"Pass the Cookie and Pivot to the Clouds"

By wunderwuzzi

Published: 2018-12-16 · Archived: 2026-04-06 03:15:13 UTC

Web Applications and Services use cookies to authenticate sessions and users.

An adversary can pivot from a compromised host to Web Applications and Internet Services by stealing authentication cookies from browsers and related processes. At the same time this technique bypasses most multi-factor authentication protocols.

The reason for this is that the final authentication token that the attacker steals is issued after all factors have been validated. Many users persist cookies that are valid for an extended period of time, even if the web application is not actively used. Cookies can be found on disk and also in process memory. Additionally other applications on the targets machine might store sensitive authentication tokens in memory (e.g. apps which authenticate to cloud services). This pivoting technique can be extended to bearer tokens, JWT and the likes. Pass the Cookie is a post-exploitation technique to perform session hijacking.

So, let’s Pass the Cookie and Pivot to the Clouds.

Update October 2019: This tactic is now part of the MITRE ATT&CK Matrix, in particular:

- Credential Access - Steal Web Session Cookie
- Lateral Movement - Web Session Cookie

Update December 2018: Pass the Cookie at the Chaos Communication Congress (35C3)

Watch the 35C3 Lightning Talk here:

Attack Chain

Disclaimer: Always make sure you have proper authorization before pen testing.

Pass the Cookie is done via the following steps (variations exist):

1. Acquire the cookie from the victims browser or other processes (e.g. via process dump, or accessing the cookie storage on disk)
2. Exfiltrate the necessary authentication cookies
3. Open Firefox on the attackers machine (or any other machine)
4. Navigate to the resource to access (the domain the cookie is valid for)
5. Use the Developer Console and set the cookie via document.cookie="key=value", or use the UI
6. Refresh the page and observe being logged in as the victim.

The appendix shows examples for Github and Google Cloud Platform by using Google Chrome to pass the cookie.

Detections

When it comes to detections a few things come to mind:

- One can monitor on the client side for applications that perform process dumps on browser processes or others.
- Monitor for unusual activity on critical web assets (like cloud provider management consoles, etc,..)
- Monitor for login anomalies (location, time, unusual access patterns)
- Leverage features that cloud providers and web apps provide (Threat Detection, Access logs,…)

- Perform authorized adverserial emulation in your organization to test detections

Mitigations

To protect from these attacks its important to stay up to date with security patches, etc. to ensure your host does not get compromised. As seasoned security engineer you assume the worst, and here are some ideas on how to mitigate implications of an attack:

- Regularly delete persistent cookies, so they get removed from hard drive to limit exposure.
- Delete session cookies as well
- **Be the only Administrator on your machine**
- Leverage features that cloud providers offer (Threat Detection, IAM, RBAC, Firewalls,…)
- Browse sensitive sites (high value assets) from isolated or dedicated machines
- Seperateion of duties
- Requiring further authentication proof for sensitive operations can help limit the damage
- Requiring client side certificates makes it also more difficult to pass the cookie

Aquiring Cookies, Tools and Techniques

In case you don't won't to write your own toolset, there are a couple of options available to gain access to cookies:

- **firefox_creds** - Access the SQL Lite Cookie Databases
- **cookie_crimes** - Neat way to grab cookies from Chrome on Macs (also Windows and Linux)
- **ProcDump** - Swiss army knife to dump strings from any process

There are also good articles online describing how to access and decrypt the cookies in the SQL Lite databases yourself - if you'd like to do your own research or tool.

Pass The Cookie - Cheat Sheet

Below is a list of some "cookies of interest" for valuable web applications your organization might use. An adversary might be after those and you could emulate to see if your organization catches the attack.

This list might change over time or have inaccuracies - feel to provide feedback or help amend.

