How Tracking Companies Circumvented Ad Blockers Using WebSockets

Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda,
William Robertson, Christo Wilson
Northeastern University

Presented by Eric Newberry
University of Michigan

Slides (mostly) by Muhammad Ahmad Bashir

Online Tracking

Online Tracking

Surge in online advertising (internet economy)

- Ad networks pour in billions of dollars.
- Value for their investment?
- Extensive tracking to serve targeted ads.

Online Tracking

Surge in online advertising (internet economy)

- Ad networks pour in billions of dollars.
- Value for their investment?
  - Extensive tracking to serve targeted ads.

User concern over tracking

- Led to the proliferation of ad blocking extensions

Online Tracking

Surge in online advertising (internet economy)

- Ad networks pour in billions of dollars.
- Value for their investment?
  - Extensive tracking to serve targeted ads.

User concern over tracking

- Led to the proliferation of ad blocking extensions

Ad networks fight back

- E.g Using anti-ad blocking scripts

Google & Safari

- Google evaded Safari's third-party cookie blocking policy (Jonathan Mayer)
- … by submitting a form in an invisible iFrame
- Google fined $22.5M by FTC

This Talk

How **Ad Networks** leveraged a bug in the Chrome API to **bypass Ad Blockers** using **WebSockets**

This Talk

How **Ad Networks** leveraged a bug in the Chrome API to **bypass Ad Blockers** using **WebSockets**

1. What caused this?
2. How was this bug leveraged by ad networks?

Web Sockets

Web Sockets

HTTP/S

Web Sockets

HTTP/S

Web Sockets

HTTP/S

Web Sockets

HTTP/S

Web Sockets

HTTP/S

Web Socket

Web Sockets

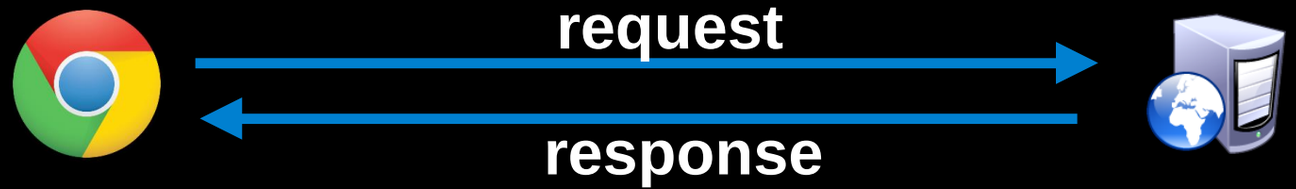
HTTP/S

Web Socket

- Both client and server can send/receive data
- This is a persistent connection

Web Sockets

HTTP/S



Web Socket

- Both client and server can send/receive data
- This is a persistent connection

Ad Blockers

Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests

Ad Blockers

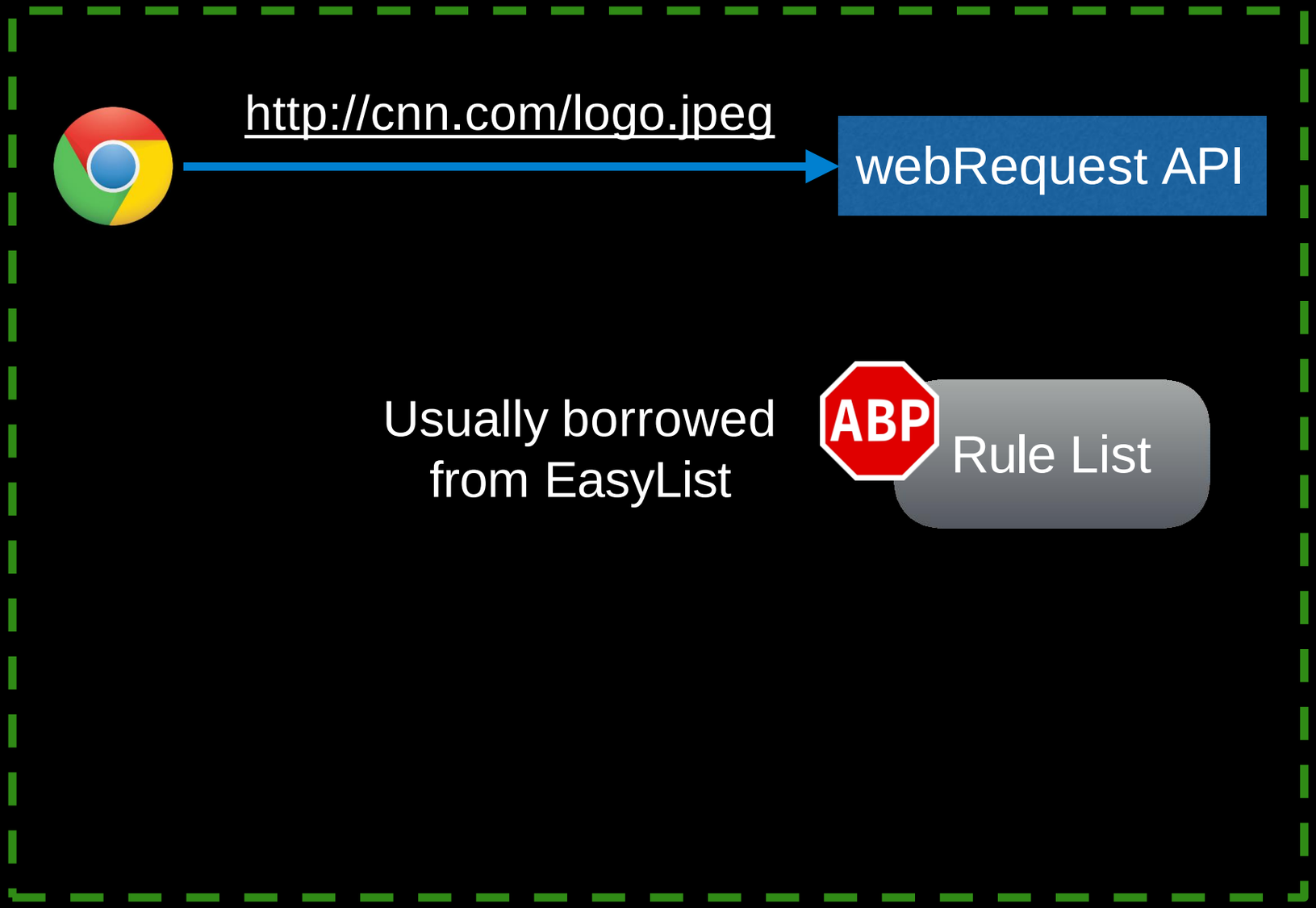
- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests

Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests

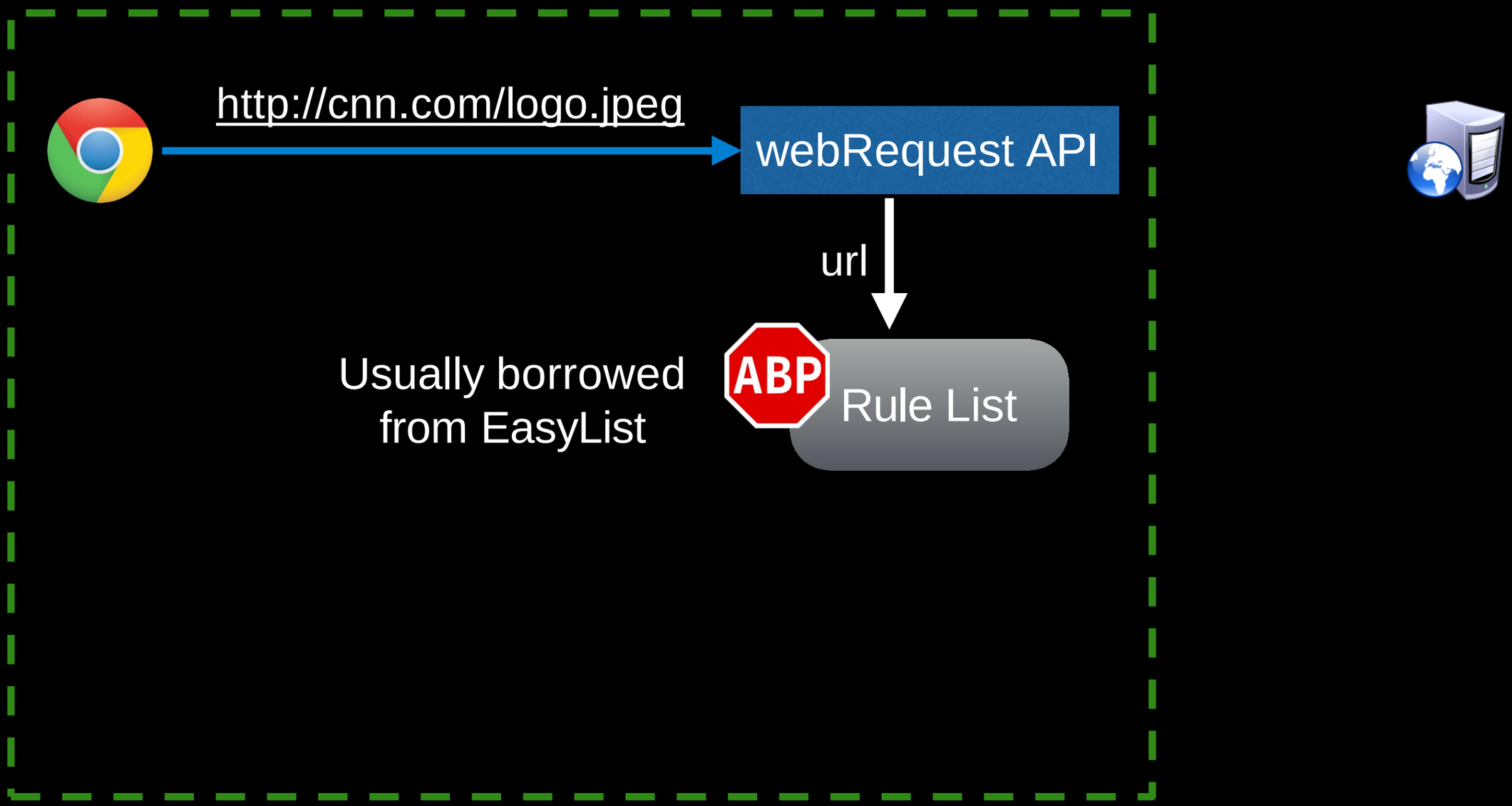
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



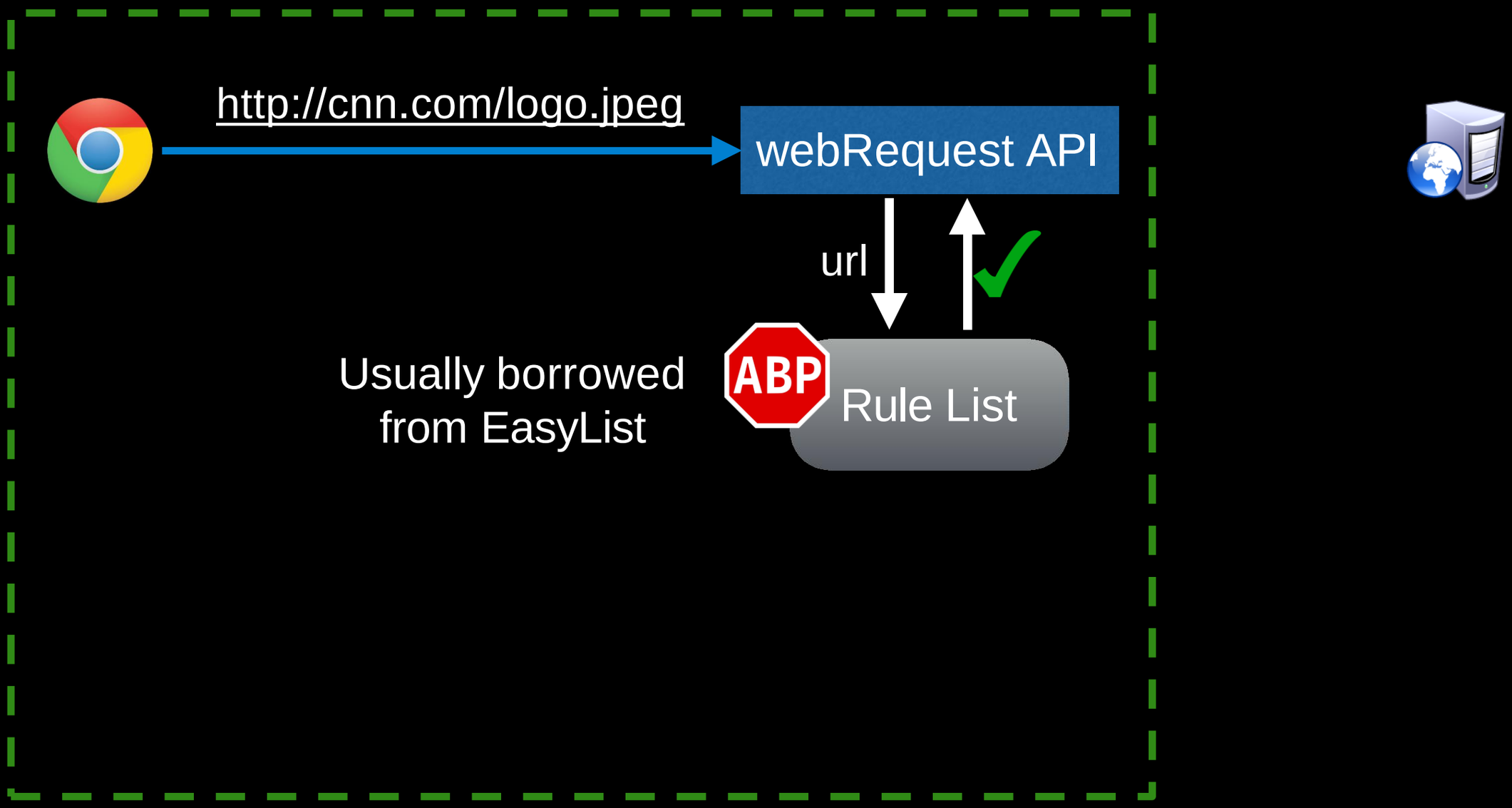
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



