Elad Shuster
Senior Security Researcher, Team Lead
Akamai

PASSIVE FINGERPRINTING OF HTTP/2 CLIENTS

Before we begin….

Agenda

- Usage Statistics on Akamai's Platform
- HTTP/2 Overview
- Passive Client Fingerprinting
- HTTP/2 Fingerprinting and it's Use Cases
- HTTP/2 Threat landscape

```
[  1:11PM ]  [ eshuster@tlv-mpixn:~ ]
 $
```

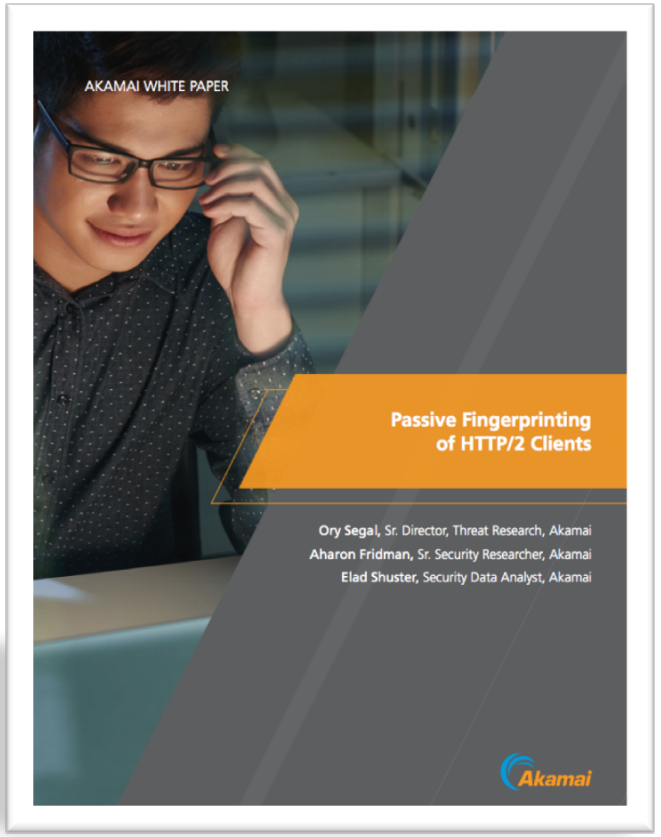
- Uptime ~ 37 years
- Threat Research Team @ Akamai Technologies
- Enjoying Big-Data
- Love Single Malt Whiskeys!
- CPA(il), MBA

Acknowledgments

This research was led by:

Ory Segal
Sr. Director Threat Research
Akamai

Aharon Friedman
Sr. Security Researcher
Akamai



http://akamai.me/2qWIqON

DATA COLLECTION

AKAMAI

The Intelligent Platform

- 220,000+ Edge Servers
- 3,315+ Locations
- 1200+ Cities
- 129 Countries
- 1,227+ Networks
- 60 Tbps at last peak

The Data

- 3 trillion hits per day
- 1 Billion unique IPs seen quarterly
- 13+ trillion log lines per day
- 260+ TB of compressed daily logs

15 - 30% of all web traffic

http://wwwnui.akamai.com/gnet/globe/

http://tech.akamai.com/attack-globe/

Akamai

WEB APPLICATION ATTACKS

SQL Injection (71%)
273,037

Remote File Inclusion (15.2%)
58,458

Cross-site Scripting (12.3%)
47,310

PHP Injection (1.2%)
4,609

Command Injection (0.4%)
1,542

31.3%

9,229,381

Attack events in the last **24** hours

HTTP/2 Usage Statistics

27.2M
Unique IP Addresses

1 Billion
Daily requests

10%
Of Total Traffic

675.3K
User Agents

15.7K
Hosts

413.4M
Login requests

HTTP/2 OVERVIEW

HTTP 1.x

GET /index.html HTTP/1.1

Host: www.fdsa.co

Connection: keep-alive

User-Agent: Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml

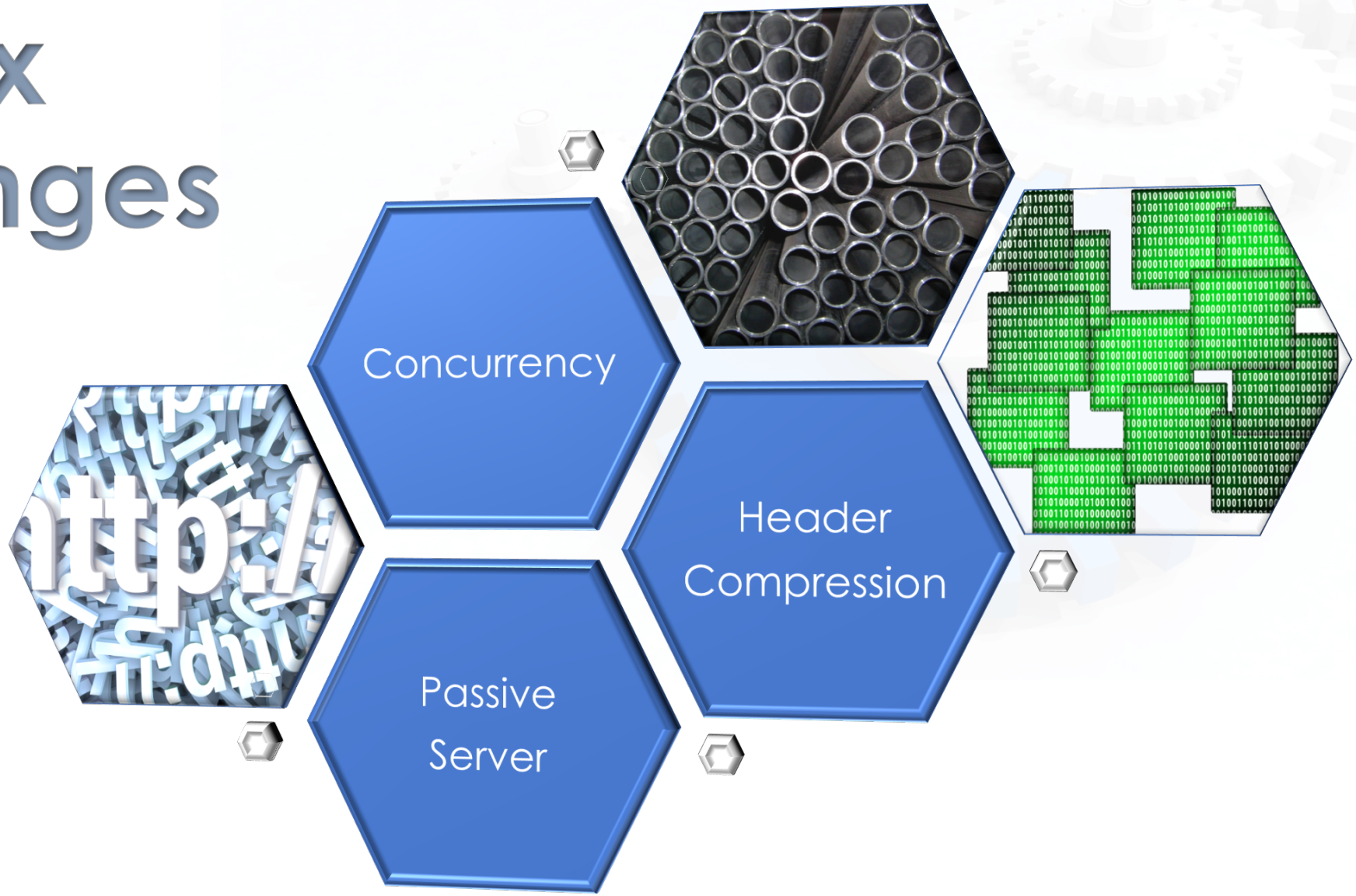
Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9,he;q=0.8