Application	Cookie Name	Domain	Notes
Amazon Web Services	aws-userInfo, aws-creds	.amazon.com	https://console.aws.amazon.com

Application	Cookie Name	Domain	Notes
Google Cloud Platform	OSID, HSID, SID, SSID, APISID, SAPISID, LSID	.google.com	https://console.cloud.google.com OSID has to be set on console.cloud.google.com, others on .google.com LSID needed for cross app auth (e.g. GCP to Gmail).
Microsoft Online	ESTSAUTHPERSISTENT	.microsoftonline.com	
Facebook for Work	c_user, cs	.facebook.com	Also works for regular Facebook
OneLogin	sub_session_onelogin.com	.onelogin.com	
GitHub	user_session	.github.com	
Hotmail, Calendar, People	RPSSecAuth	.live.com	Access to hotmail,… (No OneDrive)
Gmail	OSID, HSID, SID, SSID, APISID, SAPISID, LSID	.google.com	https://mail.google.com For basic mail access only first 4 seem needed.

Notice: When setting cookies through the web console, each cookie has to be set individually via document.cookie="". You can always view the currently set cookies via document.cookie

Also when setting cookies ensure to set them on the correct domain. If in doubt you can try setting them on the root domain.

Conclusion

Pass the Cookie is a powerful post-exploitation technique to pivot from on-premise machines to cloud assets. It can be leveraged to bypass 2FA techniques as the cookie is in the end still a single factor.

Hopefully this was helpful, so you can build better detections, improvements and tests into your infrastructure to catch malicious activity.

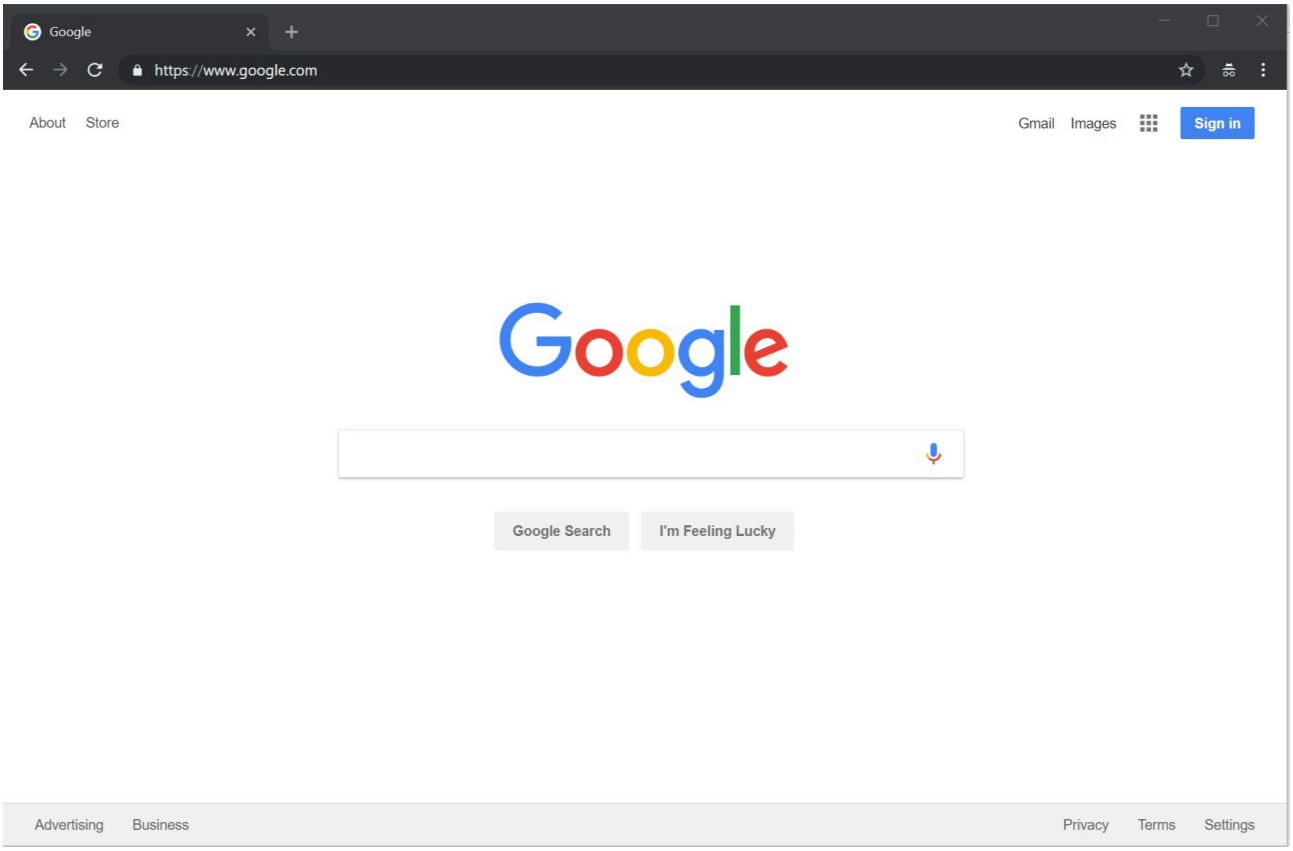
If you have any questions or ideas feel free to send me an email at security@wunderwuzzi.net.

