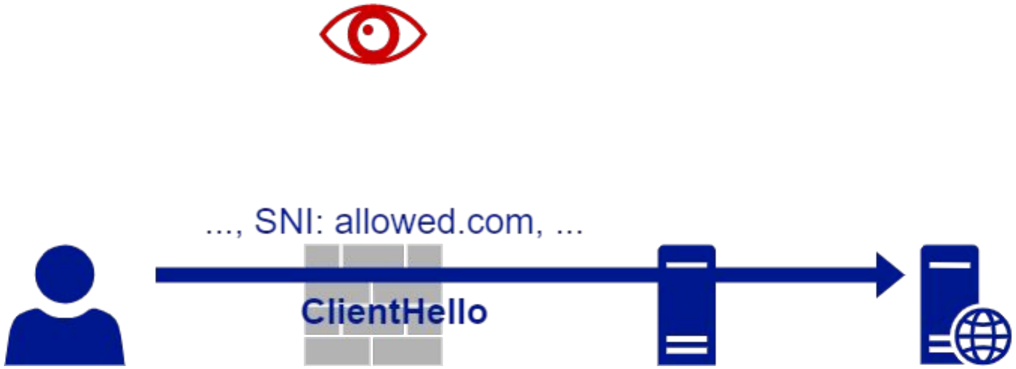
- TLS (Server) Certificate
  - Signed by a CA for a specific party (domain, organization, company, etc.)
  - Used in Public Key Infrastructure(PKI) to establish encrypted connections

