Local Frames: Exploiting Inherited Origins to Bypass Content Blockers

Alisha Ukani*, Hamed Haddadi^, Alex C. Snoeren*, Peter Snyder^
*UC San Diego, ^Brave

To Appear at CCS 2025
IETF-123 PEARG
July 25, 2025

This talk will show how websites can easily bypass most content blockers...

And that 56% of popular websites have the capability to do so!

Outline

- **Background**
  - Local frames, origins, and content blockers
- Content blocker capabilities
- Content blocker evasion via local frames
- Measuring local frame prevalence
- Conclusion

Local Frames

- Local frames are iframes with a non-standard source like “about:blank”

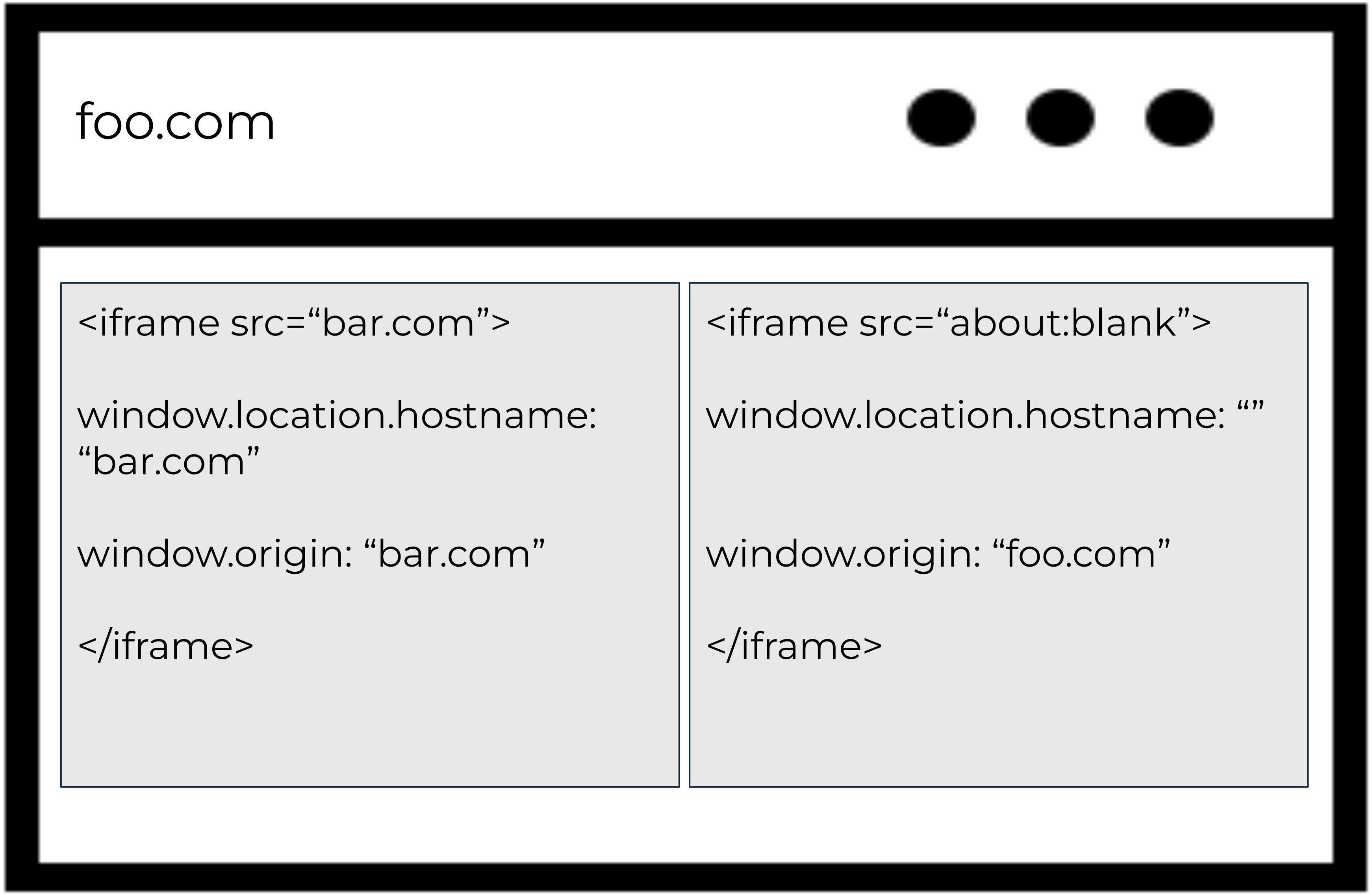
```
<iframe src="about:blank">
    <!-- Your existing code here -->
</iframe>
```

- All iframes get an empty HTML document and a clean environment (e.g. CSS rules, JS environment)
  - Useful isolation mechanism

Origins

- Origin = protocol + hostname + port
- The origin of https://foo.com/bar is https://foo.com:443
- Uses:
  - Same-origin policy
  - Content blocking: whether to allow or block
- Normally derived from frame's URL…but local frames **inherit** the origin of their parent frames

It’s very easy to handle local frames incorrectly!