You can also follow or DM me on Twitter: @wunderwuzzi23

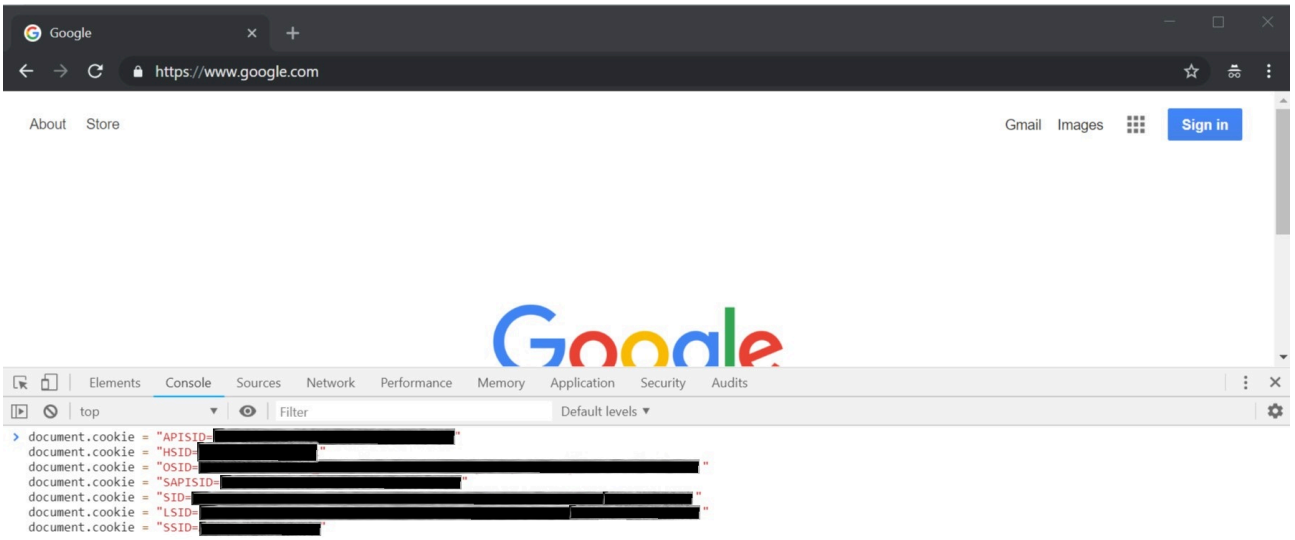
Appendix - Pass the Cookie Examples

Example 1) Google Cloud Platform, Gmail,…

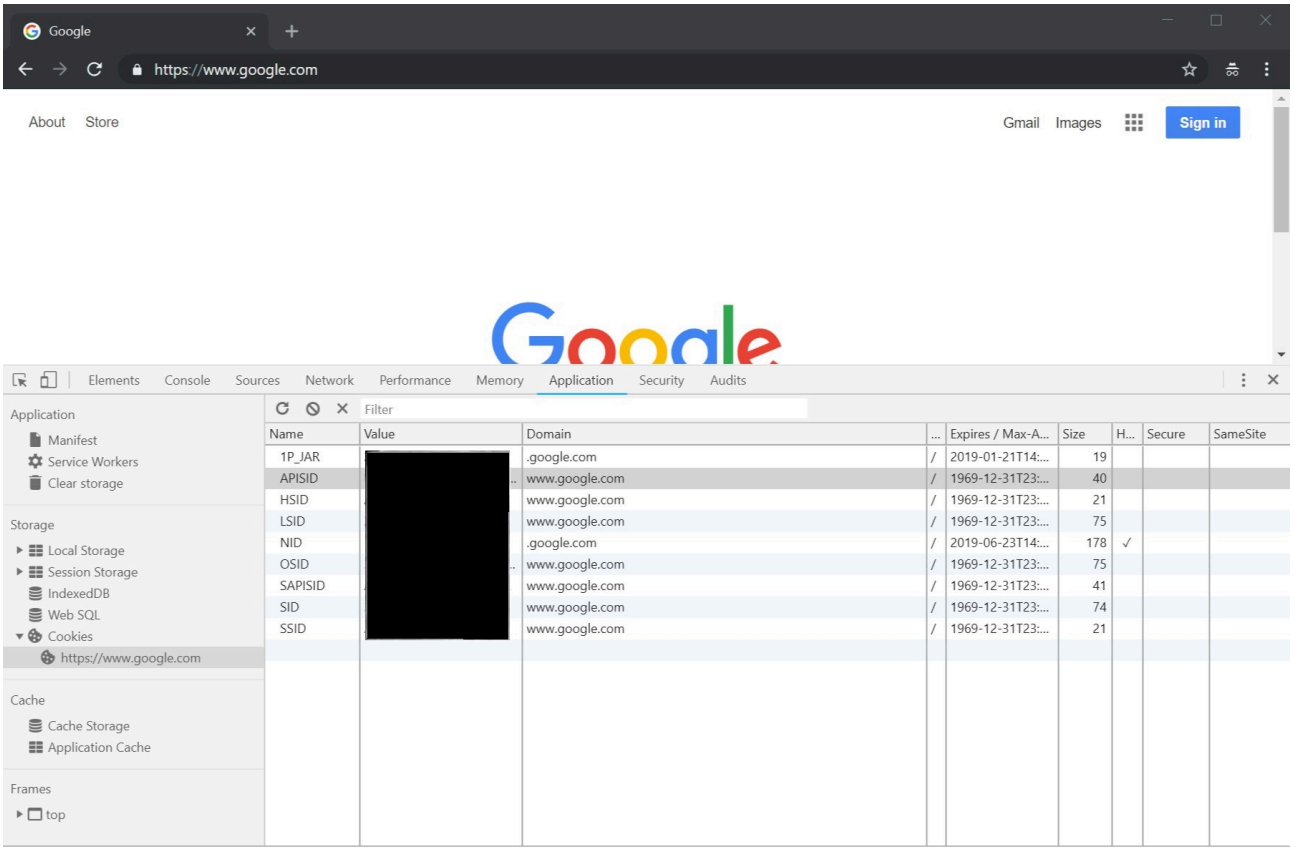
Browse to www.google.com in private mode. We aren't logged in.



Open Developer Console and set the appropriate cookies (see cheat sheet for cookie details)

