Network Detection

Implementing and Analysis of JA3 data for TLS Fingerprinting using Bro/Zeek Data

By: Ahmed Techini,
M.Eng., Eng., CISSP, CEH, CCNA

Whoami

- Security Solution Architect at Bank of Canada
- Teaching Cyber Security at École Polytechnique of Montréal

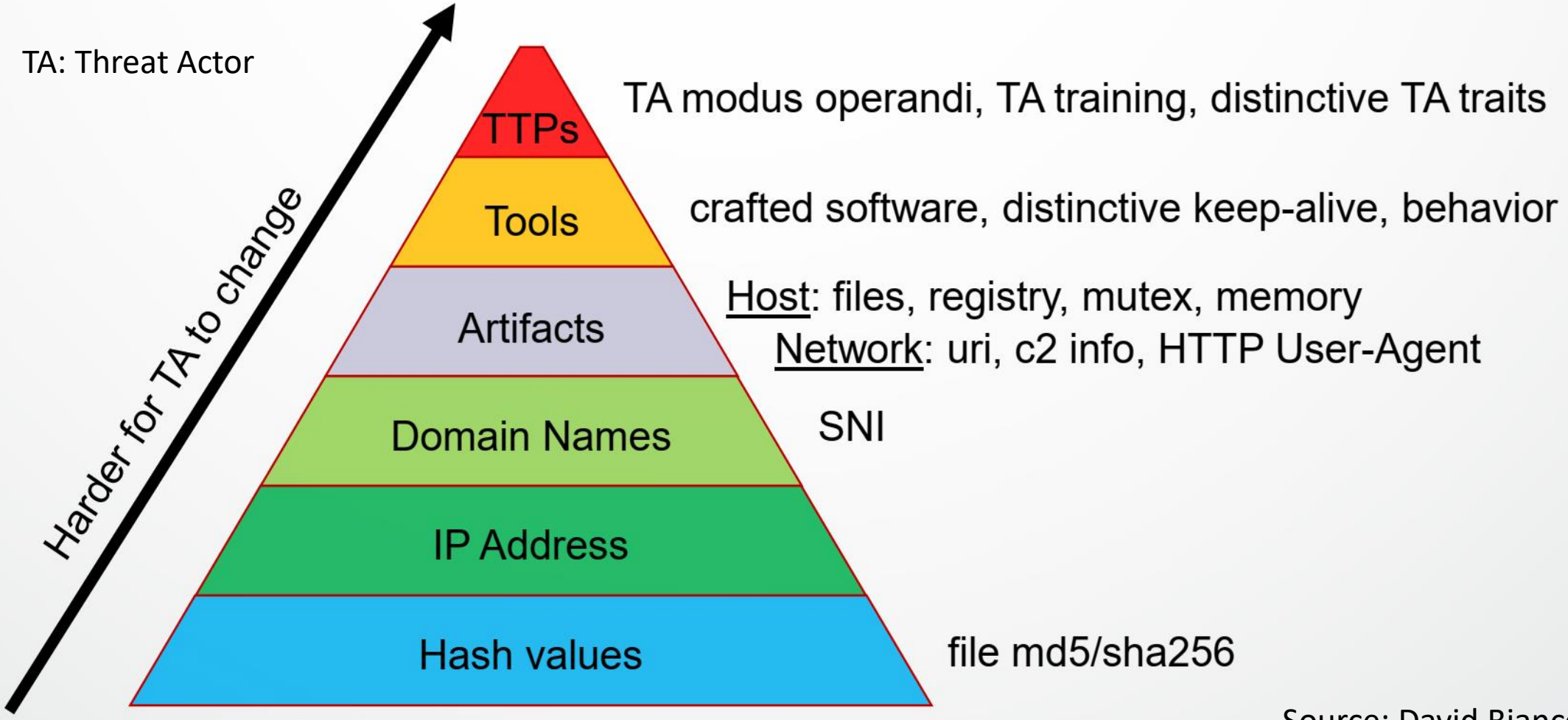
Problem Statement

Lack of visibility for encrypted traffic.

Effectiveness of network detection approach.

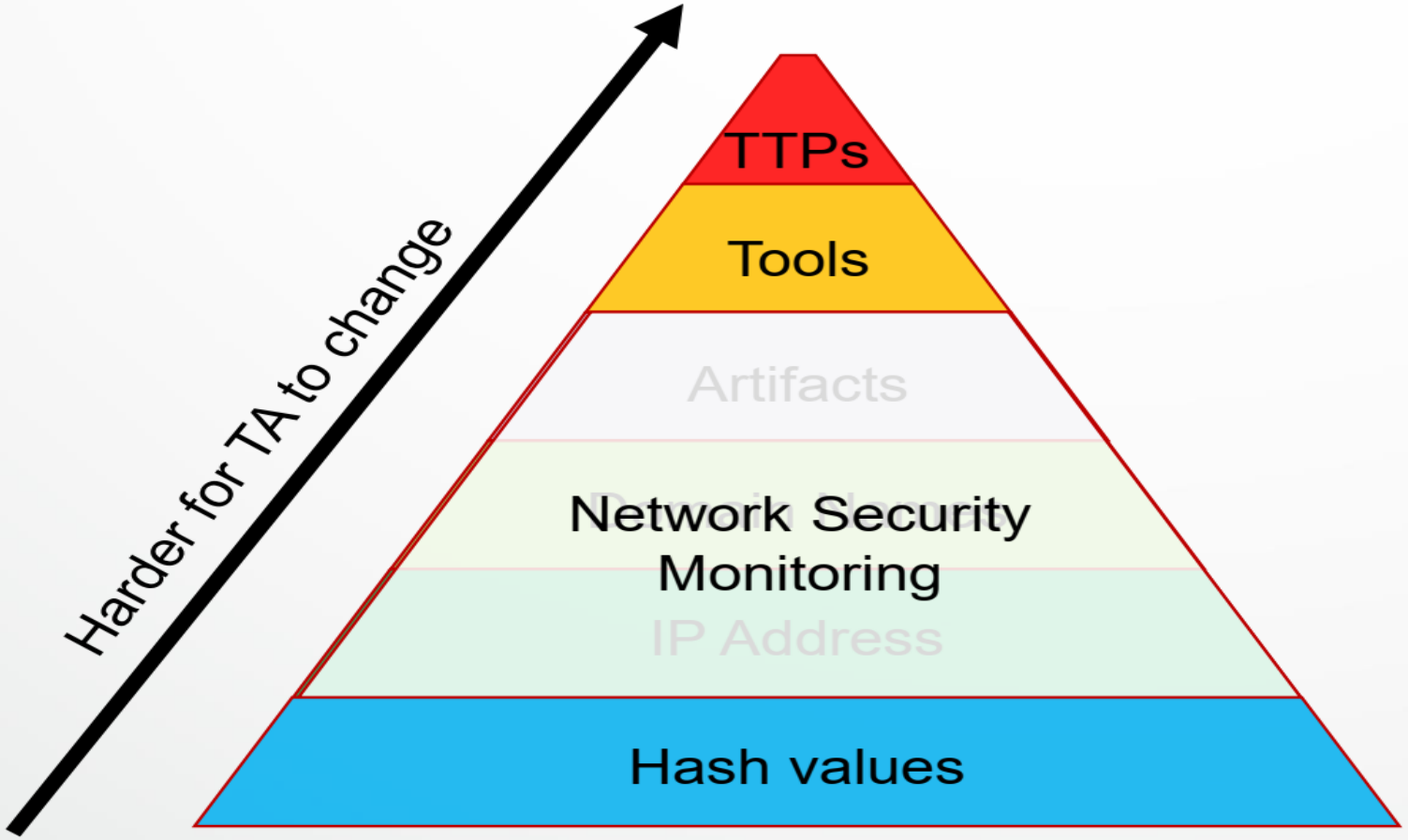
No correlation between host telemetry, network telemetry ,etc.