HTTP/2 Overview

- Based on the SPDY Protocol (develop by Google)
- Published during 2015:
  - RFC 7540 Hypertext Transfer Protocol Version 2 (HTTP/2)
  - RFC 7541 HPACK: Header Compression for HTTP/2
- Binary Protocol
- Addresses (performance) challenges in HTTP/1.x

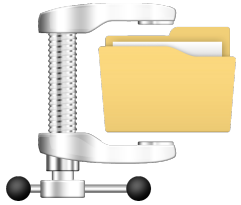
HTTP/1.x Challenges



Enter HTTP/2…

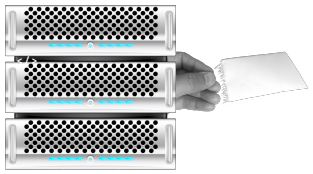
- Concurrency

Allows interleaving of request and response messages on the same TCP connection



- Compression

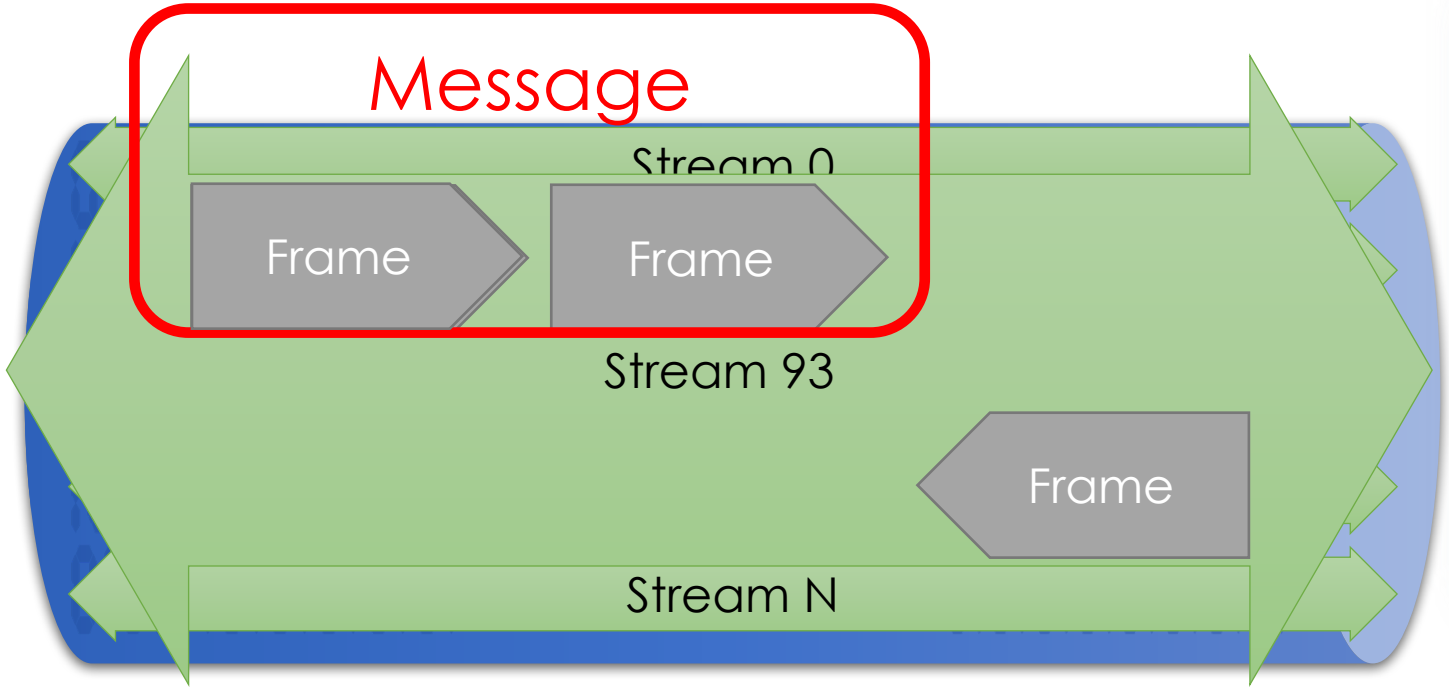
Uses an efficient coding for HTTP header fields, as well as header compression (HPACK)



- Server push

Adds a new interaction mode whereby a server can push responses to a client, if it thinks the client will need them

HTTP/2 Connection



Stream

an independent, bidirectional sequence of frames exchanged between the client and server