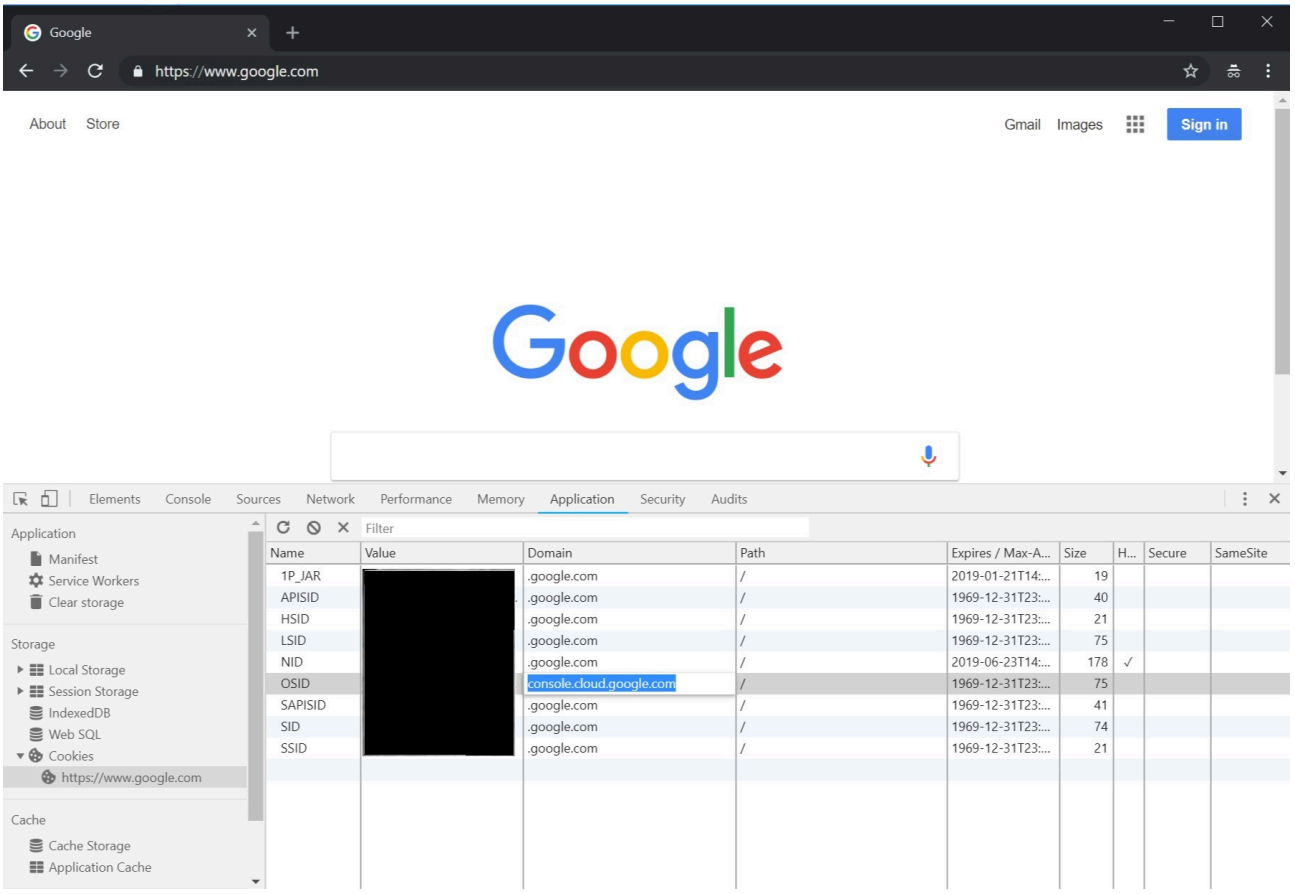


Switch to the Applications tab and look at the cookies. You can see that they got set on www.google.com, which is not what we want.

