Who Left Open the Cookie Jar?
A Comprehensive Evaluation of Third-Party Cookie Policies

Gertjan Franken, Tom Van Goethem, Wouter Joosen, KU Leuven

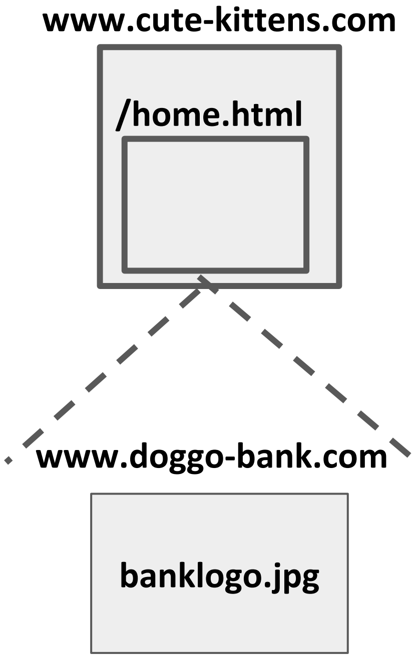
Motivation

What are Cookies Doing on Internet?

Cookies are basically meant to:

- Maintain user state.
- Authenticate & identify the user.
- Sent along all the requests.
- Protected by Same-Origin Policy.

Same Origin Policy (SOP)

