Slow rate denial of service attacks against HTTP/2 and detection

Critical review

Bastien Dhiver

February 22, 2018

bfrd2@kent.ac.uk

Table of contents

Authors

Authors

Indian Institute of Technology Indore, India

Nikhil Tripathi

- Ph.D. student
- Network Security, Computer Networks
- Slow rate DoS attacks

Neminath Hubballi

- Assistant professor
- Nikhil's supervisor
- Network Security, System Security
- Worked at HP, Infosys Lab, Samsung R&D

Background

(Distributed) Denial-of-Service attacks

Transport-level DDoS flooding attacks

- Exhausting network bandwidth
- Consumes excess amount of victim's ressources
- Exploiting implementation bugs of transport layer
- Reflection and amplification (ICMP Echo, Smurf)
- Require a lot of malicious client's bandwidth
- Can be easily detected

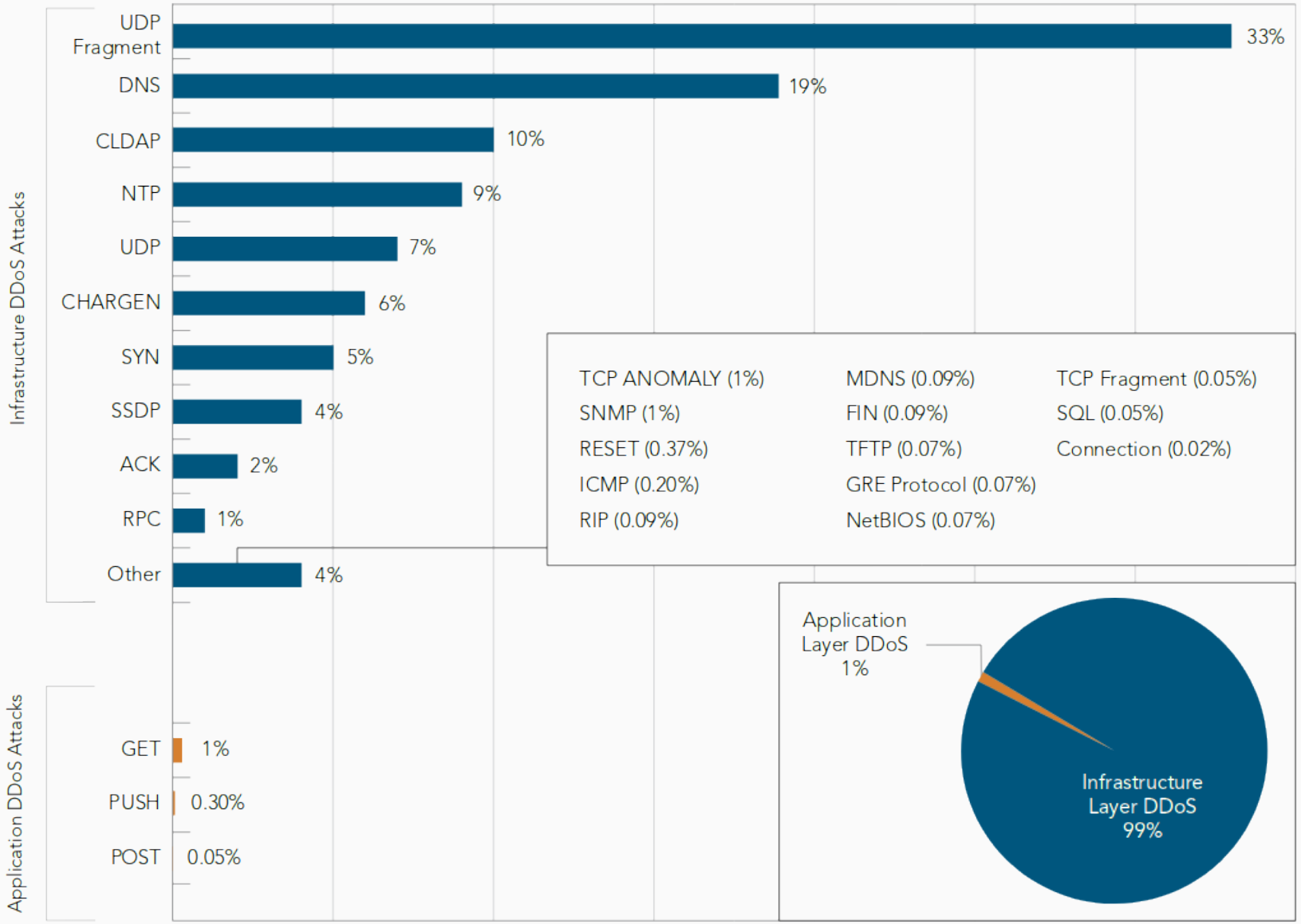

Figure 1: DDoS Attack Vector Frequency, Q4 2017 from Akamai[1]

[1] https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q4-2017-state-of-the-internet-security-report.pdf

Application-level DDoS flooding attacks

- Exhausting target ressources
- Less bandwidth, stealthier
- Reflection (VoIP), Amplification (DNS), Protocol Specific (HTTP flooding)
- Complete requests at a very big rate
- Harder to distinguish from normal traffic

Application-specific slow rate DoS attacks

- Very small of incomplete requests
- Interacts very slowly with the server
- Minimal bandwidth
- Highly stealth
- HTTP/1.1 Slowloris attack
- The proposed attacks belong to this category

Application layer protocol independent slow rate DoS attacks

Meta attacks (FTP, SMTP, HTTP, ...)