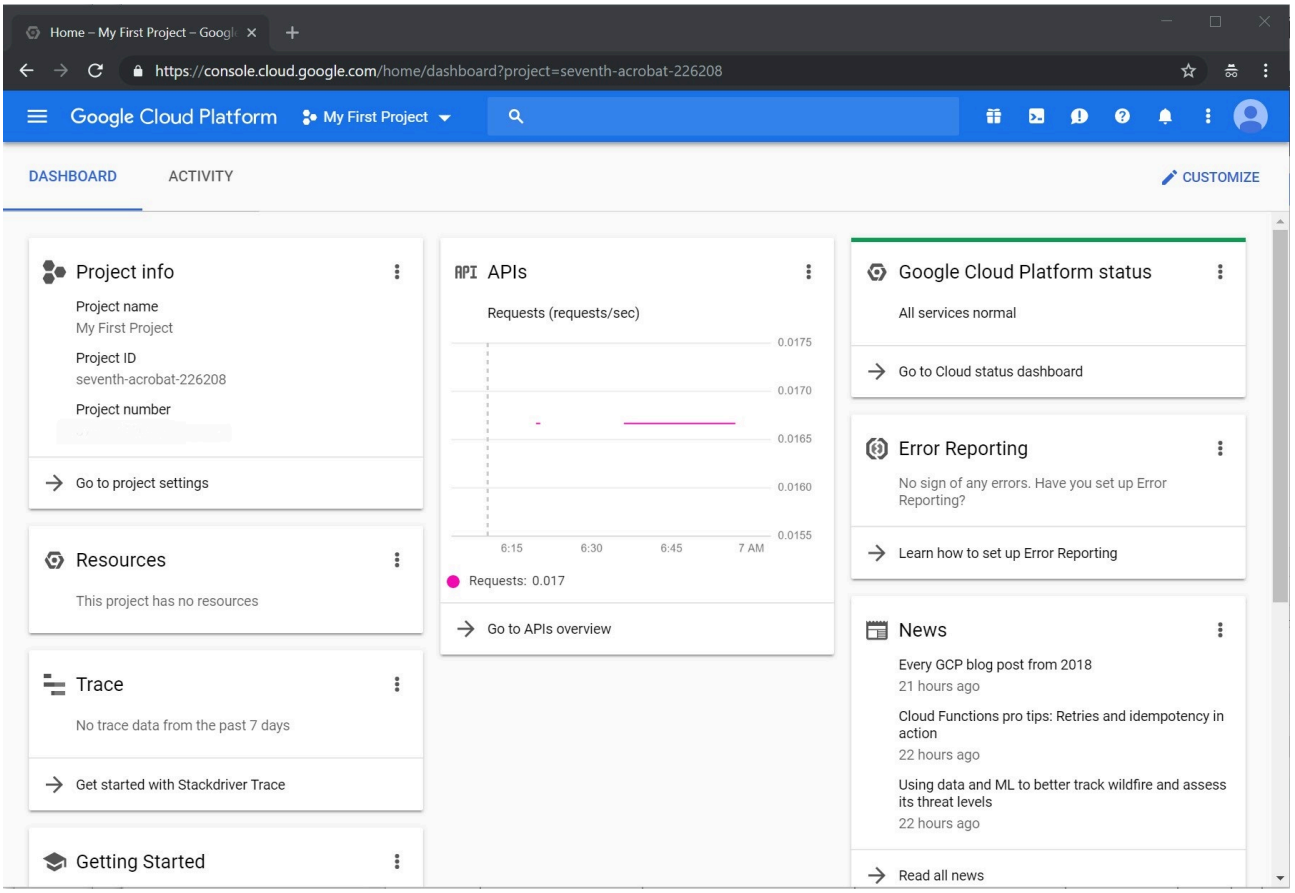


Update the domain setting of the cookies to .google.com. The cookie for OSID has to be set to console.cloud.google.com for GCP (it works on .google.com as well, but you might observe cookie mismatch errors later if you want to go to different services outside of GCP).

So this can be a bit of a hiccup at times.