Pyramid of Pain: Effectiveness Measure



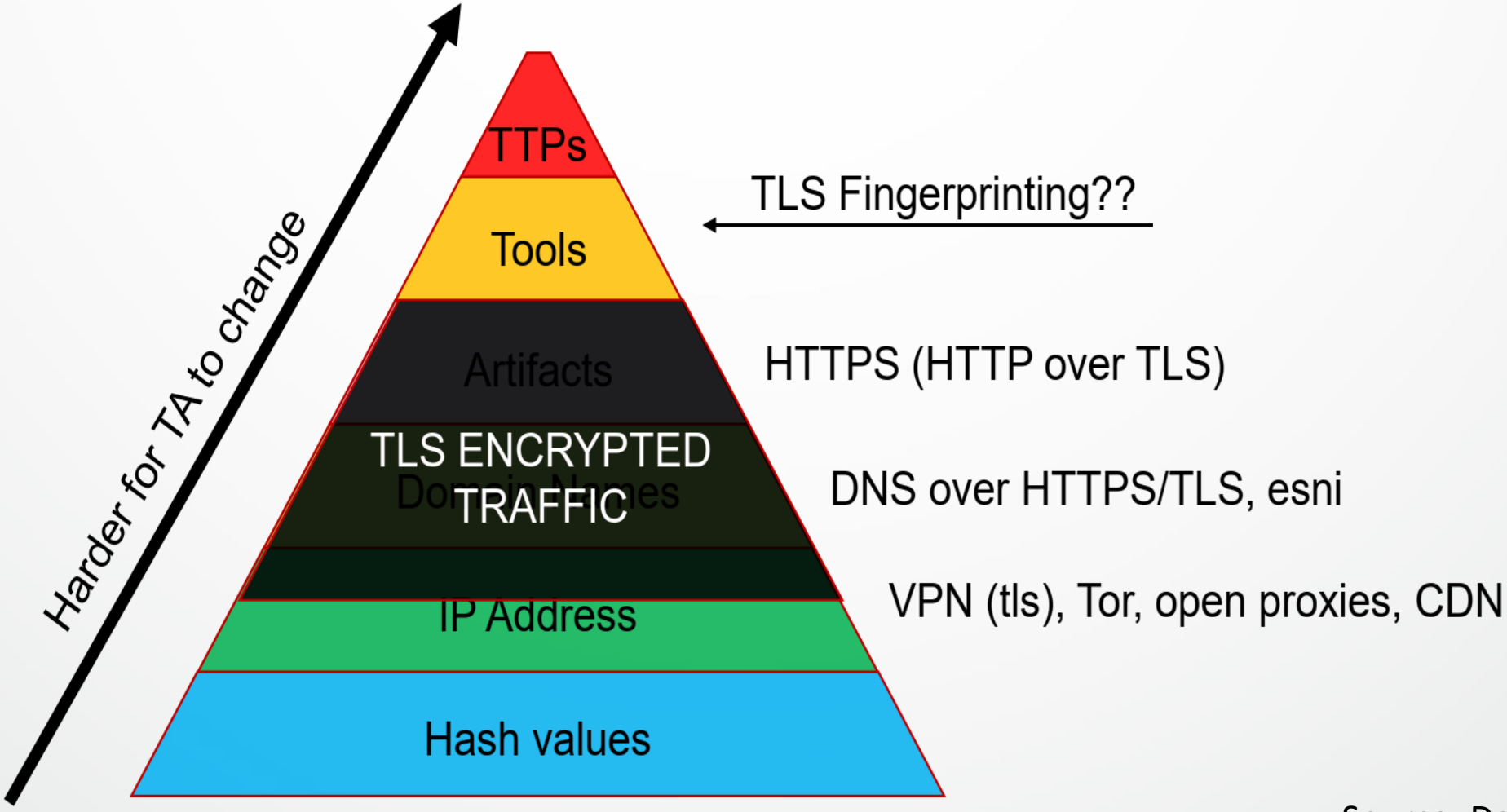
Source: David Bianco

Pyramid of Pain: NSM Effectiveness



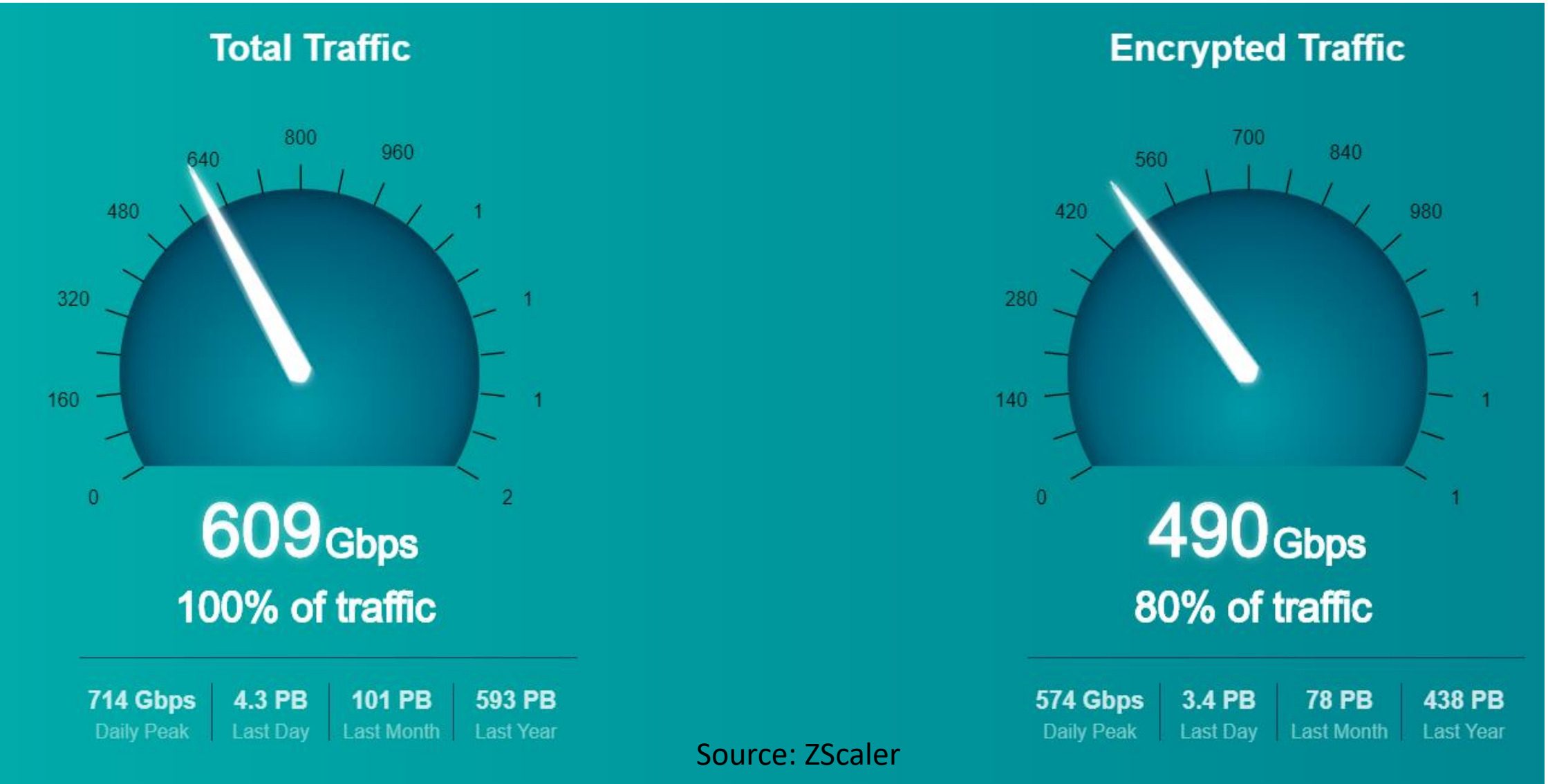
Source: David Bianco

Pyramid of Pain: Encrypted Traffic



Source: David Bianco

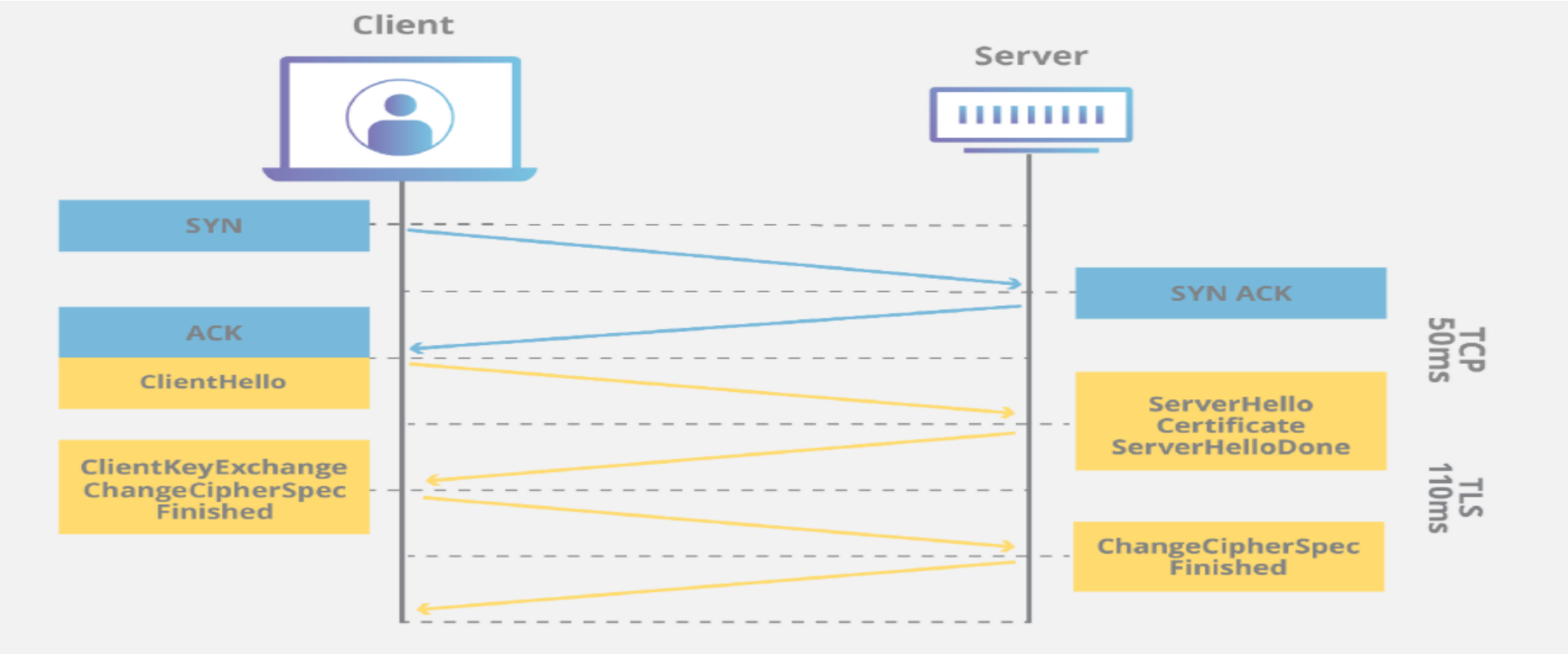
Encrypted traffic: The rise of TLS (~50%-90%)



Source: ZScaler

TLS Handshake

- TLS is an encryption protocol designed to secure Internet communications. A TLS handshake is the process that kicks off a communication session that uses TLS encryption.



TLS Fingerprinting: Is it new?

"A technique to identify a client application or a library based on parameters in the TLS traffic without decryption". [1]

BLOG: IVAN RISTIĆ

« Security researchers ask Google to enable SSL encryption by default | Main | Improved handling of SSL warr

HTTP client fingerprinting using SSL handshake analysis

June 17, 2009

SSL fingerprinting for p0f

17 June 2012

synackpse / tls-fingerprinting

forked from LeeBrotherston/tls-fingerprinting

Code | Pull requests 1 | Projects 0

TLS Fingerprinting with JA3 and JA3S

John Althouse Follow

Jan 15 · 10 min read

TLS Fingerprinting- The JA3 Method

```
▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 224
  ▼ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 220
      Version: TLS 1.2 (0x0303)
    ▶ Random
      Session ID Length: 0
      Cipher Suites Length: 38
    ▶ Cipher Suites (19 suites)
      Compression Methods Length: 1
    ▶ Compression Methods (1 method)
      Extensions Length: 141
    ▶ Extension: server_name
    ▶ Extension: elliptic_curves
    ▶ Extension: ec_point_formats
    ▶ Extension: signature_algorithms
    ▶ Extension: next_protocol_negotiation
    ▶ Extension: Application Layer Protocol Negotiation
    ▶ Extension: status_request
    ▶ Extension: signed_certificate_timestamp
    ▶ Extension: Extended Master Secret

0060  1a e1 15 00 00 26 00 ff  c0 2c c0 2b c0 24 c0 23   .....&.. .,.+.$.#
0070  c0 0a c0 09 c0 30 c0 2f  c0 28 c0 27 c0 14 c0 13   .....0./ .(.'....
0080  00 9d 00 9c 00 3d 00 3c  00 35 00 2f 01 00 00 8d   .....=.< .5./....
0090  00 00 00 18 00 16 00 00  13 63 6c 69 65 6e 74 73   ........ .clients
00a0  31 2e 67 6f 6f 67 6c 65  2e 63 6f 6d 00 0a 00 08   1.google .com....
00b0  00 06 00 17 00 18 00 19  00 0b 00 02 01 00 00 0d   ........ ........
00c0  00 12 00 10 04 01 02 01  05 01 06 01 04 03 02 03   ........ ........
```