HTTP/2 Key Elements

Frame

smallest unit of communication in HTTP/2

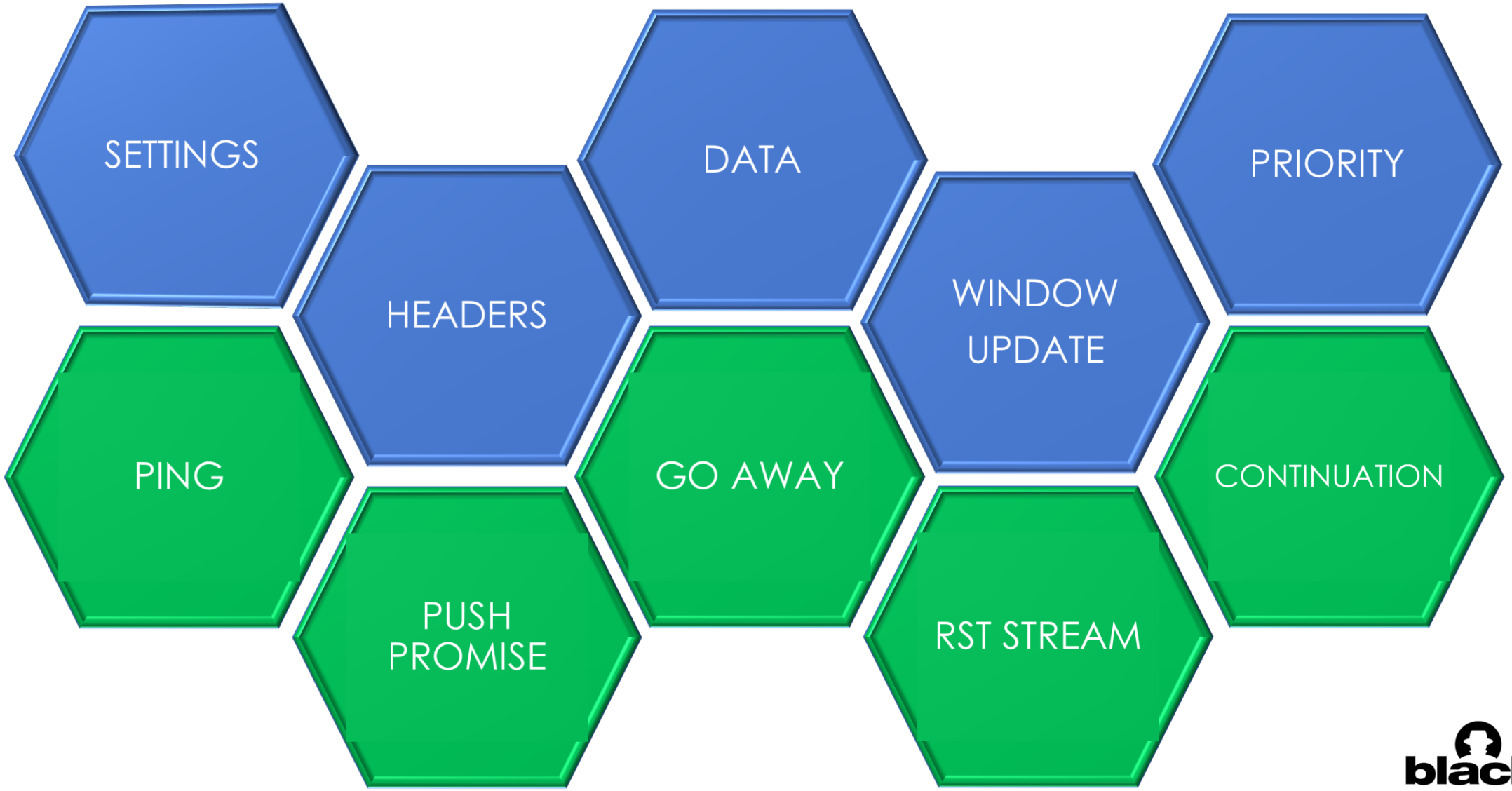
Stream

bidirectional flow of frames within an established connection - Assigned with a Unique ID and a Priority

Message

sequence of frames that map to a logical request or response

Frame Types



Frame Structure

```
+-----------------------------------------------+
|                 Length (24)                   |
+---------------+---------------+---------------+
|   Type (8)    |   Flags (8)   |
+-+-------------+---------------+-------------------------------+
|R|                 Stream Identifier (31)                      |
+=+===============================================================+
|                   Frame Payload (0...)                      ...
+---------------------------------------------------------------+
```

Figure 1: Frame Layout

Frame Structure - Example

Length (24 bit) = XXXXX

Type = 0x1
HEADERS_FRAME

Flags = 0x25
END_HEADERS (0x4) ,
END_STREAM (0x1),
PRIORITY (0x20)

R

Stream Identifier (Stream ID = 73)

Pad Length = 0

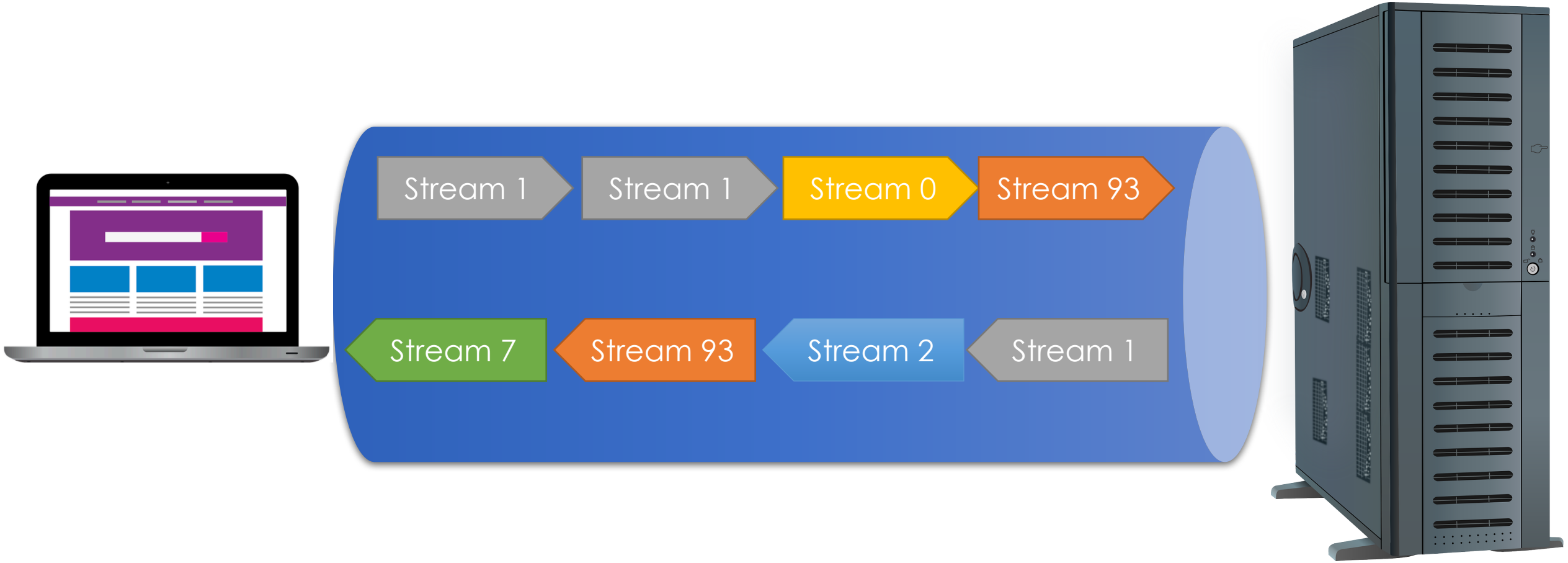
E = 1

Stream Dependency = 0

Weight = 220

```
:method: GET
:authority: http2.akamai.com
:scheme: https
:path: /resources/h2.css
user-agent: Mozilla/5.0 (......) Chrome/62.0.3202.75
```

Single TCP Connection



Akamai

HTTP/2 is the future of the Web, and it is here!

Your browser supports HTTP/2!

This is a demo of HTTP/2's impact on your download of many small tiles making up the Akamai Spinning Globe.

HTTP/1.x	HTTP/2
Text	Binary
Clear Text OR Encrypted	Clear Text OR Encrypted
Multiple TCP Connections	Single TCP connection
Pipelining of requests	Request Multiplexing
-	HPACK Header Compression
-	Server Push Enabled

Keep in mind…

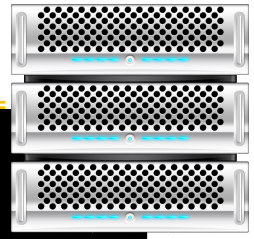
- HTTP/2 is binary (you can't use netcat to draft traffic)
- HTTP/2 implementations use TLS
- Most intercepting proxies (e.g. Burp) don't support H2

```
00000238  17 03 03 07 f5 00 00 00  00 00 00 00 01 d7 10 1d  ........ ........
00000248  bc 1c e3 c9 3a b1 b0 53  32 f4 31 e6 34 4b 57 e9  ....:..S 2.1.4KW.
00000258  68 b2 0a 93 fe 3d 7a b5  94 fe a0 df 5d d4 d5 22  h....=z. ....]..".
00000268  a2 e6 d2 81 66 bc 68 64  85 75 fa 4b aa c3 a1 fc  ....f.hd .u.K....
00000278  02 d9 94 21 df 5d e9 74  09 d2 bd 85 f2 94 65 01  ...!.].t ......e.
00000288  5b 65 20 8e 46 1a 8c 65  8a 6a eb 01 26 26 f5 a2  [e .F..e .j..&&..
00000298  6f 5e bb 5f a5 25 96 b4  f1 85 f5 63 bc 95 64 08  o^._.%.. ...c..d.
000002A8  9b 21 06 60 97 66 14 fd  ca 2c 31 4f 90 a1 16 e8  .!.`.f.. .,1O....
000002B8  f7 05 5c 05 2a a7 99 7b  9d 33 5e 2f df 32 c9 17  ..\.*..{ .3^/.2..
000002C8  fc 40 40 83 94 03 fa f2  e2 77 0e df fc e9 b9 5a  .@@..... .w.....Z
000002D8  8f bd c6 6a 9a 93 06 cc  1d 4a 01 bf 50 10 a6 a3  ...j.... .J..P...
000002E8  34 b3 08 fe 4a c7 91 ae  6b 08 39 ff b5 09 de 52  4...J... k.9....R
000002F8  e9 03 3e 05 4d 37 04 bd  ec 08 13 d1 70 2c 0a 1e  ..>.M7.. ....p,..
00000308  f9 c1 a9 34 23 62 c9 a6  02 ea 83 16 bb 18 6d a7  ...4#b.. ......m.
00000318  93 d0 f3 66 58 cd 2a 88  b6 bc 30 38 b6 32 21 3b  ...fX.*. ..08.2!;
00000328  bf 40 da 1e 7b 00 30 60  90 a4 81 53 6e 4a 02 22  .@..{.0` ...SnJ."
00000338  a8 c3 4a f9 a1 03 c5 79  81 e2 63 d2 1b e5 23 7a  ..J....y ..c...#z
00000348  5c bb 69 b6 a3 6b 4a a4  7c 38 7e f2 2f 0a 94 f1  \.i..kJ. |8~./...
```