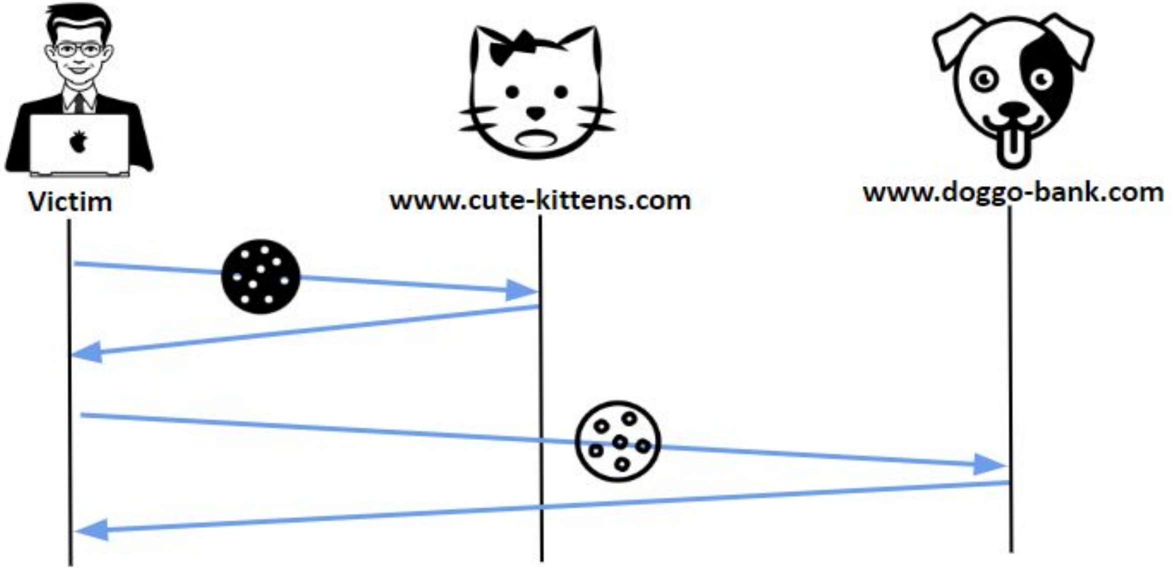


www.cute-kittens.com

/home.html

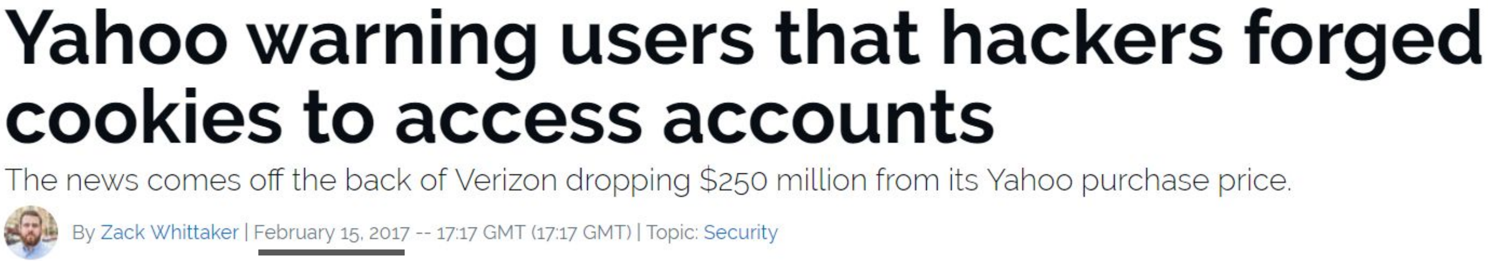
www.doggo-bank.com

banklogo.jpg



So, cookies are safe & sweet, right?

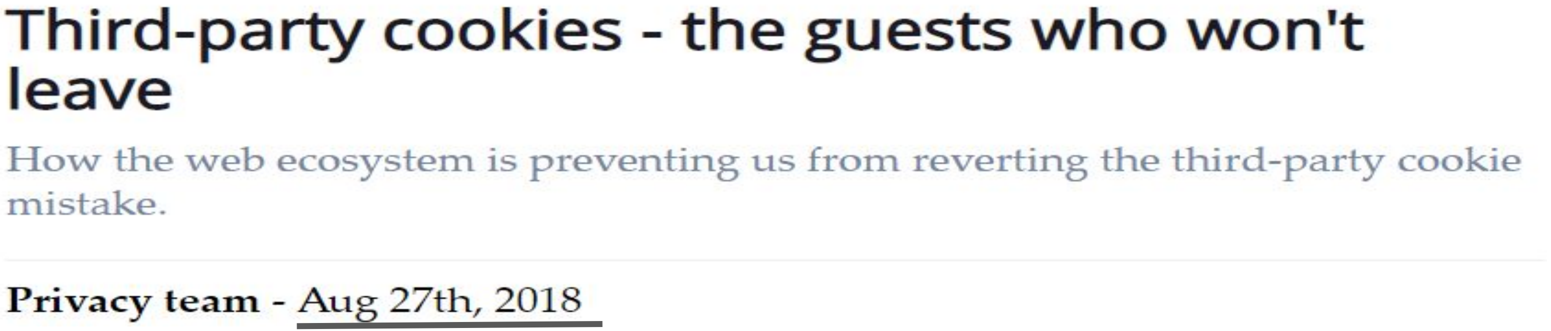
Unfortunately, In This Case...



Yahoo warning users that hackers forged cookies to access accounts

The news comes off the back of Verizon dropping $250 million from its Yahoo purchase price.

By Zack Whittaker | February 15, 2017 -- 17:17 GMT (17:17 GMT) | Topic: Security



Third-party cookies - the guests who won't leave

How the web ecosystem is preventing us from reverting the third-party cookie mistake.

Privacy team - Aug 27th, 2018

Source I: https://www.zdnet.com/article/yahoo-warning-users-that-hackers-forged-cookies-to-access-accounts/