Fingerprinting TLS - The JA3 Method

Version,Ciphers,Extensions,EllipticCurves,ECPointFormats

771,49172-157-156-61-53-47-10,0-5-10-11-13,29-23-24,0

MD5 hash

JA3 = `f4c4f050188e15839a6cd3af798b6c77`

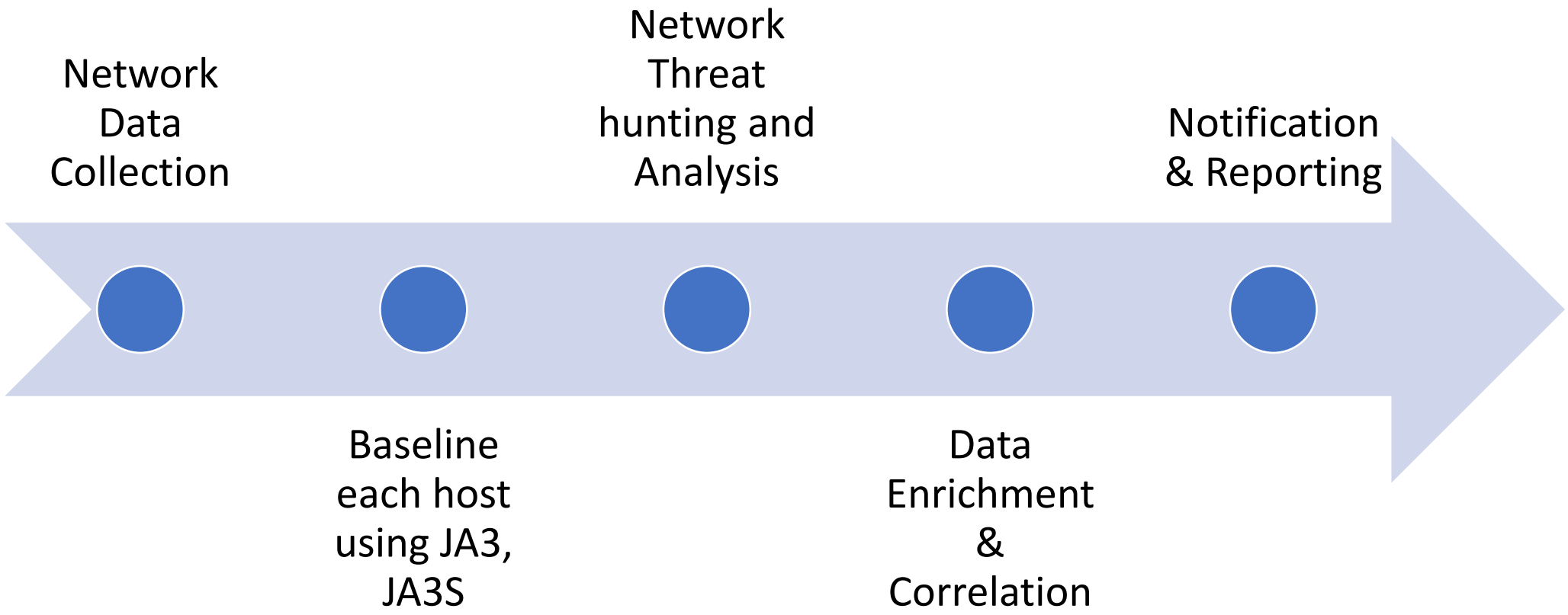
TLS Fingerprinting- The JA3 Method

The JA3 method is used to gather the decimal values of the bytes for the following fields in the Client Hello packet:

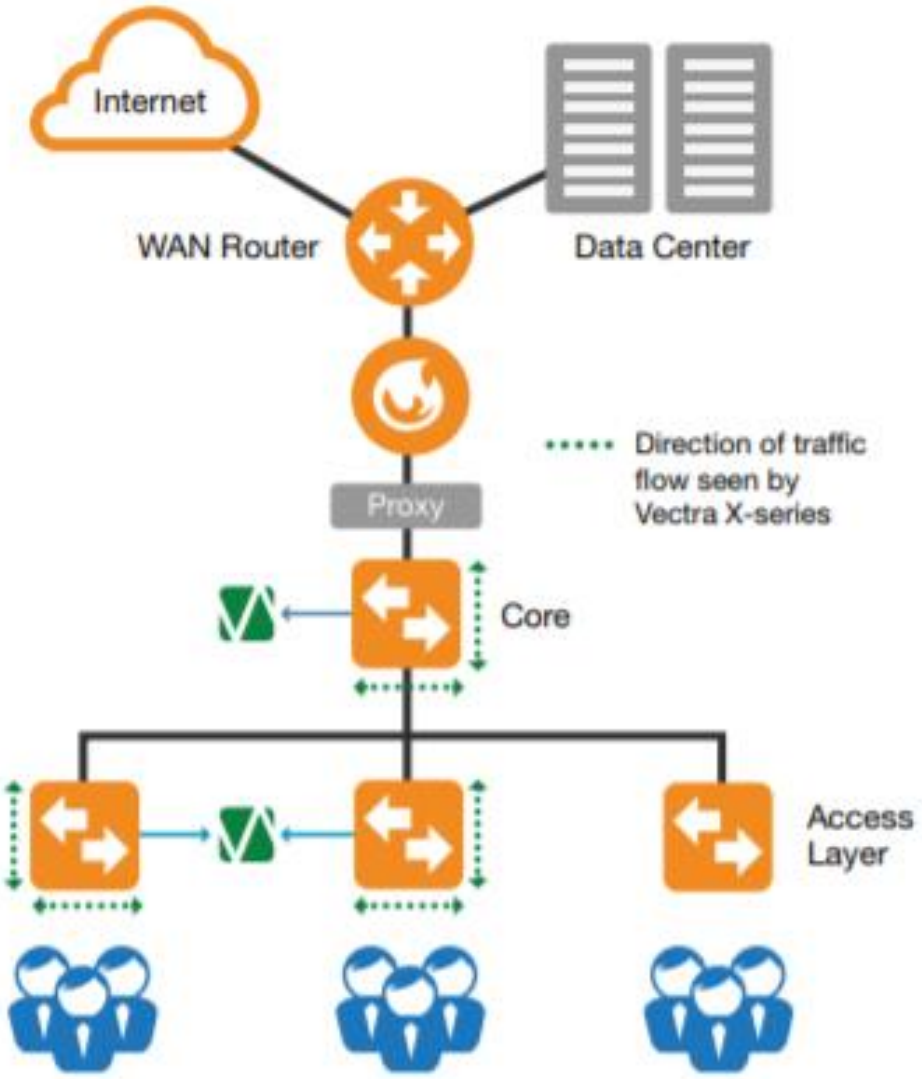
Version, Accepted Ciphers, List of Extensions, Elliptic Curves, and Elliptic Curve Formats.

These strings are then MD5 hashed to produce an easily consumable and shareable 32 character fingerprint. This is the JA3 TLS Client Fingerprint.