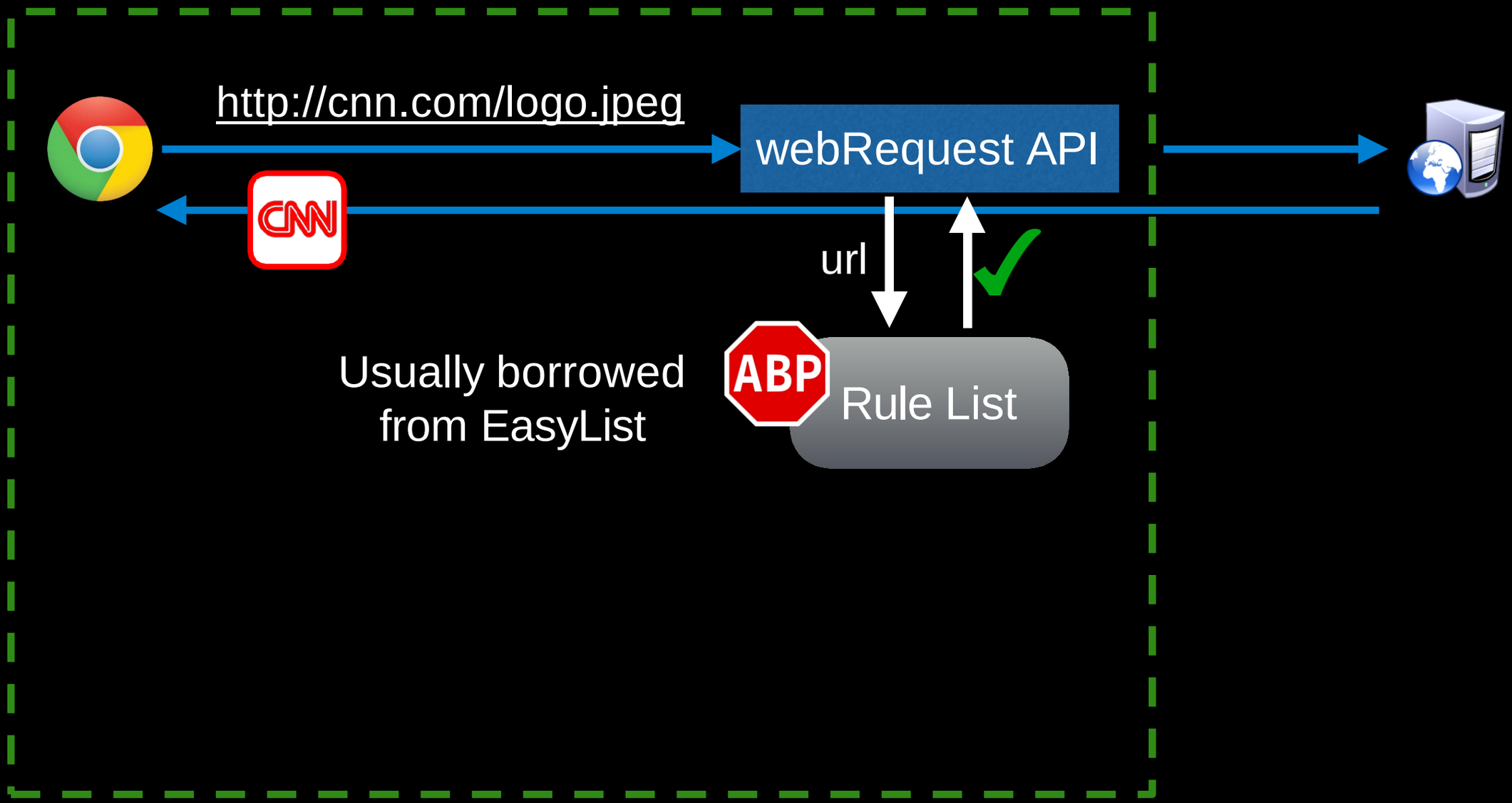
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



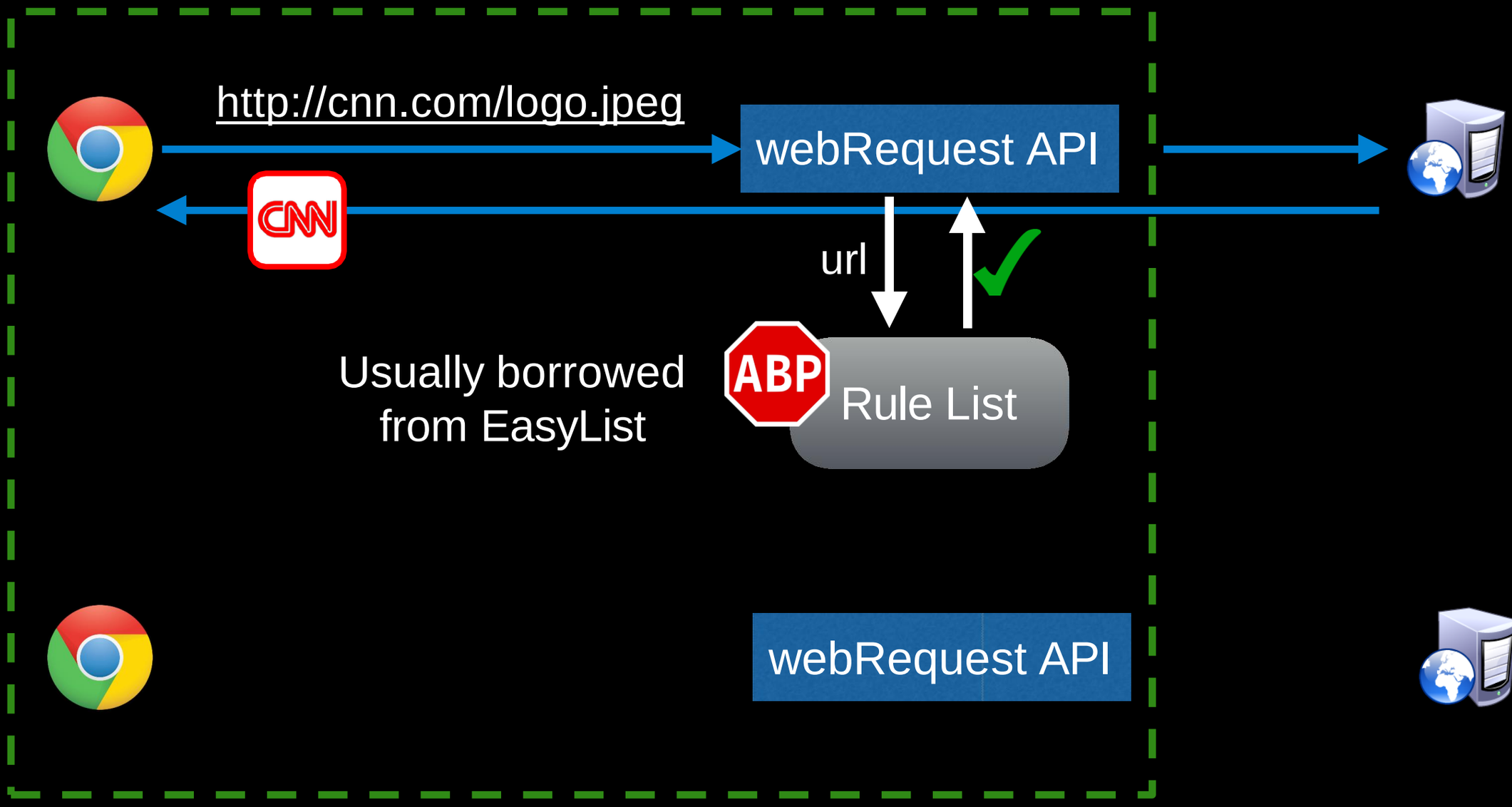
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



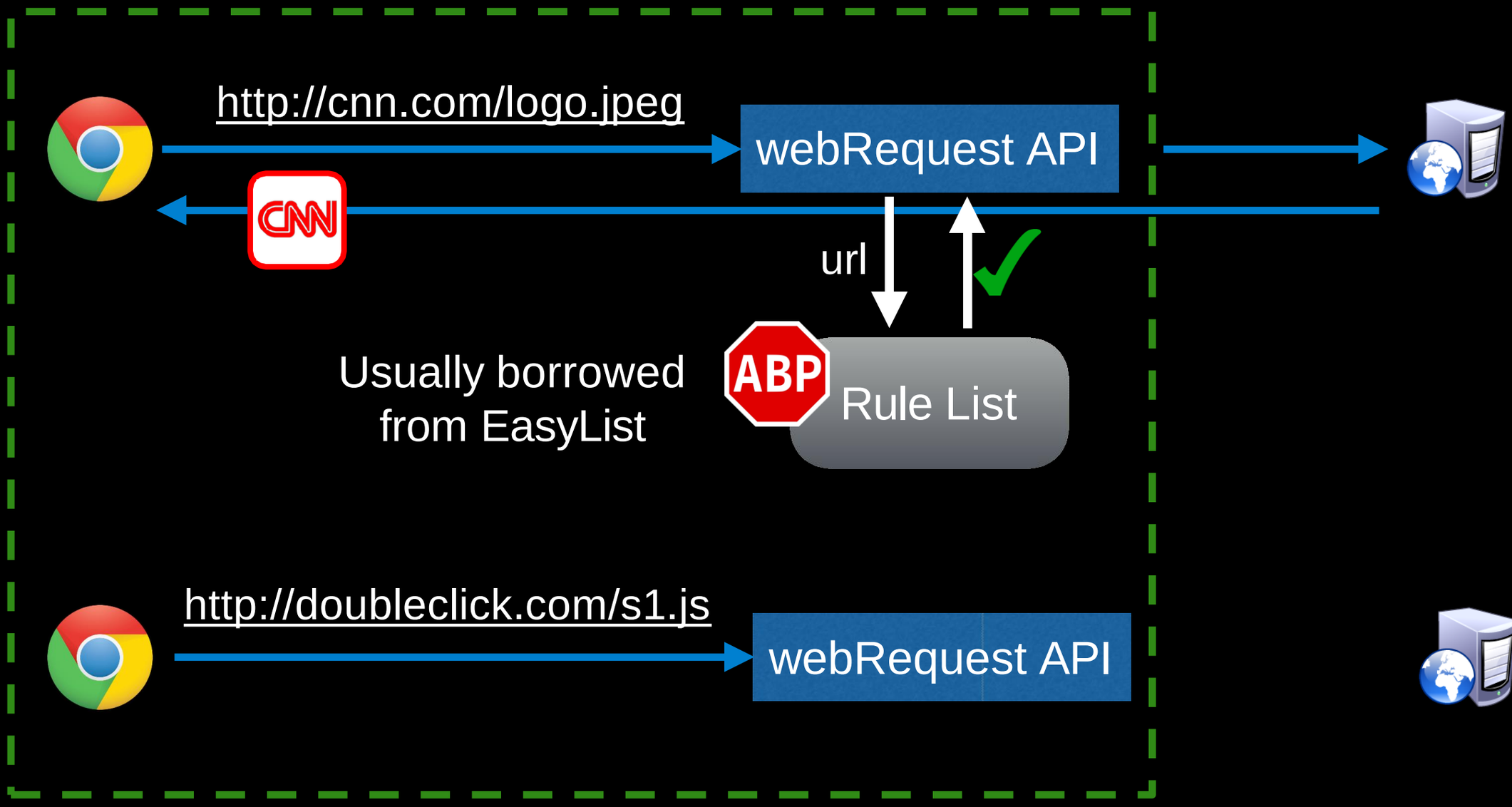
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



