When Push Comes to Shove: Empirical Analysis of Web Push Implementations in the Wild

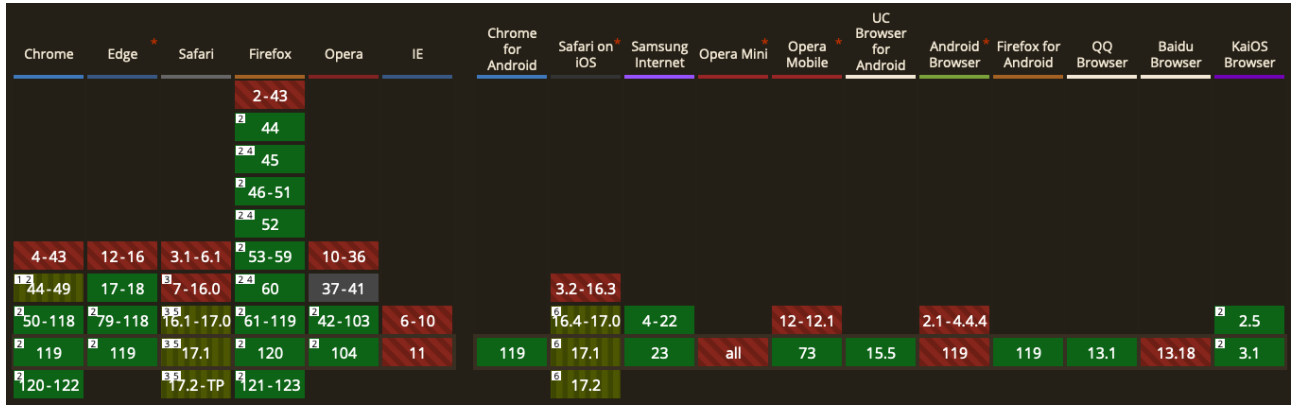
Alberto Carboneri, **Mohammad Ghasemisharif**, Soroush Karami, and Jason Polakis