Inspection
Tools

Server Side

Server side



```
IPv4: listen 0.0.0.0:4433
IPv6: listen :::4433
[ALPN] client offers:
 * h2
 * http/1.1
[ALPN] client offers:
 * h2
 * http/1.1
SSL/TLS handshake completed
The negotiated protocol: h2
[id=1] [ 42.888] send SETTINGS frame <length=6, flags=0x00, stream_id=0>
          (niv=1)
          [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
SSL/TLS handshake completed
The negotiated protocol: h2
[id=2] [ 43.877] send SETTINGS frame <length=6, flags=0x00, stream_id=0>
          (niv=1)
          [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
[id=1] [ 44.130] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
          (niv=3)
          [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
          [SETTINGS_INITIAL_WINDOW_SIZE(0x04):131072]
          [SETTINGS_MAX_FRAME_SIZE(0x05):16384]
[id=1] [ 44.131] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=0>
          (window_size_increment=12517377)
[id=1] [ 44.131] recv PRIORITY frame <length=5, flags=0x00, stream_id=3>
```

Web server debug logs

Client Side

Client side

chrome://net-internals/#events × HTTP/2: the Future of the Inter ×

Chrome | chrome://net-internals/#events&q=id:16690

capturing events (65521)

- Capture
- Import
- Proxy
- Events
- Timeline
- DNS
- Sockets
- Alt-Svc
- HTTP/2
- QUIC
- SDCH
- Cache
- Modules
- Domain Security Policy
- Bandwidth
- Prerender

(?) id:16690 1 of 3733

	ID	Source Type	Description
☑	16690	HTTP2_SESSION	

16690: HTTP2_SESSION
Start Time: 2017-10-29 13:47:42.201

```
t= 7652 [st=    0]  HTTP2_SESSION_PING
                       --> is_ack = false
                       --> type = "sent"
                       --> unique_id = 1
t= 7652 [st=    0]  HTTP2_SESSION_SEND_HEADERS
                       --> exclusive = true
                       --> fin = true
                       --> has_priority = true
                       --> :method: GET
                           :authority: http2.akamai.com
                           :scheme: https
                           :path: /resources/h2.css
                           dpr: 1
                           user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6)
                           viewport-width: 1430
                           accept: text/css,*/*;q=0.1
                           referer: https://http2.akamai.com/
                           accept-encoding: gzip, deflate, br
                           accept-language: en-US,en;q=0.9,he;q=0.8,zh-TW;q=0.7,zh;q=0
                           cookie: [1033 bytes were stripped]
                       --> parent_stream_id = 0
                       --> source_dependency = 17235 (HTTP_STREAM_JOB)
                       --> stream_id = 7
                       --> weight = 256
t= 7653 [st=    1]  HTTP2_SESSION_SEND_HEADERS
```

Chrome://net-internals

Let’s get familiarized with the logs….

```
[ 43.877] send SETTINGS frame <length=6, flags=0x00, stream_id=0>
    (niv=1)
    [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
[ 44.130] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
    (niv=3)
    [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
    [SETTINGS_INITIAL_WINDOW_SIZE(0x04):131072]
    [SETTINGS_MAX_FRAME_SIZE(0x05):16384]
```

Stream 0

Source	Frame Type	Values
Server	SETTINGS	[SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]

```
[ 43.877] send SETTINGS frame <length=6, flags=0x00, stream_id=0>
    (niv=1)
    [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
[ 44.130] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
    (niv=3)
    [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
    [SETTINGS_INITIAL_WINDOW_SIZE(0x04):131072]
    [SETTINGS_MAX_FRAME_SIZE(0x05):16384]
```

Stream 0

Source	Frame Type	Values
Server	SETTINGS	[SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
Client	SETTINGS	[SETTINGS_HEADER_TABLE_SIZE(0x01):65536] [SETTINGS_INITIAL_WINDOW_SIZE(0x04):131072] [SETTINGS_MAX_FRAME_SIZE(0x05):16384]

```
[id=1] [ 44.134] recv HEADERS frame <length=211, flags=0x25, stream_id=15>
          ; END_STREAM | END_HEADERS | PRIORITY
          (padlen=0, dep_stream_id=13, weight=42, exclusive=0)
          ; Open new stream
[id=1] [ 44.135] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=15>
          (window_size_increment=12451840)

[id=1] [ 44.136] send HEADERS frame <length=45, flags=0x05, stream_id=15>
```

Stream 15

Source	Frame Type	Values
Client	HEADERS	<Flags, Headers>

```
[id=1] [ 44.134] recv HEADERS frame <length=211, flags=0x25, stream_id=15>
          ; END_STREAM | END_HEADERS | PRIORITY
          (padlen=0, dep_stream_id=13, weight=42, exclusive=0)
          ; Open new stream
[id=1] [ 44.135] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=15>
          (window_size_increment=12451840)
[id=1] [ 44.136] send HEADERS frame <length=45, flags=0x05, stream_id=15>
```

Stream 15

Source	Frame Type	Values
Client	HEADERS	<Flags, Headers>
Client	WINDOW_UPDATE	(window_size_increment=12451840)

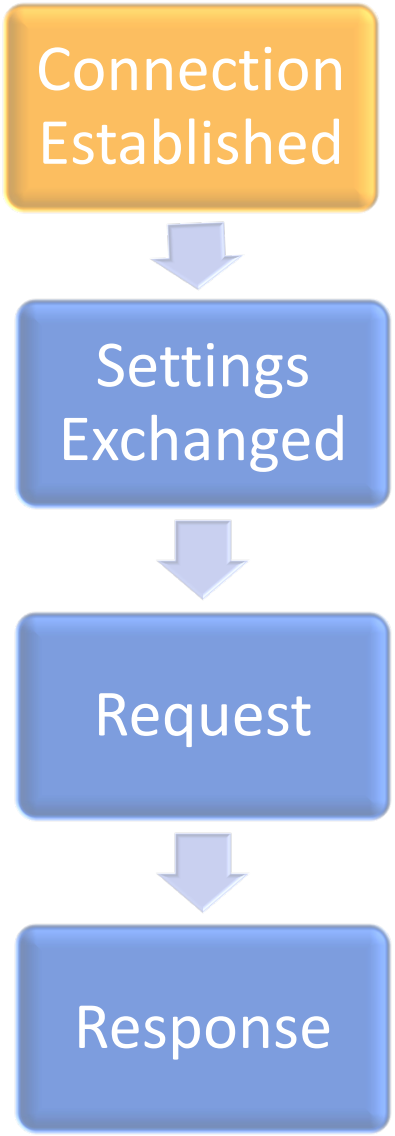
```
[id=1] [ 44.134] recv HEADERS frame <length=211, flags=0x25, stream_id=15>
          ; END_STREAM | END_HEADERS | PRIORITY
          (padlen=0, dep_stream_id=13, weight=42, exclusive=0)
          ; Open new stream
[id=1] [ 44.135] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=15>
          (window_size_increment=12451840)

[id=1] [ 44.136] send HEADERS frame <length=45, flags=0x05, stream_id=15>
```

