



A Deep Dive Into SameSite Cookies

What They Are and Why They Matter

Stephen Rees-Carter

twitter.com/valorin

Senior Developer

at Defiant / Wordfence

HTTPbis
Internet-Draft
Updates: [6265](#) (if approved)
Intended status: Standards Track
Expires: April 30, 2015

M. West
Google, Inc
October 27, 2014

First-Party Cookies
draft-west-first-party-cookies-00

Abstract

This document updates [RFC6265](#), defining the "First-Party" attribute for cookies, which allows servers to mitigate the risk of cross-site request forgery and related information leakage attacks by asserting that a particular cookie should only be sent in a "first-party" context.

<https://tools.ietf.org/html/draft-west-first-party-cookies-00>

Timeline

- **October 2014**
 - “First-Party” cookie attribute proposed
- **February 2015**
 - Attribute changed to “First-Party-Only”
- **January 2016**
 - Attribute changed to “SameSite”
- **April 2016**
 - Attribute option values added:
SameSite=Strict
SameSite=Lax
SameSite=None
- **May 2016**
 - Google Chrome 51 adds support

CSRF

Cross Site Request Forgery

User updates their own password:



Step #1
POST <https://mysite.com/account>
password=correct+horse+battery+staple



CSRF

Cross Site Request Forgery

Evil Hacker updates user's password:



Step #3
(via Javascript sent to the user's browser)
POST <https://mysite.com/account>
password=evil+hacker+passwd



Step #1
(Trick the user into visiting)
<https://anothersite.com>



Step #2
GET <https://anothersite.com>



CSRF

Defending Against Attack

1. CSRF tokens or Nonces
 - a. Required in all requests
 - b. Known-secret based protection
2. Verify Origin or Referer header
 - a. Cannot be modified by client
3. Client-side cryptographic magic
 - a. Some apps work in specific ways

+ SameSite Cookies!

SameSite cookie attribute

Set-Cookie: app_session=eyJpdiI6ImNQWTBCU3VERW...;

Set-Cookie: app_session=eyJpdiI6ImNQWTBCU3VERW...; **SameSite=Strict**

Set-Cookie: app_session=eyJpdiI6ImNQWTBCU3VERW...; **SameSite=None; Secure**

Set-Cookie: app_session=eyJpdiI6ImNQWTBCU3VERW...; **SameSite=Lax**

Can I use

SameSite

? ⚙ Settings

6 results found



#

'SameSite' cookie attribute - OTHER

Usage

% of all users

?

Global

86.19% + 7.4% = 93.59%

★

Same-site cookies ("First-Party-Only" or "First-Party") allow servers to mitigate the risk of CSRF and information leakage attacks by asserting that a particular cookie should only be sent with requests initiated from the same registrable domain.

Current aligned

Usage relative

Date relative

Filtered

All



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android
	12-15		4-50			3.2-11.4					
	¹ 16-17	2-59	51-79	3.1-11.1	10-38	⁵ 12-12.4					
6-10	18-84	60-80	³ 80-84	⁴ 12-13.1	39-70	13-13.7		2.1-4.4.4	12-12.1		
¹ ² 11	85	81	³ 85	⁵ 14	71	14	all	81	59	³ 85	79
		82-83	³ 86-88	TP							

SameSite=Strict

Cross-Site/Third-Party Requests

Blocks all CSRF attacks (when cookies are required)

Embedded Content

`<iframe>` ``



Unsafe Requests

`POST/PUT/DELETE/...`



Safe Requests

`GET/HEAD`



SameSite=None; Secure

Cross-Site/Third-Party Requests

(HTTPS Only)

Blocks no CSRF attacks

Embedded Content

`<iframe>` ``



Unsafe Requests

POST/PUT/DELETE/...



Safe Requests

GET/HEAD



SameSite=None

(Without "Secure")

Will not be sent on any request, HTTPS or HTTP.



SameSite=Lax

Cross-Site/Third-Party Requests

Blocks CSRF attacks on “unsafe” requests.

Embedded Content

`<iframe>` ``



Unsafe Requests

POST/PUT/DELETE/...



Safe Requests

GET/HEAD





Story Time...

⚠ A cookie associated with a cross-site resource at <https://hackwp.valorin.dev/> was set without the `(index):1` ``SameSite`` attribute. It has been blocked, as Chrome now only delivers cookies with cross-site requests if they are set with ``SameSite=None`` and ``Secure``. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <https://www.chromestatus.com/feature/5088147346030592> and <https://www.chromestatus.com/feature/5633521622188032>.

Improving privacy and security on the web

Tuesday, May 7, 2019

Improving cookie controls in Chrome

We announced at I/O that we will be updating Chrome to provide users with more transparency about how sites are using cookies, as well as simpler controls for cross-site cookies. We will preview these new features later this year.

We are making a number of upcoming changes to Chrome to enable these features, starting with modifying how cookies work so that developers need to explicitly specify which cookies are allowed to work across websites — and could be used to track users. The mechanism we use builds on the web's SameSite cookie attribute, and you can find the technical details on web.dev.