JA3 Hunting Methodology



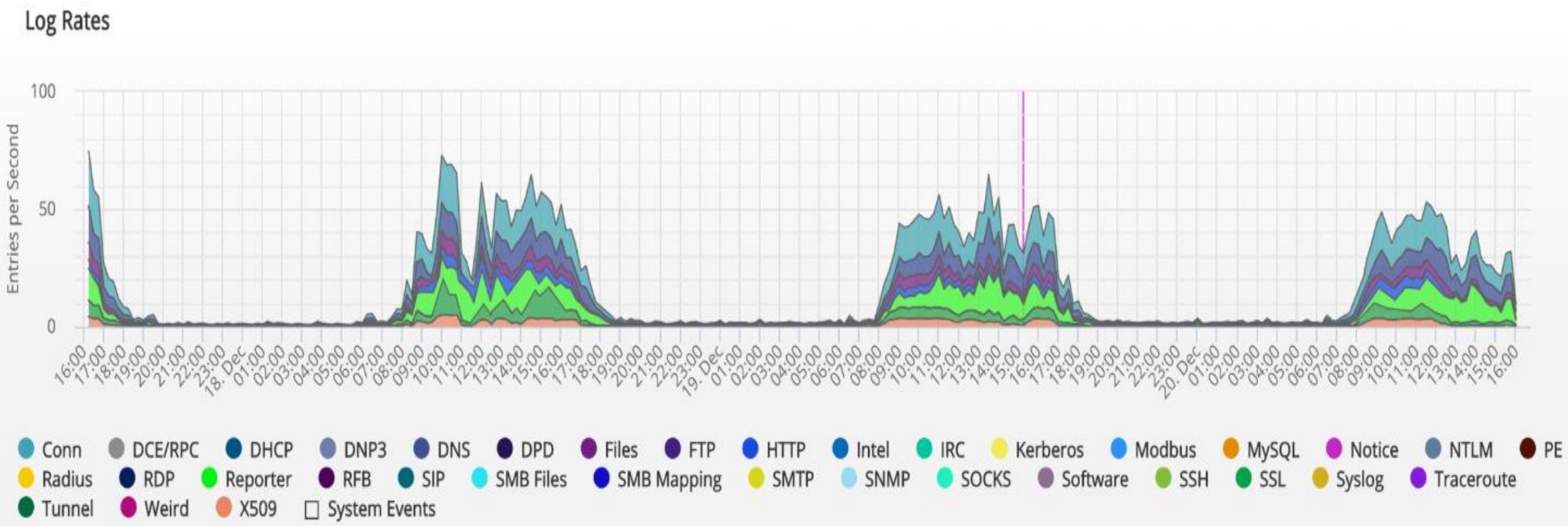
Network Data Collection



Traffic type	Purpose
User to Internet	Detect C&C connections, botnet monetization, click fraud, data exfiltration
User to data center	Detect reconnaissance, data acquisition, data exfiltration
User to user	Detect reconnaissance, lateral movement, data acquisition, data exfiltration
User to authentication servers	Detect brute force login attempts, lateral movement. Also used for host identification
DHCP	Identify hosts