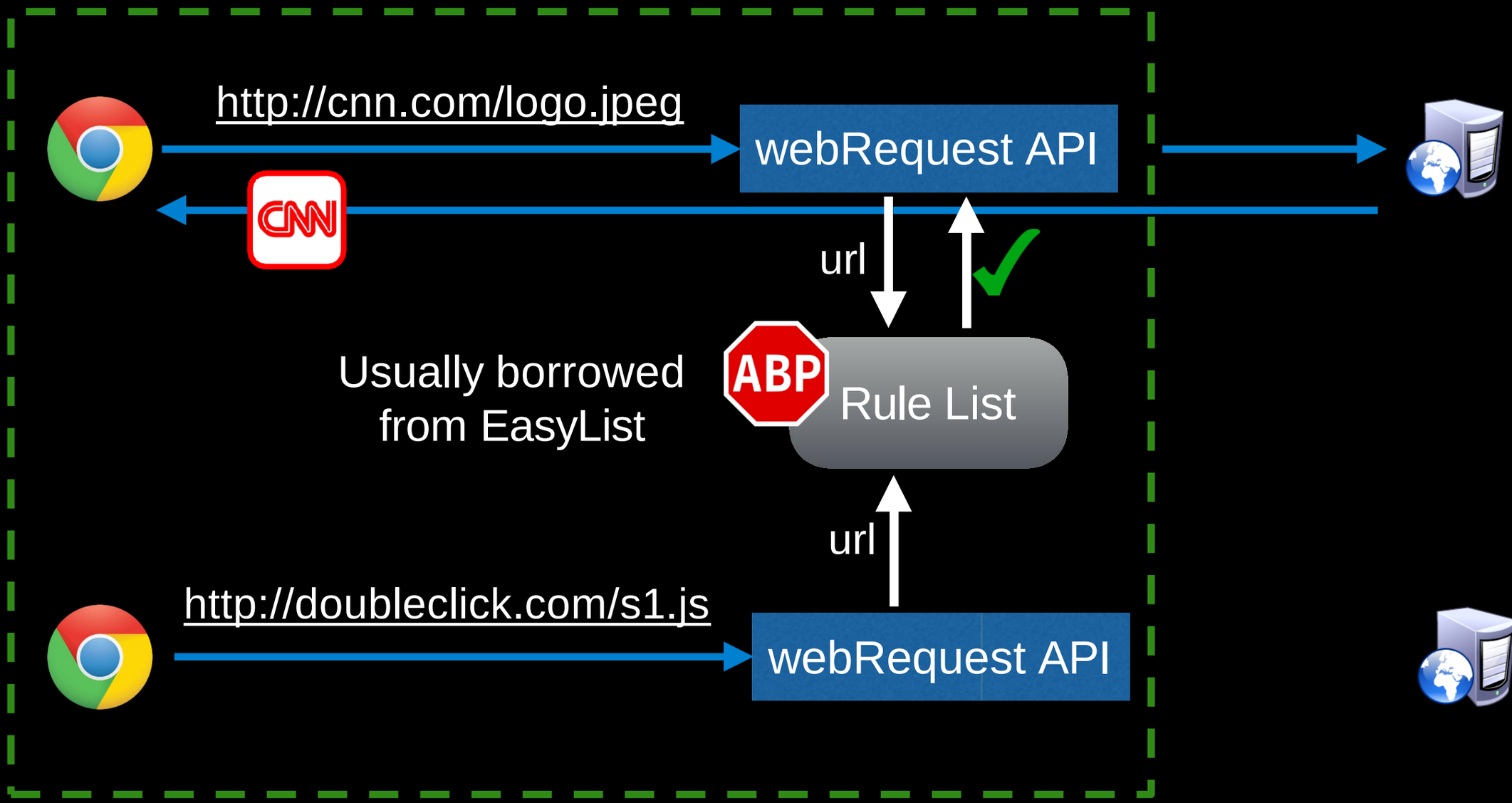
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



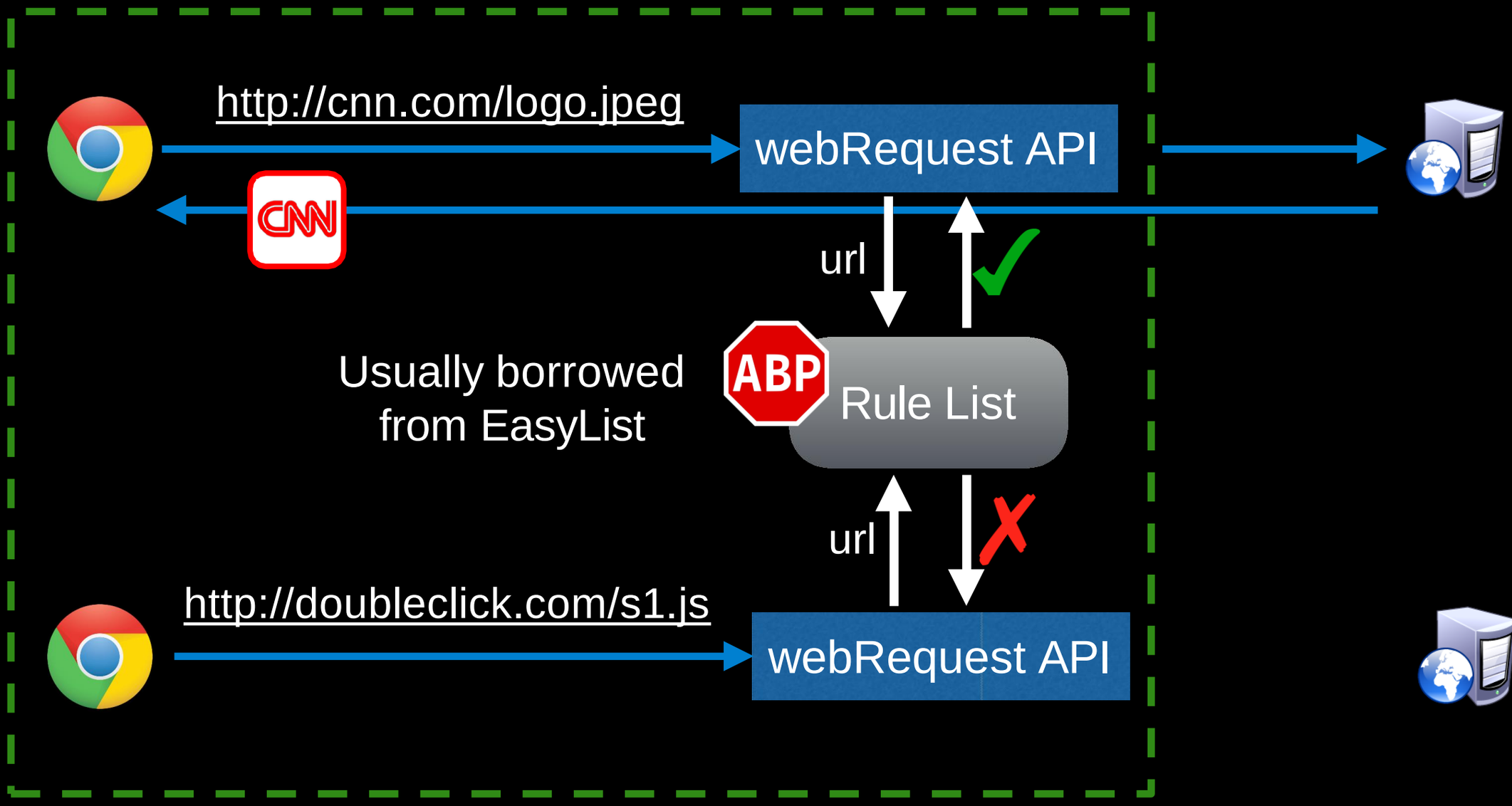
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



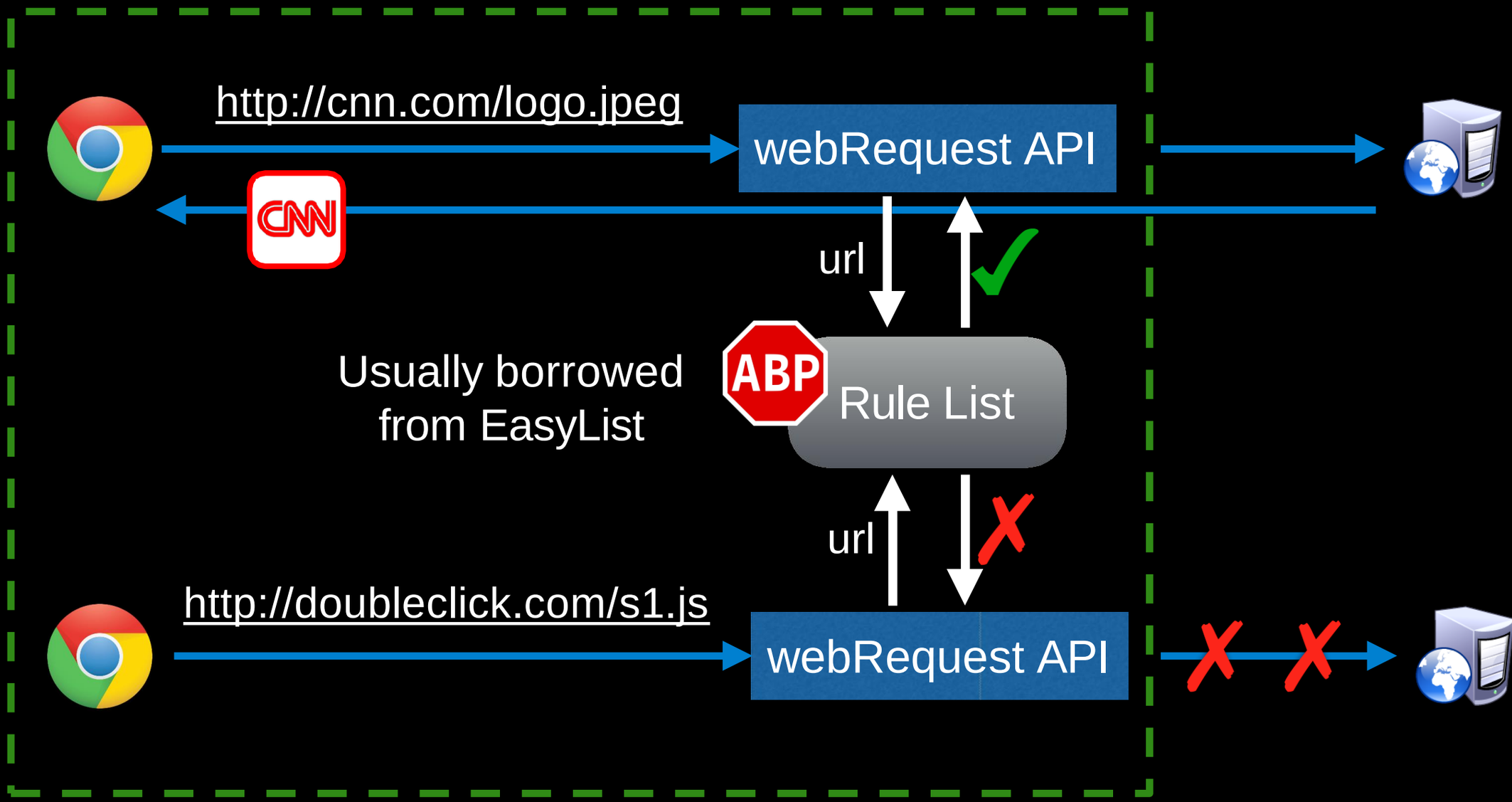
Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



Ad Blockers

- Chrome extension **chrome.webRequest** API
  - Extension can inspect / modify / drop outgoing requests



AdBlock Evasion

AdBlock Evasion

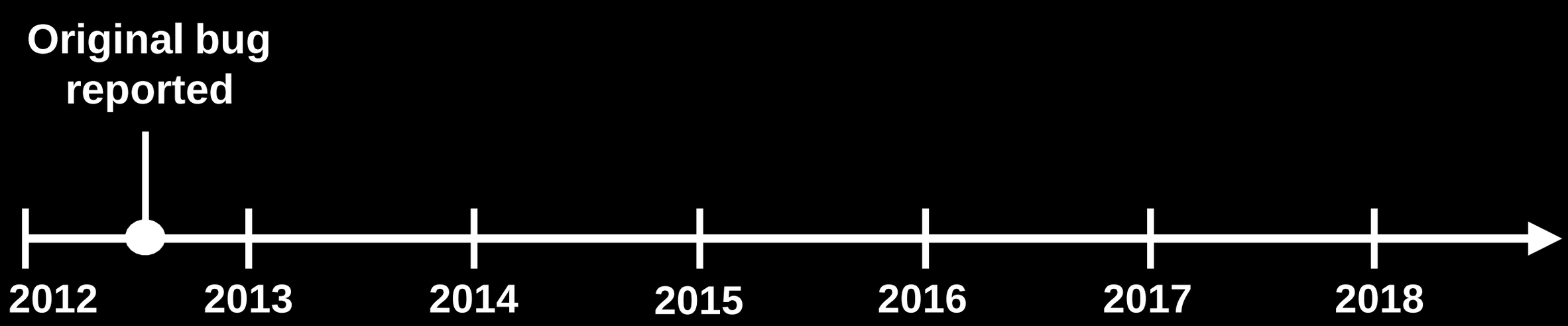
- Bug in `webRequest API`
  - ws/wss requests did not trigger the API