Developers: Get Ready for New SameSite=None; Secure Cookie Settings

Wednesday, October 23, 2019

Chrome Enforcement Starting in February 2020

With Chrome 80 in February, Chrome will treat cookies that have no declared SameSite value as `SameSite=Lax` cookies. Only cookies with the `SameSite=None; Secure` setting will be available for external access, provided they are being accessed from secure connections. The Chrome Platform Status trackers for [SameSite=None](#) and [Secure](#) will continue to be updated with the latest launch information.

Mozilla has affirmed their support of the new cookie classification model with their [intent to implement](#) the `SameSite=None; Secure` requirements for cross-site cookies in Firefox. Microsoft recently [announced](#) plans to begin implementing the model starting as an experiment in Microsoft Edge 80.

- **SameSite by default cookies**

Treat cookies that don't specify a SameSite attribute as if they were SameSite=Lax. Sites must specify SameSite=None in order to enable third-party usage. – Mac, Windows, Linux, Chrome OS, Android

[#same-site-by-default-cookies](#)

Enabled



- **Cookies without SameSite must be secure**

If enabled, cookies without SameSite restrictions must also be Secure. If a cookie without SameSite restrictions is set without the Secure attribute, it will be rejected. This flag only has an effect if "SameSite by default cookies" is also enabled. – Mac, Windows, Linux, Chrome OS, Android

[#cookies-without-same-site-must-be-secure](#)

Enabled



SameSite Cookie Changes in February 2020: What You Need to Know

Monday, February 3, 2020

With the stable release of Chrome 80 this month, Chrome will begin enforcing a new secure-by-default cookie classification system, treating cookies that have no declared SameSite value as `SameSite=Lax` cookies. Only cookies set as `SameSite=None; Secure` will be available in third-party contexts, provided they are being accessed from secure connections.

Temporarily rolling back SameSite Cookie Changes

Friday, April 3, 2020

With the stable release of [Chrome 80 in February](#), Chrome began enforcing secure-by-default handling of third-party cookies as part of our [ongoing](#) effort to improve privacy and security across the web. We've been [gradually](#) rolling out this change since February and have been closely monitoring and evaluating ecosystem impact, including proactively reaching out to individual websites and services to ensure their cookies are labeled correctly.

However in light of the extraordinary global circumstances due to COVID-19, we are temporarily rolling back the enforcement of SameSite cookie labeling, starting today. While most of the web ecosystem was prepared for this change, we want to ensure stability for websites providing essential services including banking, online groceries, government services and healthcare that facilitate our daily life during this time. As we roll back enforcement, organizations, users and sites should see no disruption.



Gianni D'Hooge
@giannidhooge



Replying to @valorin and @LaraconEU

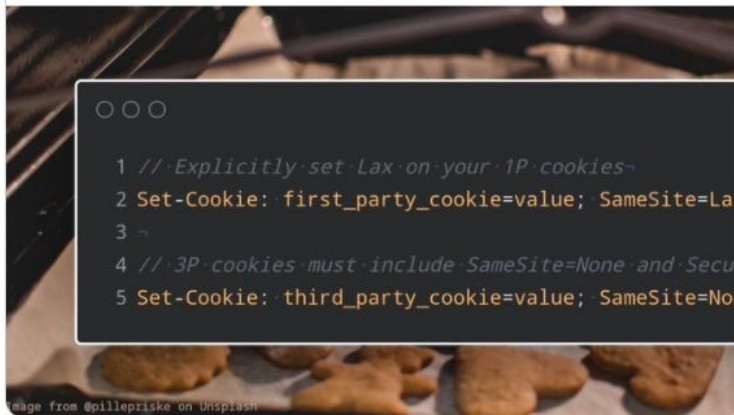
"At some point the default is going to change" you said during the talk. A few hours later, Google announced that they will re-enable Lax by default:



Chrome Developers @ChromiumDev · May 29

Cookie news 🍪 We are going to start re-enabling `SameSite=Lax` by default and requiring `SameSite=None; Secure` with the Chrome 84 stable release on July 14. Make sure you're ready and [📖](#) read on for resources.
blog.chromium.org/2020/05/resumi...

[Show this thread](#)