Stream 15

Source	Frame Type	Values
Client	HEADERS	<Flags, Headers>
Client	WINDOW_UPDATE	(window_size_increment=12451840)
Server	HEADERS	<Flags, Headers>

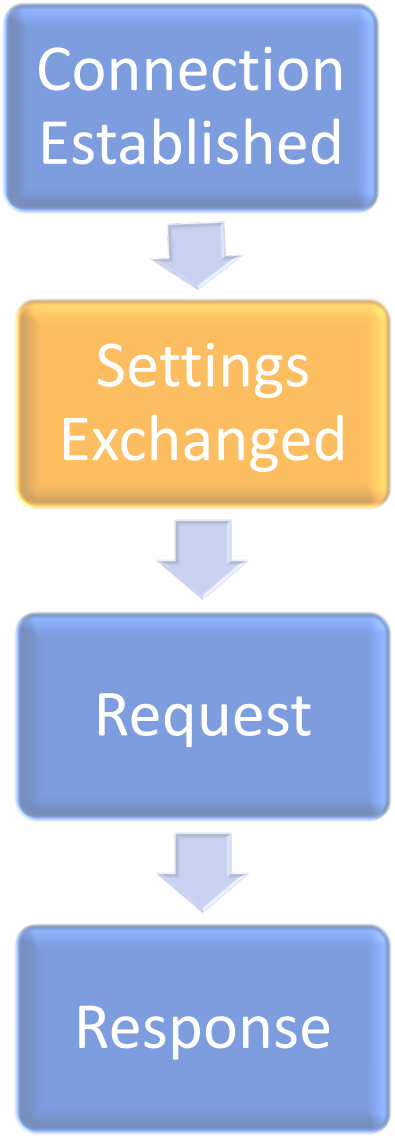
HTTP/2 Conversation




```
[ALPN] client offers:
 * h2
 * http/1.1
[ALPN] client offers:
 * h2
 * http/1.1
SSL/TLS handshake completed
The negotiated protocol: h2
```

- HTTP/2 is negotiated via the TLS ALPN extension (Application Level Protocol Negotiation)

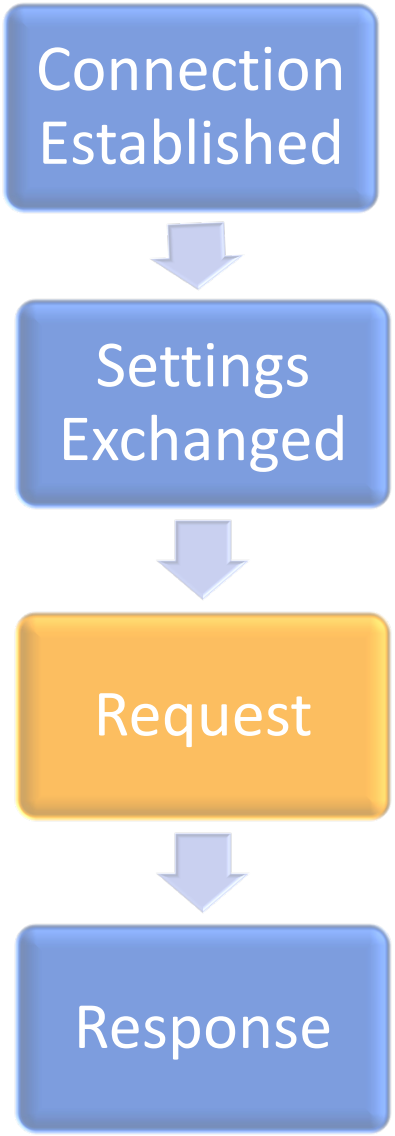
HTTP/2 Conversation




```
[id=26] [35556.047] send SETTINGS frame <length=6, flags=0x00, stream_id=0>
          (niv=1)
          [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
SSL/TLS handshake completed
The negotiated protocol: h2
[id=27] [35556.053] send SETTINGS frame <length=6, flags=0x00, stream_id=0>
          (niv=1)
          [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
[id=26] [35556.053] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
          (niv=3)
          [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
          [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):1000]
          [SETTINGS_INITIAL_WINDOW_SIZE(0x04):6291456]
[id=26] [35556.054] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=0>
          (window_size_increment=15663105)
[id=26] [35556.054] send SETTINGS frame <length=0, flags=0x01, stream_id=0>
          ; ACK
          (niv=0)
```

- SETTINGS – Always Stream ID = 0

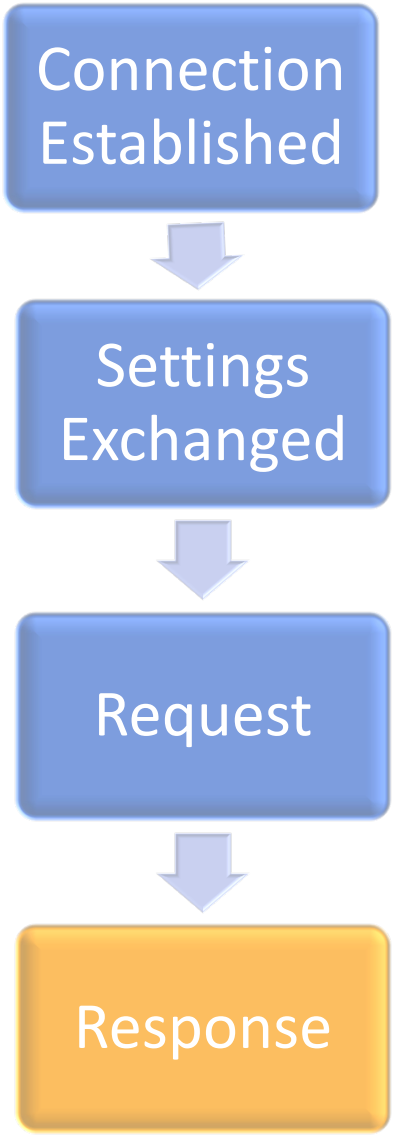
HTTP/2 Conversation




```
[35556.057] recv (stream_id=1) :method: GET
[35556.057] recv (stream_id=1) :authority: www.f
[35556.058] recv (stream_id=1) :scheme: https
[35556.058] recv (stream_id=1) :path: /robots.txt
[35556.058] recv (stream_id=1) user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)
[35556.058] recv (stream_id=1) accept: */*
[35556.058] recv (stream_id=1) accept-encoding: gzip, deflate, br
[35556.058] recv (stream_id=1) accept-language: en-US,en;q=0.8,he;q=0.6
[35556.059] recv HEADERS frame <length=170, flags=0x25, stream_id=1>
  ; END_STREAM | END_HEADERS | PRIORITY
  (padlen=0, dep_stream_id=0, weight=220, exclusive=1)
  ; Open new stream
```