- Allowlist enforced in Quanzhou(Ch'üan-chou), Fujian Province, China

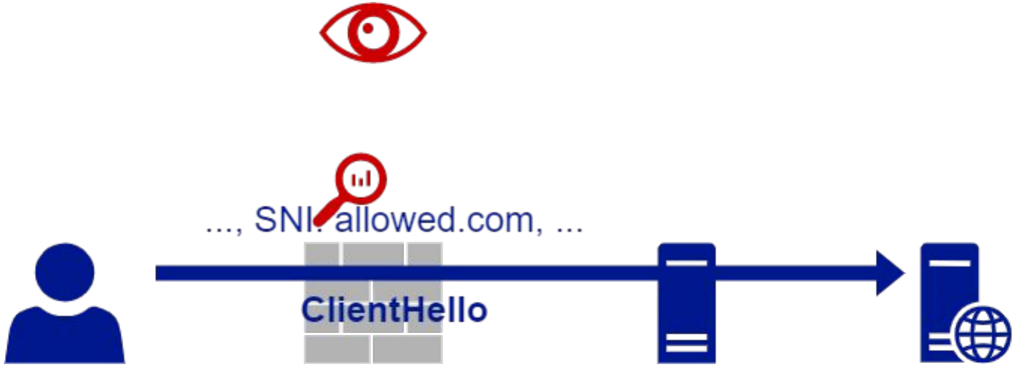
ShadowTLS: Steps



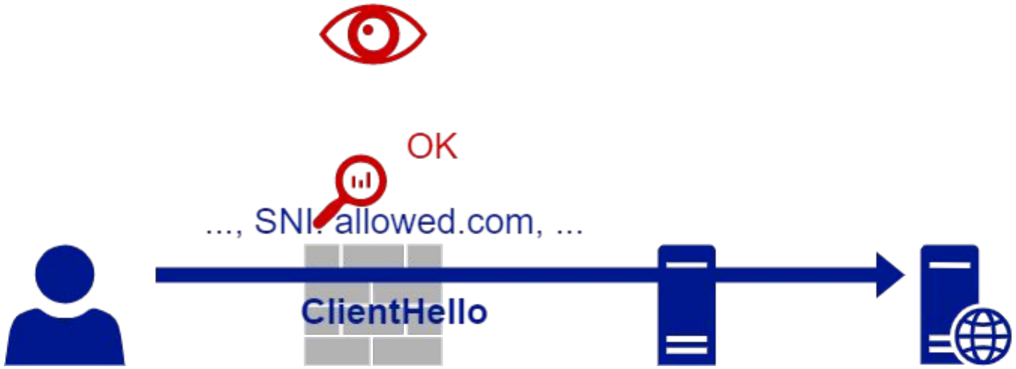
ShadowTLS: Steps



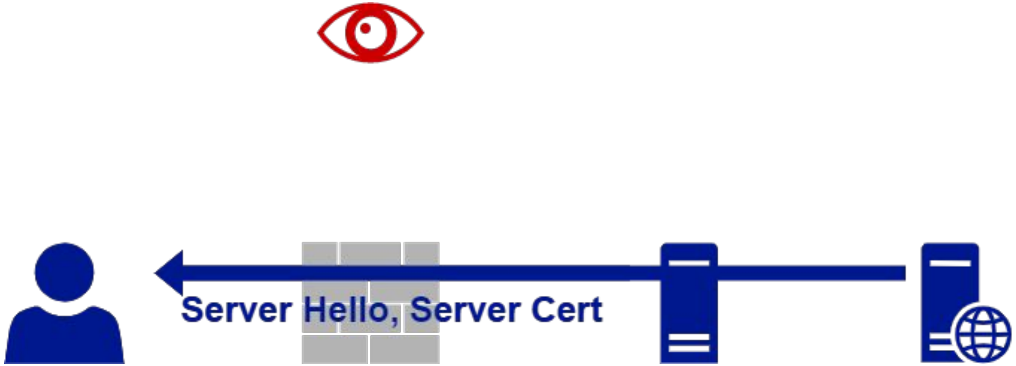
ShadowTLS: Steps



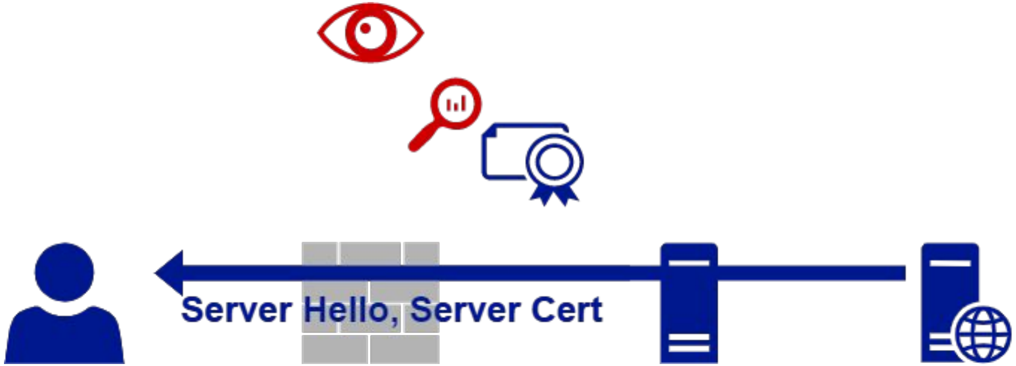
ShadowTLS: Steps



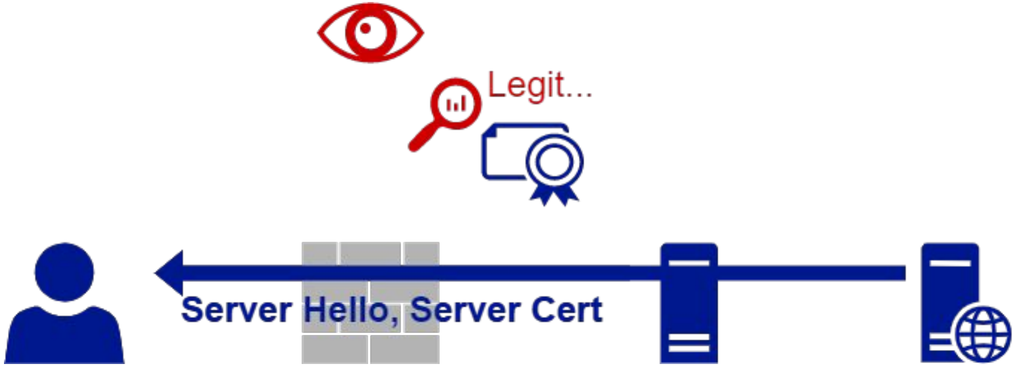
ShadowTLS: Steps



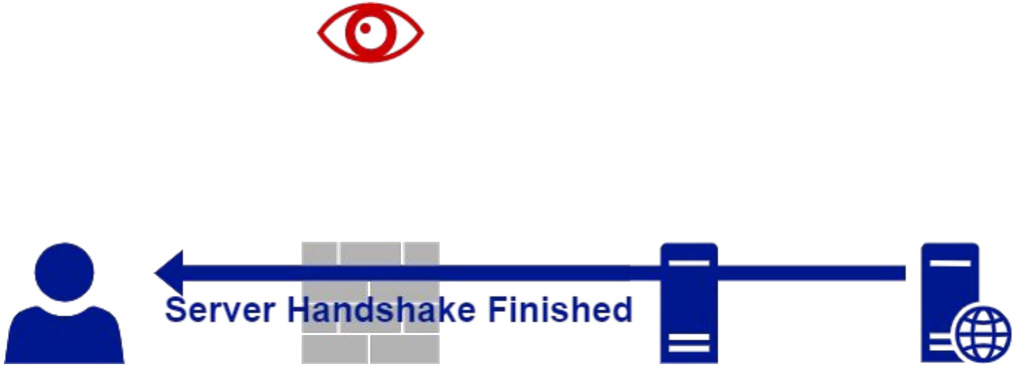