ACSAC 2023

UIC UNIVERSITY OF ILLINOIS CHICAGO

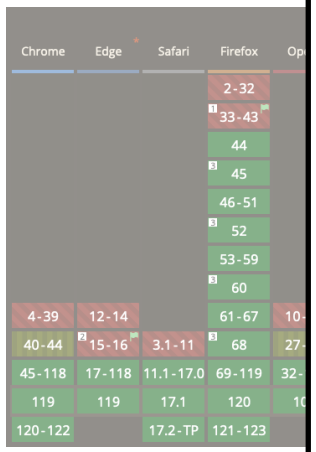
Browser Features Adoption

Service Workers



Web Push

Browser Features Adoption



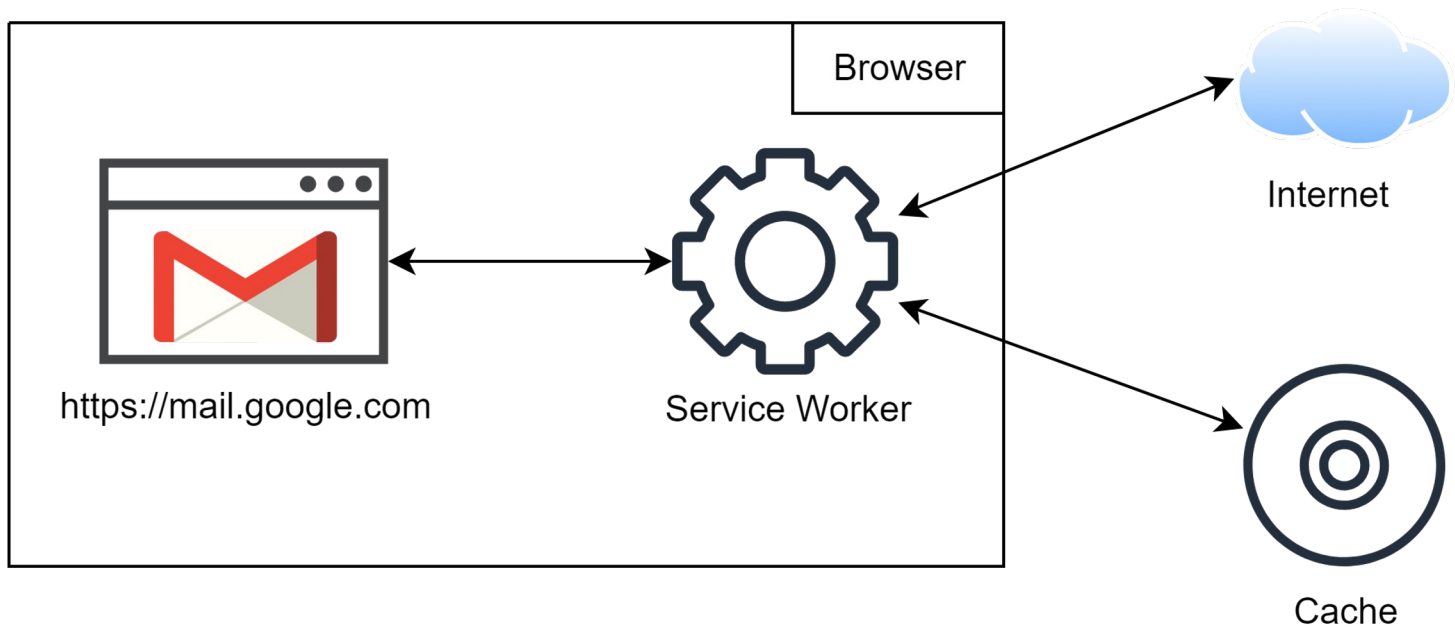
In this work, we focus on the correct deployment of **Web Push** and potential vulnerabilities.