- Client Send a HEADERS frame
- Stream ID = 1

HTTP/2 Conversation




```
[35556.060] send HEADERS frame <length=65, flags=0x04, stream_id=1>
  ; END_HEADERS
  (padlen=0)
  ; First response header
  :status: 404
  server: nghttpd nghttp2/1.7.1
  date: Tue, 19 Sep 2017 09:20:59 GMT
  content-type: text/html; charset=UTF-8
[35556.060] send DATA frame <length=146, flags=0x01, stream_id=1>
  ; END_STREAM
```

- Server Responds with a Message
- Message = HEADERS and DATA frames
- Stream ID = 1

Passive Client Fingerprinting

Passive collection of attributes that might expose consistent unique behavior

Transport layer
Session layer
Application layer

Fingerprinting software clients NOT end users

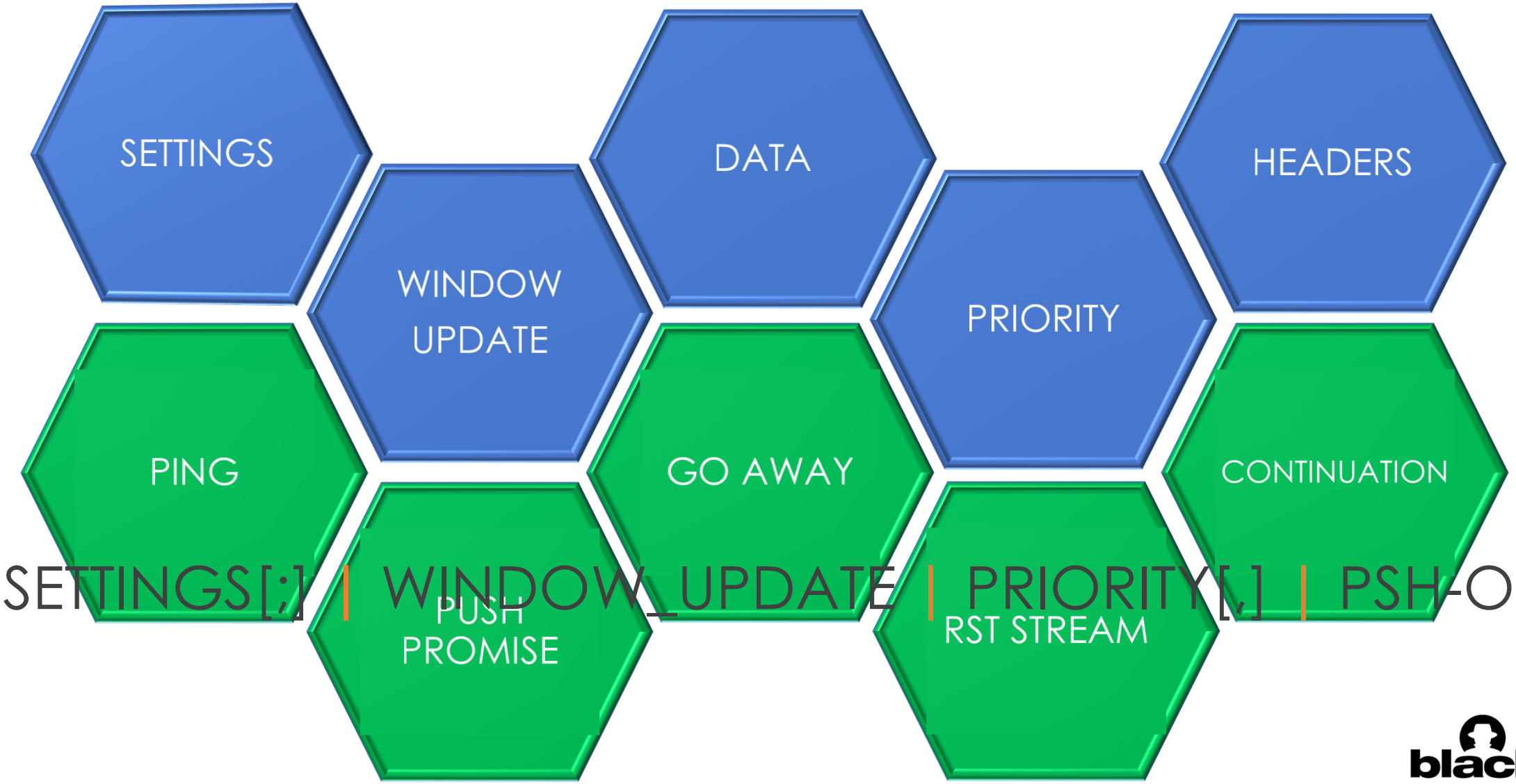
Deduce about up-time, OS (type and version), Running Software, etc…

Passive Client Fingerprinting

- Observe client's behaviors while establishing a connection
- Attributes sent by the client that might expose consistent unique behavior:
  - ✓ Initial connection settings
  - ✓ Initial flow control settings
  - ✓ Prioritization
  - ✓ (Pseudo) Header Order

HTTP/2 Passive Client Fingerprinting

Proposed Fingerprint Elements



SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

- SETTINGS frame Conveys configuration parameters
- SETTINGS MUST be sent by BOTH endpoints at the start of a connection
- Parameter default values vary between implementations
- Stream identifier for a SETTINGS frame MUST be zero

SETTINGS PARAMETERS

Parameter Name	Scope
SETTINGS_HEADER_TABLE_SIZE (0x1) (0x1)	Allows the sender to inform the remote endpoint of the maximum size of the header compression table used to decode header blocks, in octets.
SETTINGS_ENABLE_PUSH (0x2) (0x2)	This setting can be used to disable server push (Section 8.2).
SETTINGS_MAX_CONCURRENT_STREAMS (0x3)	Indicates the maximum number of concurrent streams that the sender will allow.
SETTINGS_INITIAL_WINDOW_SIZE (0x4)	Indicates the sender's initial window size (in octets) for stream-level flow control. The initial value is 216-1 (65,535) octets.
SETTINGS_MAX_FRAME_SIZE (0x5)	Indicates the size of the largest frame payload that the sender is willing to receive, in octets.
SETTINGS_MAX_HEADER_LIST_SIZE (0x6)	This advisory setting informs a peer of the maximum size of header list that the sender is prepared to accept, in octets.

Firefox/55.0 - Mac OS X 10.11.6

```
[ 44.130] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
  (niv=3)
  [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):131072]
  [SETTINGS_MAX_FRAME_SIZE(0x05):16384]
```

Safari 10.1.2 - Mac OS X 10.11.6

```
[23003.408] recv SETTINGS frame <length=12, flags=0x00, stream_id=0>
  (niv=2)
  [SETTINGS_ENABLE_PUSH(0x02):0]
  [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):100]
```

EDGE 15.15063 – Windows 10

```
[28297.704] recv SETTINGS frame <length=12, flags=0x00, stream_id=0>
  (niv=2)
  [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):1024]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):10485760]
```

Chrome 60 – Android 8.0.0 Pixel XL

```
[30336.100] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
  (niv=3)
  [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
  [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):1000]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):6291456]
```

User-Agent	MAX CONCURRENT STREAMS	HEADER TABLE SIZE	MAX HEADER LIST SIZE	MAX FRAME SIZE	INITIAL WINDOW SIZE	ENABLE PUSH
Mozilla/5.0 (Android 6.0; Mobile; rv:52.0) Gecko/52.0 Firefox/52.0	[]	['4096']	[]	['16384']	['32768']	[]
Mozilla/5.0 (Android 6.0.1; Tablet; rv:47.0) Gecko/47.0 Firefox/47.0	[]	[]	[]	['16384']	['32768']	[]
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729; McAfee)	['1024']	[]	[]	[]	['10485760']	[]
Mozilla/5.0 (Linux; Android 7.1; Pixel XL...	['100']	['4096']	['131072']	['16384']	['163840']	['0']


forging a fingerprint....