SlowReq and SlowConn

- Incomplete and pending requests
- Detect connection closes
- Re-establish

SlowNext

- Valid and legitimate requests
- Maintaining established connection (keep alive)
- Stealth ++

Background

HTTP/2

HTTP/2 in ~one slide

- RFC 7540, May 2015
- Binary protocol
- Efficient use of TCP (one connection)
- Message multiplexing (frames, streams)
- Prediction of ressource requirement (PUSH)
- Header compression (HPACK)
- TLS as a De Facto requirement

HTTP/2 frames

Connection Preface

Initial settings for a HTTP/2 connection

WINDOW_UPDATE

Number of bytes that the sender is willing to accept

GOAWAY

- Indicate to tear off an established connection
- Carry an error code

HEADERS and CONTINUATION

Transmit the headers

DATA

Carry message body

SETTINGS

Negotiate connection parameters

- Initial window size
- Max concurent streams

Multiplexing

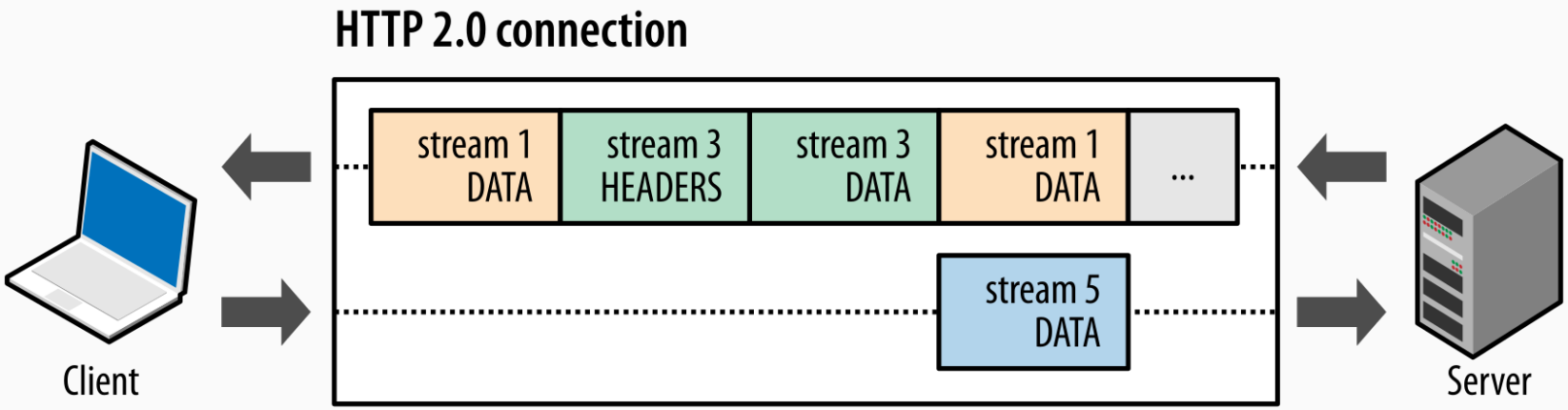

Figure 2: HTTP/2 request and response multiplexing within a shared connection [2]

HTTP/1.1 HoL (Head-of-line) blocking solved

In real life?

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.17.0.1	172.17.0.2	TCP	74	44158 → 443 [SYN] Seq=0 Win=2
2	0.000020060	172.17.0.2	172.17.0.1	TCP	74	443 → 44158 [SYN, ACK] Seq=0
3	0.000060896	172.17.0.1	172.17.0.2	TCP	66	44158 → 443 [ACK] Seq=1 Ack=1
4	0.005162759	172.17.0.1	172.17.0.2	TLSv1.2	260	Client Hello
5	0.005173770	172.17.0.2	172.17.0.1	TCP	66	443 → 44158 [ACK] Seq=1 Ack=1
6	0.009652765	172.17.0.2	172.17.0.1	TLSv1.2	2068	Server Hello, Certificate, Se
7	0.009663343	172.17.0.1	172.17.0.2	TCP	66	44158 → 443 [ACK] Seq=195 Ack
8	0.010059132	172.17.0.1	172.17.0.2	TLSv1.2	192	Client Key Exchange, Change C
9	0.010234476	172.17.0.2	172.17.0.1	TLSv1.2	117	Change Cipher Spec, Finished
10	0.010272763	172.17.0.2	172.17.0.1	HTTP2	135	SETTINGS, WINDOW_UPDATE
11	0.010328702	172.17.0.1	172.17.0.2	HTTP2	119	Magic
12	0.010345182	172.17.0.1	172.17.0.2	HTTP2	122	SETTINGS
13	0.010356067	172.17.0.1	172.17.0.2	HTTP2	108	WINDOW_UPDATE
14	0.010357516	172.17.0.2	172.17.0.1	TCP	66	443 → 44158 [ACK] Seq=2123 Ac
15	0.010378007	172.17.0.2	172.17.0.1	HTTP2	104	SETTINGS
16	0.010381360	172.17.0.1	172.17.0.2	HTTP2	131	HEADERS
17	0.010409364	172.17.0.1	172.17.0.2	HTTP2	104	SETTINGS
18	0.010485333	172.17.0.2	172.17.0.1	HTTP2	841	HEADERS, DATA
19	0.010649587	172.17.0.1	172.17.0.2	TLSv1.2	97	Alert (Level: Warning, Descri
20	0.010717403	172.17.0.2	172.17.0.1	TCP	66	443 → 44158 [FIN, ACK] Seq=29
21	0.011092971	172.17.0.1	172.17.0.2	TCP	66	44158 → 443 [FIN, ACK] Seq=60
22	0.011098358	172.17.0.2	172.17.0.1	TCP	66	443 → 44158 [ACK] Seq=2937 Ac

Figure 3: Decrypted HTTP/2 exchange between Nginx and curl displayed in Wireshark

The client IP address is 172.17.0.1

The paper [1]

Proposed attacks

Proposed attacks

- Five novel Slow Rate HTTP/2 DoS attacks
- Number of **free connections slots** available is targeted
- **Hold back** established connections for a long duration
- Tested on four popular web servers