ShadowTLS: Steps



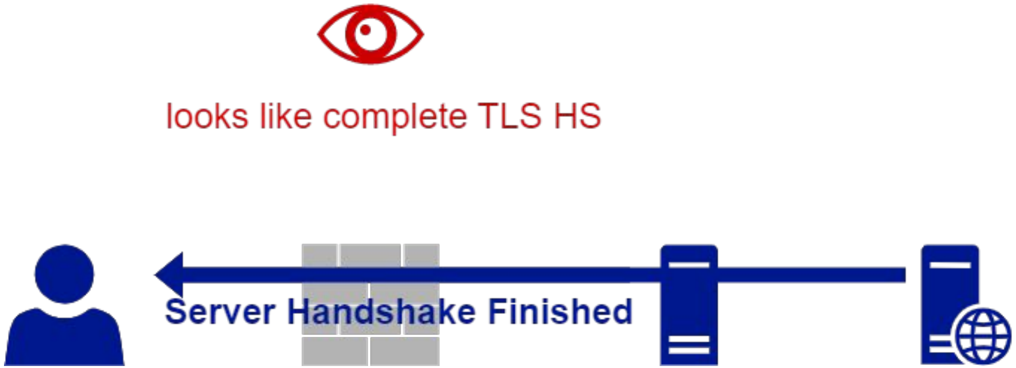
ShadowTLS: Steps



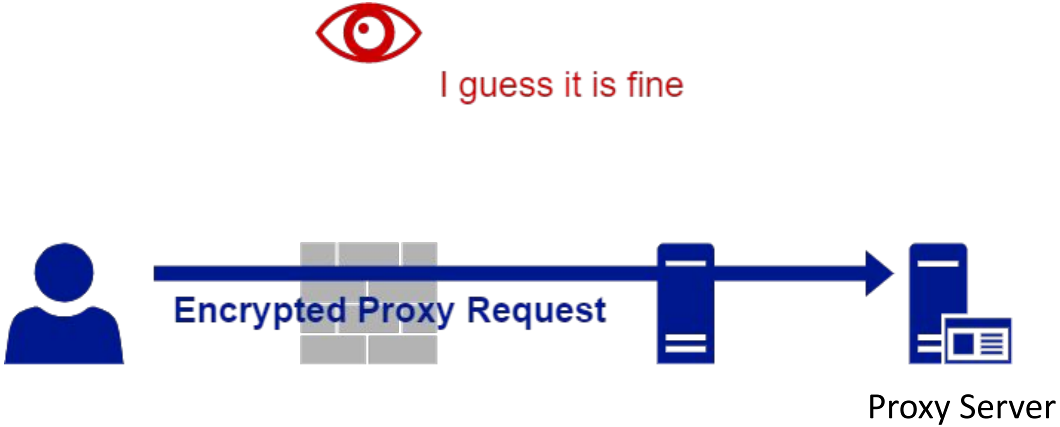
ShadowTLS: Steps



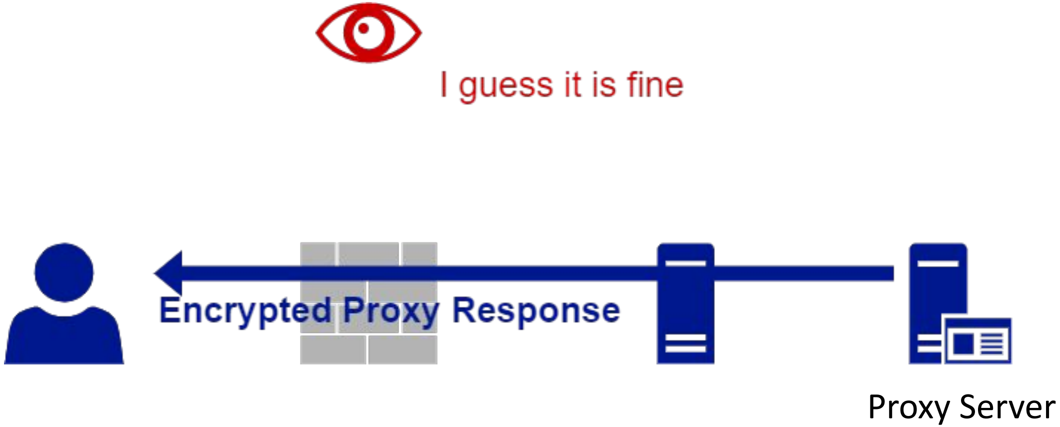
ShadowTLS: Steps



ShadowTLS: Steps

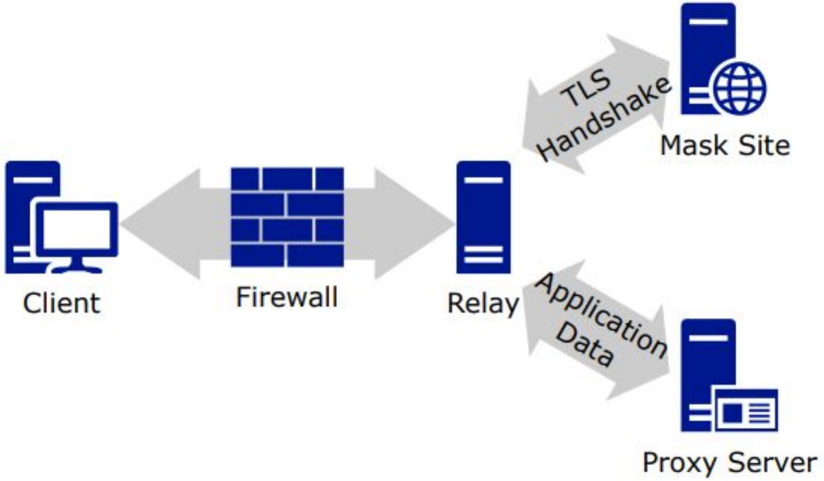


ShadowTLS: Steps



ShadowTLS

- Perform real TLS Handshake with…
  - A website that CANNOT be blocked
  - e.g., www.colorado.edu