AdBlock Evasion

- Bug in webRequest API
  - ws/wss requests did not trigger the API

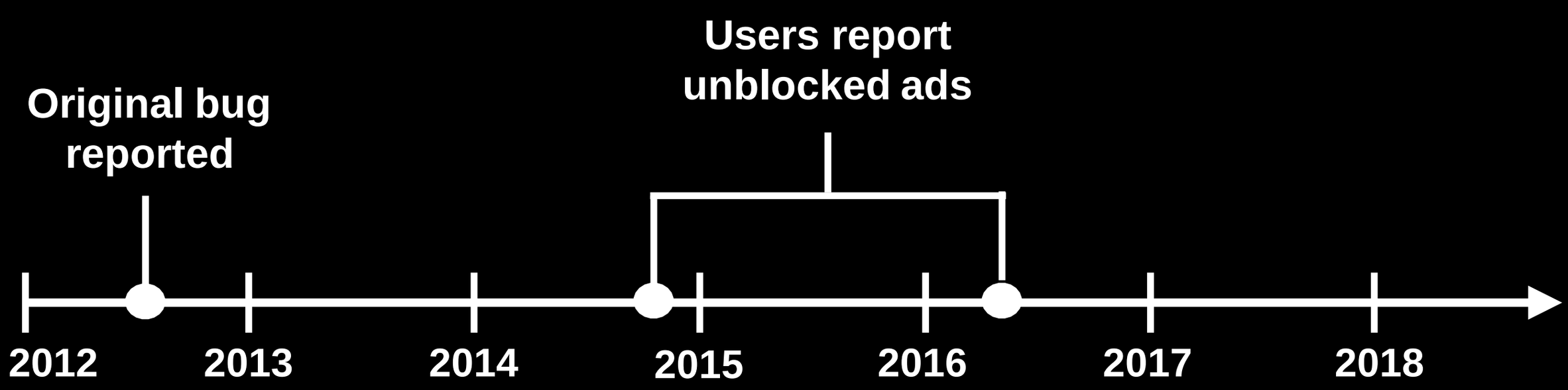
AdBlock Evasion

- Bug in `webRequest API`
  - ws/wss requests did not trigger the API



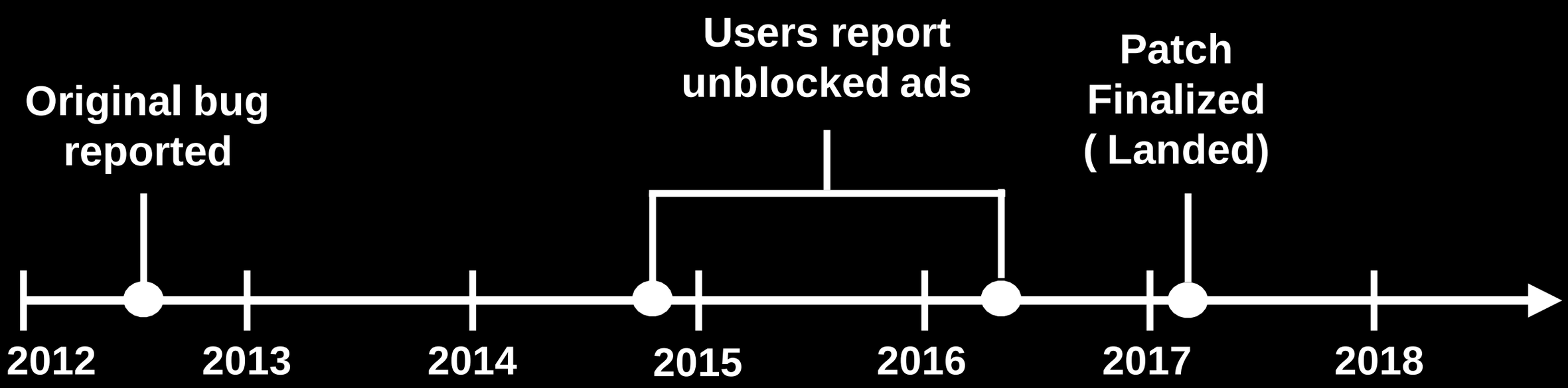
AdBlock Evasion

- Bug in `webRequest API`
  - ws/wss requests did not trigger the API



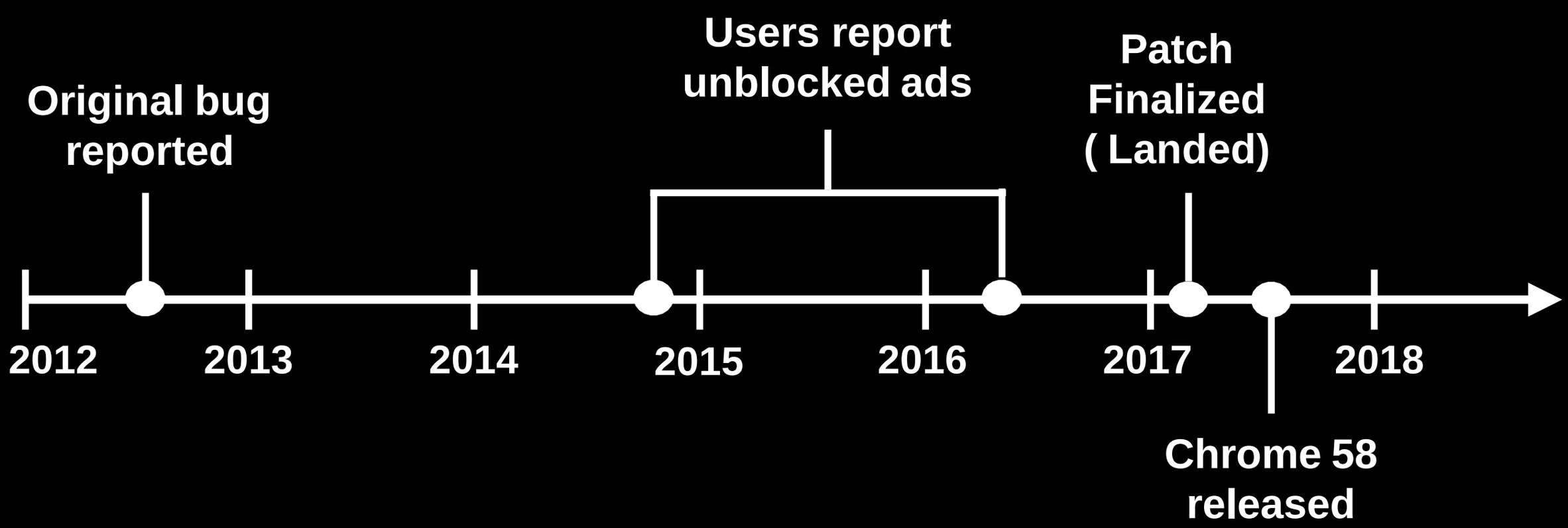
AdBlock Evasion

- Bug in `webRequest API`
  - ws/wss requests did not trigger the API



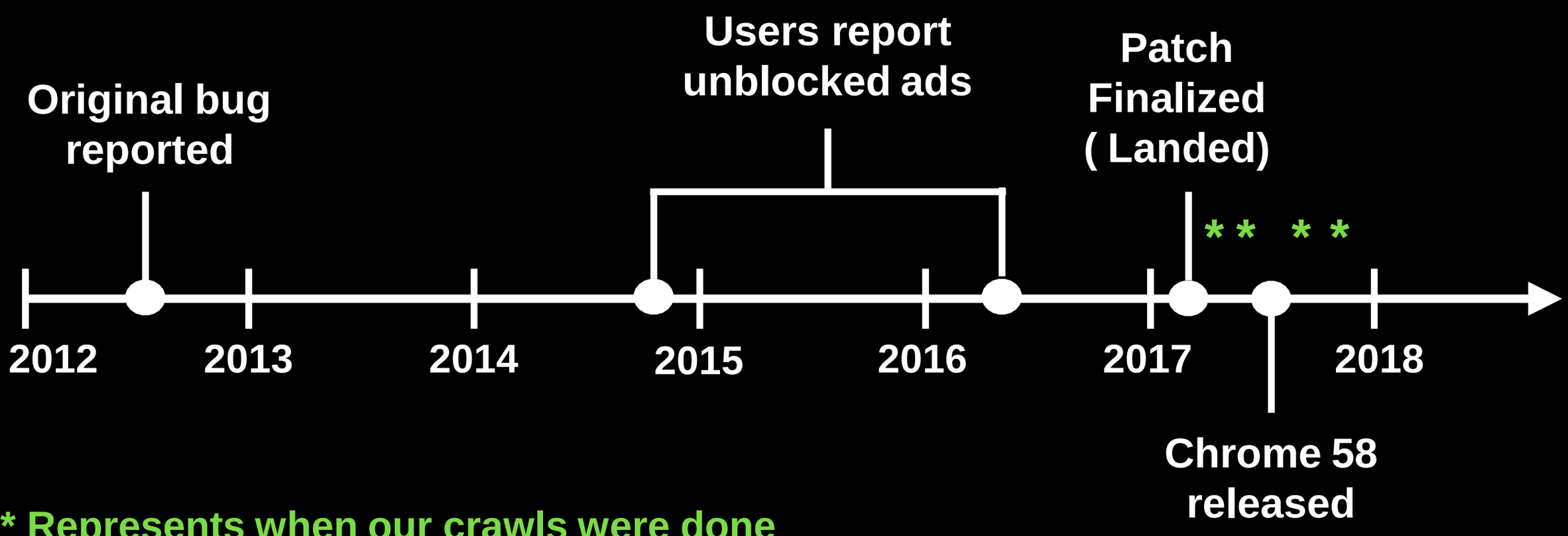
AdBlock Evasion

- Bug in `webRequest API`
  - ws/wss requests did not trigger the API



AdBlock Evasion

- Bug in `webRequest API`
  - ws/wss requests did not trigger the API



*** Represents when our crawls were done**

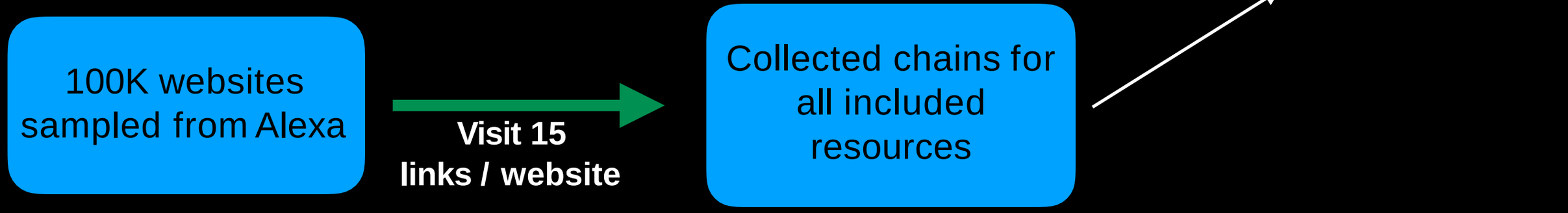
Data Crawling

Data Crawling

100K websites
sampled from Alexa

Data Crawling

Data Crawling



Data Crawling

100K websites sampled from Alexa

Visit 15 links / website

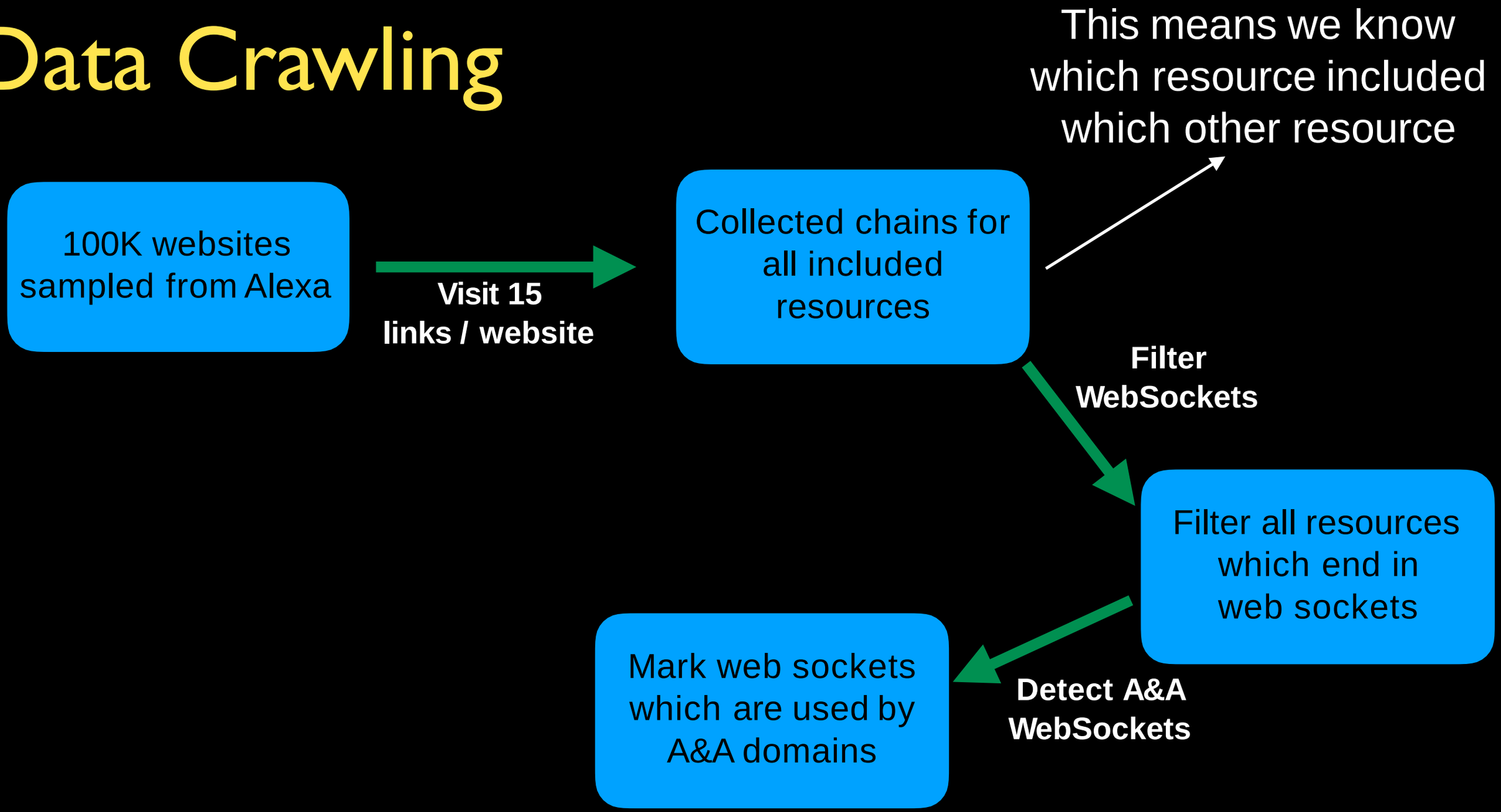
Collected chains for all included resources