Apache, Nginx, H2O and Nghttp2

Attack №1

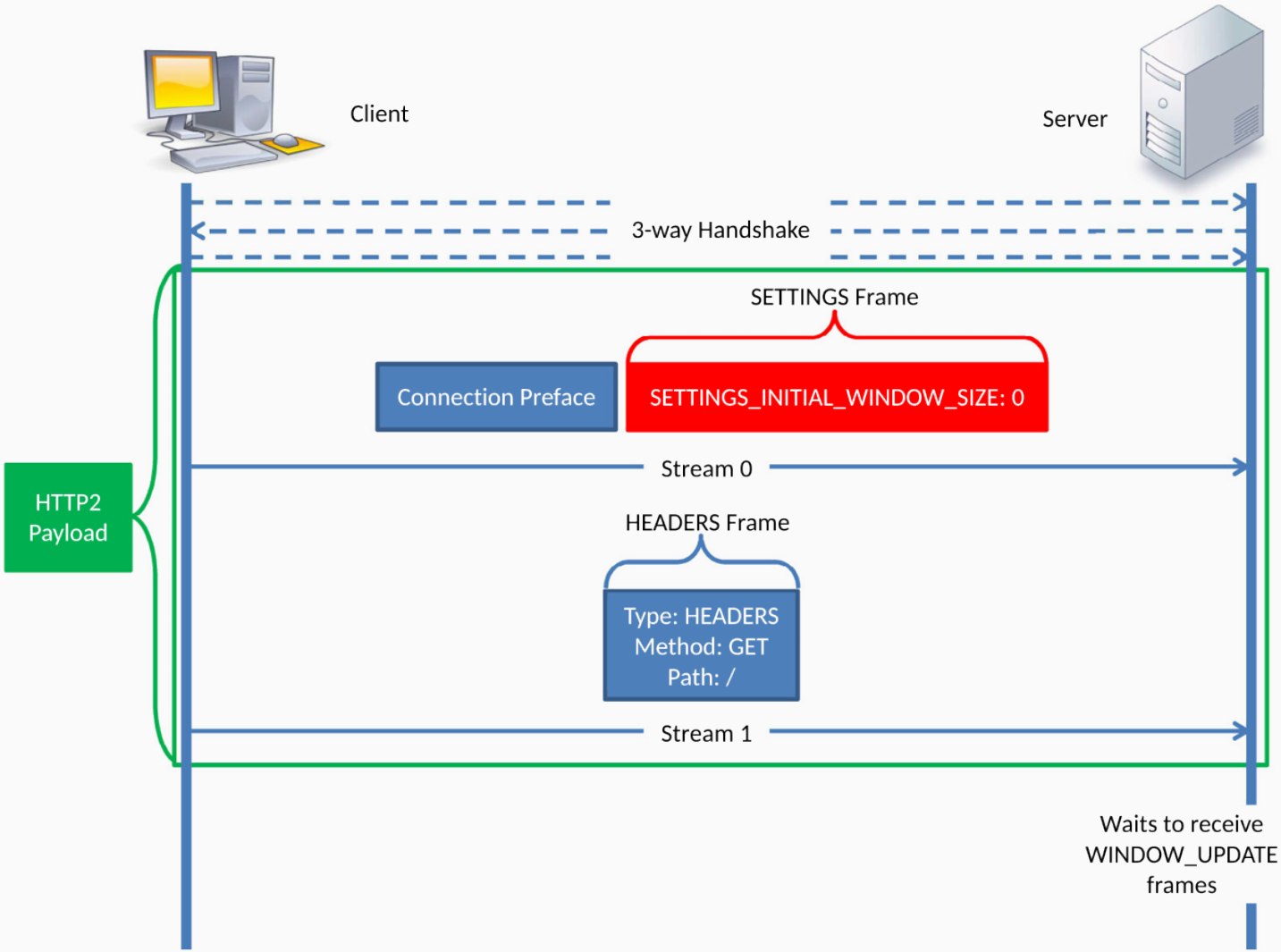

Figure 4: Attack-1. SETTINGS frame with INITIAL_WINDOW_SIZE set to 0

Attack №2

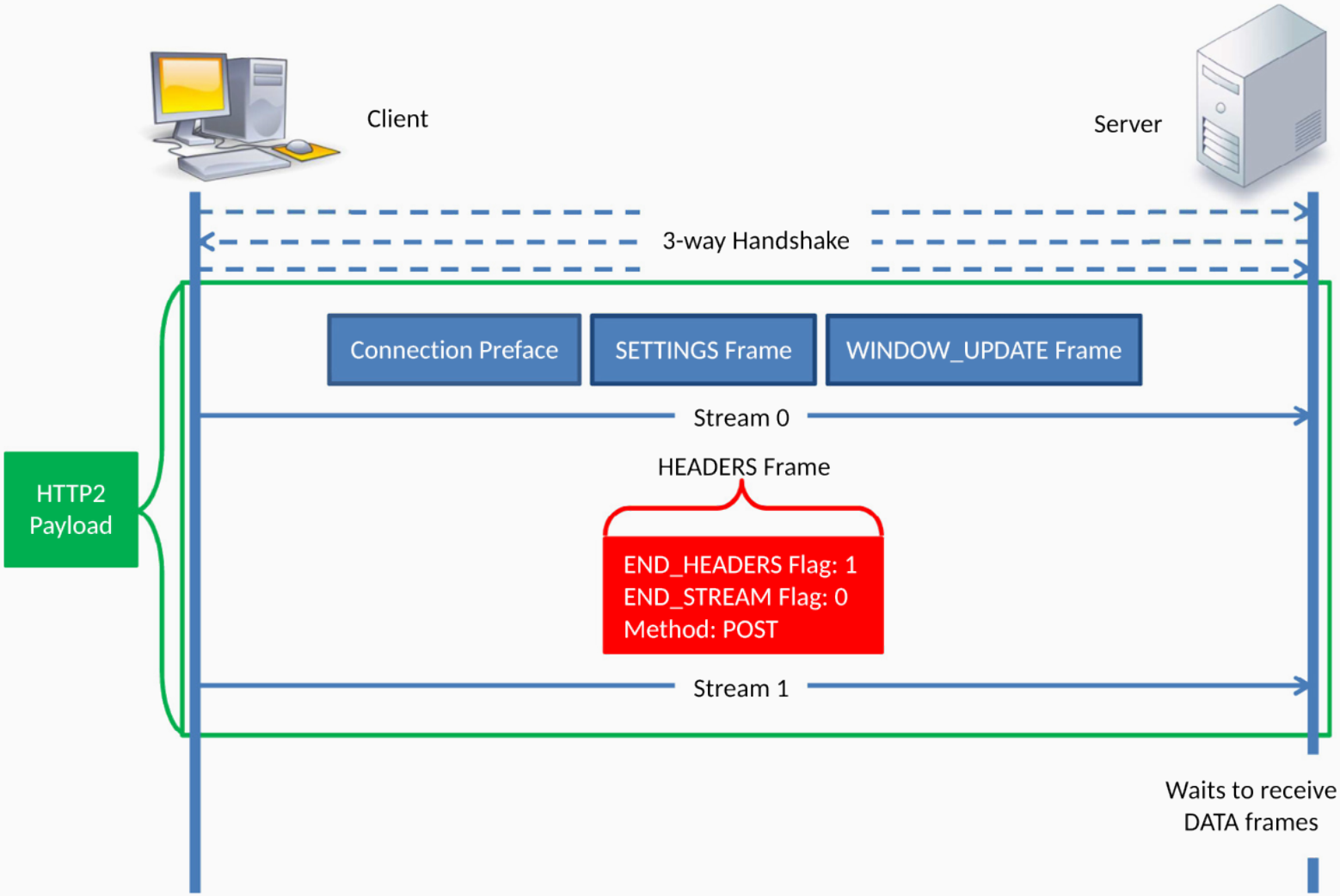

Figure 5: Attack-2. HEADERS frame with END_HEADERS set and END_STREAM reset

Attack №3

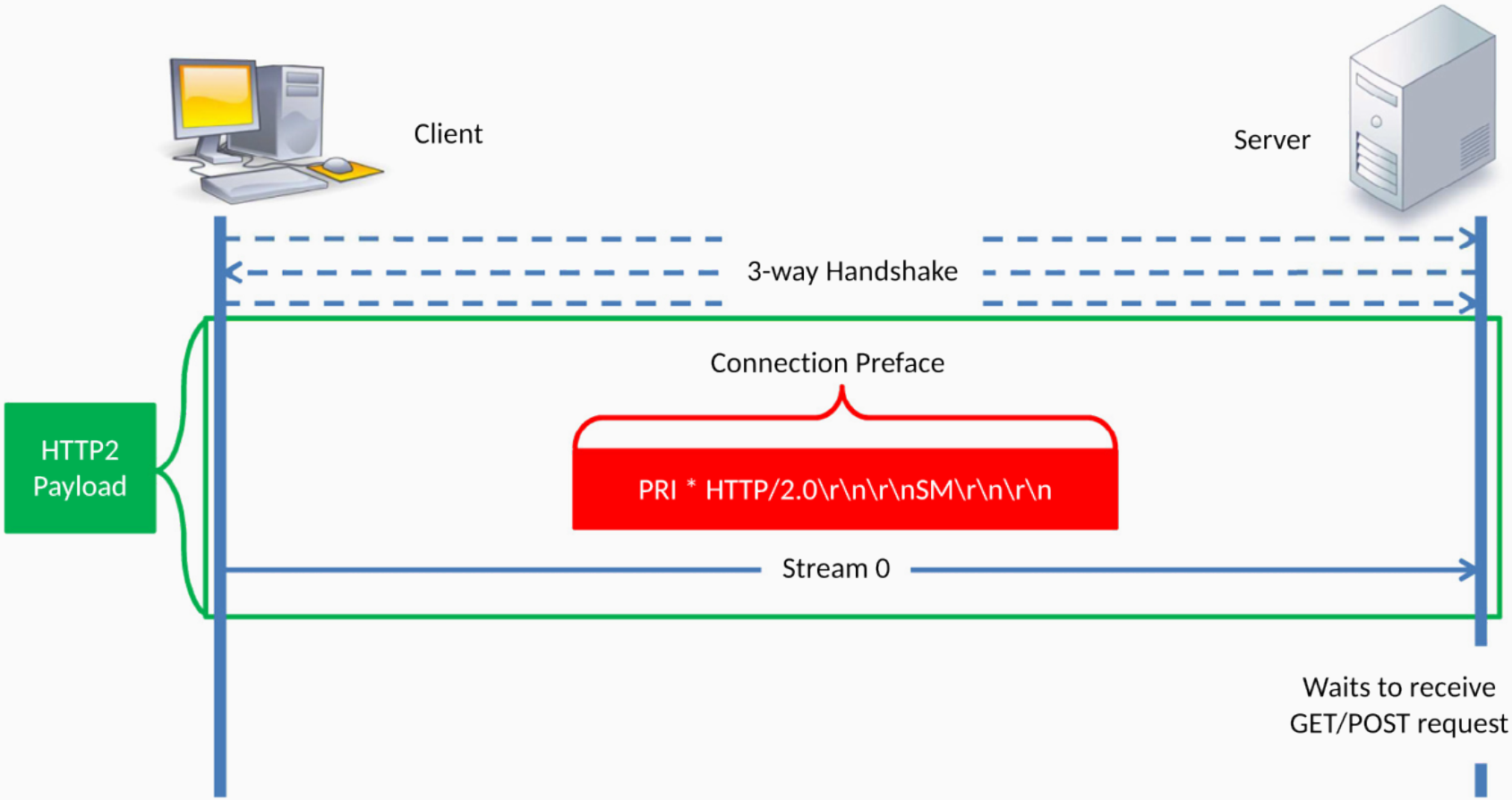

Figure 6: Attack-3. First HTTP/2 payload with only Connection Preface

Effects

Table 1: Connection waiting time in seconds and № at servers for the five attacks

Server	A1	A2	A3	A4	A5	№ conn
Apache	300	600-∞	300-300	300-∞	5-5	150
Nginx	60	30-∞	30-∞	90-90	180-180	2060
H2O	∞	10-∞	10-10	10-∞	10-10	1024
Nghttp2	60	10-975	10-975	10-60	10-975	1142

Same effects over TLS

The paper [1]

Proposed detection mechanism

Chi-square test

Distance measurement technique

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Figure 7: Chi-square equation

n: Number of categories
O_i: Observed cases in i_{th} category
E_i: Expected cases in i_{th} category
i: number of the category

To be set

- Feature selection
- Significance level α

How does it work?