- Client handshakes with the Relay
- Relay forwards to Mask Site
- … Until the end of Handshake, then forwards to Proxy Server

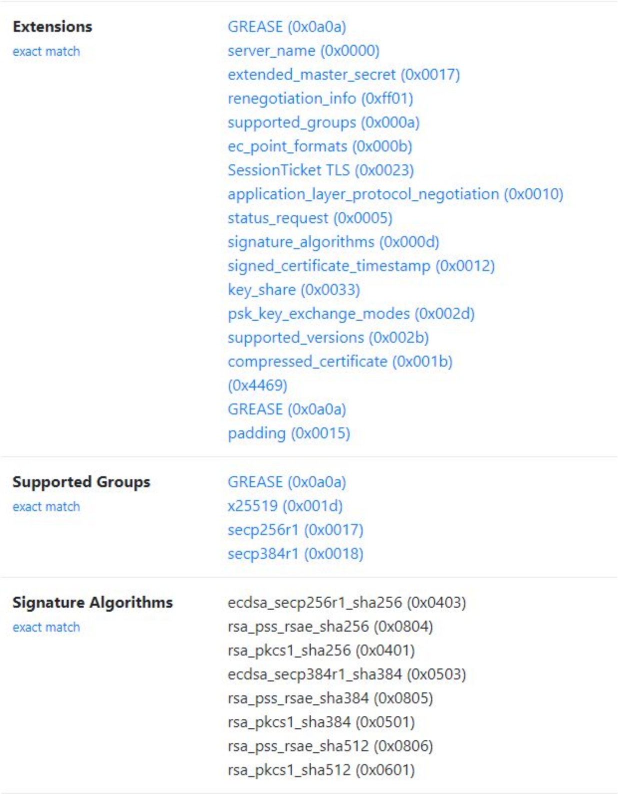
Threat Model

- Censor: the Great Firewall of China
  - Passive: Observe connections
  - Active: Modify TCP stream, active probing

- Assumptions about the censor
  - Unwilling to block all TLS traffic
  - May maintain an **allow list** of domains, and block other connections
  - Doesn’t know **shared secret** between client and relay

Passive Analysis - I

- TLS Fingerprinting
  - Fingerprint-able info in ClientHello
  - Well-known/popular fingerprints
  - ShadowTLS: **unique TLS Fingerprint**
    - `ebaa863800590426`
  - Fix: use uTLS to mimic