This means we know which resource included which other resource

Filter WebSockets

Filter all resources which end in web sockets

Data Crawling



A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

Data Crawling

100K websites sampled from Alexa

Visit 15 links / website

Collected chains for all included resources

This means we know which resource included which other resource

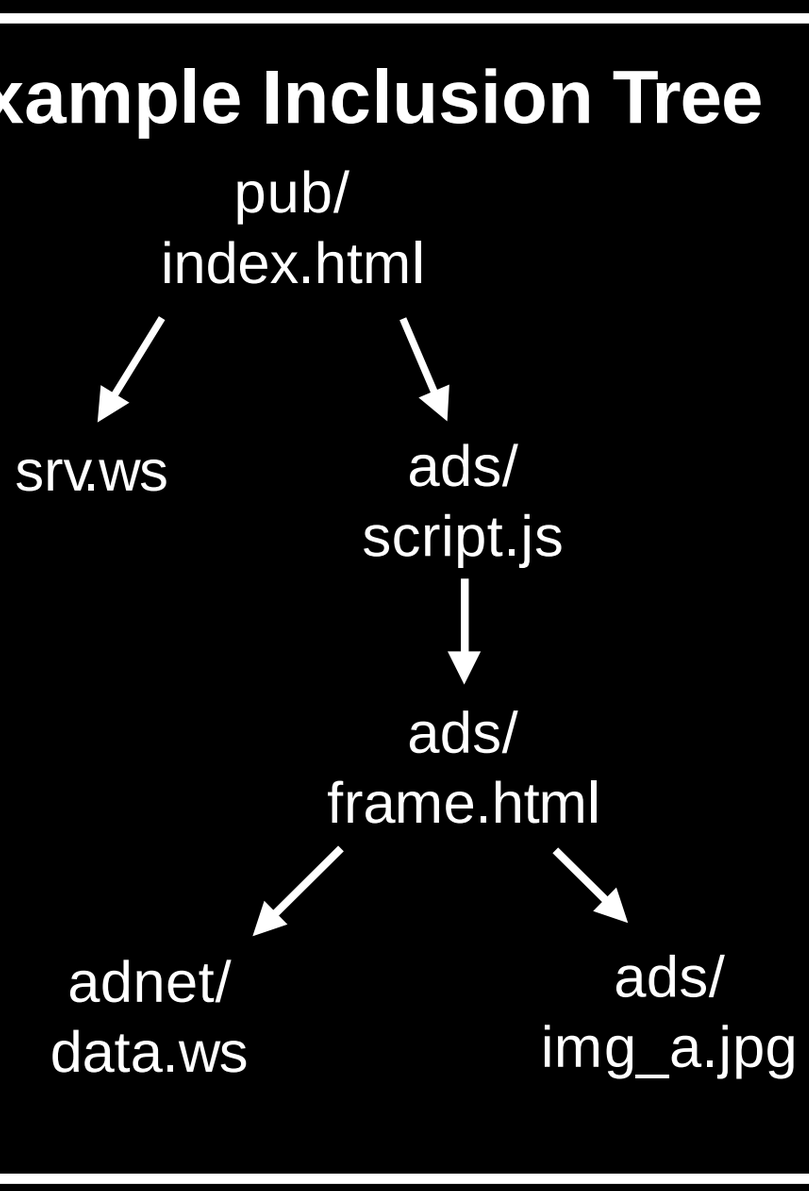
Filter WebSockets

Filter all resources which end in web sockets

Detect A&A WebSockets

Mark web sockets which are used by A&A domains

Example Inclusion Tree



A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

Data Crawling

100K websites sampled from Alexa

Visit 15 links / website

Collected chains for all included resources

This means we know which resource included which other resource

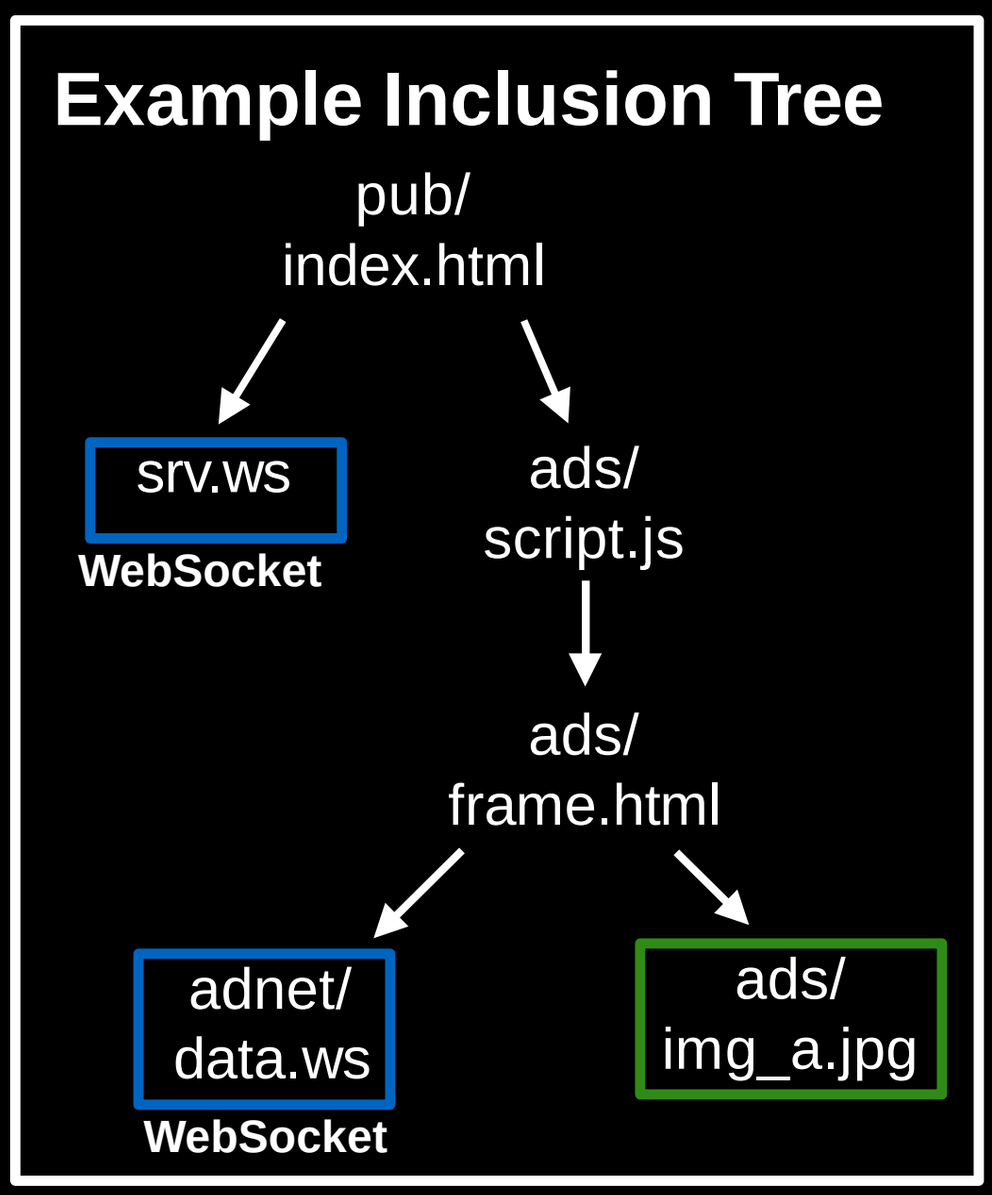
Filter WebSockets

Filter all resources which end in web sockets

Detect A&A WebSockets

Mark web sockets which are used by A&A domains

Example Inclusion Tree



A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

Data Crawling

100K websites sampled from Alexa

Visit 15 links / website

Collected chains for all included resources

This means we know which resource included which other resource

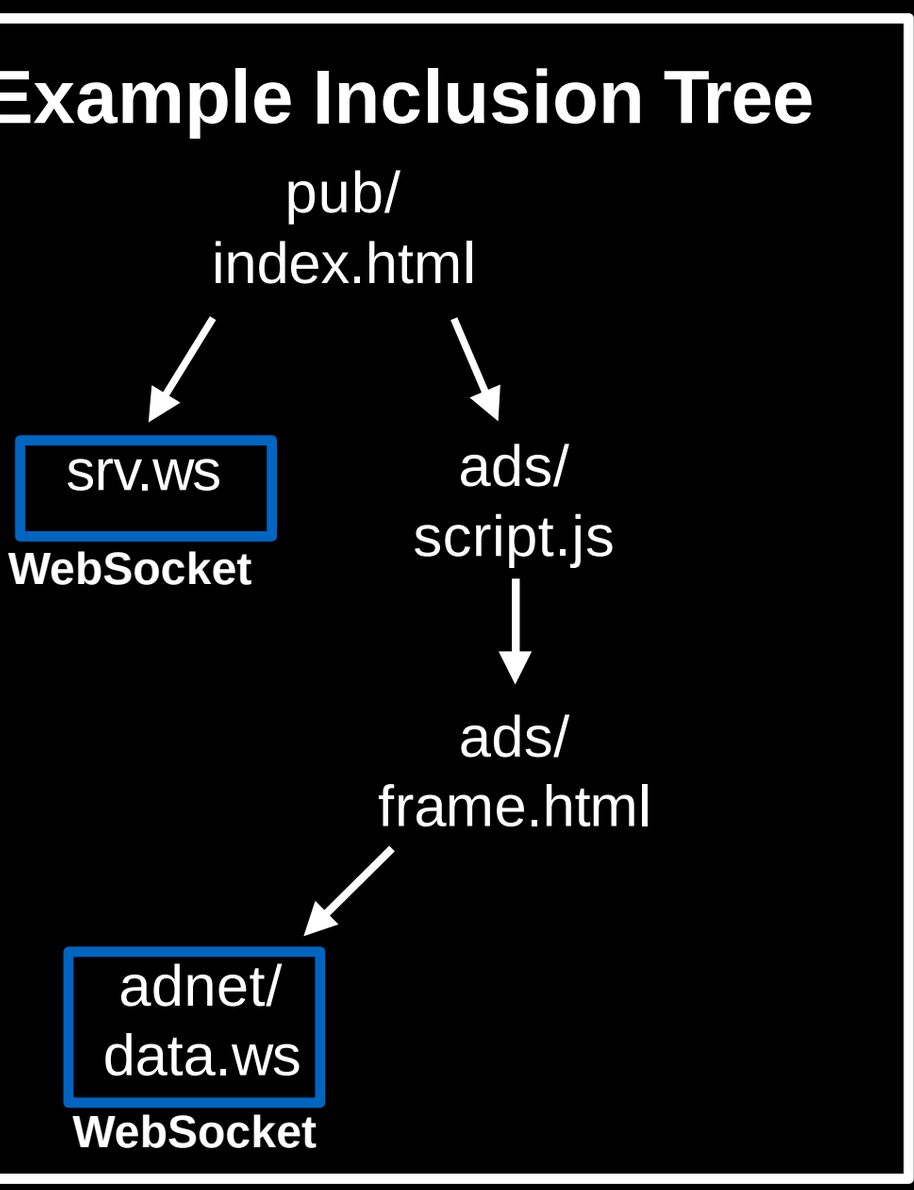
Filter WebSockets

Filter all resources which end in web sockets

Detect A&A WebSockets

Mark web sockets which are used by A&A domains

Example Inclusion Tree



A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

Data Crawling

100K websites sampled from Alexa

Visit 15 links / website

Collected chains for all included resources

This means we know which resource included which other resource

Filter WebSockets

Filter all resources which end in web sockets

Detect A&A WebSockets

Mark web sockets which are used by A&A domains

Example Inclusion Tree

pub/
index.html

ads/
script.js

ads/
frame.html

adnet/
data.ws

WebSocket

A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

High-Level Numbers

High-Level Numbers

	Crawl Dates	%Websites with sockets	% Sockets with A&A Initiators	% Sockets with A&A Receivers	#Unique A&A Initiators	#Unique A&A Receivers
Before Chrome 58	Apr 02-05, 2017	2.1	60.6	73.7	75	16
	Apr 11-16, 2017	2.4	61.3	74.6	63	18

High-Level Numbers

	Crawl Dates	%Websites with sockets	% Sockets with A&A Initiators	% Sockets with A&A Receivers	#Unique A&A Initiators	#Unique A&A Receivers
Before Chrome 58	Apr 02-05, 2017	2.1	60.6	73.7	75	16
	Apr 11-16, 2017	2.4	61.3	74.6	63	18
After Chrome 58	May 07-12, 2017	1.6	60.2	69.7	19	15
	Oct 12-16, 2017	2.5	63.4	63.7	23	18

High-Level Numbers

	Crawl Dates	%Websites with sockets	% Sockets with A&A Initiators	% Sockets with A&A Receivers	#Unique A&A Initiators	#Unique A&A Receivers
Before Chrome 58	Apr 02-05, 2017	2.1	60.6	73.7	75	16
	Apr 11-16, 2017	2.4	61.3	74.6	63	18
After Chrome 58	May 07-12, 2017	1.6	60.2	69.7	19	15
	Oct 12-16, 2017	2.5	63.4	63.7	23	18

- ~2% websites use web sockets.