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

EDGE 15.15063 – Windows 10

```
[28297.704] recv SETTINGS frame <length=12, flags=0x00, stream_id=0>
  (niv=2)
  [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):1024]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):10485760]
```

[3:1024 ;]

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

EDGE 15.15063 – Windows 10

```
[28297.704] recv SETTINGS frame <length=12, flags=0x00, stream_id=0>
  (niv=2)
  [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):1024]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):10485760]
```

[3:1024 ; 4:10485760]

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Firefox/55.0 - Mac OS X 10.11.6

```
[ 44.130] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
  (niv=3)
  [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):131072]
  [SETTINGS_MAX_FRAME_SIZE(0x05):16384]
```

[1:65536 ; 4:131072 ; 5:16384]

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

```
[35556.054] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=0>
  (window_size_increment=15663105)
```

- Flow control element
- Window size can be set for entire connection or per stream
  - Connection – Initial size can be set in SETTINGS
  - RFC set default window sizes if not specified

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Chrome 60 – Android 8.0.0 Pixel XL

```
[35556.053] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
  (niv-3)
  [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
  [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):1000]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):6291456]
[35556.054] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=0>
  (window_size_increment=15663105)
```

[1:65536 ; 4:131072 ; 5:16384 | 15663105]

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Chrome 60 – Android 8.0.0 Pixel XL

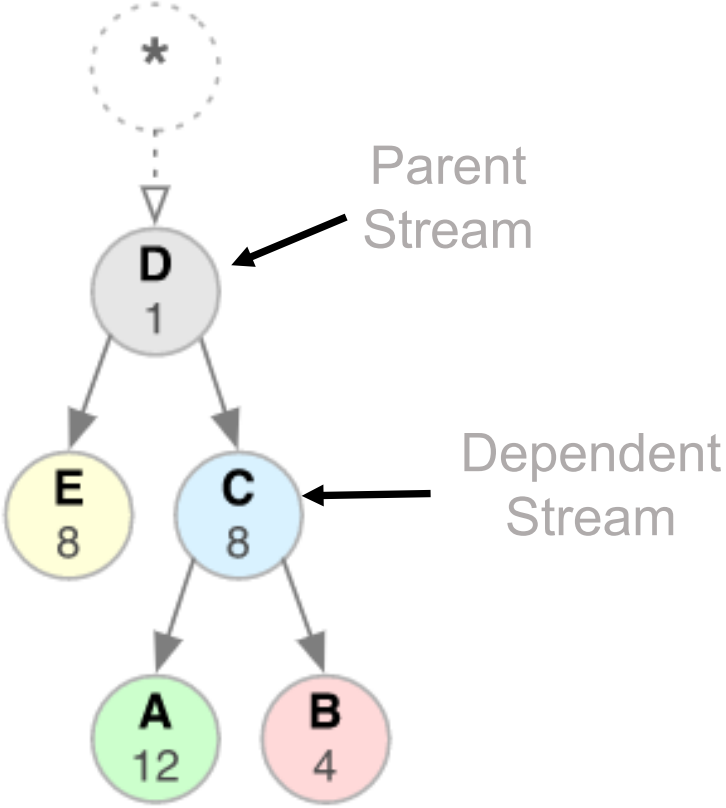
```
[35556.053] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
  (niv=3)
  [SETTINGS_HEADER_TABLE_SIZE(0x01):65536]
  [SETTINGS_MAX_CONCURRENT_STREAMS(0x03):1000]
  [SETTINGS_INITIAL_WINDOW_SIZE(0x04):6291456]
[35556.054] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=0>
  (window_size_increment=15663105)
```

[1:65536 ; 4:131072 ; 5:16384 | 15663105]

* If frame is not sent – use 0 instead

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

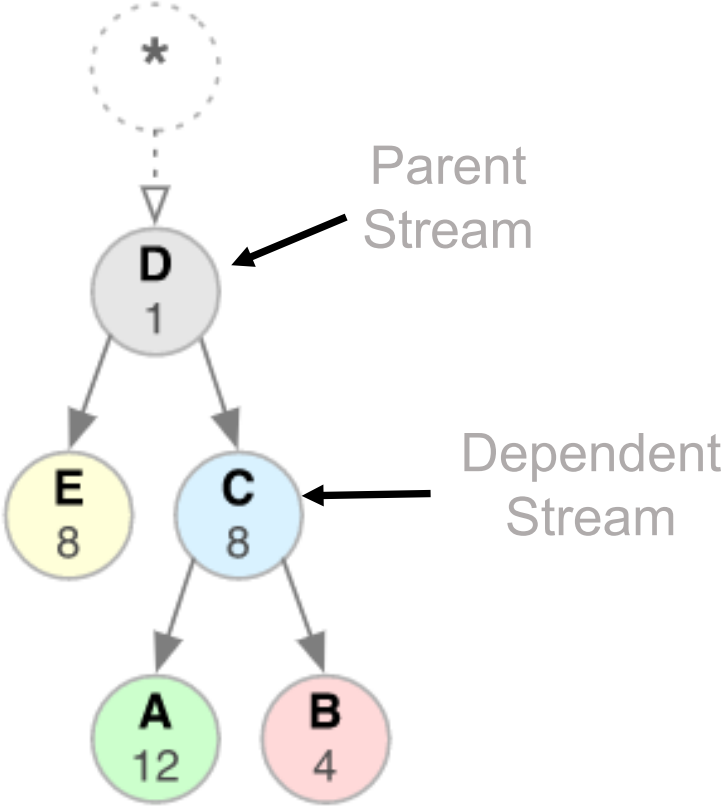


- Set stream dependencies and priorities
- Priority is set by assigning weights to streams
- Weights express preference of resources allocation
- No guarantees

"only a suggestion"

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

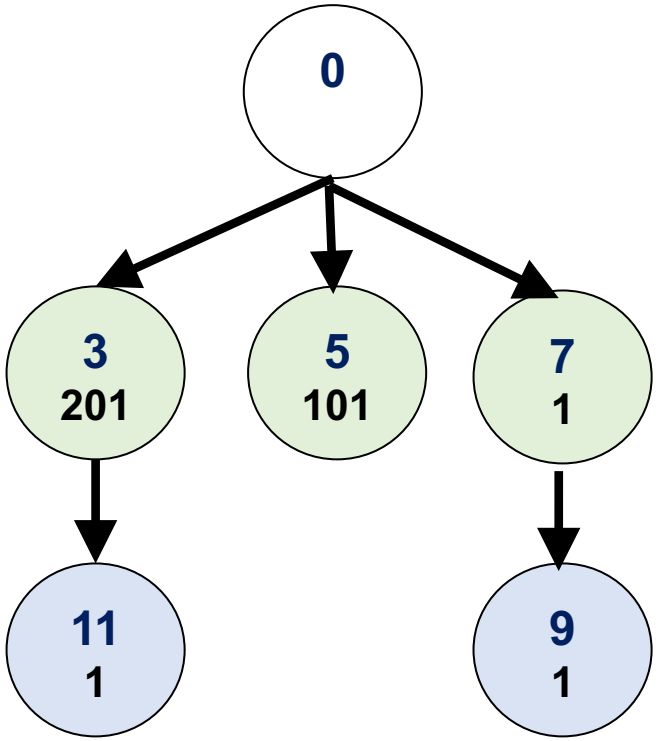


- Used by some at the beginning of each connection
- Each frame has three fields:
  - Weight
  - Stream Dependency
  - Exclusivity Bit