Content Blockers

Content blockers are tools that:

- Hide ads
- Block tracking scripts
- Reduce mobile data usage

Popular content blockers are used by **millions of people**

Image credit: 1) ABP logo: https://www.logo.wine/logo/Adblock_Plus 2) uBO logo: https://en.m.wikipedia.org/wiki/File:UBlock_Origin.svg 3) Brave logo: https://brave.com/brave-branding-assets/ 4) DuckDuckGo logo: https://www.logo.wine/logo/DuckDuckGo

Outline

- Background
  - Local frames, origins, and content blockers
- **Content blocker capabilities**
- Content blocker evasion via local frames
- Measuring local frame prevalence
- Conclusion

An Example Website

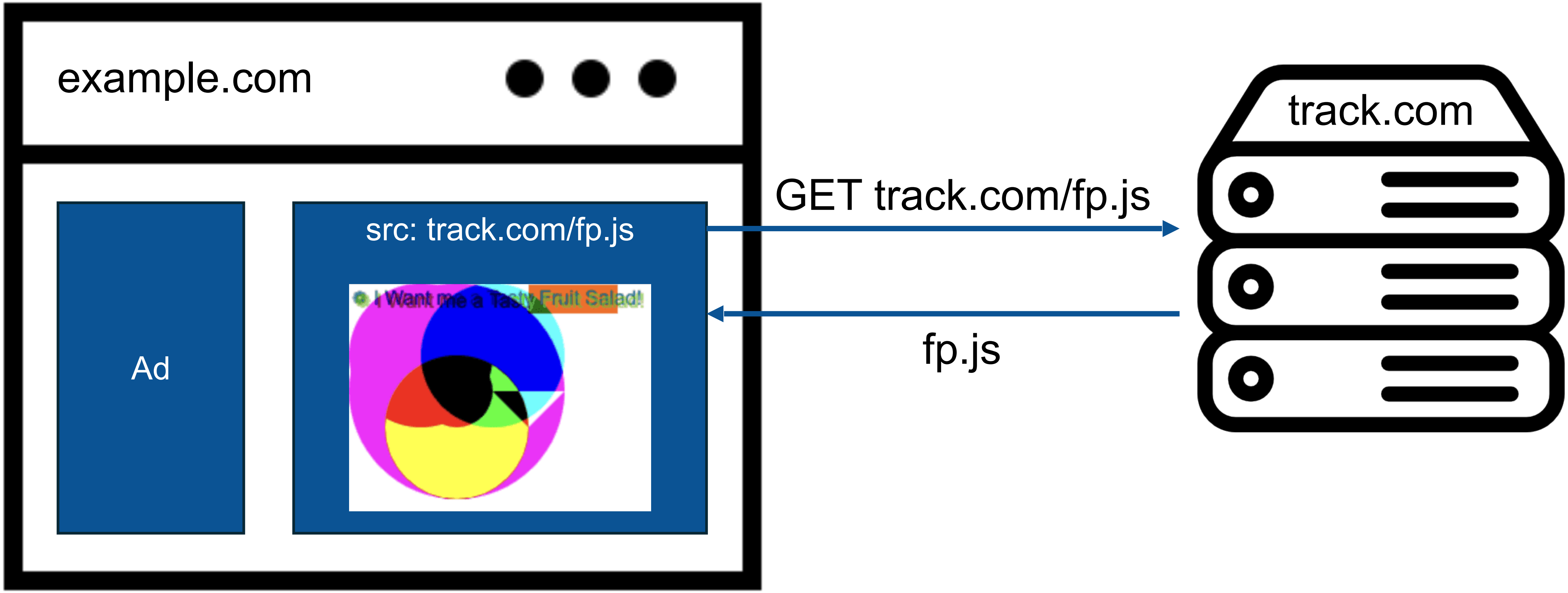


Image credit: Pixel Perfect

Capability 1: Request Blocking

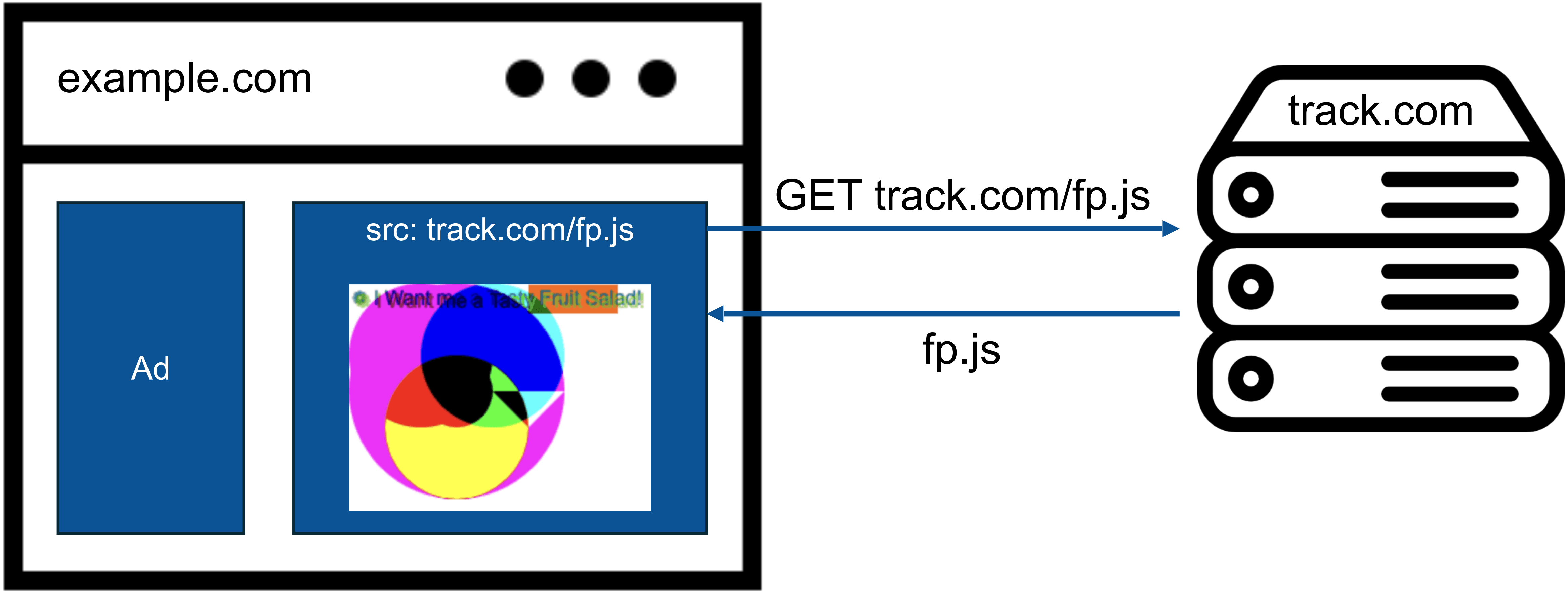