Source: Vectra Deployment

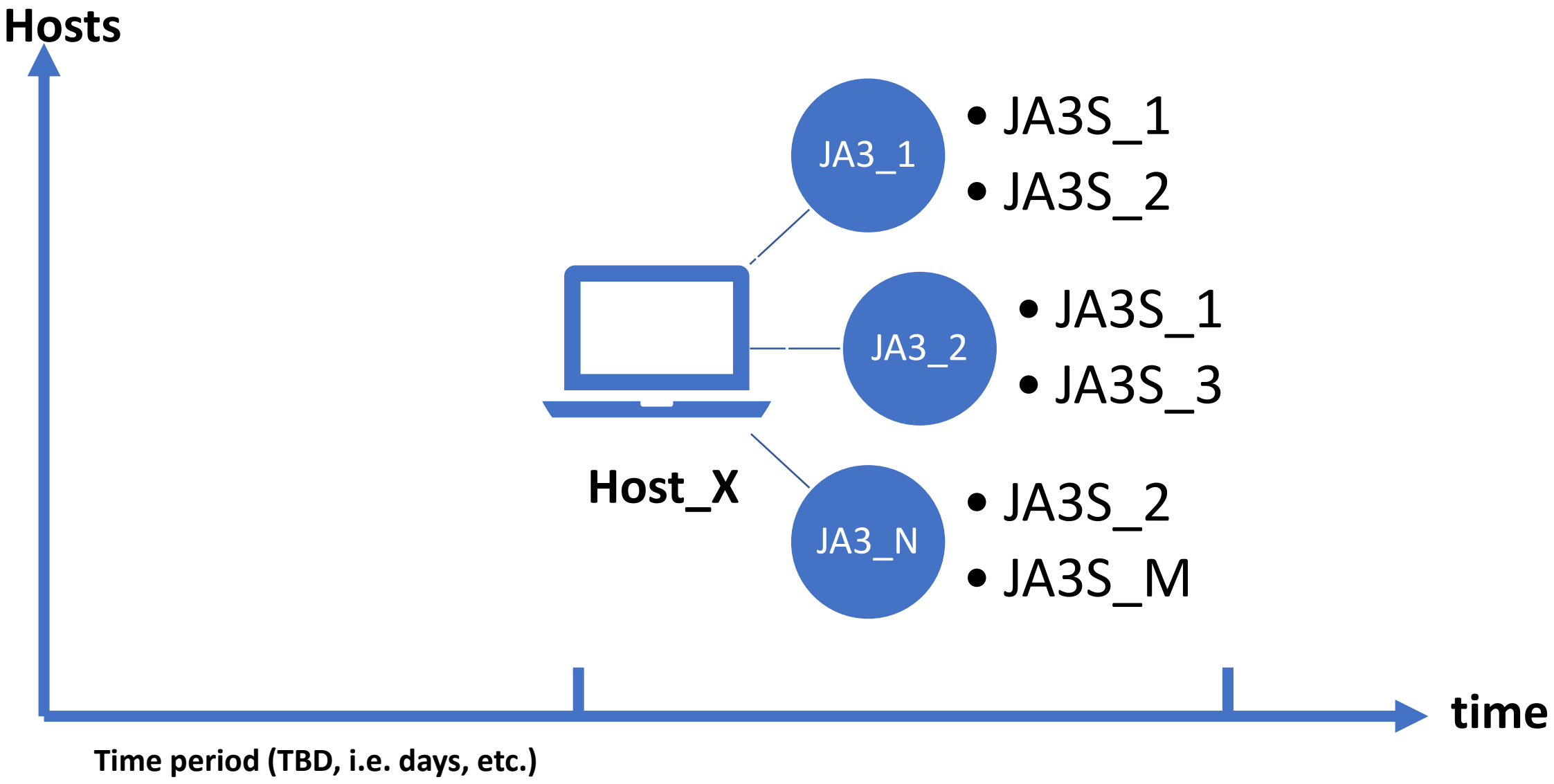
Zeek/Bro: balanced network visibility



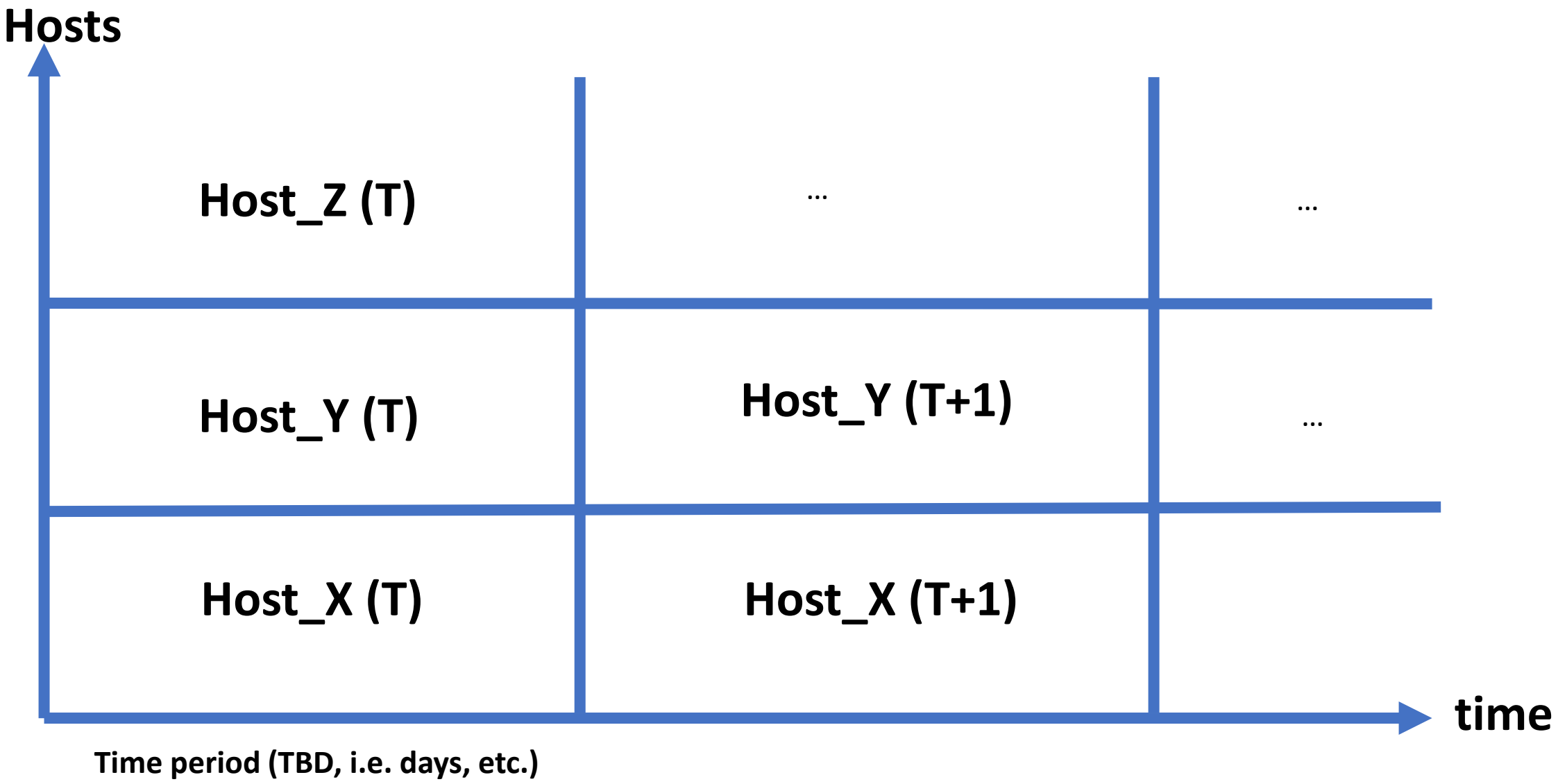
100x richer than Netflow / 100x smaller than PCAP / 50+ data types and protocols.

Source: Corelight Website

Baseline each host using JA3, JA3S

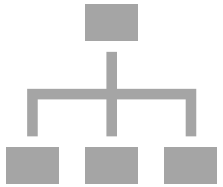


Baseline each host using JA3, JA3S



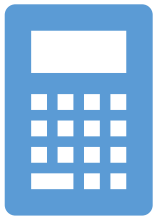
Network Threat hunting and Analysis

Searching



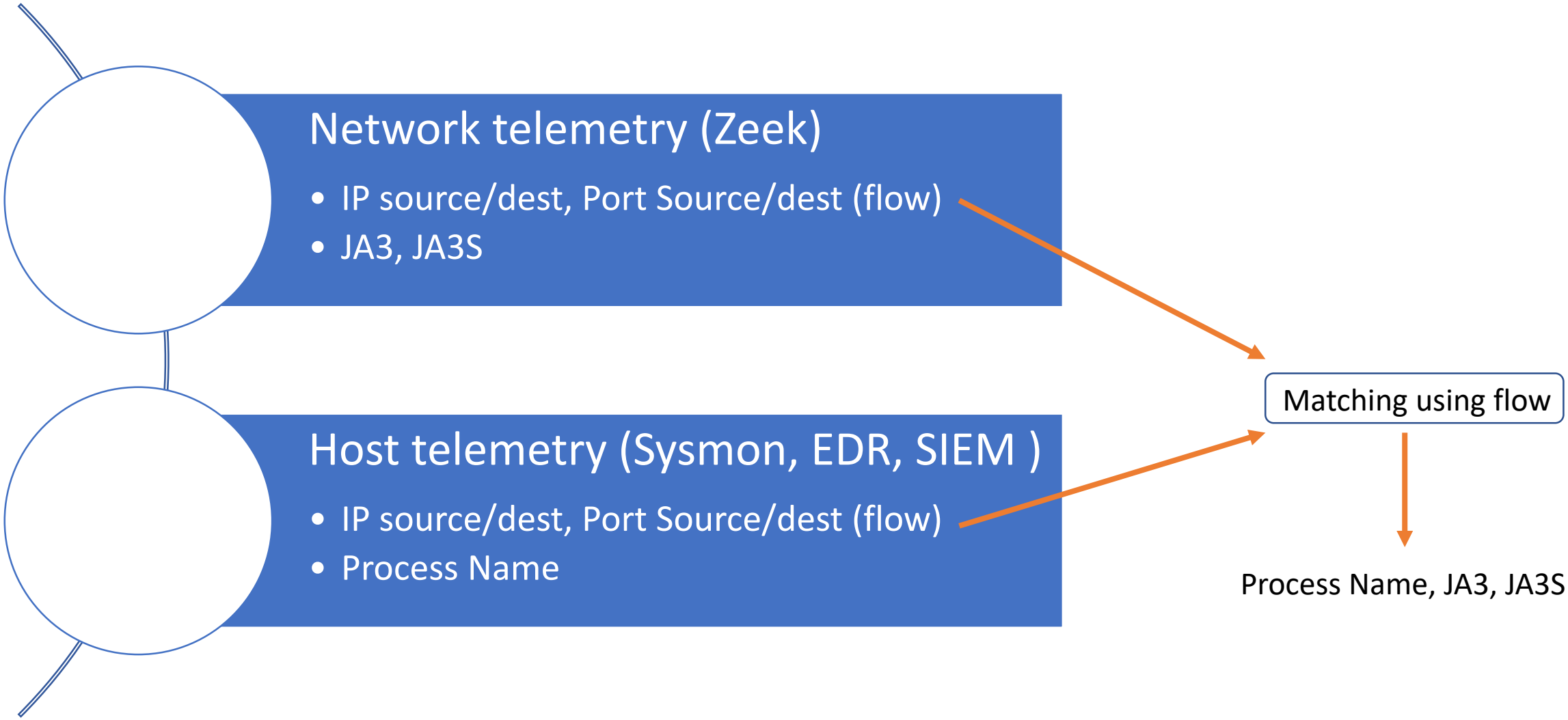
Clustering

Grouping



Stack Counting

Data Enrichment & Correlation



Notification and Reporting

Benign JA3 list

Malicious JA3 list

Suspicious JA3 list

Unknown JA3 list

Application of the methodology (PoC)

