black hat
USA 2019
AUGUST 3-8, 2019
MANDALAY BAY / LAS VEGAS
Monsters in the Middleboxes

Building Tools for Detecting HTTPS Interception

Luke Valenta
lvalenta@cloudflare.com
Twitter: @lukevalenta

Gabbi Fisher
gabbi@cloudflare.com
Twitter: @gabbifish
How HTTPS Interception Works

1. Administrator installs new root certificate
2. Middlebox generates certificate for example.com
3. Server has certificate for example.com
4. Middlebox inspects inner HTTP content
Types of HTTPS Interception

Antivirus/Corporate/Government

- Detect malware
- Detect C&C traffic
- Detect exfiltration
- Anti-terrorism
- Censorship

Project Zero calls out Kaspersky AV for SSL interception practices
Using an SSL proxy that simplisticly stored certificates, Kaspersky Anti-Virus left its users open to TLS certificate collisions.

Blue Coat SSL Visibility Appliance contains multiple vulnerabilities
Vulnerability Note VU#498348
Original Release Date: 2015-05-29 | Last Revised: 2015-06-02

Kazakhstan government is now intercepting all HTTPS traffic
Kazakh government first wanted to intercept all HTTPS traffic way back in 2016, but they backed off after several lawsuits.
By Catalin Cimpanu for Zero Day | July 18, 2019 -- 19:30 GMT (12:30 PST) | Topic: Security
Types of HTTPS Interception

Antivirus/Corporate/Government

- Mac malware intercepts encrypted web traffic for ad injection

- Alert (TA15-051A)
  - Lenovo Superfish Adware Vulnerable to HTTPS Spoofing

Malware

- Inject ads
- Steal private data

Censorship
Types of HTTPS Interception

Antivirus/Corporate/Government
- Detect malware
- Detect C&C traffic
- Detect exfiltration
- Anti-terrorism
- Censorship

Leaky Proxies
- Product features
- Convenience

Vulnerability Report – CVE-2018-17612
Certificate Management Vulnerability in Sennheiser HeadSetup
Hans-Joachim Knobloch, André Domnick
Secorvo Security Consulting GmbH
Version 1.2
Date October 31, 2018
Types of HTTPS Interception

Antivirus/Corporate/Government

Incident report on memory leak caused by Cloudflare parser bug
23 Feb 2017 by John Graham-Cumming

CLOUDTEST VULNERABILITY (CVE-2019-11011)

By Akamai InfoSec June 17, 2019 10:00 AM
0 Comments

Ticketbleed (CVE-2016-9244)
Ticketbleed is a software vulnerability in the TLS/SSL stack of F5 BIG-IP appliances allowing a remote attacker to extract up to 31 bytes of uninitialized memory at a time.

Malware

- Inject ads
- Steal private data

Reverse Proxies

- Security
- Performance
- Reliability
Detecting HTTPS Interception [Durumeric et al., 2017]
Identifying HTTP and TLS Clients

**HTTP**
Parse User Agent Header
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36

**TLS**
No identifying field is present in the protocol. Instead, use known techniques for fingerprinting browsers based on TLS Client Hello.

1. [Ristić; 2009]
2. [Majkowski; 2012]
3. [Brotherston; 2015]
4. [Anderson, McGrew; 2016]
5. [Durumeric et al.; 2017]
6. [Althouse & Atkinson, Atkins; 2017]
7. [Frolov, Wustrow; 2019]
8. [...and others]
TLS Handshake (RFC 5246)

struct {
  ProtocolVersion client_version;
  Random random;
  SessionID session_id;
  CipherSuite cipher_suites<2..2^16-2>;
  CompressionMethod compression_methods<1..2^8-1>;
  select (extensions_present) {
    case false:
      struct {};
    case true:
      Extension extensions<0..2^16-1>;
  }
} ClientHello;

TLS libraries tend to keep these fields the same!
TLS Fingerprinting based on Client Hello

- **Cipher Suites (17 suites)**
  - Cipher Suite: Reserved (GREASE) (0xeaea)
  - Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
  - Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  - Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)