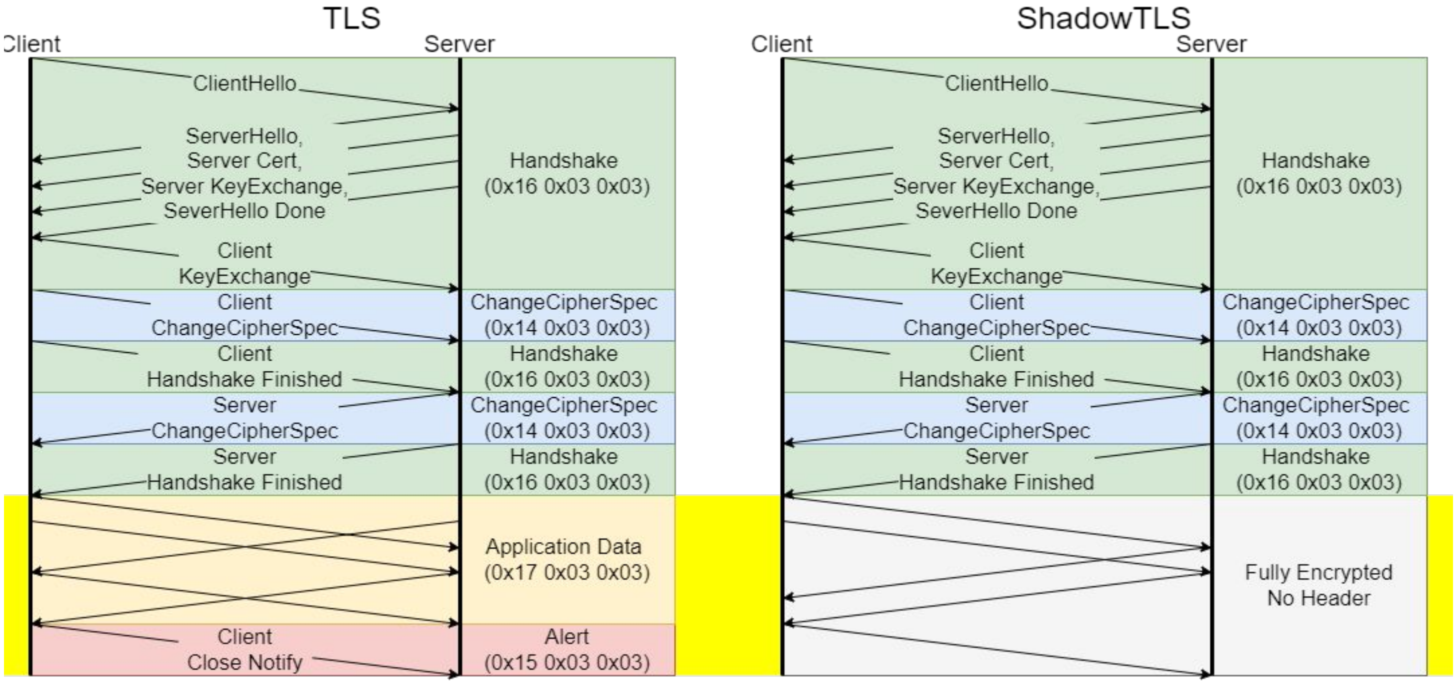


Extensions exact match	GREASE (0x0a0a) server_name (0x0000) extended_master_secret (0x0017) renegotiation_info (0xff01) supported_groups (0x000a) ec_point_formats (0x000b) SessionTicket TLS (0x0023) application_layer_protocol_negotiation (0x0010) status_request (0x0005) signature_algorithms (0x000d) signed_certificate_timestamp (0x0012) key_share (0x0033) psk_key_exchange_modes (0x002d) supported_versions (0x002b) compressed_certificate (0x001b) (0x4469) GREASE (0x0a0a) padding (0x0015)
Supported Groups exact match	GREASE (0x0a0a) x25519 (0x001d) secp256r1 (0x0017) secp384r1 (0x0018)
Signature Algorithms exact match	ecdsa_secp256r1_sha256 (0x0403) rsa_pss_rsae_sha256 (0x0804) rsa_pkcs1_sha256 (0x0401) ecdsa_secp384r1_sha384 (0x0503) rsa_pss_rsae_sha384 (0x0805) rsa_pkcs1_sha384 (0x0501) rsa_pss_rsae_sha512 (0x0806) rsa_pkcs1_sha512 (0x0601)

Passive Analysis - II

- TLS Stream Reassembly
  - Collecting all packets in the TCP stream and resembling them later
  - TLS header is expected in every packet starting from the TLS Handshake
  - ShadowTLS demonstrates Zero-Copy, no decoration to proxy packets

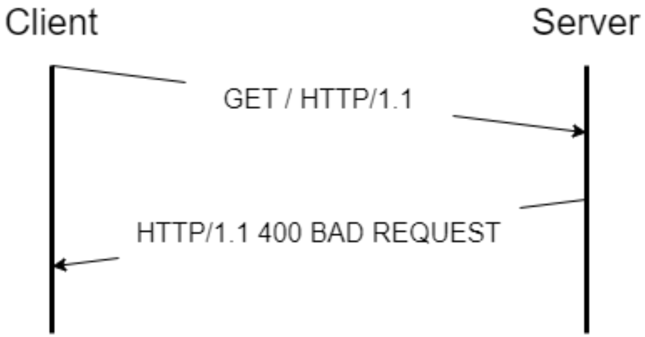
TLS vs ShadowTLS



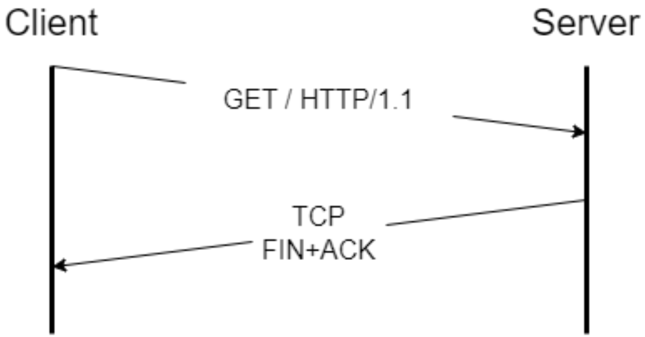
Active Probing - I

- Alternative Protocols
    - TLS largely used in HTTPS
    - HTTPS Server may respond to raw **HTTP**
    - Some respond with HTTP Page
    - Others may RESET the TCP Connection
    - ShadowTLS:
      closes connection (`FIN+ACK`)