5 Hours of TLS traffic : PCAP of 522 Gb (avg 2G/min)

5,3 millions TLS flows

5,3 millions JA3 ->

360 unique JA3 / 660 unique JA3S

4,4 millions certs -> 12k unique certs

Data extracted from Zeek SSL/TLS Logs

- Timestamp
- IP source
- Port Source
- IP dest
- Port dest
- JA3
- JA3S

Stack Counting of JA3 (5,3 M TLS Flows)

Top Occurrence JA3 : Usually Safe

JA3	***occurrence***
ce5f3254611a8c095a3d821d44539877	1323228
5068eff2e6fc3d406f6ad2251509a6fe	724655
66918128f1b9b03303d77c6f2eefd128	620609
10ee8d30a5d01c042afd7b2b205facc4	552603
2d71e639d5d074ea1ba48192a1923bc2	537740
39667e20aca2bc283b4e515485c25c28	213248
a0e9f5d64349fb13191bc781f81f42e1	192625
7375c86ede5d928ba34a0622e4ac0dcd	178151
8f41a697eff27e008f969cf7b5ba4117	154677
11db1cd0dcb0d21f00b603b0dd305495	132733
e539cf186447573e3ac2e0b10e262bd1	70442
bd0bf25947d4a37404f0424edf4db9ad	52919
28a2c9bd18a11de089ef85a160da29e4	35742
bc6c386f480ee97b9d9e52d472b772d8	14875
3b5074b1b5d032e5620f69f9f700ff0e	14639
0eecb7b1551fba4ec03851810d31743f	13727
4abce01f2924d18db5f8e939c9fab036	12612
3d0e94714ddaa4c6e9eb690f529c55ce	10574
851235d5e9d490f3e2b43db94ac71961	9724
2a88f3ec3327fda9aa219682b236e46f	7019
b20b44b18b853ef29ab773e921b03422	5728
f58f4c92d50fe2785d35d6e2eb8756f2	5570
e34fdea9216922aa5e85307c6d5f38d6	4976
f22bdd57e3a52de86cda40da2d84e83b	3569
f8128c51dc8d1f49da1d61267353 00d5	3486
98eaec8c8ef8baab245d0b65f788be91	3000
5c60d9b844ba6734958d357bff4aff60	2932
e2966889cdcd2470b40132a164e09acb	2918
5182f54f9c6e99d117d9dde3fa2b4cff	2360
0ffee3ba8e615ad22535e7f771690a28	2240

Stack Counting of JA3 (5,3 M TLS Flows)

Rare Occurrence JA3 : Need Investigation

JA3	***occurrence***
5bf43fbca3454853c26df6d996954aca	1
edb680d2136f0abc774e3369bcfccdc7	1
b70a512a93b8619a0480e6a876e1171c	1
a1a3466d7652d5e397683a68ef86325a	1
1fbe5382f9d8430fe921df747c46d95f	1
adf55f61efe61bf8f83857927f1fd6ad	1
7aab5a832d843ee9533ad66a0325c8f1	1
977c5ca224fe72c8bf953d48e27edf70	1
4e623d918a8aa1361c613dadbbba6ae9	1
1b1d75104f3c2482f02436b06f97596e	1
ca70a9058215562270f2ad91601bc072	1
2d2eac5c36c8f8f955afaf10878548f1	1
f436b9416f37d134cadd04886327d3e8	1
0ae18052c288c1bd39910255598ed827	1
e5f9eaf6372aa79e37487ebb85889441	1
7d47b5ecec79c45643b52510b55baa02	1
c376061f96329e1020865a1dc726927d	1
20c9baf81bfe96ff89722899e75d0190	1
307f08ff4a51a297be3b32882ffce30e	1
a20fe054526698db4feb61cdbd53b092	1
3ed575e4a4c08727f7ea8bbd16379d18	1
d470a3fa301d80227bc5650c75567d25	1
54e5f5ba8a7a24eec56c45258dc5f424	1
2c1a25ebb1942336cb86cc5183f89271	1
3663e1b4300b292192f61c1004646781	1
369fcde4652bd00b4365c9c56ea30e1e	1
b081ba34faac9e4b7d109d7666185ba3	1
9c726efb125aff72a33e17a34c83f4d8	1
afd3aee9bab304c82a6efacb5c0a6c5d	1
06815b74b41fda94c217a66281bccd36	1

Stack Counting of JA3S (5,3 M TLS Flows)

Top Occurrence JA3S : Usually Safe

JA3S	***occurrence***
986571066668055ae9481cb84fda634a	1319205
303951d4c50efb2e991652225a6f02b1	457147
364ff14b04ef93c3b4cfa429d729c0d9	357391
f9a66afdd1f499d415ca470974ec00c8	346072
28ef90cc3d9d08c96a8a2cb6f365a79e	183049
fbe78c619e7ea20046131294ad087f05	149454
a704460bd0a887c62e4f462bf1bba96b	140423
410b9bedaf65dd26c6fe547154d60db4	124828
15381d64ba148f31a70eb87b53085230	107821
35af4c8cd9495354f7d701ce8ad7fd2d	100879
9a022b14200c7389ebbb1436d5cf1339	82177
98bf23c62ffddc3907c57a6712ada3ad	76915
eca9b8f0f3eae50309eaf901cb822d9b	57904
699a80bdb17efe157c861f92c5bf5d1d	54893
8d2a028aa94425f76ced7826b1f39039	53061
7bee5c1d424b7e5f943b06983bb11422	47593
704239182a9091e4453fdbfe0fd17586	45085
4cf820cab8f5a2bf61be14f5493233ae	42127
860fcf58fd757e26aa8911e5eaff6b53	41621
5badad76fbdd6e8b6296e2e9f4024401	38072
9d9ce860f1b1cbef07b019450cb368d8	33991
02bdc318d9f618eea3e10d0a7ba25ba0	33798
0debd3853f330c574b05e0b6d882dc27	33247
61be9ce3d068c08ff99a857f62352f9d	32227
5a1d5fe94bd964277aa8109fa53618d3	32148
42ec7b1db61428bf1cc6e01b9ef02b04	31764
1d0e57c6ae42e9204defa51e5e1cdc4c	30336
4560a2cba28d0d5a9c9208d56ff6b439	29888
a9e3ed16ee3208291487c8d2aa2ad924	29178
15c4d139d9f284ce5a6e4380e77c1f5c	27799