Image credit: Pixel Perfect

Capability 1: Request Blocking

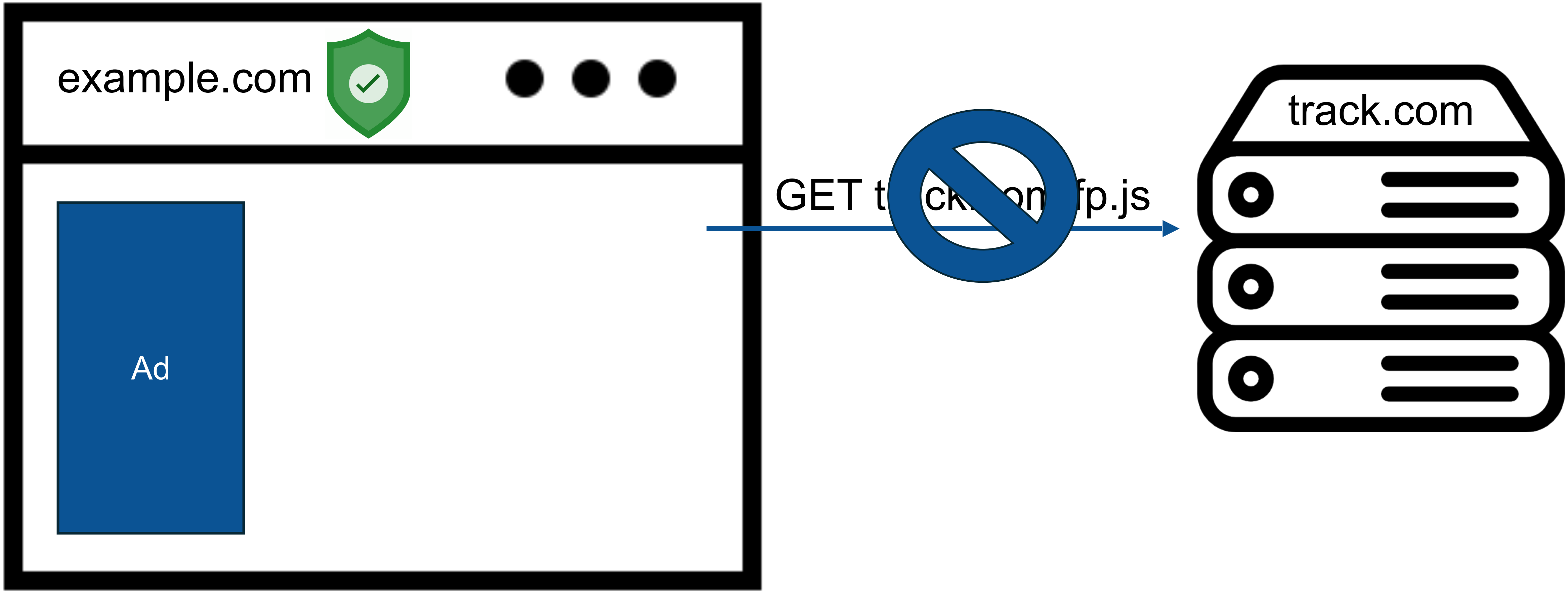


Image credit: Pixel Perfect

Capability 2: Resource Replacement

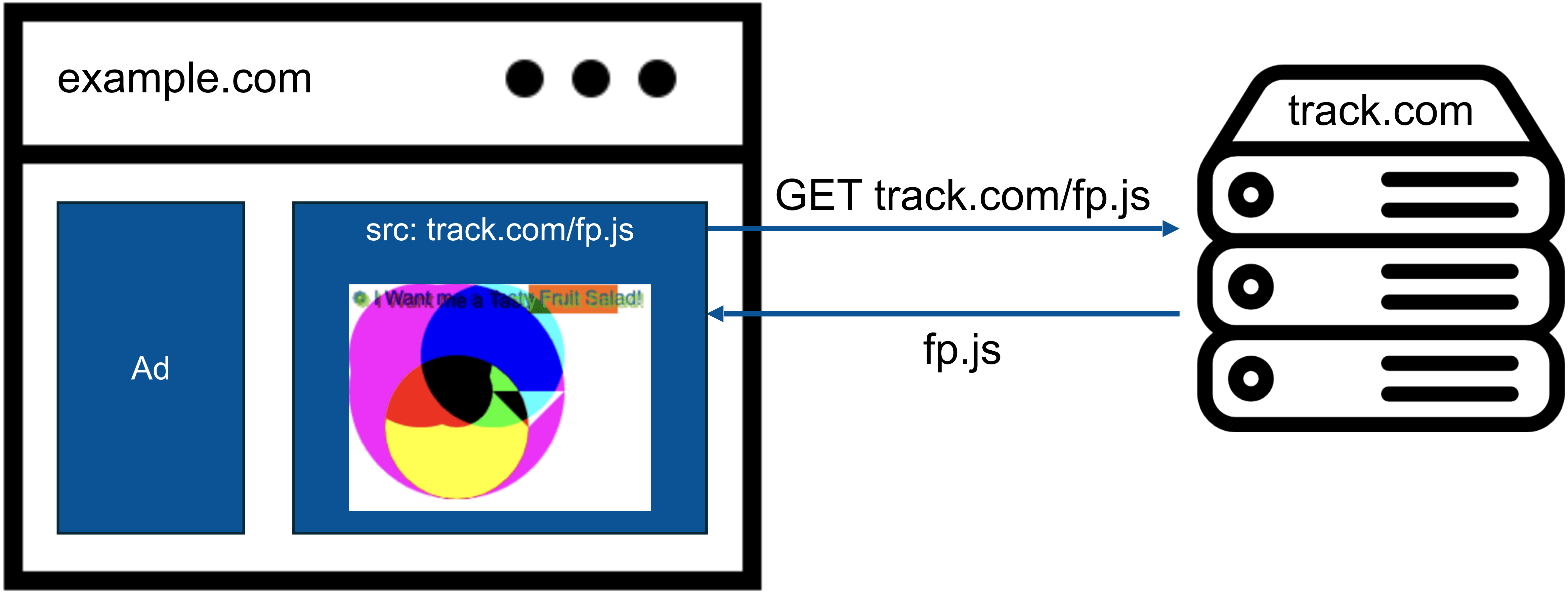


Image credit: Pixel Perfect

Capability 2: Resource Replacement

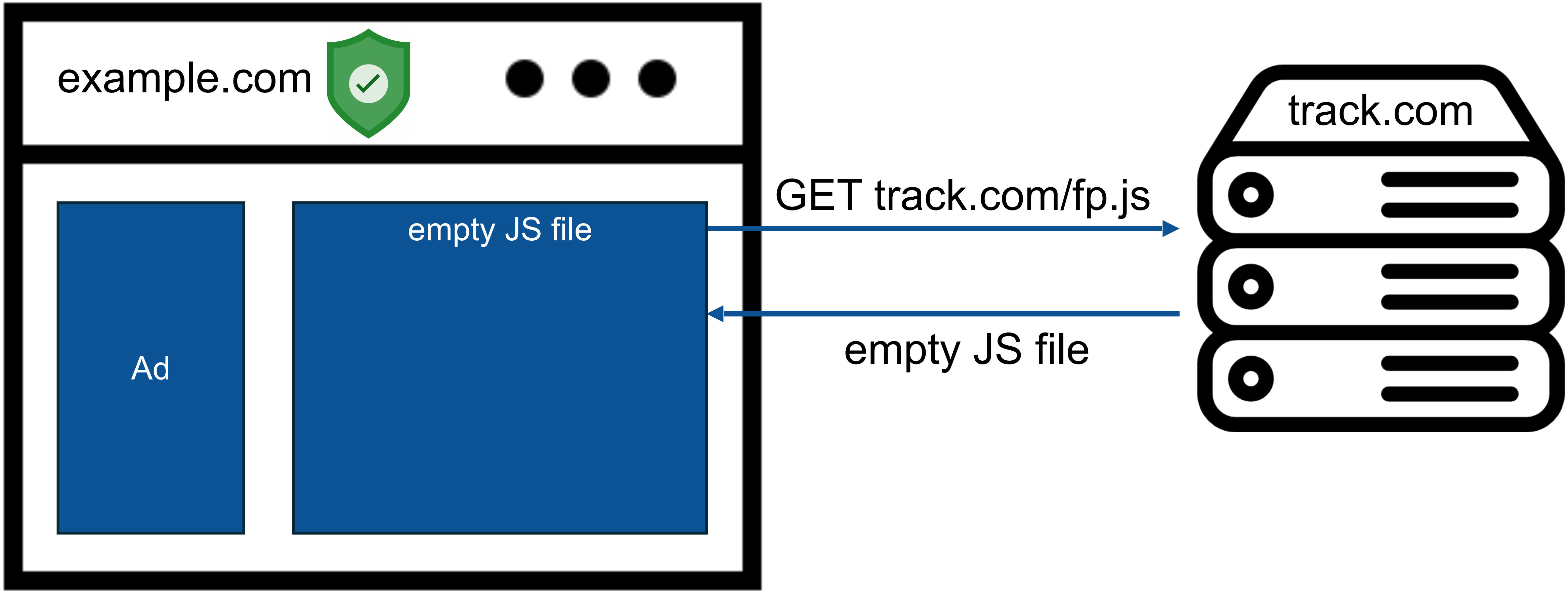


Image credit: Pixel Perfect

Capability 3: Scriptlet Injection

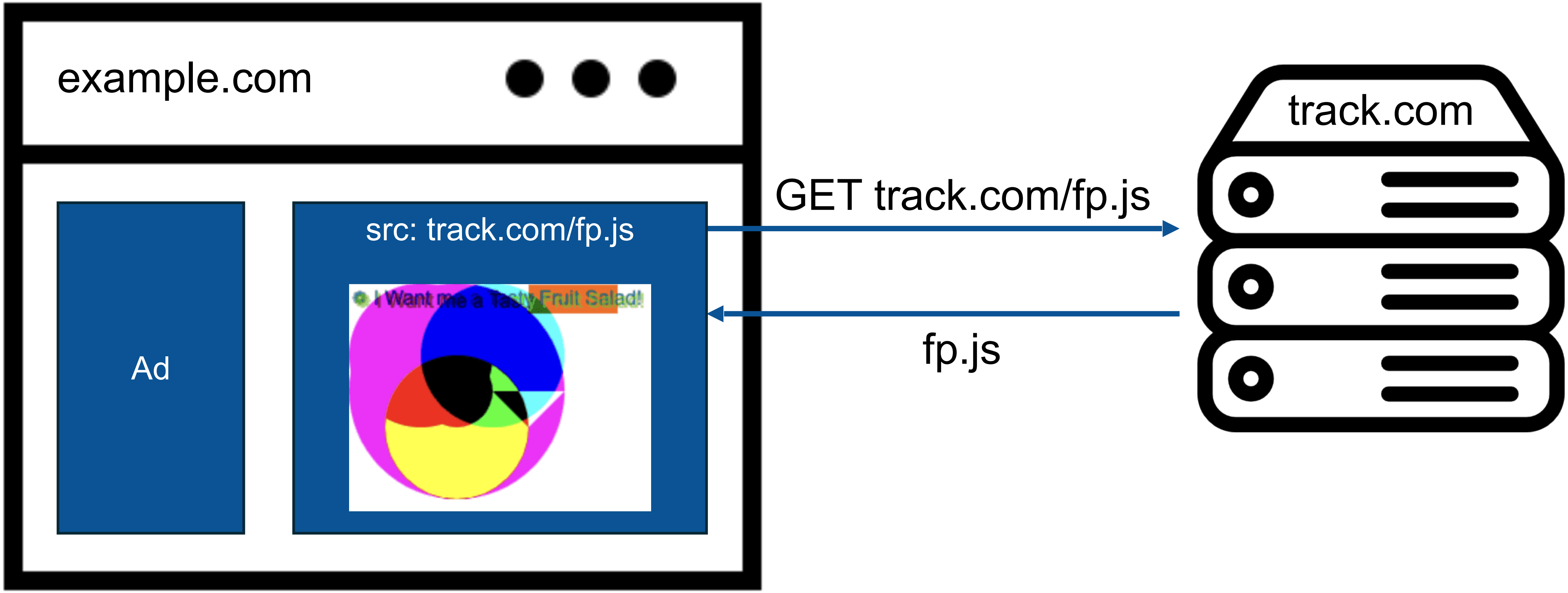


Image credit: Pixel Perfect

Capability 3: Scriptlet Injection

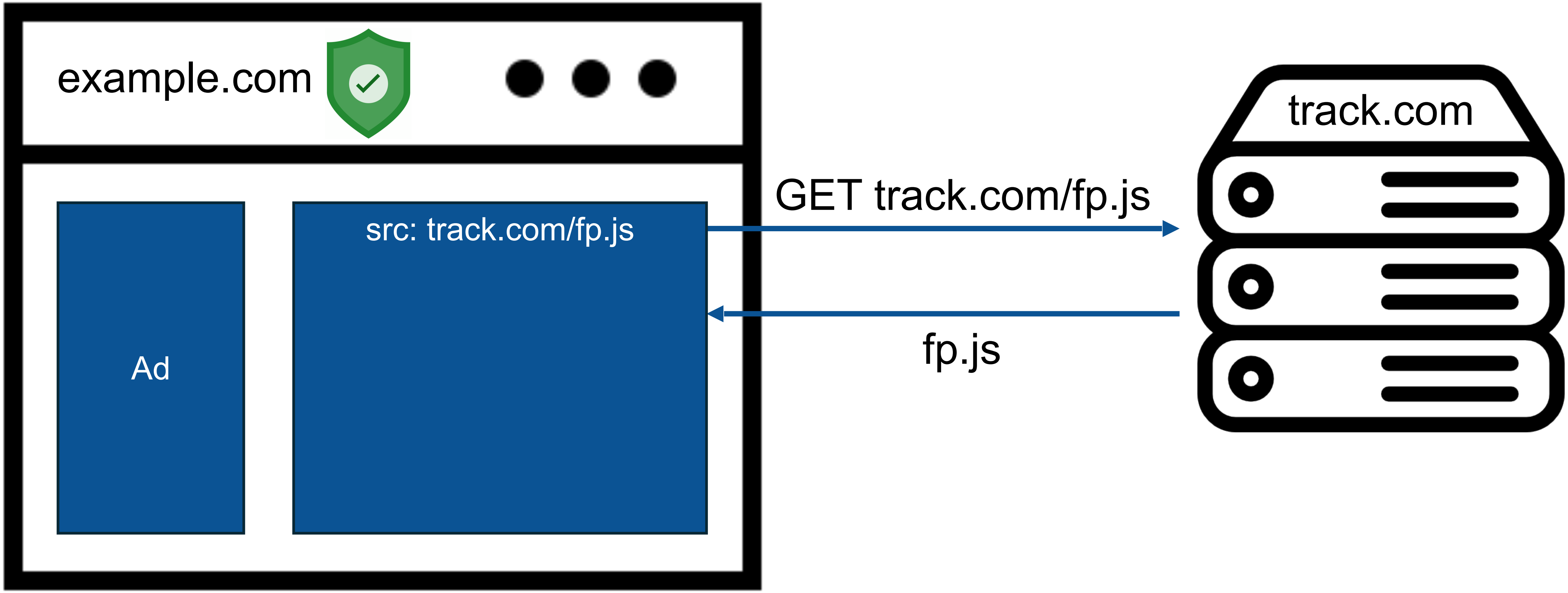


Image credit: Pixel Perfect

Capability 4: Cosmetic Filtering

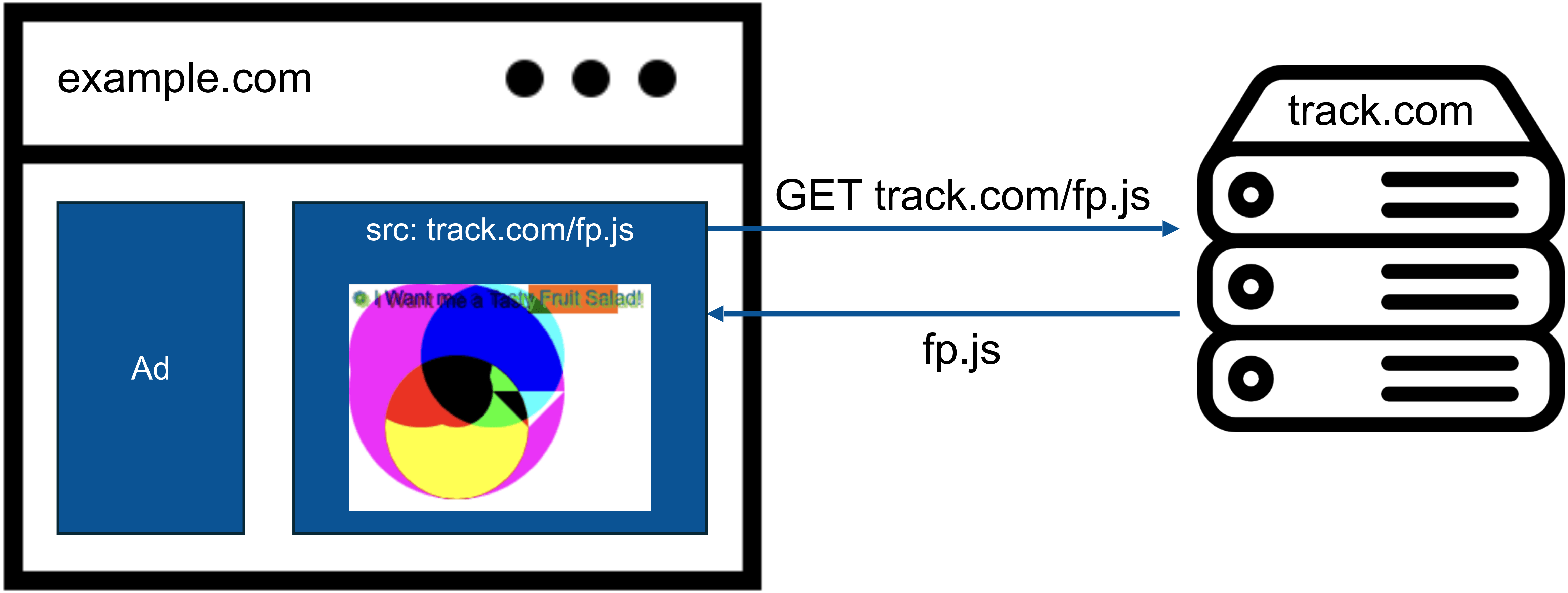


Image credit: Pixel Perfect

Capability 4: Cosmetic Filtering

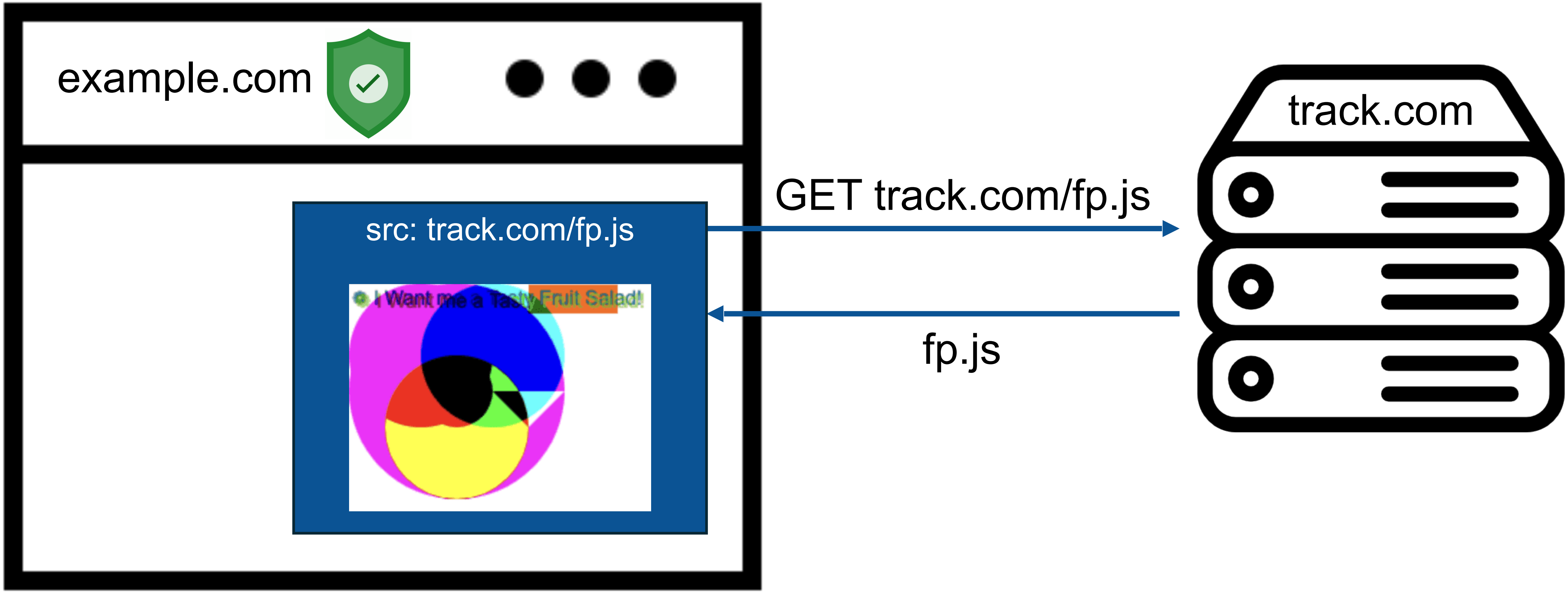