Nginx TLS



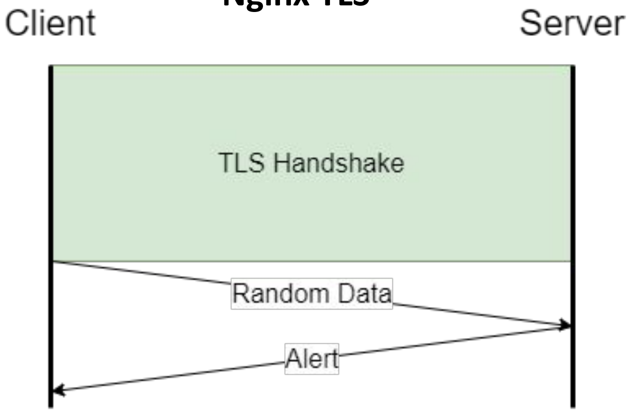
ShadowTLS



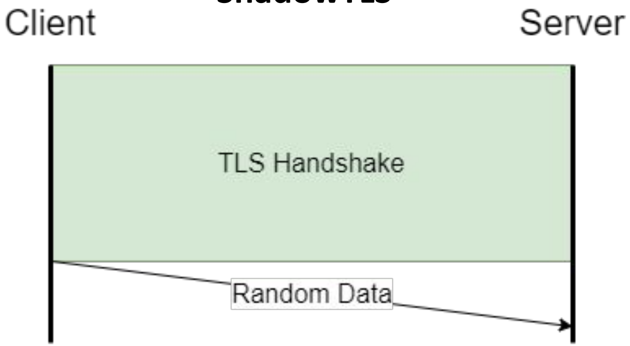
Active Probing - II

- TLS Handshake followed by arbitrary Non-TLS payload
  - Undefined behavior by RFC
  - Most replies `TLS Fatal Alert`
  - STLS forwards all packets to proxy (e.g., Shadowsocks)
  - Shadowsocks remains silent

Nginx TLS



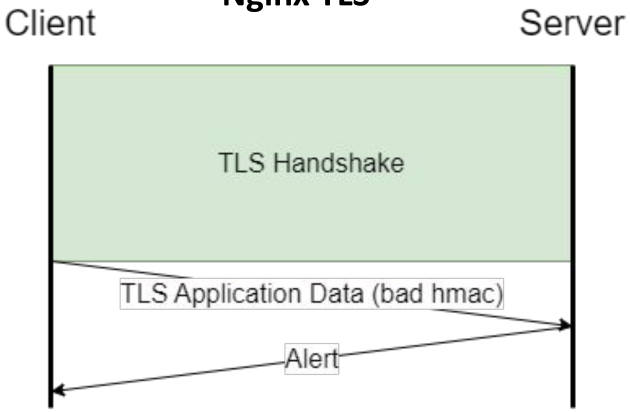
ShadowTLS



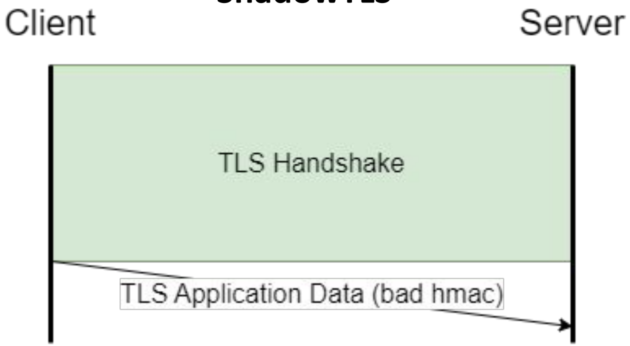
Active Probing - III

- TLS Handshake followed by *Corrupted* TLS payload
    - RFC: must send `Fatal Alert`
    - Most servers sends `Fatal Alert`
    - STLS stays silent

Nginx TLS



ShadowTLS



Evaluation

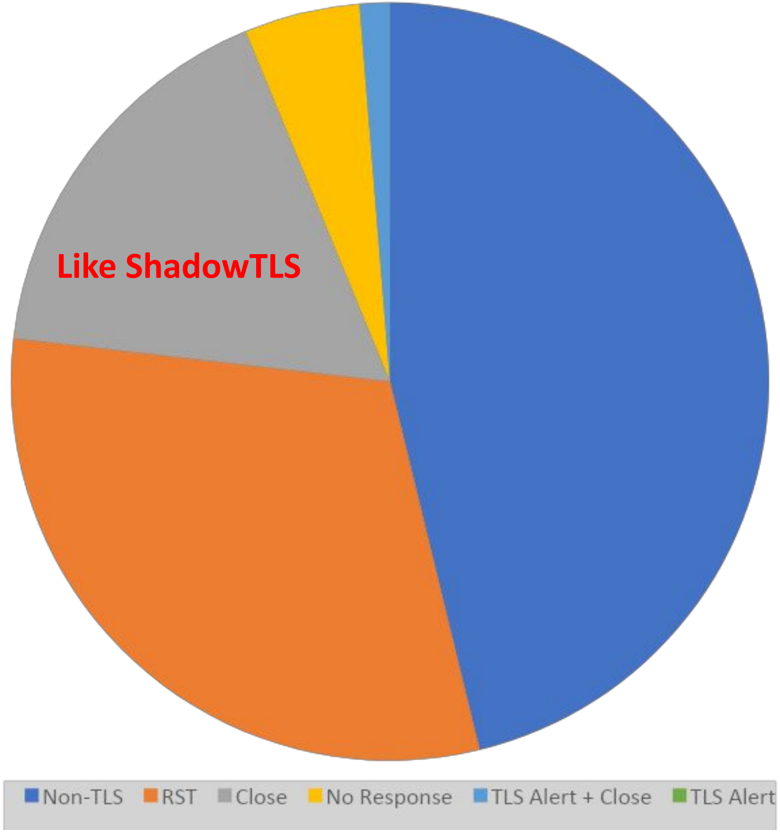
- Scanned the Internet with each, for TLS 1.2+ compatible server on port 443
  - Alternative Protocols
  - Handshake then Non-TLS
  - Handshake then Corrupted TLS

- How many TLS Servers respond like a ShadowTLS relay?