- **Supported Groups (4 groups)**
  - Supported Group: Reserved (GREASE) (0x3a3a)
  - Supported Group: x25519 (0x800d)
  - Supported Group: secp256r1 (0x0017)
  - Supported Group: secp384r1 (0x0018)

- **Extensions Length**: 401
  - Extension: Reserved (GREASE) (len=0)
  - Extension: server_name (len=28)
  - Extension: extended_master_secret (len=0)
  - Extension: renegotiation_info (len=1)

- **Cipher Suites (48 suites)**
  - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
  - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
  - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc0e8)
  - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc0e4)

- **Supported Groups (3 groups)**
  - Supported Group: x25519 (0x001d)
  - Supported Group: secp256r1 (0x0017)
  - Supported Group: secp384r1 (0x0018)

- **Extensions Length**: 54
  - Extension: ec_point_formats (len=2)
  - Extension: supported_groups (len=8)
  - Extension: session_ticket (len=0)
  - Extension: signature_algorithms (len=28)
HTTPS Interception Detection Process

1. Build database of HTTP and TLS browser fingerprints
2. Check HTTP and TLS fingerprints of incoming requests against database
MITMEngine: HTTPS Interception Detection Library

Open sourced at https://github.com/cloudflare/mitmengine. PRs welcome!

● Goal #1: Maintainability
  ○ Fingerprints quickly go stale with browser updates
  ○ Time-consuming to generate new fingerprints manually
  ○ Goal is to automatically generate ground truth fingerprints from Cloudflare’s network

● Goal #2: Flexibility
  ○ Currently support a flexible fingerprint format to model a variety of browser behavior
  ○ Plan to add support for other TLS fingerprint formats (JA3, tlsfingerprint.io)

● Goal #3: Performance
  ○ The system should be fast enough to deploy at scale
  ○ Currently deployed on a sample of Cloudflare traffic
MALCOLM: HTTPS Interception on Cloudflare’s Network


- Provides insight into HTTPS Interception observed by Cloudflare
- Powered by MITMEngine
- Allows for filtering by OS, browser, HTTPS interception tool, etc.
MALCOLM: Overall HTTPS Interception Rates
# MALCOLM: HTTPS Interception by Browser/OS

The following table displays recognized browsers.

<table>
<thead>
<tr>
<th>BROWSER / USER AGENT</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>61.81%</td>
</tr>
<tr>
<td>Safari</td>
<td>11.84%</td>
</tr>
<tr>
<td>IE</td>
<td>10.26%</td>
</tr>
<tr>
<td>Firefox</td>
<td>7.01%</td>
</tr>
<tr>
<td>Other</td>
<td>9.08%</td>
</tr>
</tbody>
</table>

The following table displays recognized operating systems.

<table>
<thead>
<tr>
<th>OPERATING SYSTEM</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>38.95%</td>
</tr>
<tr>
<td>Windows 10</td>
<td>19.58%</td>
</tr>
<tr>
<td>iOS</td>
<td>10.93%</td>
</tr>
<tr>
<td>Windows 7</td>
<td>10.23%</td>
</tr>
<tr>
<td>Other</td>
<td>20.31%</td>
</tr>
</tbody>
</table>
Case Study: Kazakhstan

- Summary of incident w/ live results at [https://censoredplanet.org/kazakhstan](https://censoredplanet.org/kazakhstan)
- KZ chooses which domains to MITM based on SNI (server name indication)
- Encrypted SNI (eSNI) is on the horizon -- makes selective interception harder
- New MALCOLM feature (soon): filter MITM results by country
Black Hat Sound Bytes

TLS-terminating middleboxes pose serious threats to network security

Heuristics based on HTTP and TLS fingerprints can be effective at detecting HTTPS interception

Our new open source tool and public dashboard provide new insights into the state of HTTPS interception on the Internet
References

- [Majkowski; 2012] SSL fingerprinting for p0f. https://idea.popcount.org/2012-06-17-ssl-fingerprinting-for-p0f
- [Frolov, Wustrow; 2019]. Tlsfingerprint.io