Image credit: Pixel Perfect

Outline

- Background
  - Local frames, origins, and content blockers
- Content blocker capabilities
- **Content blocker evasion via local frames**
- Measuring local frame prevalence
- Conclusion

Testing Evasion

- **Request blocking:** if we try to block a request made inside of a local frame, does it get blocked?
- **Resource replacement:** if we redirect the response of an AJAX request to an empty file, does the redirection happen?
- **Scriptlet injection:** if we set a property of the window object, do we read the correct value?
- **Cosmetic filters:** if we try to hide an element, is the correct element hidden?

19 Vulnerabilities Found

Every tool that hides ads had some vulnerable offering

Tool	Platform	Request Blocking (RQ1)	Resource Replacement (RQ2)	Scriptlet Injection (RQ3)	Cosmetic Filters (RQ4)
Safari Content Blocker	MacOS	○	N/A	N/A	●
AdBlock Plus	Chrome Extension	○	○	◐	○
	Firefox Extension	○	○	○	○
	iOS	○	N/A	N/A	●
uBlock Origin	Chrome Extension	○	○	◐	○
	Firefox Extension	○	○	○	○
	Chrome MV3 (uBlock Origin Lite)	○	○	○*	●
AdGuard	Chrome Extension	○	○	●	●
	Firefox Extension	○	○	●	●
	iOS	○	N/A	N/A	●
Brave Browser	Desktop	○	○	●	●
	iOS	●	●	●	●
	Android	○	○	●	●
DuckDuckGo	Chrome Extension	○	○	N/A	N/A
	Firefox Extension	○	○	N/A	N/A
	Desktop	●†	○	N/A	N/A
	iOS	○	○	N/A	N/A
	Android	○	○	N/A	N/A

All Brave iOS protections could be bypassed

Disclosure

- We disclosed our findings to all affected parties
- Brave, Apple, AdGuard, and DuckDuckGo finished their patches
  - **15/19 vulnerabilities fixed**