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Firefox/54.0



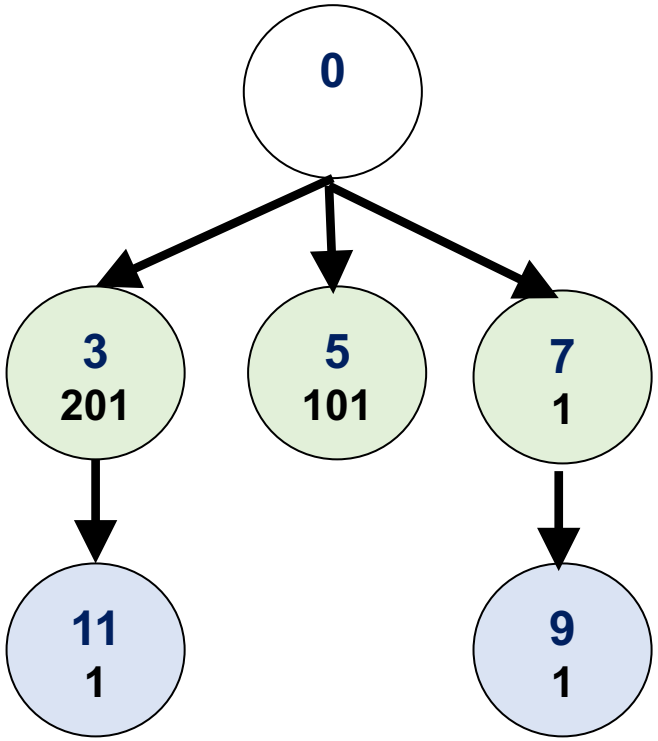

```
[39000.283] recv PRIORITY frame <length=5, flags=0x00, stream_id=3>
  (dep_stream_id=0, weight=201, exclusive=0)
[39000.283] recv PRIORITY frame <length=5, flags=0x00, stream_id=5>
  (dep_stream_id=0, weight=101, exclusive=0)
[39000.283] recv PRIORITY frame <length=5, flags=0x00, stream_id=7>
  (dep_stream_id=0, weight=1, exclusive=0)
[39000.283] recv PRIORITY frame <length=5, flags=0x00, stream_id=9>
  (dep_stream_id=7, weight=1, exclusive=0)
[39000.284] recv PRIORITY frame <length=5, flags=0x00, stream_id=11>
  (dep_stream_id=3, weight=1, exclusive=0)
[39000.284] recv PRIORITY frame <length=5, flags=0x00, stream_id=13>
  (dep_stream_id=0, weight=241, exclusive=0)
```

- Collect dependency, weight, exclusivity

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Firefox/54.0



❒ Http2Session.cpp

```
// The Hello is comprised of
// 1] 24 octets of magic, which are designed to
// flush out silent but broken intermediaries
// 2] a settings frame which sets a small flow control window for pushes
// 3] a window update frame which creates a large session flow control window
// 4] 5 priority frames for streams which will never be opened with headers
//    these streams (3, 5, 7, 9, b) build a dependency tree that all other
//    streams will be direct leaves of.
```

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

User-Agent	SETTINGS	WINDOW UPDATE	PRIORITY
Chrome 58.0 Mac OS X	1:65536 ; 3:1000 ; 4:6291456	15663105	0
okhttp/3.6.0	4:16777216	16711681	0
curl/7.54.0	3:100 ; 4:1073741824	1073676289	0
nghttp2/1.22.0	3:100 ; 4:65535	00	3:0:0:20,5:0:0:101, 7:0:0:1,9:0:7:1,11:0:3:1


ALMOST THERE...
STAY ON TARGET.
memegenerator.net

Nice.
But still…

not enough entropy

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Pseudo Headers

Request Pseudo Headers

- ❒ :method
- ❒ :scheme
- ❒ :authority
- ❒ :path

Response Pseudo Headers

- ❒ :status

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

- **HTTP/1.1 Request**

  GET / HTTP/1.1

  Host: www.example.com

  User-Agent: Mozilla/5.0

  Accept: text/html

- **HTTP/2 Request**

  :method: GET

  :path: /

  :authority: www.example.com

  :scheme: https

  User-Agent: Mozilla/5.0

  Accept: text/html

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Client / Implementation	Pseudo Headers Name Order
Google Chrome (58.0.3029.110 on Mac OS X)	`:method, :authority, :scheme, :path`
Firefox v53.0 (Mac OS X)	`:method, :path, :authority, :scheme`
Safari v10.1 (Mac OS X)	`:method, :scheme, :path, :authority`
Curl v7.54.0 (Mac OS X)	`:method, :path, :scheme, :authority`
Go-http-client v2.0	`:authority, :method, :path, :scheme`
Jetty HTTP2 Client v9.3.4.v20151007	`:scheme, :method, :authority, :path`

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

Example from Chrome's source code:

```
void CreateSpdyHeadersFromHttpRequest(const HttpRequestInfo& info,
                                      const HttpRequestHeaders& request_headers,
                                      bool direct,
                                      SpdyHeaderBlock* headers) {
  (*headers)[":method"] = info.method;
  if (info.method == "CONNECT") {
    (*headers)[":authority"] = GetHostAndPort(info.url);
  } else {
    (*headers)[":authority"] = GetHostAndOptionalPort(info.url);
    (*headers)[":scheme"] = info.url.scheme();
    (*headers)[":path"] = info.url.PathForRequest();
  }
```

Proposed Fingerprint

SETTINGS[;] | WINDOW_UPDATE | PRIORITY[,] | PSH-Order

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:53.0) Gecko/20100101 Firefox/53.0

HTTP/2 fingerprint:

`1:65536;4:131072;5:16384|12517377|3:0:0:201,5:0:0:101,7:0:0:1,9:0:7:1,11:0:3:1|m,p,a,s`

SETTINGS | **Window Update** | **PRIORITY** | **Pseudo Header Order**

USE CASES

Use Cases

- Positive Security
- Detect Browser Impersonators
- Tool Detection
- Anonymous Proxy / VPN Detection

* Fingerprinting should also combine other layers

black hat EUROPE 2017

HTTP/2 THREAT LANDSCAPE

Most security tools lack H2 support:

- ✘ Burp Suite
- ✘ Zed Attack Proxy
- ✘ Fiddler
- ✘ SQLmap
- ✘ Acunetix
- ✘ AppScan
- ✘ NetSparker
- ✘ SentryMBA
- ✘ THC-Hydra

Why ?

- Not enough incentive for Attackers
  - Web servers support both HTTP/1.X and HTTP/2
  - HTTP/2 libraries are not common
  - Cost exceeds the Gain
- Server Implementation Weaknesses found in 2016
  - Handling of Compression, Stream management

Key Takeaways

- Basic understanding of how HTTP/2 works
- Key differences between HTTP 1.x and 2.0
- Passive Fingerprinting
- Proposed fingerprint mechanism and Use Cases
- (Lack of) Threat Landscape

Questions

THANK YOU

Elad Shuster

eshuster@akamai.com