- A perfect detection would minimize False Positive Rate

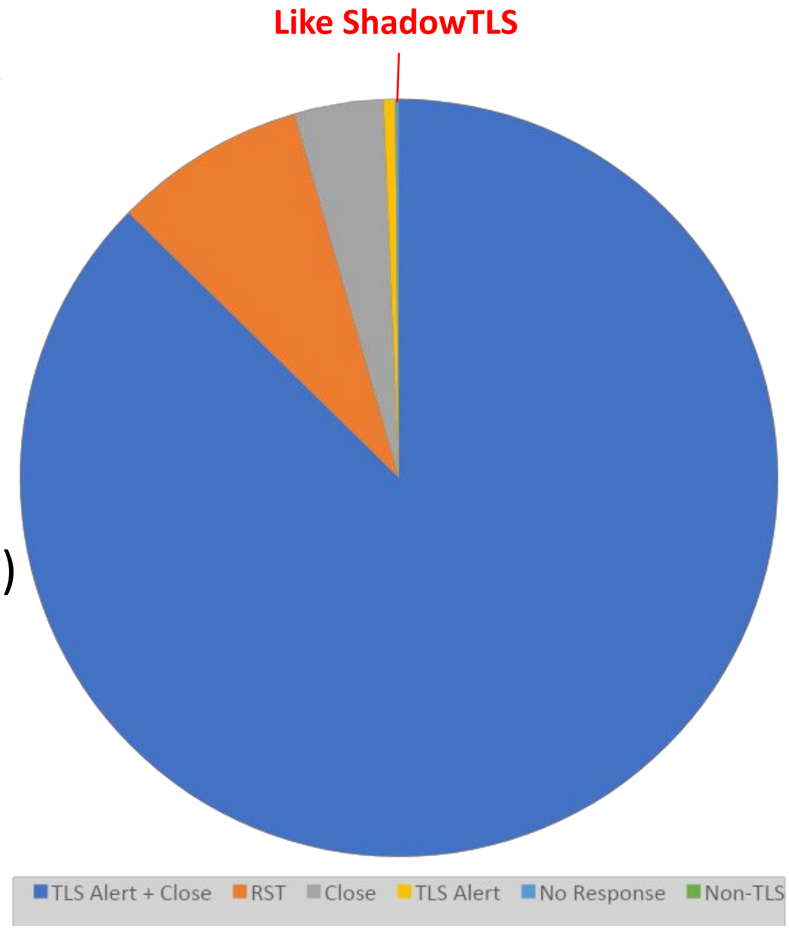
Evaluation
I - Alternative Protocols

- 46% Non-TLS (mostly HTTP)
- 31% RST
- 17% Closed Conn (like ShadowTLS)