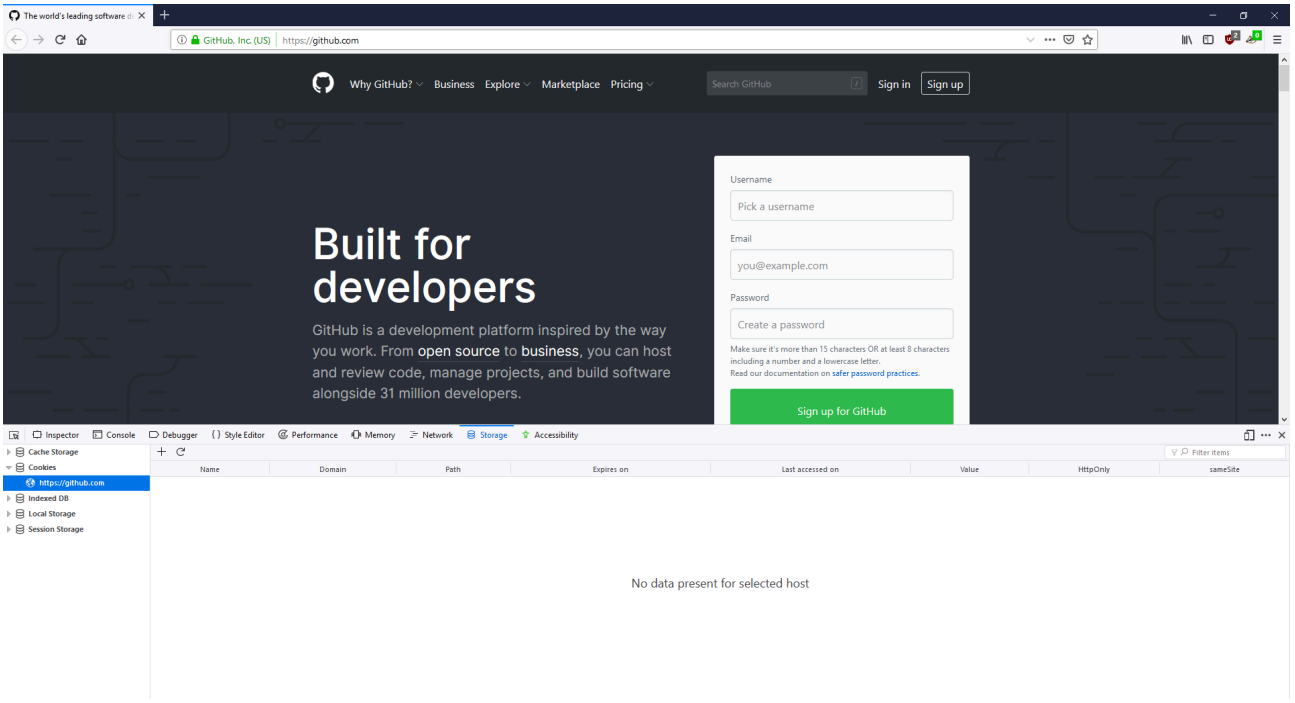


Finally, navigate to https://console.cloud.google.com and observe being magically logged in. If you set the LSID cookie you can also go to GMail or the Accounts settings page.