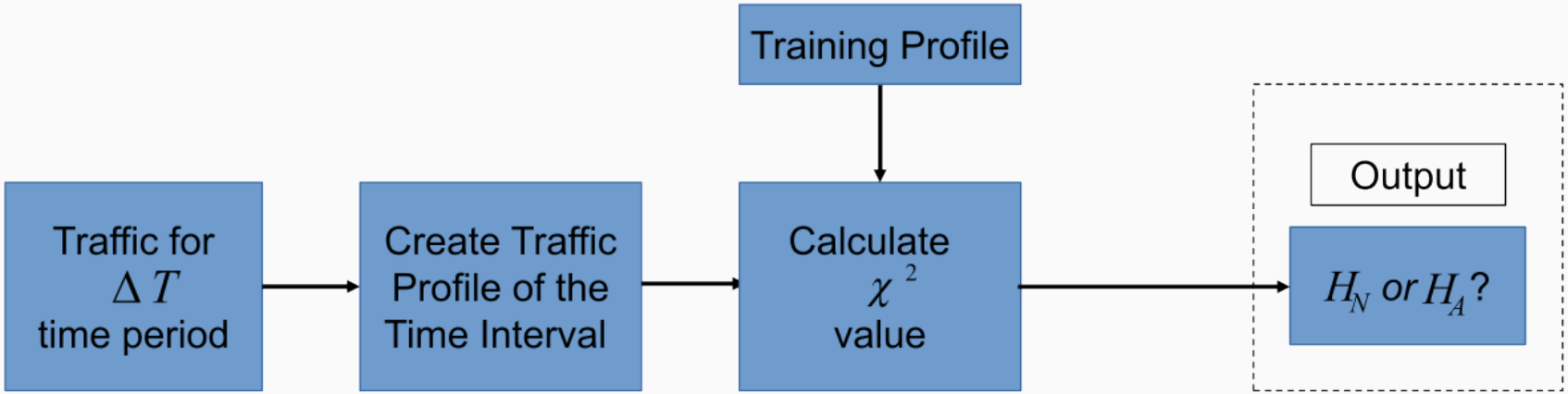

Figure 8: Detector working

Training and testing phases

Detection performance

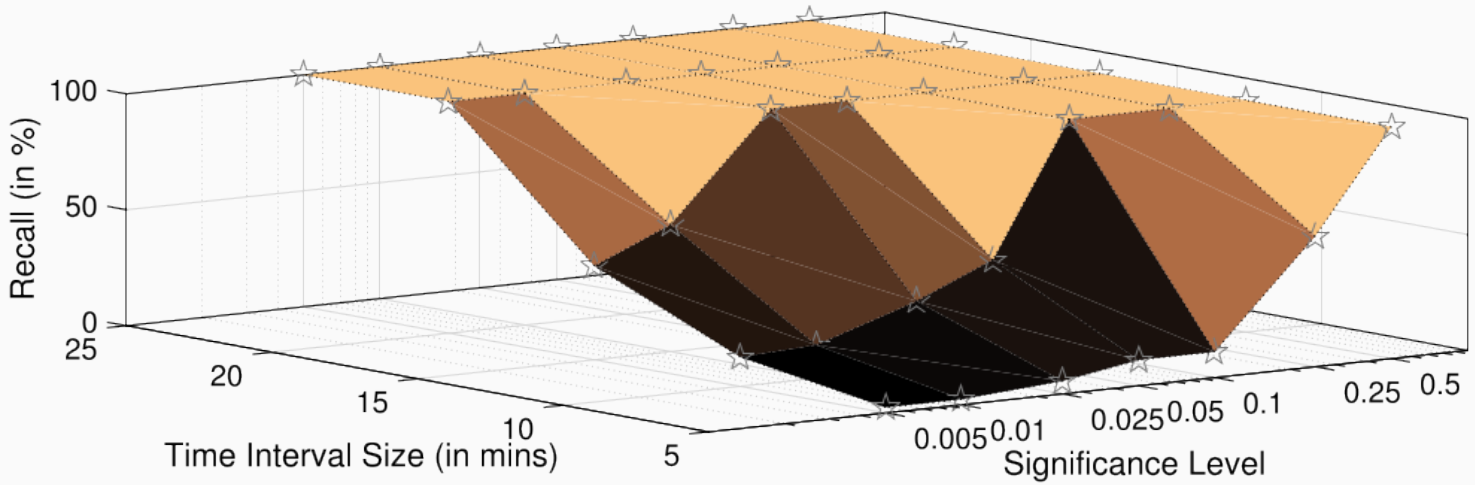

Figure 9: Recall rate

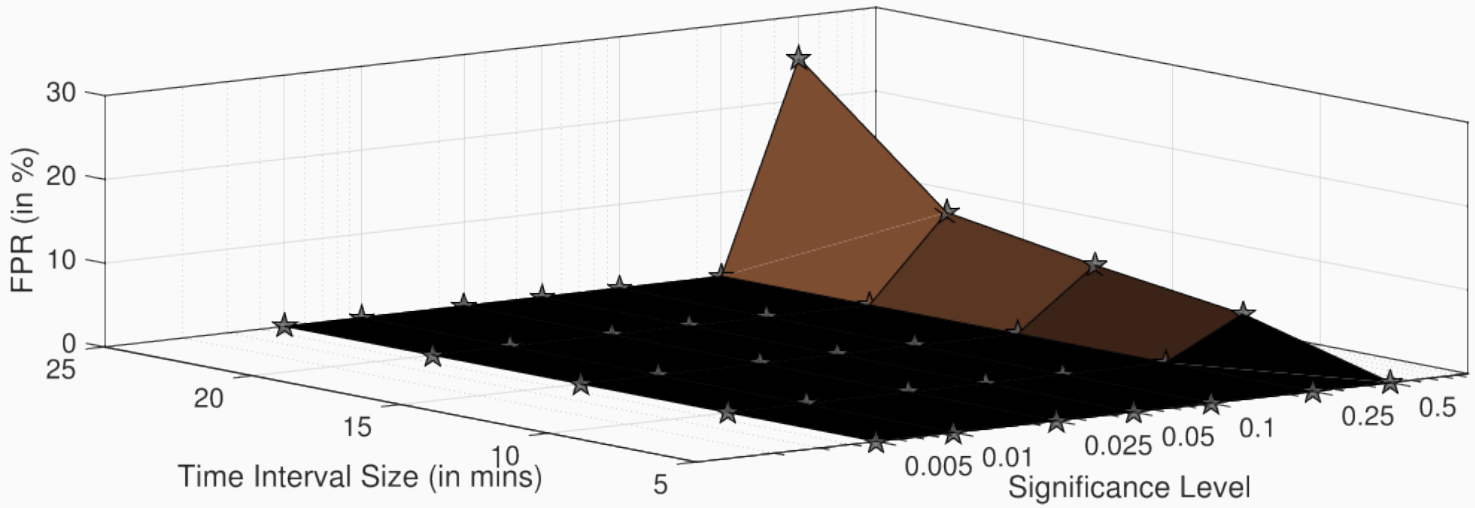

Figure 10: False positive rate

Prior work

Prior work

Vulnerabilities in HTTP/2 protocol

- Two papers by Adi and al [3] [4]
- Flood attack, reduction of Quality (RoQ) only
- Impreva report
- Slow Read, HPACK, Dependency DoS, Stream abuse
  $\Rightarrow$patched

Anomalies in encrypted network traffic

- Observing inter-packet arrival time, time gaps, ...