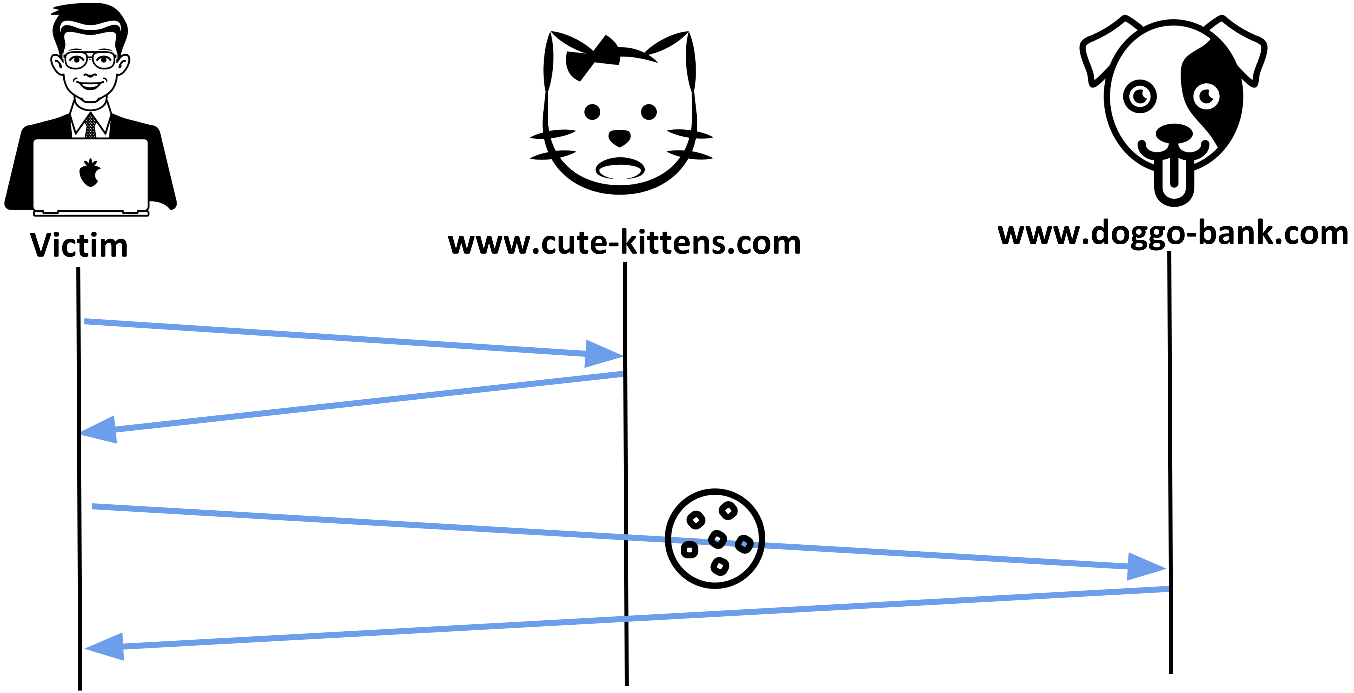
Source II: https://whotracks.me/blog/block-third-party-cookies.html

Cookies are BitterSweet...

Cookies Invade user Security & Privacy by:

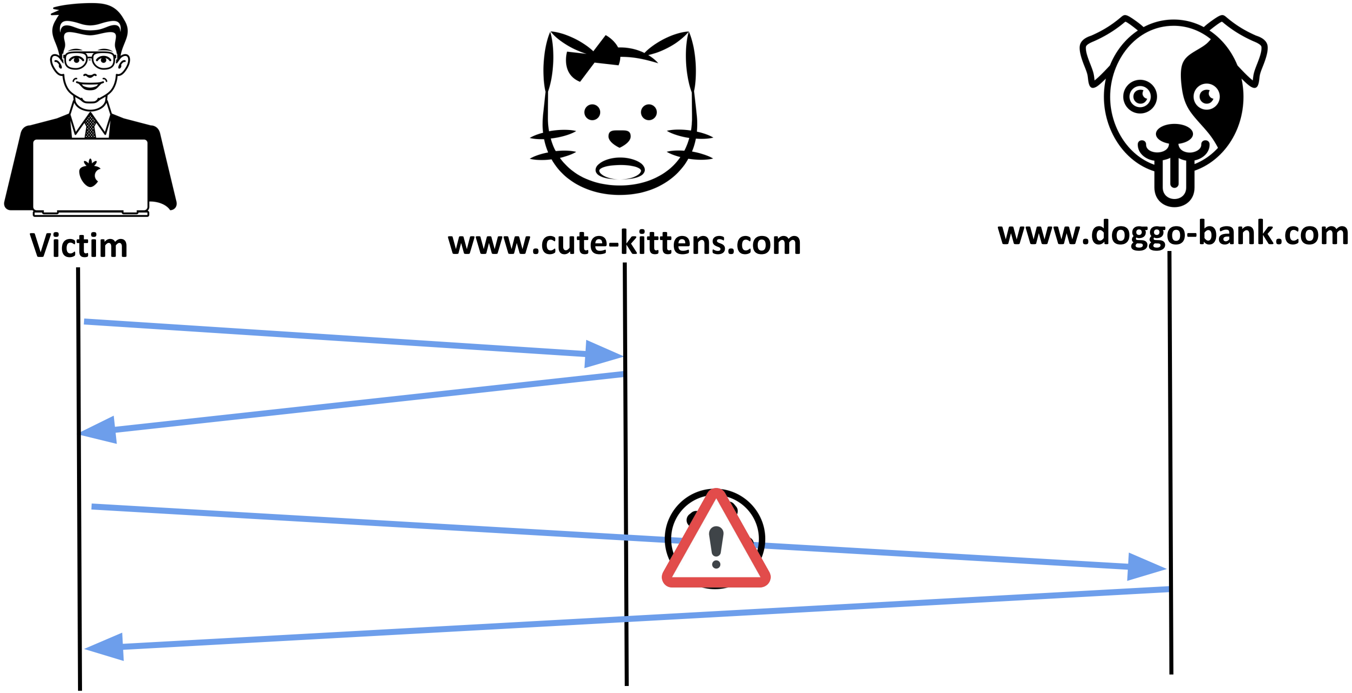
- Cross-Site Attacks
  - Cross-Site Request Forgery
  - Cross-Site Script Exclusion
- Third-Party Tracking.