High-Level Numbers

	Crawl Dates	%Websites with sockets	% Sockets with A&A Initiators	% Sockets with A&A Receivers	#Unique A&A Initiators	#Unique A&A Receivers
Before Chrome 58	Apr 02-05, 2017	2.1	60.6	73.7	75	16
	Apr 11-16, 2017	2.4	61.3	74.6	63	18
After Chrome 58	May 07-12, 2017	1.6	60.2	69.7	19	15
	Oct 12-16, 2017	2.5	63.4	63.7	23	18

- ~2% websites use web sockets.
- ~61 % sockets are initiated by A&A domains

A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

High-Level Numbers

	Crawl Dates	%Websites with sockets	% Sockets with A&A Initiators	% Sockets with A&A Receivers	#Unique A&A Initiators	#Unique A&A Receivers
Before Chrome 58	Apr 02-05, 2017	2.1	60.6	73.7	75	16
	Apr 11-16, 2017	2.4	61.3	74.6	63	18
After Chrome 58	May 07-12, 2017	1.6	60.2	69.7	19	15
	Oct 12-16, 2017	2.5	63.4	63.7	23	18

- ~2% websites use web sockets.
- ~61 % sockets are initiated by A&A domains
- ~71 % sockets contact an A&A domain

A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

High-Level Numbers

	Crawl Dates	%Websites with sockets	% Sockets with A&A Initiators	% Sockets with A&A Receivers	#Unique A&A Initiators	#Unique A&A Receivers
Before Chrome 58	Apr 02-05, 2017	2.1	60.6	73.7	75	16
	Apr 11-16, 2017	2.4	61.3	74.6	63	18
After Chrome 58	May 07-12, 2017	1.6	60.2	69.7	19	15
	Oct 12-16, 2017	2.5	63.4	63.7	23	18

- ~2% websites use web sockets.
- ~61 % sockets are initiated by A&A domains
- ~71 % sockets contact an A&A domain
- # Initiators drop after Chrome 58 release.

A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

High-Level Numbers

	Crawl Dates	%Websites with sockets	% Sockets with A&A Initiators	% Sockets with A&A Receivers	#Unique A&A Initiators	#Unique A&A Receivers
Before Chrome 58	Apr 02-05, 2017	2.1	60.6	73.7	75	16
	Apr 11-16, 2017	2.4	61.3	74.6	63	18
After Chrome 58	May 07-12, 2017	1.6	60.2	69.7	19	15
	Oct 12-16, 2017	2.5	63.4	63.7	23	18

- ~2% websites use web sockets.
- ~61 % sockets are initiated by A&A domains
- ~71 % sockets contact an A&A domain
- # Initiators drop after Chrome 58 release.
- Small but persistent A&A receivers.

A&A = Advertising and Analytics
e.g. DoubleClick, Criteo, Adnxs

Initiators and Receivers

Initiators and Receivers

Initiators and Receivers

Initiators and Receivers

Initiators and Receivers

Top A&A Initiators

A&A Initiator	#A&A Receivers
facebook	11
google	11
doubleclick	9
youtube	8
addthis	8
hotjar	7
googlesyndication	6
twitter	5
sharethis	4
adnxs	3

Initiators and Receivers

Top A&A Initiators

A&A Initiator	#A&A Receivers
facebook	11
google	11
doubleclick	9
youtube	8
addthis	8
hotjar	7
googlesyndication	6
twitter	5
sharethis	4
adnxs	3

Initiators and Receivers

Top A&A Initiators

A&A Initiator	#A&A Receivers
facebook	11
google	11
doubleclick	9
youtube	8
addthis	8
hotjar	7
googlesyndication	6
twitter	5
sharethis	4
adnxs	3

Top A&A Receivers

A&A Receiver	#A&A Initiators
realtime	27
33across	19
intercom	16
disqus	13
zopim	12
hotjar	11
feedjit	10
lockerdome	8
inspectlet	6
smartsupp	4

Initiators and Receivers

Top A&A Initiators

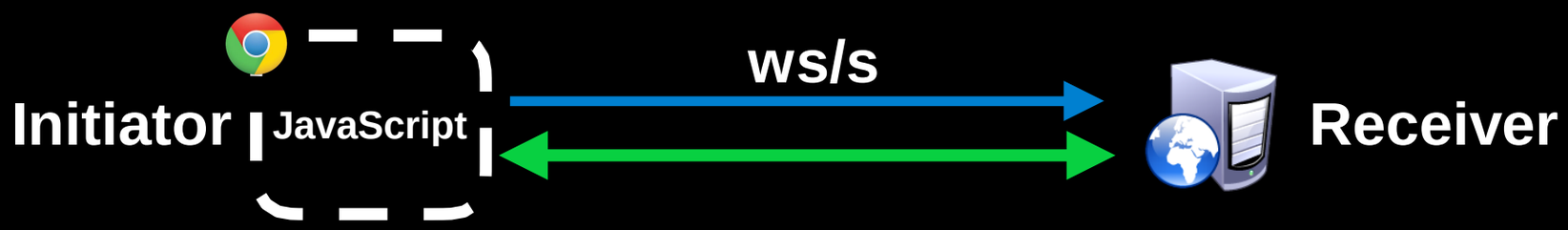
A&A Initiator	#A&A Receivers
facebook	11
google	11
doubleclick	9
youtube	8
addthis	8
hotjar	7
googlesyndication	6
twitter	5
sharethis	4
adnxs	3

Top A&A Receivers

A&A Receiver	#A&A Initiators
realtime	27
33across	19
intercom	16
disqus	13
zopim	12
hotjar	11
feedjit	10
lockerdome	8
inspectlet	6
smartsupp	4

- **Disqus** provides comment board services.

Initiators and Receivers



Top A&A Initiators

A&A Initiator	#A&A Receivers
facebook	11
google	11
doubleclick	9
youtube	8
addthis	8
hotjar	7
googlesyndication	6
twitter	5
sharethis	4
adnxs	3

Top A&A Receivers

A&A Receiver	#A&A Initiators
realtime	27
33across	19
intercom	16
disqus	13
zopim	12
hotjar	11
feedjit	10
lockerdome	8
inspectlet	6
smartsupp	4

- **Disqus** provides comment board services.
- **Zopim, Intercom, Smartsupp** provide live chat services.

Initiators and Receivers

Top A&A Initiators

A&A Initiator	#A&A Receivers
facebook	11
google	11
doubleclick	9
youtube	8
addthis	8
hotjar	7
googlesyndication	6
twitter	5
sharethis	4
adnxs	3

Top A&A Receivers

A&A Receiver	#A&A Initiators
realtime	27
33across	19
intercom	16
disqus	13
zopim	12
hotjar	11
feedjit	10
lockerdome	8
inspectlet	6
smartsupp	4

- **Disqus** provides comment board services.
- **Zopim, Intercom, Smartsupp** provide live chat services.
- **33across** & **Lockerdome** are advertising platforms.

Initiators and Receivers

Top A&A Initiators

A&A Initiator	#A&A Receivers
facebook	11
google	11
doubleclick	9
youtube	8
addthis	8
hotjar	7
googlesyndication	6
twitter	5
sharethis	4
adnxs	3

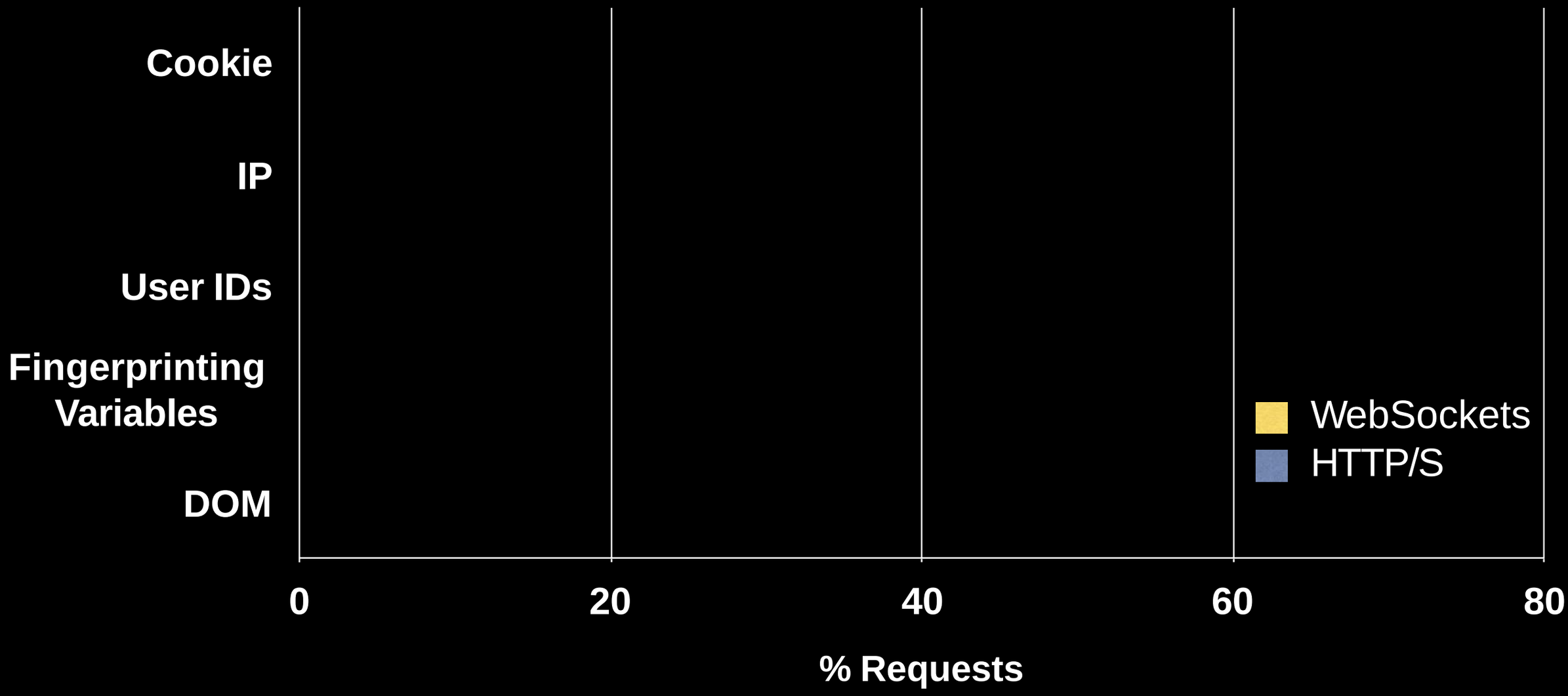
Top A&A Receivers

A&A Receiver	#A&A Initiators
realtime	27
33across	19
intercom	16
disqus	13
zopim	12
hotjar	11
feedjit	10
lockerdome	8
inspectlet	6
smartsupp	4