Chi-square test

- Used to detect intrusion, port scan, bot servers (C2)

Applicability of research

Applicability of research

- Submitted in 2017, published in 2018 (ACM)
- HTTP/2 deployement ↗
- Real web servers used
- Real effects shown
- Credible testbed

Criticisms

Proof-of-Concept

- ”we implemented these attacks in python”...
- I implemented them
- Published soon
- Serveral differences with the authors’ results

```
[me@pc hyper-h2]$ SSLKEYLOGFILE=/tmp/keylogfile.txt curl -i -k --connect-timeout 4 https://172.17.0.2/
curl: (28) Operation timed out after 4001 milliseconds with 0 out of 0 bytes received
```

Figure 11: An effective attack caused curl connection timeout

"Denial-of-service attack modelling and detection for HTTP/2 services" [5]

- Four novel attacks
- Flood attacks
- Four machine learning techniques used (suitable for detection)
- Features used: number of connections, flow information, ...

Personnal opinion

- Where are the Proofs-of-Concept??
- Examples shown with HTTP/2 plaintext protocol $\neq$ real life
- Only chi-square statistical test is mentionned
- Very specific to HTTP/2
- Good idea to focus on the number of available slot connection

Thank you!

Thank you! 

Attack №4

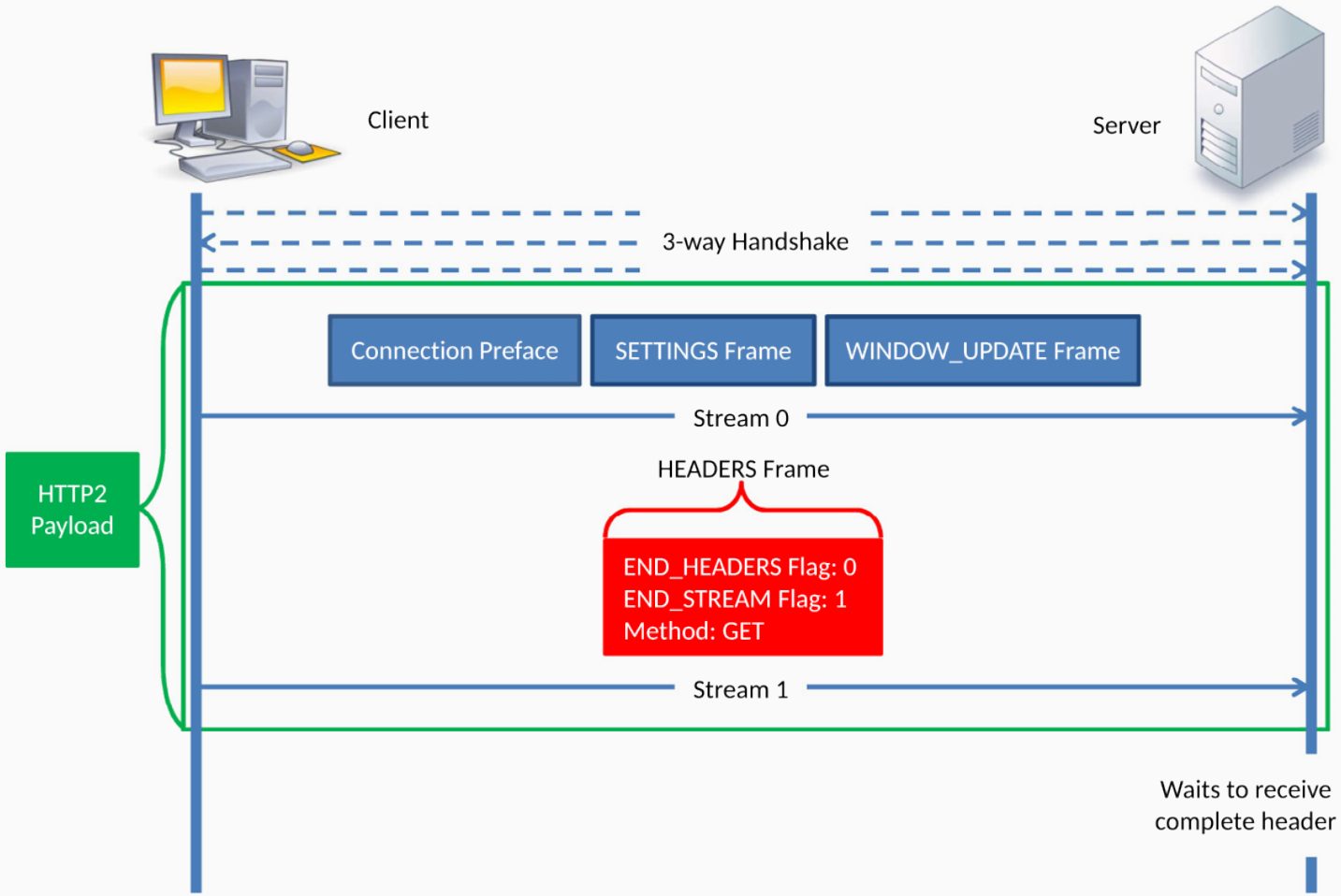

Figure 12: Attack-4. HEADERS frame with END_HEADERS reset and END_STREAM set

Attack №5

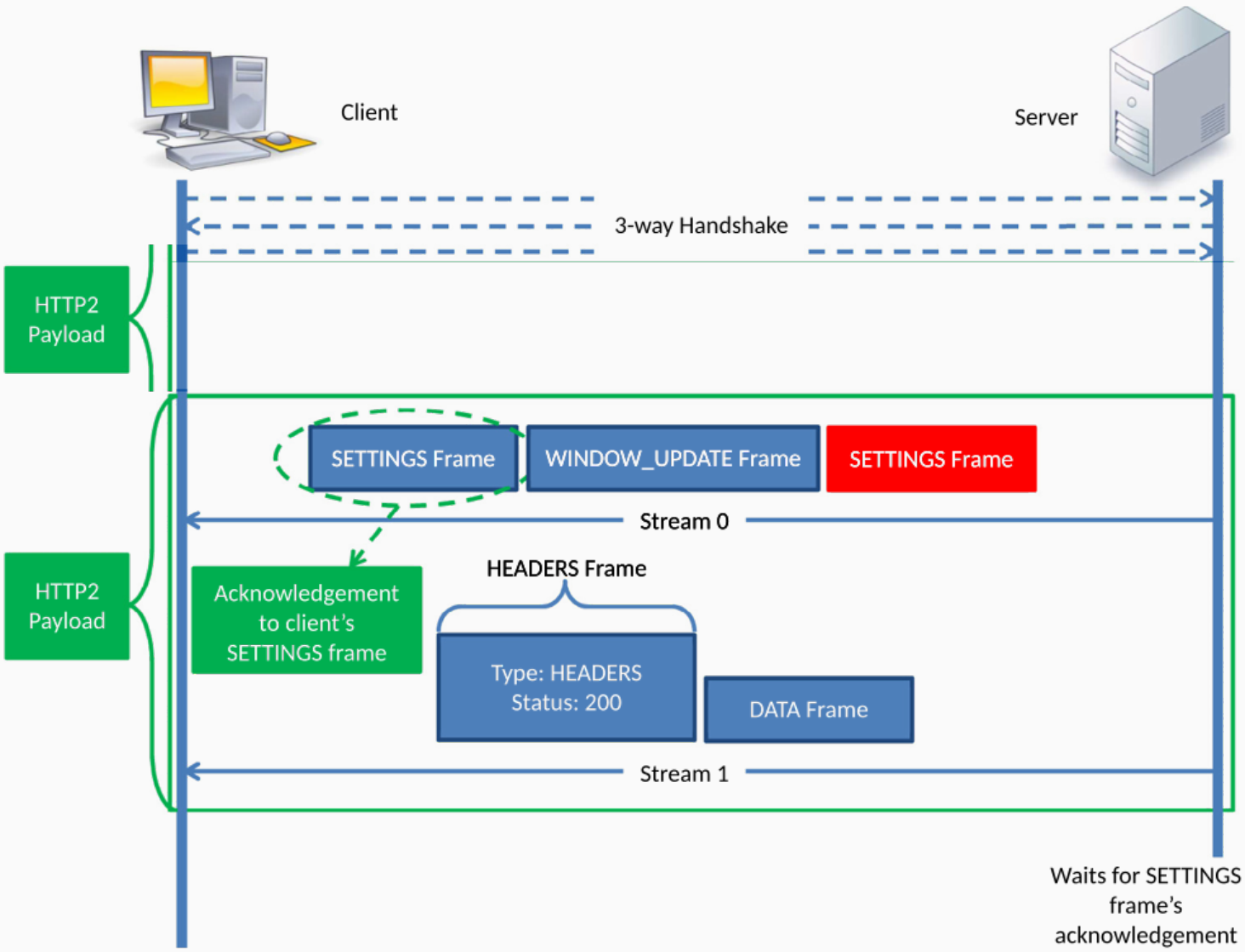

Figure 13: Attack-5. Client never acknowledges SETTINGS frame sent by server

Web servers usage

Accessed: 21 Febuary 2018

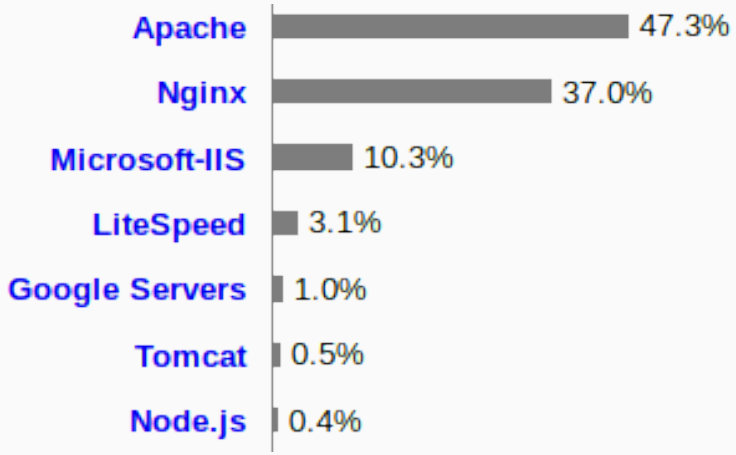

Figure 14: Market share of active sites by W3Techs.com

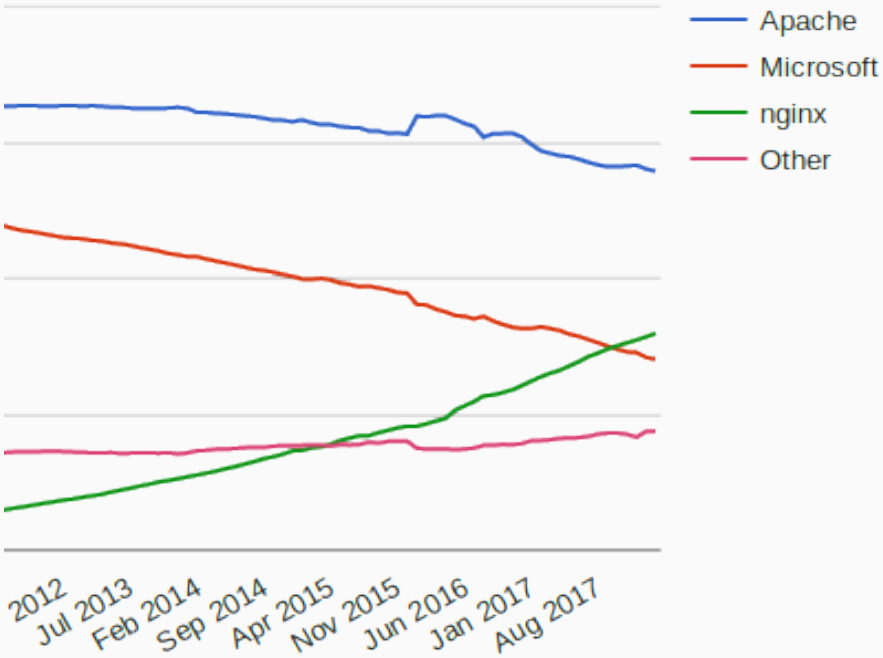

Figure 15: Percentages of websites using various web servers by Netcraft

HTTP/2 websites adoption

Accessed: 21 Febuary 2018

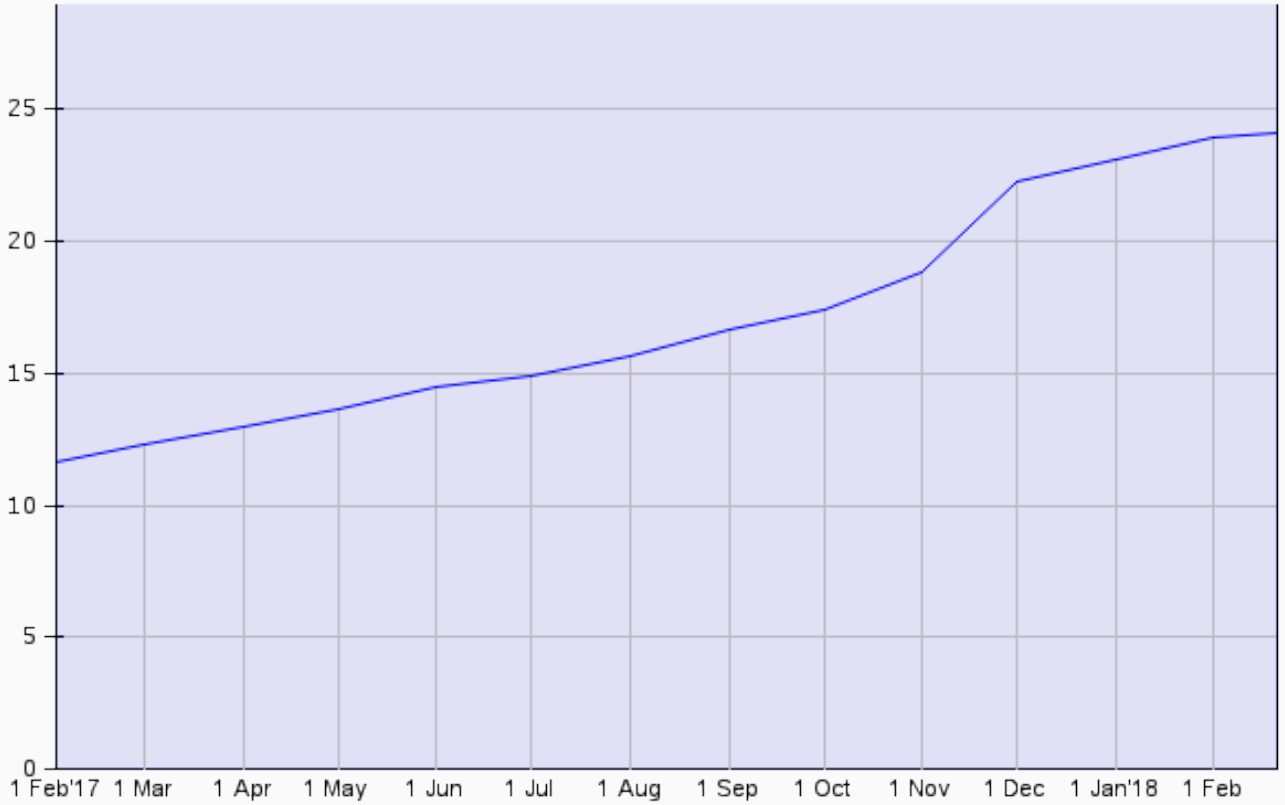

Figure 16: Usage of HTTP/2 for websites by W3Techs.com

Can I use HTTP/2?

Accessed: 21 Febuary 2018