Cross-Site Attacks - Example



http://doggo-bank.com?transfer.php?amount=450&recipient=goodtutor

Cross-Site Attacks - Example



<img src=”http://doggo-bank.com?transfer.php?amount=1000000&recipient=evilcat”>

Existing Defenses

Researchers and Browser Vendors, over the time, came across with defense strategy against these vulnerabilities.

- ➢ Browser-based inbuilt protection.
  - ○ Opera, Firefox & Safari
- ➢ Additional Anti-Tracking browser extensions.
  - ○ Ad-Blocking & Privacy Protection Extensions
- ➢ Same-Site cookies.

Assumption - Ability to intercept every possible type of requests

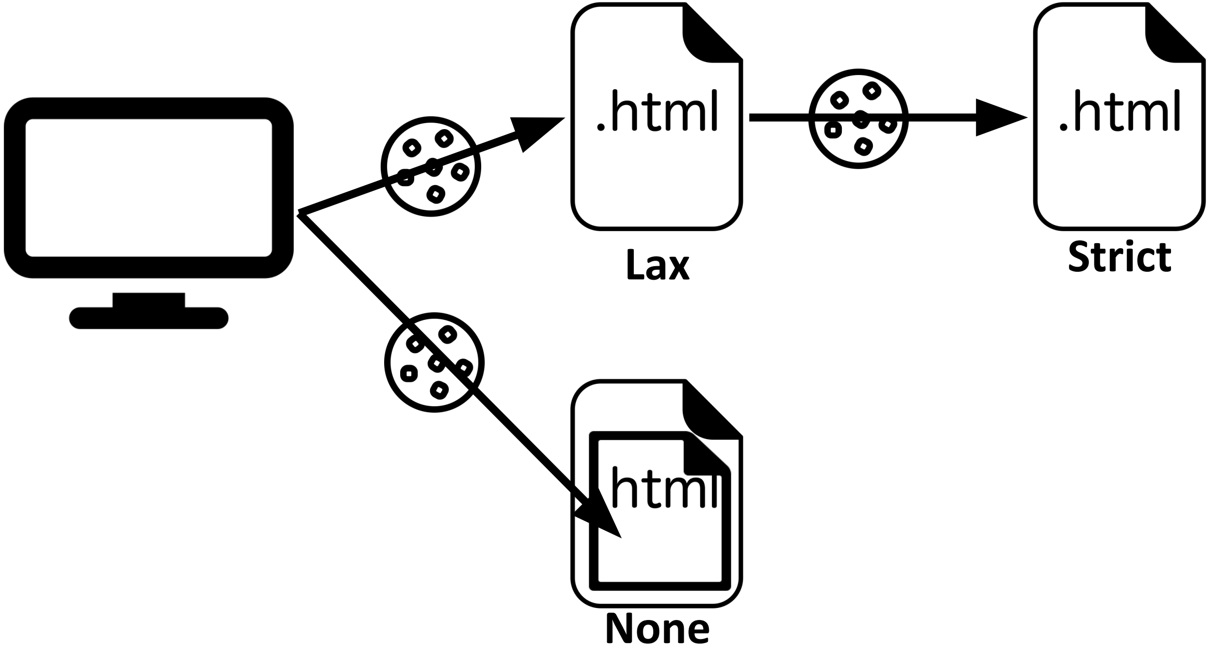
Same-Site Cookies

Cookies with additional attribute as: SameSite = {strict, lax, none}

- ➢ Strict = No Cross-Site Requests
- ➢ Lax = Only Top-Level GET Requests, exception - “prerender”
- ➢ None = No Restriction