Stack Counting of JA3S (5,3 M TLS Flows)

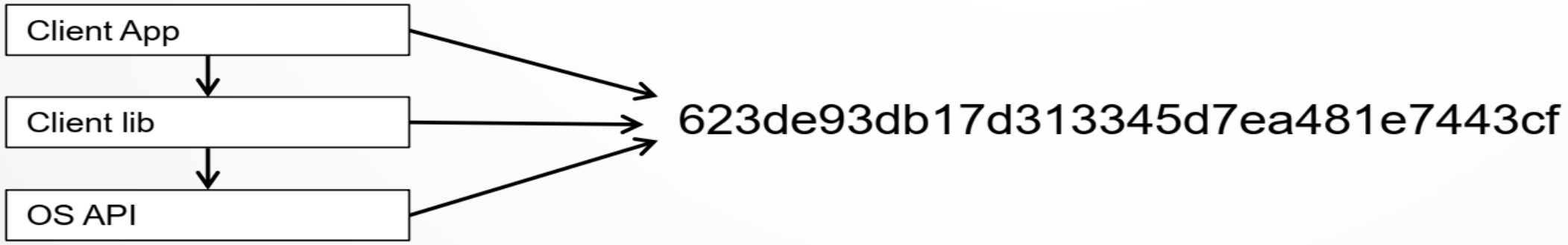
Rare Occurrence JA3S : Need Investigation

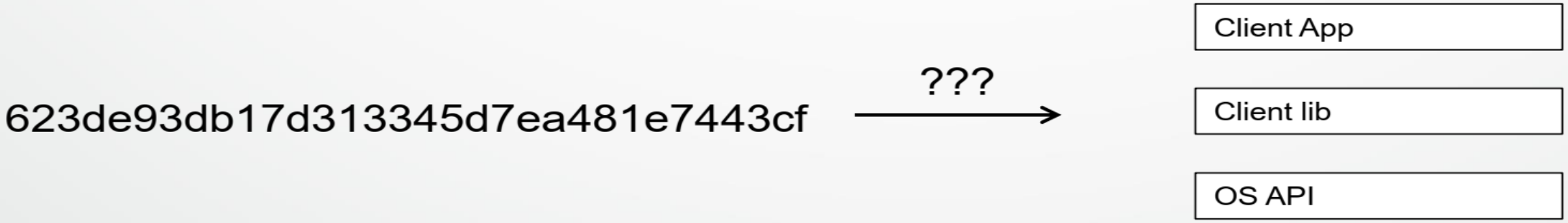
JA3S	***occurrence***
0d87f3e89ef826ca8d9c6043d1154c90	1
bb3714ede90db64a2b838d08f5a38557	1
71d9ce75f347e6cf54268d7114ae6925	1
d351504c2ea3e95c394b07f1be3b61c7	1
3402c7c295e3358839390282420b35be	1
7fc9a50daacf140c755e84a3080a19d8	1
dd4c815e611f9c16dba59a334f75a06b	1
1e60202b4001a190621caa963fb76697	1
d4d745c7326b18dc3357be0f163f2fd3	1
a56d7d52bdc65ea749d74addccfb0d36	1
6060b01ce5682281fc30979175981713	1
cbb432e9f6c8c1093ca5bb0639db1f66	1
b7bd51222a09f3ad66a340710ae9c01a	1
8503e5aae1deee435ad1a1bea3442d18	1
aaf1173ea45ea798158bbe9d37883e02	1
5677fd9197d91e09fd9b670a4699b803	1
0040ce0b9d615d0d65defe92e3122178	1
ae53107a2e47ea20c72ac44821a728bf	1
06ceee71c393081ce16f75f61a2e19af	1
5c2e91a1ad300cf70a0d920f5abea68d	1
9e357fe7424317b33da9dd18d077fa11	1
9a2c6bfd476184689a786457f33ece3a	1
aa9bd267b87f0346b39111d76f80079c	1
29ecaffc413d5f2ebae4f4b68bd486c7	1
834e9c6069aabe0f94fe274756fe7585	1
9f9bacf804c80acc4b83068d38907939	1
c58ef5526cf734959c046fcbfe8c140f	1
ad7cd3de0b03dbc32f24b13f48ea4e3a	1
3030e8776450c28632d3539aa0dc32a5	1
1fda766ae5f2b04d26fdc3d4bc06bcc5	1

To be continued

- Group by (JA3, JA3S)
- Group By (JA3, IP Source)
- Group By (IP dest, JA3S)
- Group By (IP source, JA3, IP dest, JA3S)
- Time Series Analysis
- TLS Beacon Detection

Limitation 1 : JA3 Collision





Source : Kjell Tore Fossbakk, HelseCert

Limitation 2 : Impersonating JA3

Home > Cloud Security > Bots Tampering with TLS to Avoid Detection

BOTS TAMPERING WITH TLS TO AVOID DETECTION

By Threat Research Team May 15, 2019 8:00 AM

0 Comments

Impersonating JA3

Hiding behind JA3 hash

September 27, 2019 - By Defensive Security

TLS FINGERPRINTING ...fun

Thank you

Demo/Questions

References

- https://docs.zeek.org/en/stable/script-reference/log-files.html
- https://info.vectra.ai/hubfs/no_index/compliance/cb_mitre_082318.pdf
- https://www.mbsecure.nl/blog/2019/5/dettact-mapping-your-blue-team-to-mitre-attack
- https://www.corelight.com/products/software
- https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/
- https://filtermax.hu/files/download/Vectra_DeploymentGuide.pdf
- https://www.linkedin.com/pulse/four-common-threat-hunting-techniques-sample-hunts-ely-kahn/

References

- https://github.com/cisco/joy
- https://blog.ivanristic.com/2009/06/http-client-fingerprinting-using-ssl-handshake-analysis.html
- https://idea.popcount.org/2012-06-17-ssl-fingerprinting-for-p0f/
- https://github.com/synackpse/tls-fingerprinting
- https://github.com/salesforce/ja3
- https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=449962
- https://www.cisco.com/go/anyconnect
- https://www.cisco.com/go/threatgrid
- https://www.cisco.com/go/eta

What is Transport Layer Security (TLS)?

- Transport Layer Security, or TLS, is a widely adopted security protocol designed to facilitate privacy and data security for communications over the Internet. A primary use case of TLS is encrypting the communication between web applications and servers, such as web browsers loading a website.

TLS V1.3 Handshake

Step	Client	Direction	Message	Direction	Server
1			Client Hello Supported Cipher Suites Guesses Key Agreement Protocol Key Share	>	
2		<	Server Hello Key Agreement Protocol Key Share Server Finished		
3			Checks Certificate Generates Keys Client Finished	>	

TLS V1.2 Handshake

Step	Client	Direction	Message	Direction	Server
1			Client Hello	>	
2		<	Server Hello		
3		<	Certificate		
4		<	Server Key Exchange		
5		<	Server Hello Done		
6			Client Key Exchange	>	
7			Change Cipher Spec	>	
8			Finished	>	
9		<	Change Cipher Spec		
10		<	Finished		