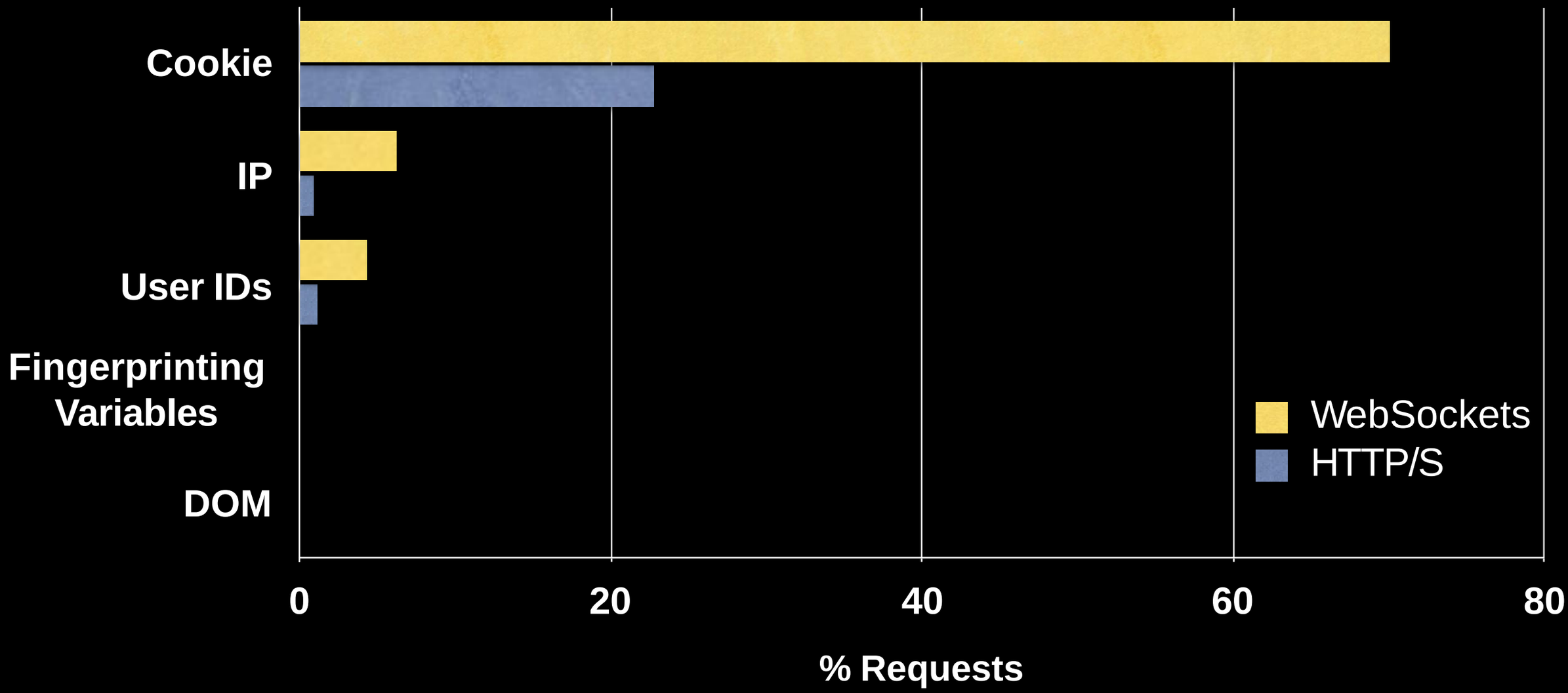
- **Disqus** provides comment board services.
- **Zopim, Intercom, Smartsupp** provide live chat services.
- **33across** & **Lockerdome** are advertising platforms.
- **Inspectlet** & **Hotjar** are session replay services.

Sent Items Over Web Sockets

Sent Items Over Web Sockets

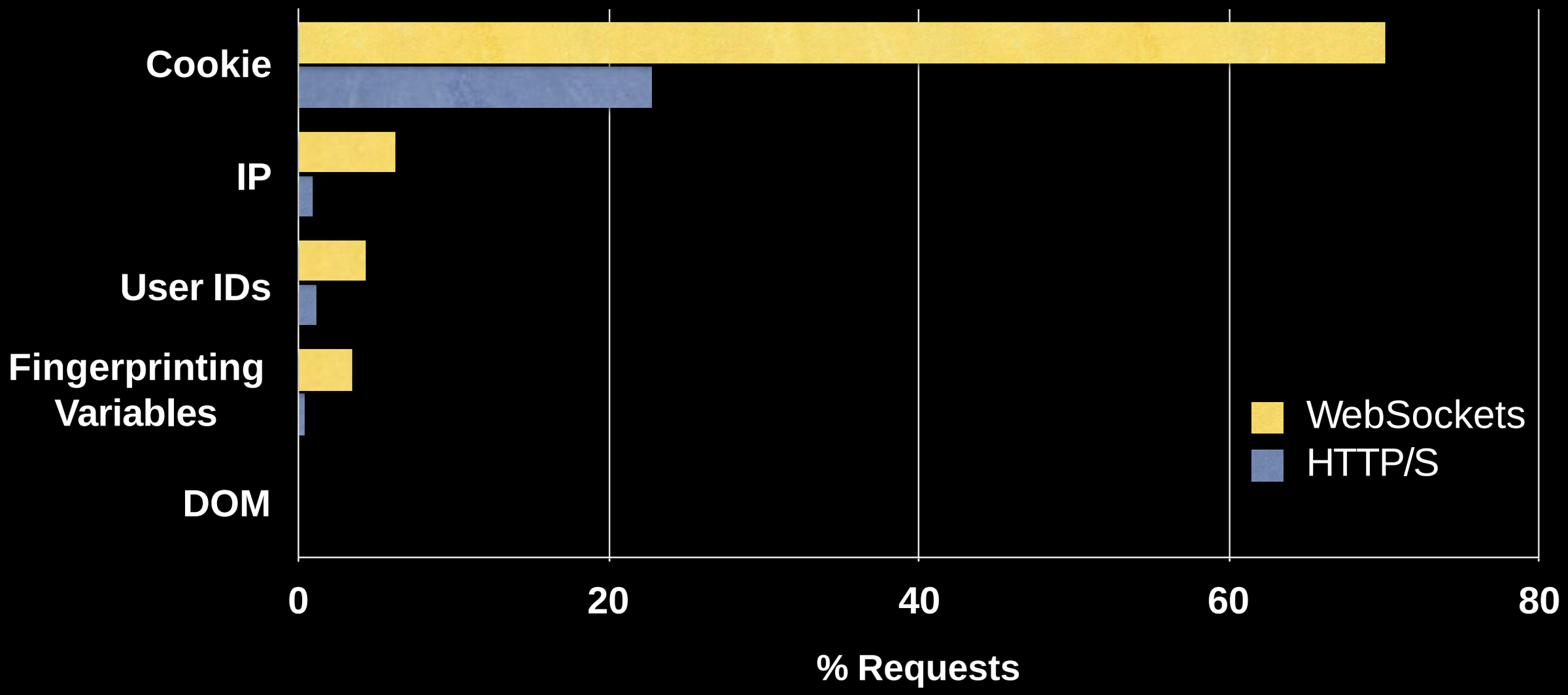


Sent Items Over Web Sockets



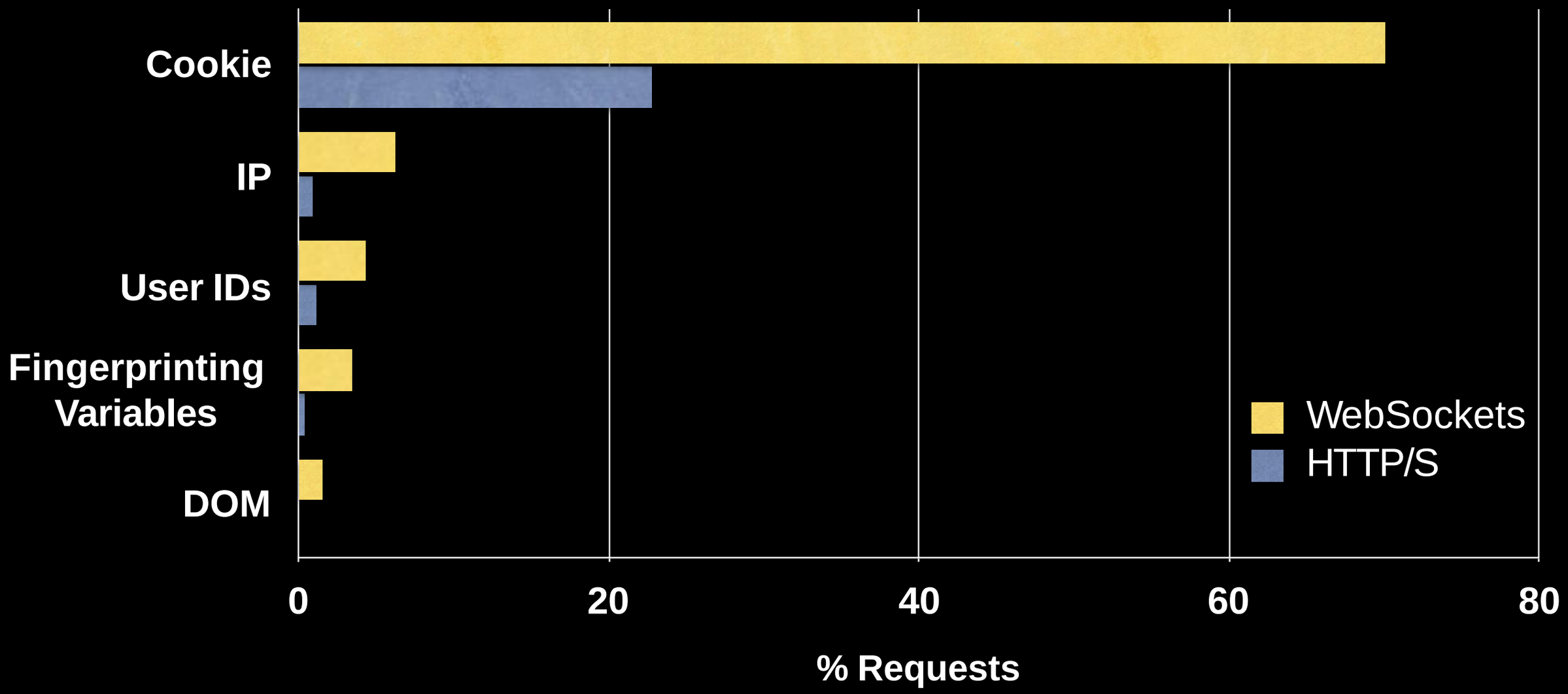
- Stateful Identifiers like Cookies and User IDs

Sent Items Over Web Sockets



- Stateful Identifiers like Cookies and User IDs
- Fingerprinting data in ~3.4% WebSockets.
  97% is **33across**