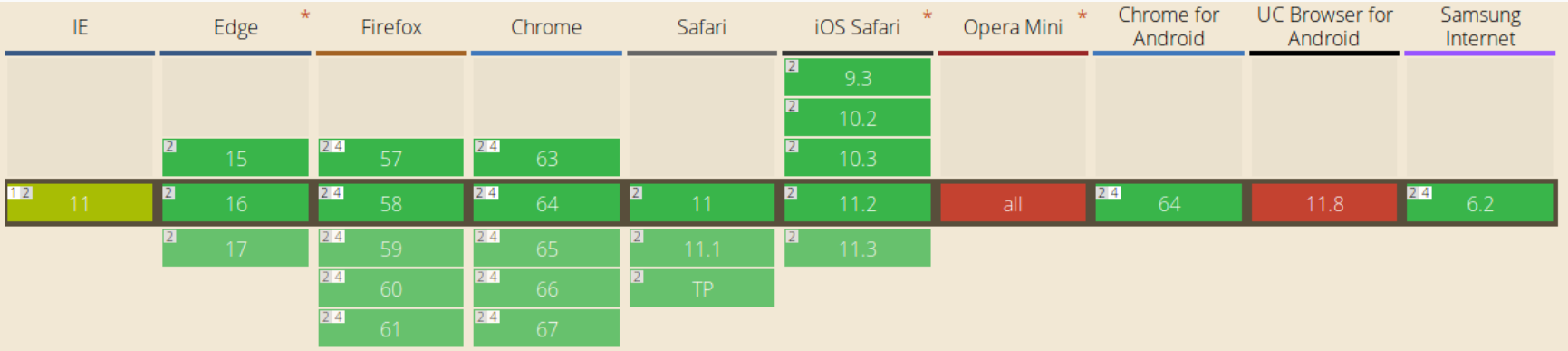

Figure 17: HTTP/2 capable clients by caniuse.com

References i

Nikhil Tripathi and Neminath Hubballi.
Slow rate denial of service attacks against HTTP/2 and detection.
Computers and Security, 72:255 – 272, 2018.

I. Grigorik.
High Performance Browser Networking: What every web developer should know about networking and browser performance.
O'Reilly Media, Incorporated, 2013.

References ii

E. Adi, Z. Baig, C. P. Lam, and P. Hingston.
Low-rate denial-of-service attacks against http/2 services.

In *2015 5th International Conference on IT Convergence and Security (ICITCS)*, pages 1–5, Aug 2015.

Erwin Adi, Zubair A. Baig, Philip Hingston, and Chiou-Peng Lam.
Distributed denial-of-service attacks against http/2 services.
Cluster Computing, 19(1):79–86, March 2016.

References iii

Erwin Adi.
Denial-of-service attack modelling and detection for HTTP/2 services.
PhD thesis, Edith Cowan University, 2017 (accessed Febuary 21, 2017).