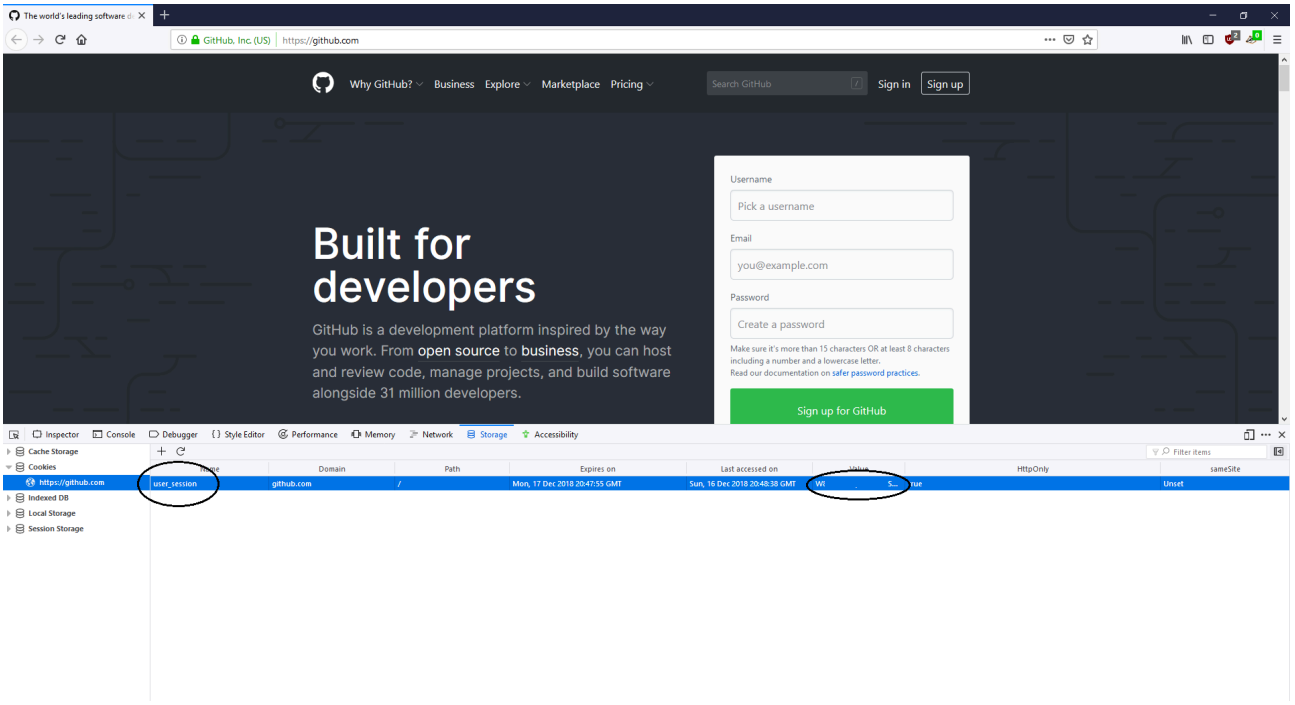


Example 2): Pass the Cookie on Github

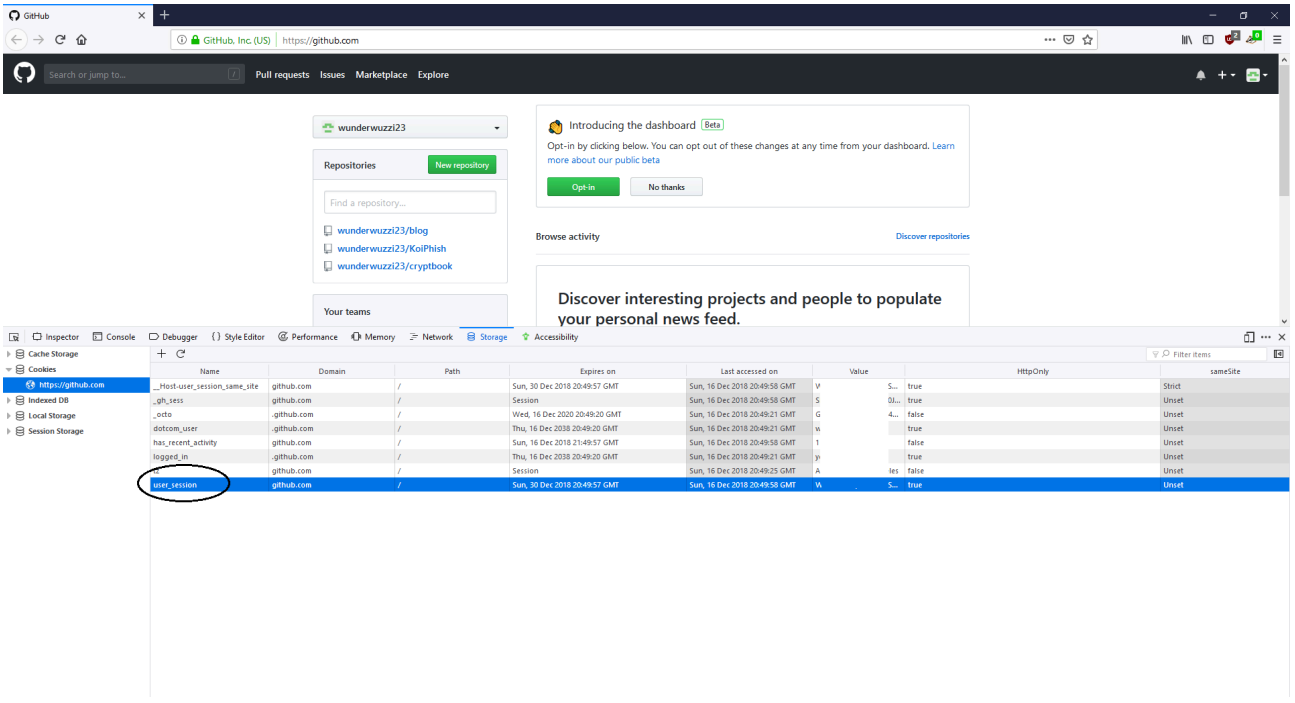
Browse to the website and observe not being authenticated. No cookies.



Set the appropriate cookie for the website domain (e.g via developer tools of the browser).



Refresh the page and observe being authenticated. :)



Source: https://wunderwuzzi23.github.io/blog/passthecookie.html