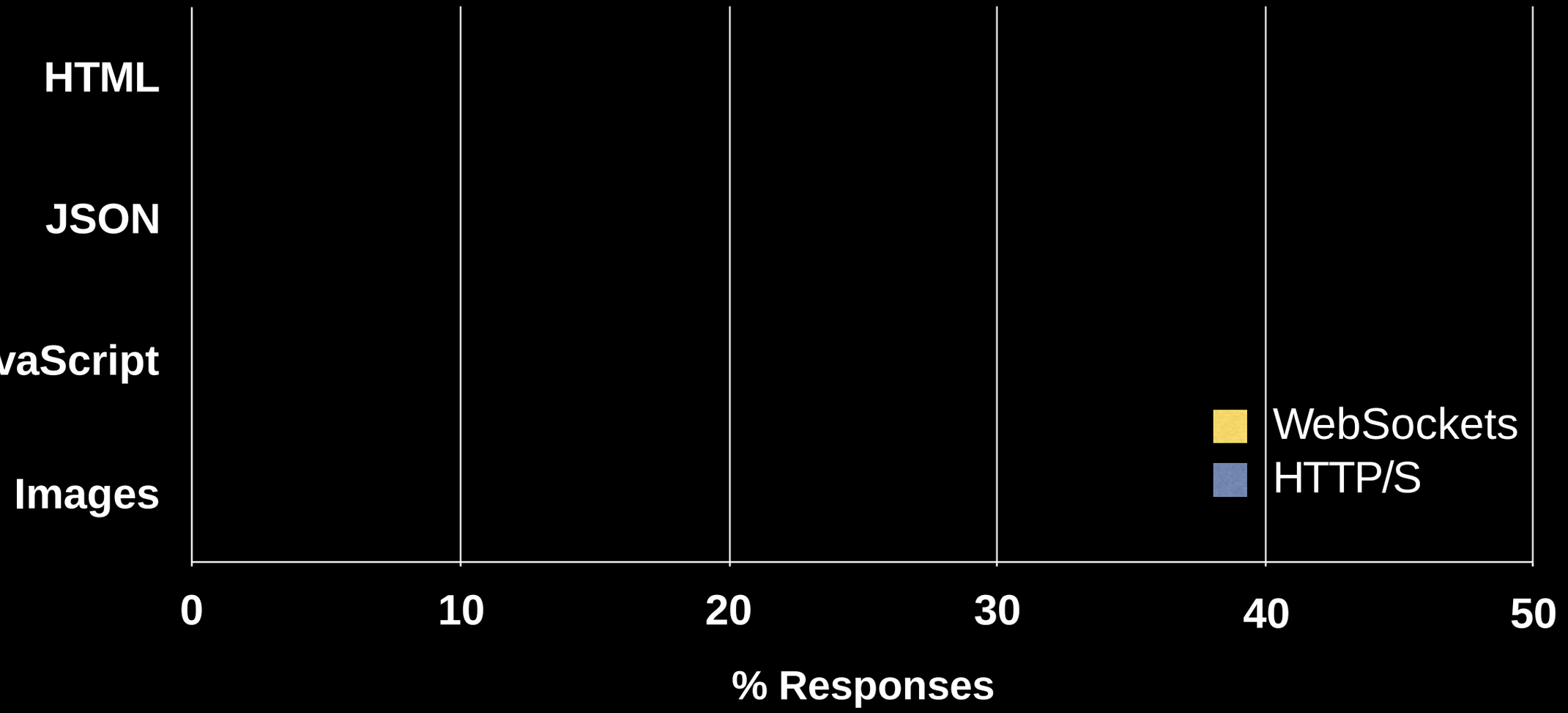
Sent Items Over Web Sockets



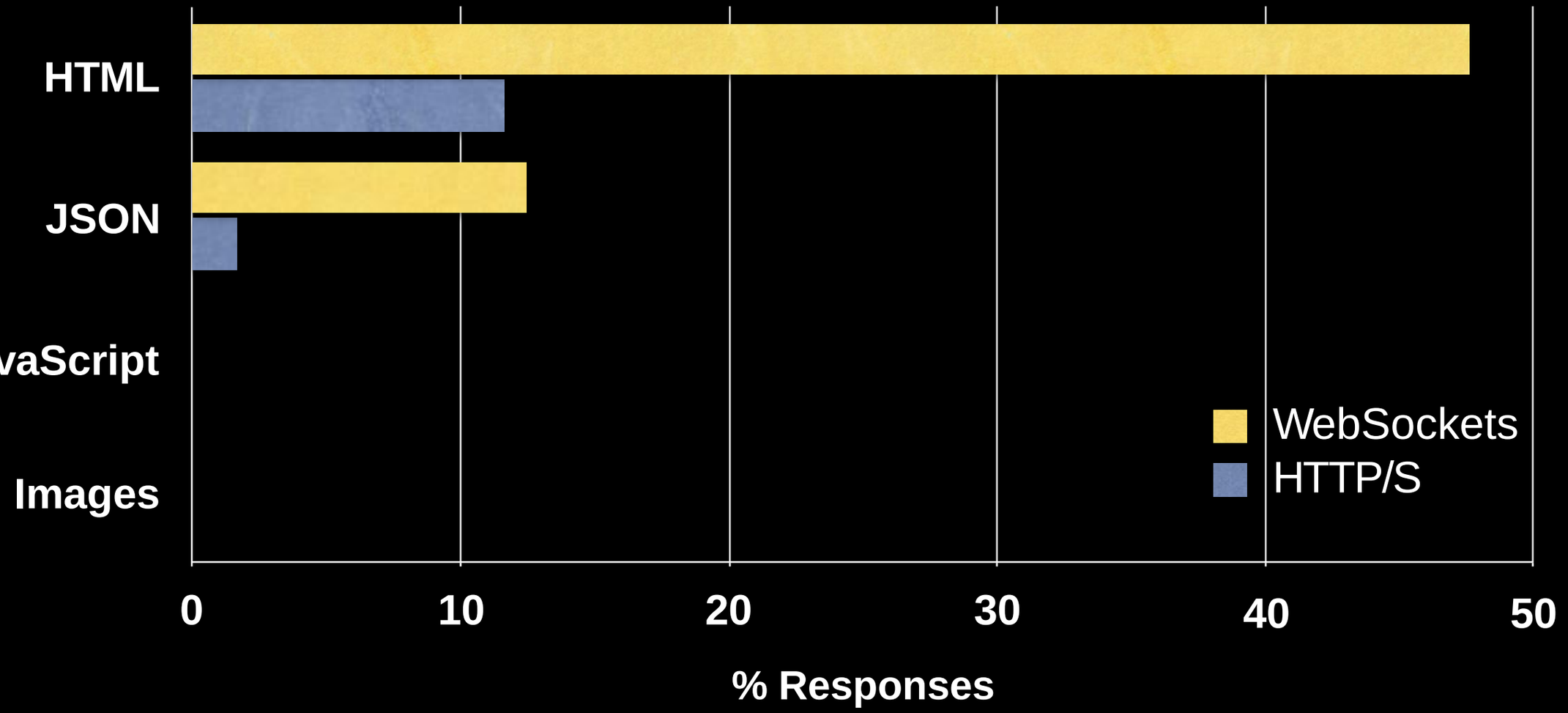
- Stateful Identifiers like Cookies and User IDs
- Fingerprinting data in ~3.4% WebSockets.
  97% is **33across**
- ~1.6% WebSockets sends the entire DOM to
  Hotjar, LuckyOrange, TruConversion

Received Items Over Web Sockets

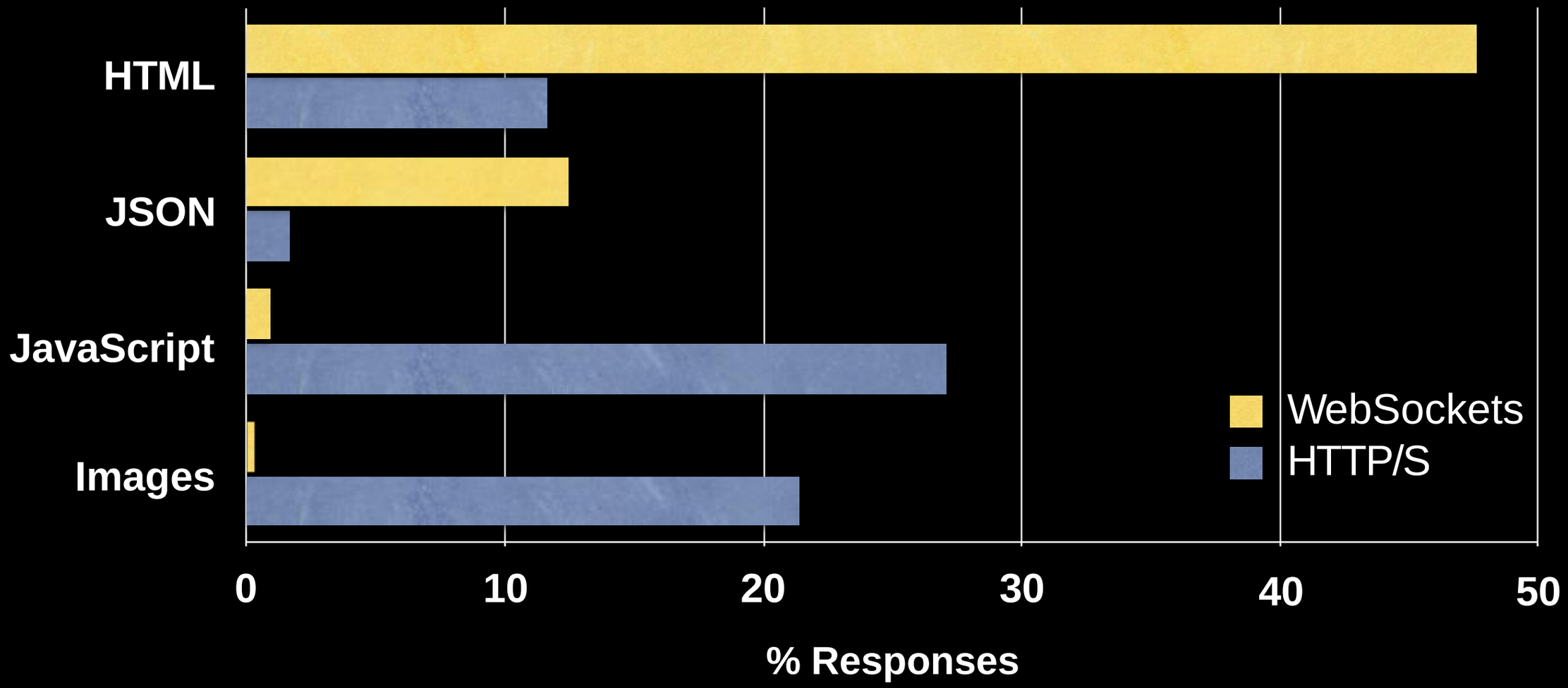
Received Items Over Web Sockets



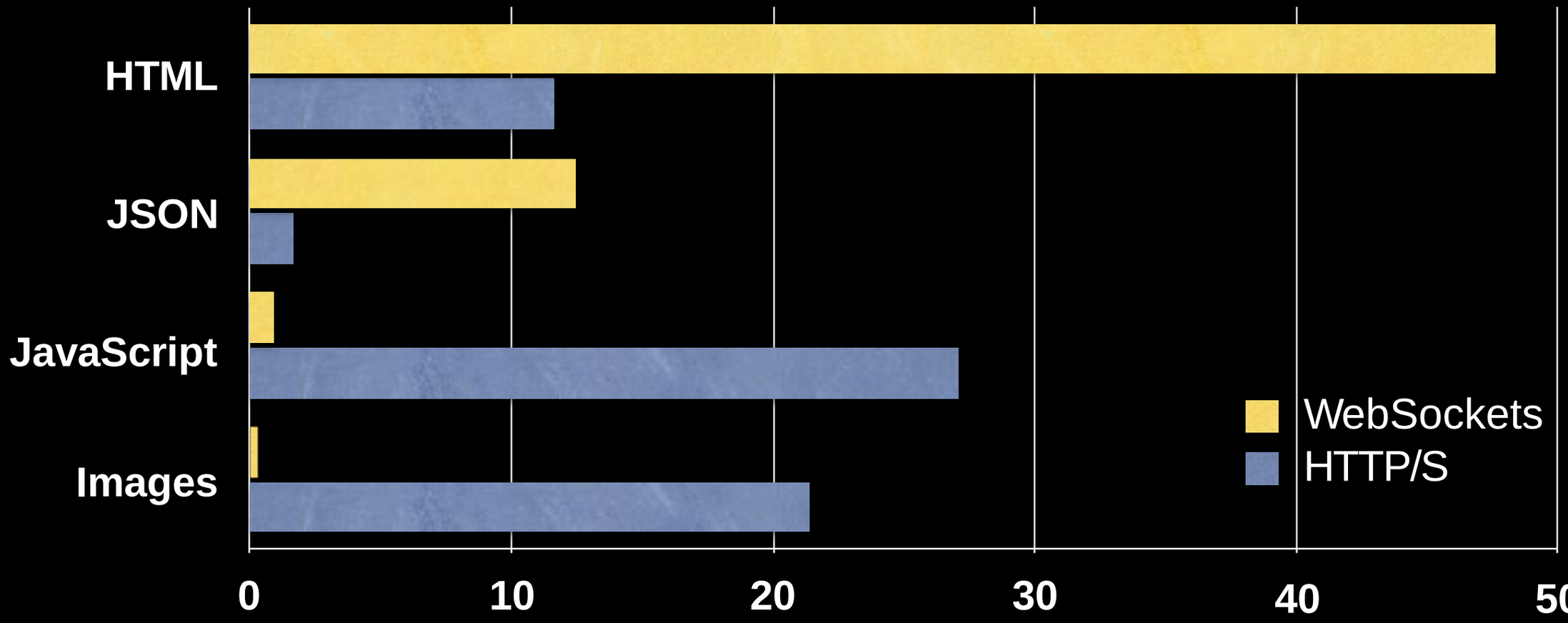
Received Items Over Web Sockets



Received Items Over Web Sockets



Received Items Over Web Sockets



Ads served from **Lockerdome**

Summary

- ~67% of socket connections are initiated or received by A&A domains.
- Major companies like Google, Facebook, Addthis adopted WebSockets. Abandoned after Chrome 58 was released.
- The culprits:
  - **33across** was harvesting fingerprinting data.
  - DOM exfiltration by **HotJar, LuckyOrange, TruConversion**
  - **Lockerdome** downloaded URLs to serve ads.
- We need to keep up with the current practices of A&A companies.

Implications

- How can we stop A&A networks from doing shady things?
- Mechanisms to help prevent tracking
  - Papaodyssefs et al (2015): private cookies to prevent tracking
  - Nikiforakis et al (2015): browser entropy to prevent fingerprinting
- However, ad blocking extensions are not always effective
  - Snyder et al (2016): extensions only blocked 65% of A&A WS connections
  - Franken et al (2018): developer mistakes affecting filters

Eric's Thoughts

- Interesting evaluation, particularly implications on A&A tactics
  - Some willing to exploit browser (security) loopholes to serve ads
  - However, most still did not do this
- Why did this loophole take so long to close (5 years)?
- Authors stated not obvious why stopped using WS after patch released
  - Correlation != causation, but…
  - Obvious educated guess: No more loophole → no more reason to use WS
- Weakness in evaluation: A&A domain matching had issues

Your Thoughts

- Measurement researchers liked it, others not so much
  - Reethika and Ram both rated it “Accept”
  - Jiacheng didn’t like it (“Weak Reject”)
  - Ben really didn’t like it (“Reject”)
  - I liked it (“Accept”)
  - Average score was 3.25 (as of earlier this afternoon)
- Jiacheng: Good that authors named A&A companies
  - Agreed! It was quite nice vs other papers that say “Company A”
- Ben: Conclusion was obvious, no suggestions on how to fix problem
  - I disagree somewhat. They were seeking to gather data about prevalence of well-known issue
  - Issue was already patched by Google as of publication
- Ed and Junpeng: Needs discussion of why WS usage ceased after patch
  - Agreed! I thought this as well during my reading
- Reethika: Alexa isn’t a good source of domains for experiments
  - This list can change rapidly (Reethika cited another IMC ‘18 paper stating this)

Your Thoughts

- Can and Ram: Why did it take Google (also does A&A) five years to fix this bug?
  - Very valid question
- Steve: They didn't need to build their own inclusion tree analysis tools
  - Steve: Two tools already exist that have been used in prior research
- Can: Authors state that many companies have legitimate uses for WS
  - Some need them for legitimate purposes, but others used them to work around user preferences
- Kevin: Observational, not empirical study → can't draw conclusions

Backup Slides

Inclusion Chain

DOM Tree

```
<html>
  <body>
    <script src="tracker/script.js" </script>
    <img src="tracker/img.jpg"> </img>

    <script src="ads/script.js"> </script>
    <iframe src="frame.html">
        <html> <body>
          <script src="script_12.js"> </script>
          <img src="img_a.jpg"> </img>
        </body> </html>
      </iframe>
    </body>
  </html>
```

Source code for ads/script_12.js

```
let ws =
  new WebSocket("ws://adnet/data.ws", …);
  ws.onopen = function (e) {ws.send("…");}
```

Inclusion Tree