Same-Site Cookies

Cookies with additional attribute as: SameSite = {strict, lax, none}



The Evaluation Framework

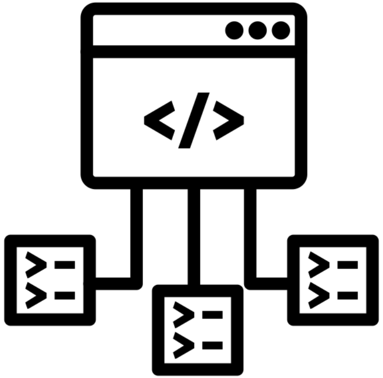
Evaluation of Third-Party Security Policies

Study on 7 different browsers & 46 extensions on their security enforcements policies specific to cookies:

- ➢ Evaluation Framework for In-Place protection mechanisms.
- ➢ Discussion on the origin of identified bypass techniques by the Framework.

FrameWork Components

Black Box Approach - Due to Complexity of Browser Source Code and large number of extensions.



- ➢ Framework Manager.
- ➢ Browser Control
- ➢ Browser Instance - with/without Extension.

System Flow

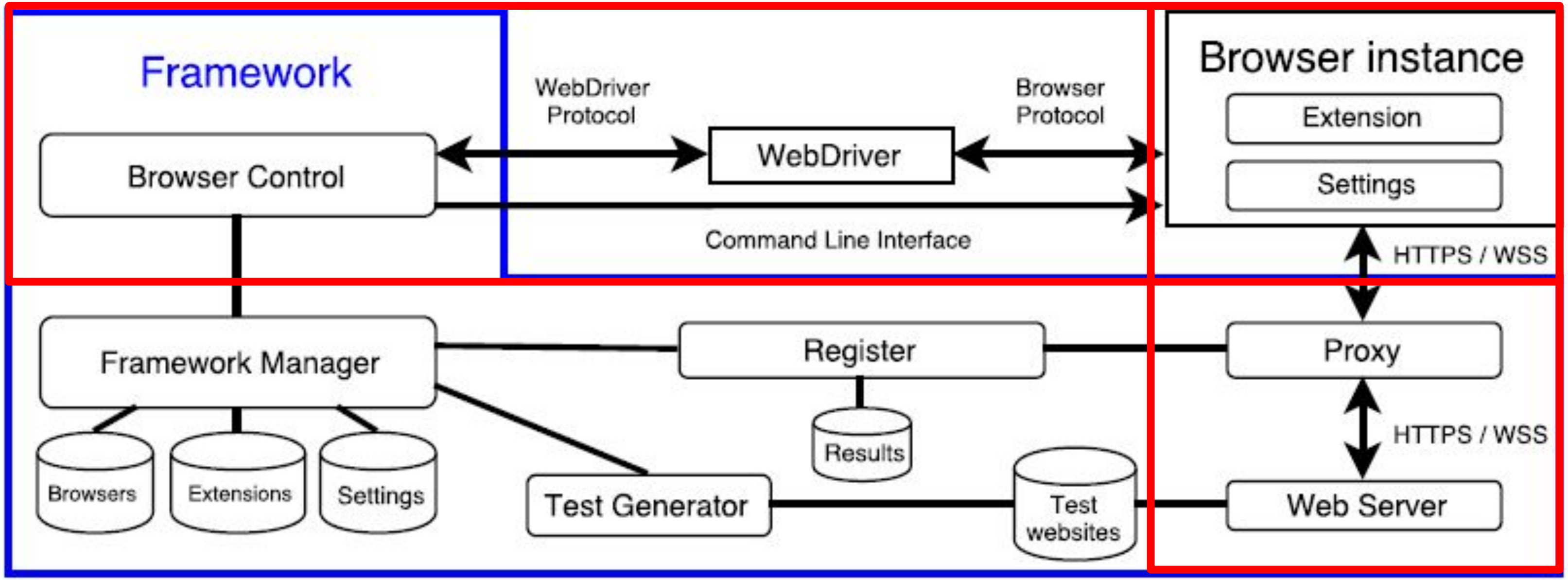
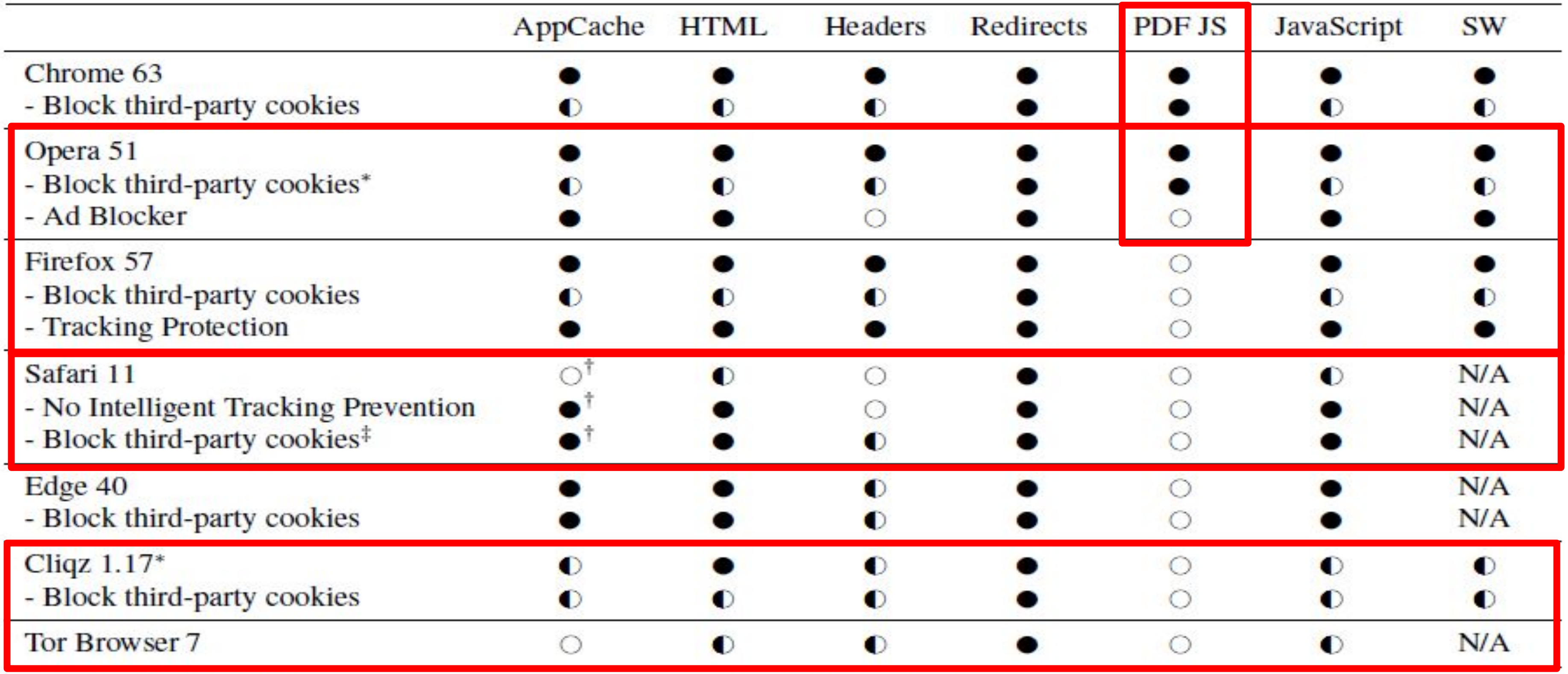


Figure 2: Design of the framework that we used to detect bypasses of imposed cross-site request policies.

Cross-Site Request Auto-Generation

- ➢ HTML Tags - <script>, <img>, <link>, etc.
- ➢ JavaScript-Based - XMLHttpRequest, Fetch, EventSource API.
- ➢ Headers - Links, CSP Policies.
- ➢ PDF JS - sendForm().
- ➢ Redirects - location, meta, etc.
- ➢ ServiceWorker API.
- ➢ AppCache API - Caching cross-site pages.