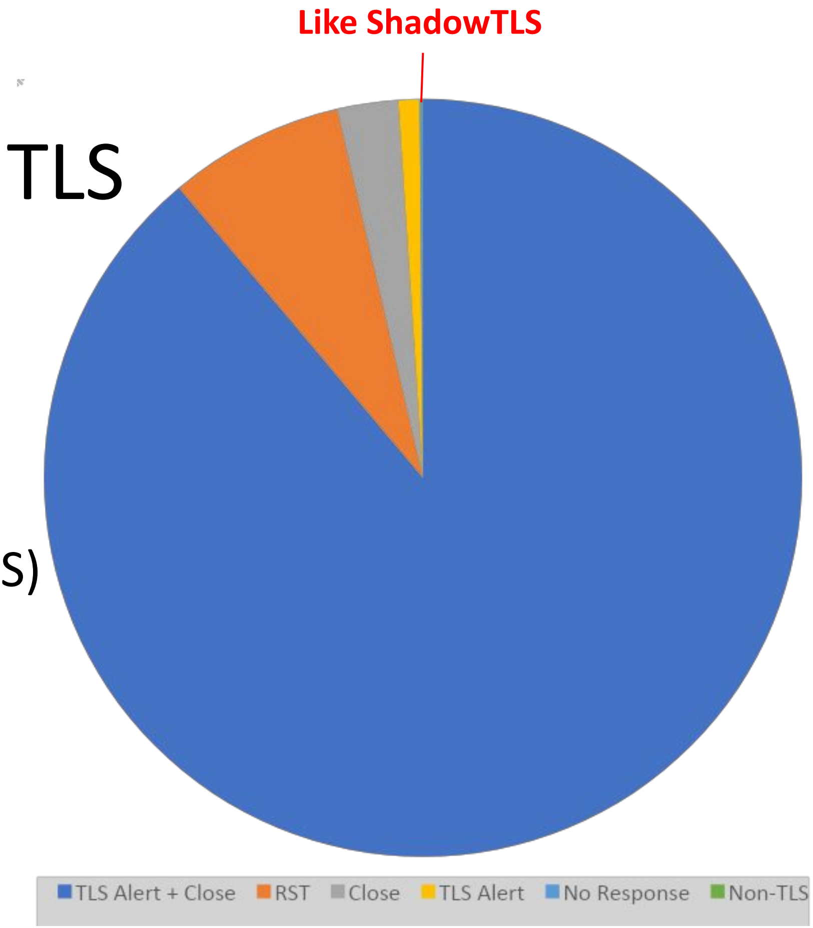
Evaluation:
II - HS then Non-TLS

- 87.3% Fatal TLS Alert
- 8.2% RST
- 0.14% No Response (like ShadowTLS)



Evaluation:
III - HS then Corrupted TLS

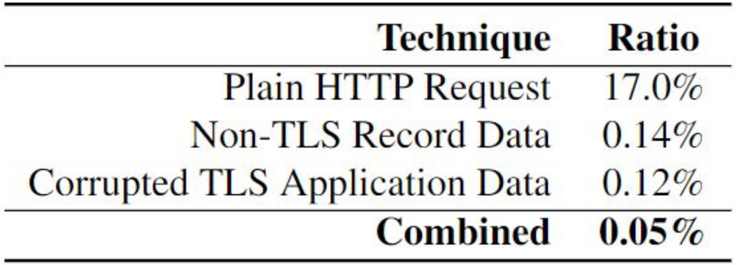
- 88.9% Fatal TLS Alert
- 7.2% RST
- 0.12% No Response (like ShadowTLS)



Evaluation

- Combining all 3 attacks
  - 15K servers (**0.05%**)

- DNS Name in default certificates
  - 5969 webex.com
  - 149 zoom.us



Technique	**Ratio**
Plain HTTP Request	17.0%
Non-TLS Record Data	0.14%
Corrupted TLS Application Data	0.12%
Combined	**0.05%**

Defenses

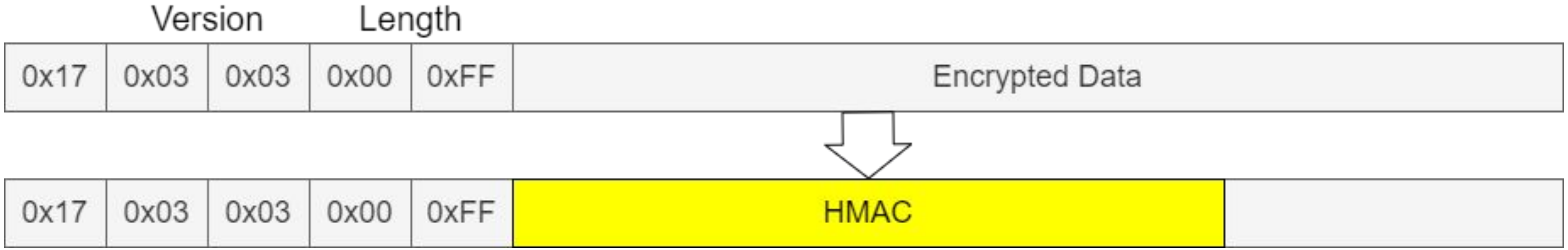
- Key Issue: Behavioral discrepancy between ShadowTLS and normal TLS

- For Passive Analysis
  - TLS Fingerprint Mimicking: uTLS
  - TLS Stream Reassembly: Add TLS Application Data Header to each proxy packet

- For Active Probing
  - Behave exactly like the Mask Site (forward all TCP packets)
  - Until the Client is **authenticated**

Defenses (Cont’d)

- Authenticating the Client
  - We include an HMAC Tag in the first TLS Application Data record after handshake
  - i.e., `Pkt[5..36] = HMAC(REPLAY_PROOF_INFO)`
  - `REPLAY_PROOF_INFO`: Some data that a censor can’t save for replay attack
    - Server Random, Client KeyShare, etc



	Version		Length		
0x17	0x03	0x03	0x00	0xFF	Encrypted Data
0x17	0x03	0x03	0x00	0xFF	HMAC

Defenses (Cont’d)

- Our `ClientAuthentication` is live since ShadowTLS V2
  - Client verify identity with Server right after TLS Handshake finishes

- Still need to patch Server to prevent other types of attacks

- Related Work:
  - Restls (Restless): An improved design based on ShadowTLS with 3-Way Auth
  - XTLS REALITY: Use real TLS with alternative certificate for valid user

Conclusion

- Detection Vulnerabilities in ShadowTLS V1 (v0.1.x)
  - Passive Analysis
  - Active Probing

- Contribution to fix issues we exposed
  - ShadowTLS V2 (v0.2.x)