- Our disclosure to uBlock Origin helped them identify new bypasses in uBlock Origin Lite

Case Study: AdGuard

- AdGuard computes the origin of third-party local frames to be the first-party website
- Consequence 1: evasion
  - Rules meant for the third-party frame are not applied
- Consequence 2: breakage
  - Rules meant for a different website are applied to these frames
  - Scriptlets can disable standard Web development tools, e.g., disabling event listeners

Case Study: Safari Content Blocker API

- API that lets app define which requests should be blocked without viewing network traffic
- Cosmetic filtering is vulnerable to evasion via local frame
- **All apps relying on this API are also vulnerable**

Case Study: Brave

- Brave tries to comprehensively protect users from tracking
- But also added code complexity, which then introduced more avenues for evasion

Case Study: DuckDuckGo

- Correctly blocks requests to trackers made inside nested local frames
- But does not report these blocked requests to the user
- Even if a capability can’t be evaded, there are many subtle ways to incorrectly handle local frames

Outline

- Background
  - Local frames, origins, and content blockers
- Content blocker capabilities
- Content blocker evasion via local frames
- **Measuring local frame prevalence**
- Conclusion

Measuring Local Frames

- We crawl 21,956 websites of the Tranco 1M most popular websites
- We measure:
  - **Prevalence** of local frames
  - **Requests** made inside local frames
  - **JavaScript** API calls
    - **Fingerprinting**-related API calls
  - **HTML** elements created inside the frames

Local Frames are Widespread

- Local frames are found on **56%** of websites (12,234 sites)
- **62%** of websites with local frames make requests to tracking resources
  - This represents 14% of all websites we crawled
- **35%** of websites with local frames make calls to fingerprinting-related APIs
- Third-party local frames are primarily created by **analytics & advertising** companies

Summary

- Local frames are widespread and could evade content blockers
- Our work led to patches in Safari, Brave, AdGuard, uBlock Origin Lite, and DuckDuckGo
- Developers need help in navigating web complexity!

Paper

Contact:

aukani@ucsd.edu

alishaukani.com

Dataset