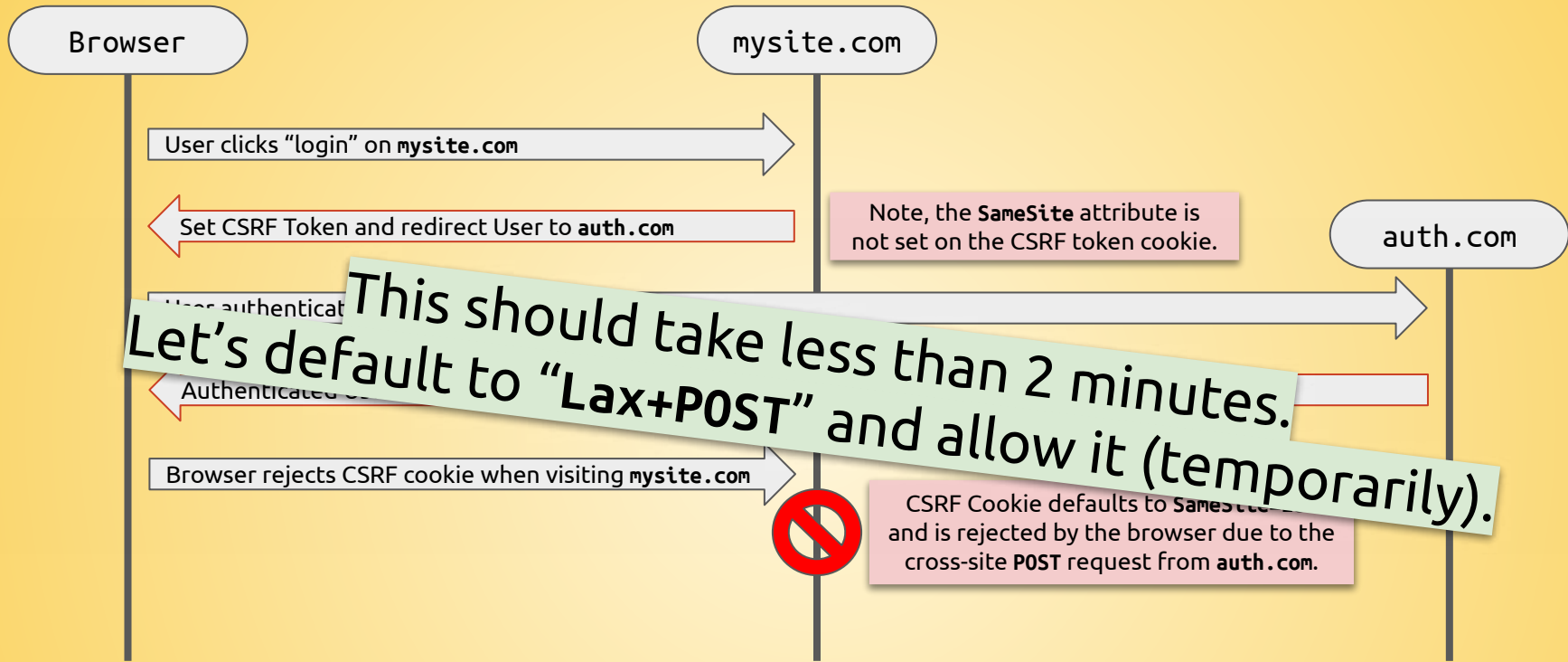


5:49 AM · May 29, 2020 · Twitter Web App

Timeline

- **October 2014**
 - “First-Party” cookie attribute proposed
- **February 2015**
 - Attribute changed to “First-Party-Only”
- **January 2016**
 - Attribute changed to “SameSite”
- **April 2016**
 - Attribute options added:
SameSite=Strict
SameSite=Lax
SameSite=None
- **May 2016**
 - Google Chrome 51 adds support
- **May 2019**
 - Google Chrome proposes *SameSite=Lax* by default in Chrome 77 (September 2019)
- **June 2019**
 - SameSite=Lax by default delayed until Chrome 80 (February 2020)
- **February 2020**
 - Chrome 80 released, *SameSite=Lax* by default rollout started
- **April 2020**
 - Rollout stopped due to COVID-19
- **May 2020**
 - Rollout announced for Chrome 84 (July 2020)
- **July 2020**
 - Rollout begins
- **August 2020**
 - Rollout complete 🎉

We're not finished yet...



Wait a sec... this will break my Auth flow! 

(Example based from the widely used OpenID Connect authentication flow used by Azure Active Directory and Microsoft Account authentication.)

"Same-Site" domains

mysite.com

static.mysite.com

account.mysite.com

“Cross-Site” domains

github.io

valorin.github.io

laravel.github.io

Demo time...

What Option Do I Use?

- Use **SameSite=Strict** if...
 - User shouldn't be automatically logged in
 - Actions must be performed over **GET** requests
- Use **SameSite=None** if...
 - **POST** requests or embedded content (`<iframe>/`) needed between third-party domains
- Use **<nothing>** if...
 - You like unexpected behaviour to confuse your users
- Otherwise, just use **SameSite=Lax**.

Is

SameSite=Lax by default
the answer?



Stephen Rees-Carter @valorin · Jan 23

Ok, web security people... Is CSRF dead now that the major browsers are going to default cookies to SameSite=Lax?



4



1



1



n00py
@n00py1

Replying to @valorin and @mmaunder

Unlikely. Even with samesite=lax 90% of my CSRF PoCs will still work.

1:46 PM · Jan 23, 2020 · [Twitter for iPhone](#)



n00py
@n00py1

Replying to @valorin and @mmaunder

Most of it is related to client work, but basically I've found that a lot of the time I can perform state changing actions using GET requests, many actions that typically send POST can also be sent as GET interchangeably.

1:56 PM · Jan 23, 2020 · [Twitter Web App](#)



Thank you!

Questions: #room-5

(Or feel free to DM me)

src.id.au/samesite

stephen@hey.com • twitter.com/valorin