Result of Evaluation Framework



	AppCache	HTML	Headers	Redirects	PDF JS	JavaScript	SW
Chrome 63	●	●	●	●	●	●	●
- Block third-party cookies	◐	◐	◐	●	●	◐	◐
Opera 51	●	●	●	●	●	●	●
- Block third-party cookies*	◐	◐	◐	●	●	◐	◐
- Ad Blocker	●	●	○	●	○	●	●
Firefox 57	●	●	●	●	○	●	●
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
- Tracking Protection	●	●	●	●	○	●	●
Safari 11	○†	◐	○	●	○	◐	N/A
- No Intelligent Tracking Prevention	●†	●	○	●	○	●	N/A
- Block third-party cookies‡	●†	●	◐	●	○	●	N/A
Edge 40	●	●	◐	●	○	●	N/A
- Block third-party cookies	●	●	◐	●	○	●	N/A
Cliqz 1.17*	◐	●	◐	●	○	◐	◐
- Block third-party cookies	◐	◐	◐	●	○	◐	◐
Tor Browser 7	○	◐	◐	●	○	◐	N/A

●: request with cookies ◐: request without cookies ○: no request

* Secure cookies were omitted in all requests.

† Safari does not permit cross-domain caching over https (only over http).

‡ Safari 10.1.2

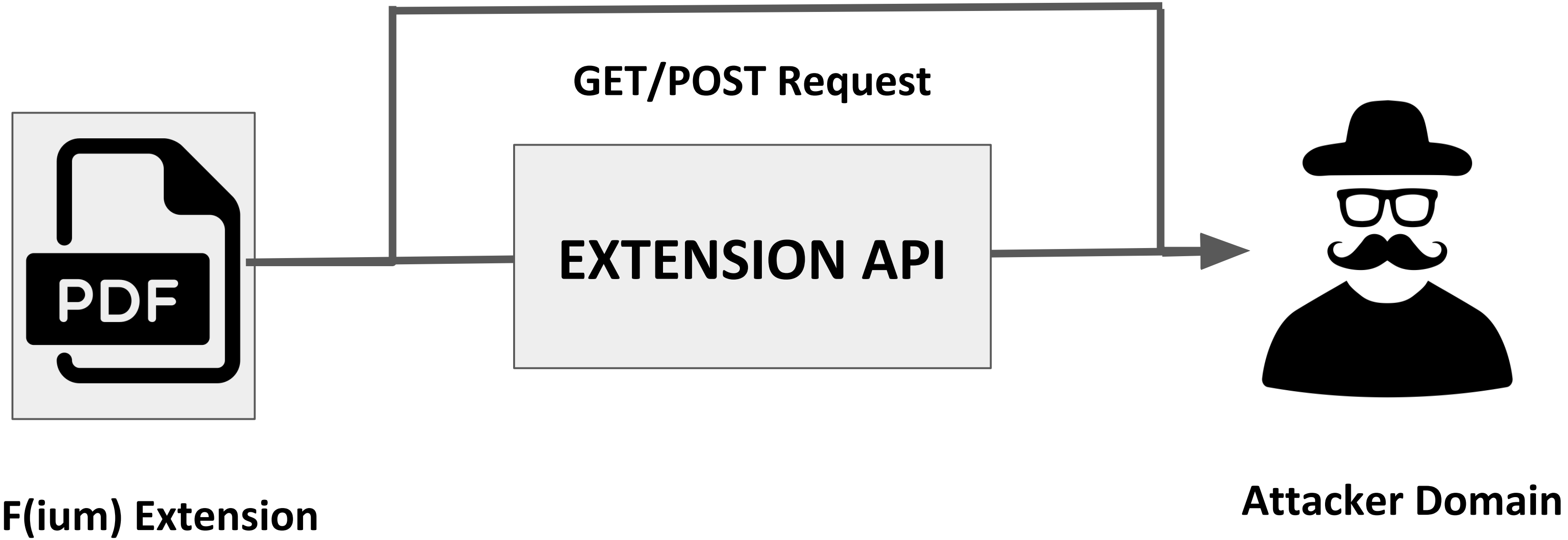
Table 1: Results from the analysis of browsers and their built-in security and privacy countermeasures.

How About Extensions?

None of them could block all types of requests!

- Insufficient APIs.
- Unclear APIs/Requests.
- Extension Development Issues.

PDF(ium) Design Vulnerability



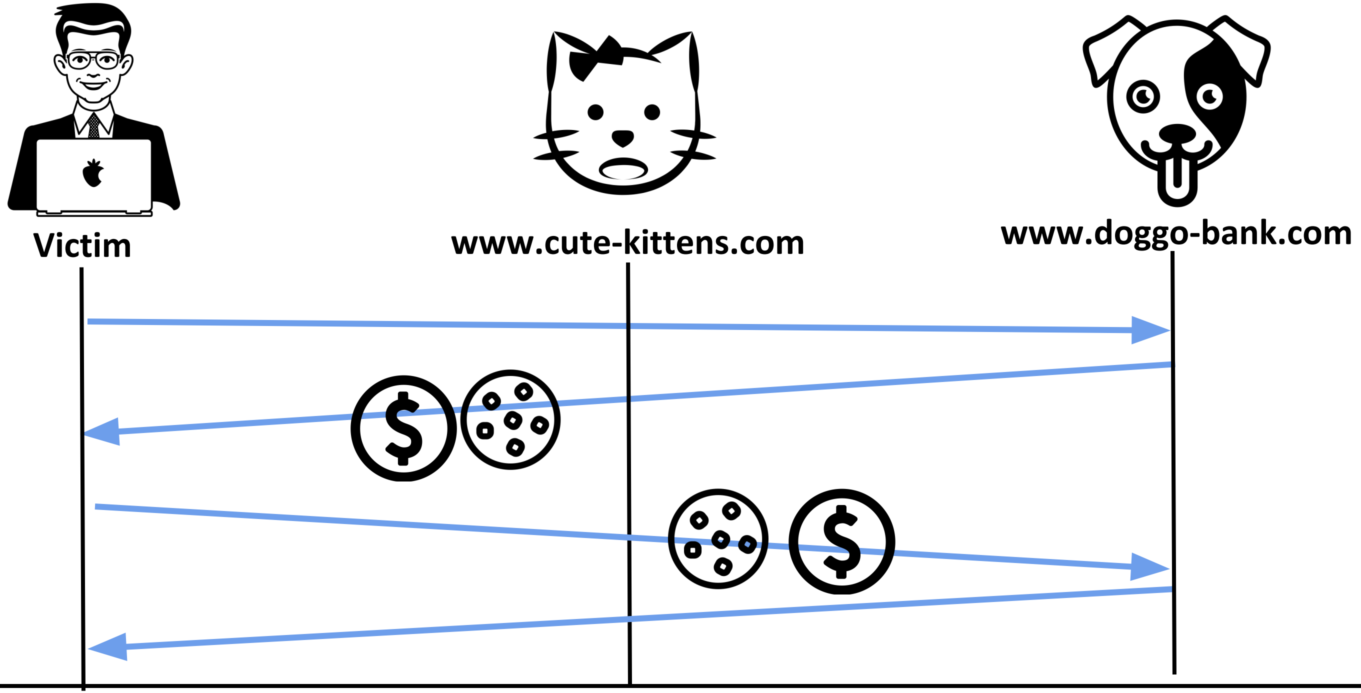
GET/POST Request

EXTENSION API

PDF(ium) Extension

Attacker Domain

Cross-Site Requests with “Prerender”



<link rel=”prerender”
href=”http://doggo-bank.com?transfer.php?amount=100000&recipient=evilcat”/>

Bypasses Exploited in the Wild

- Crawled over Alexa Top 10K Websites.
- Visited over 160,059 pages.
- Analysed intercepted HTTP(S) requests.
- None of the reported bypasses exploited yet.

Conclusion

- Developed broad evaluation framework to analyze cookie-related security policies.
- Identified significant bypasses that exist in currently proposed protection mechanisms.
- Fortunately, none of them found to be exploited among Top 10K Alexa Websites.
- Need for automated tool for analysis of security & privacy specific policies

Future Research Avenues

- Mobile Browsers can be explored further.
- Browser-specific attack surfaces can be investigated.
- Privacy Mode & Security Policies, e g. CSP can also be further analyzed in future.
- Investigation of other storage APIs and their interaction with code which carries stateful information.

Questions?