Overview

- Service Workers
- Web Push Notifications

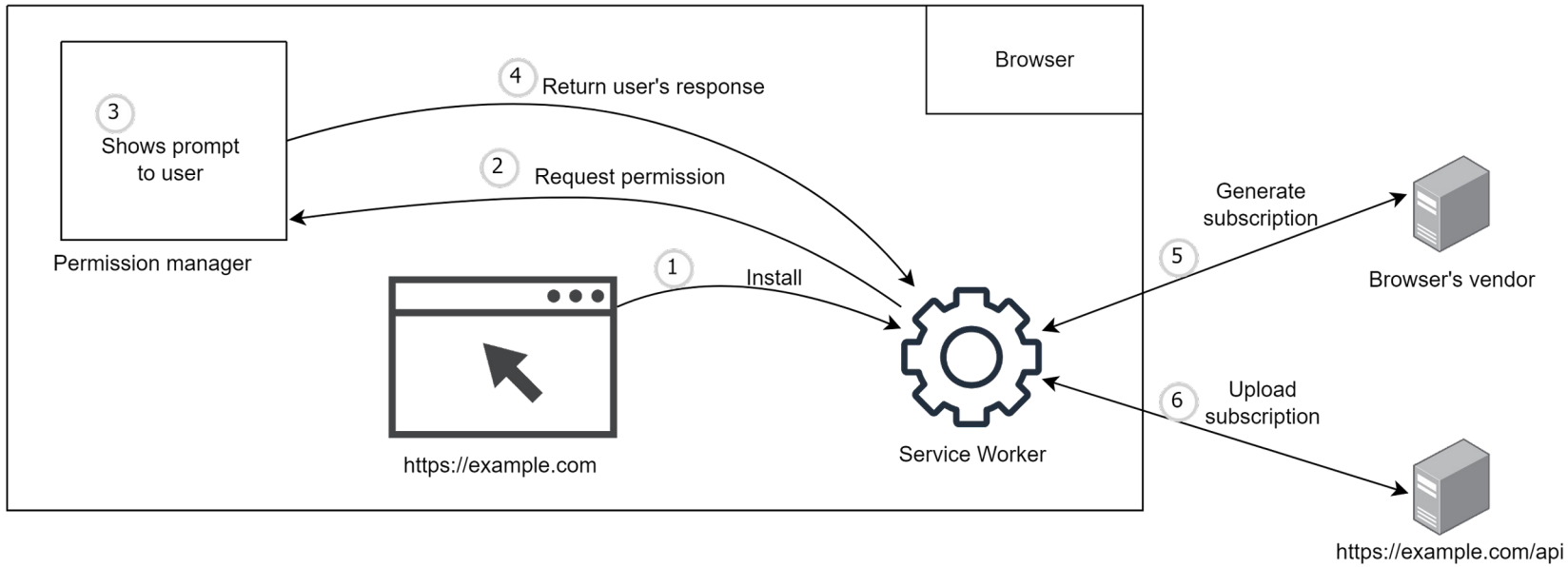
Service Workers

- Detached from the main page
- Requires HTTPS
- Can handle complex scenarios
  - Caching
  - Sync
  - **Web Push**



Web Push Notifications

- Permission based
- Requires a service worker
- Complex to implement



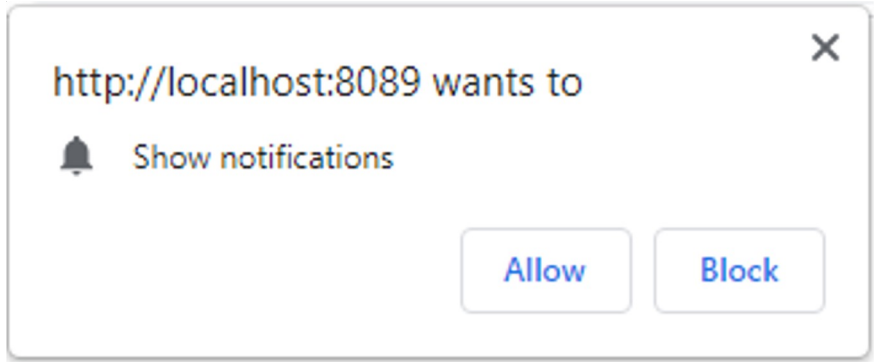
Web Push Notifications

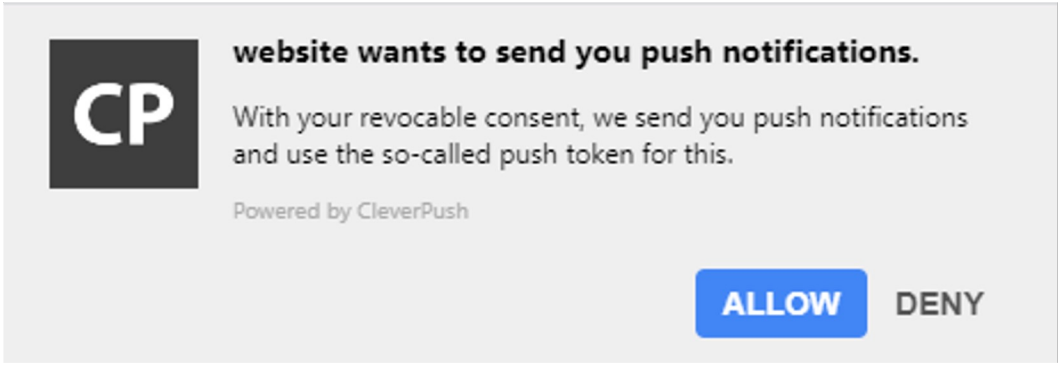
- Subscription

```
{
    endpoint: 'https://fcm.googleapis.com/fcm/send/cwEL8oM02a8:APA91 ... cg',
    expirationTime: null,
    keys: {
        p256dh: 'BPm4u ... vvNrY',
        auth: 'bv1SgdyGs9vT30x2-5BoRQ'
    }
}
```

Web Push Notifications

- Permission based
  - Direct permission request
  - Soft-ask



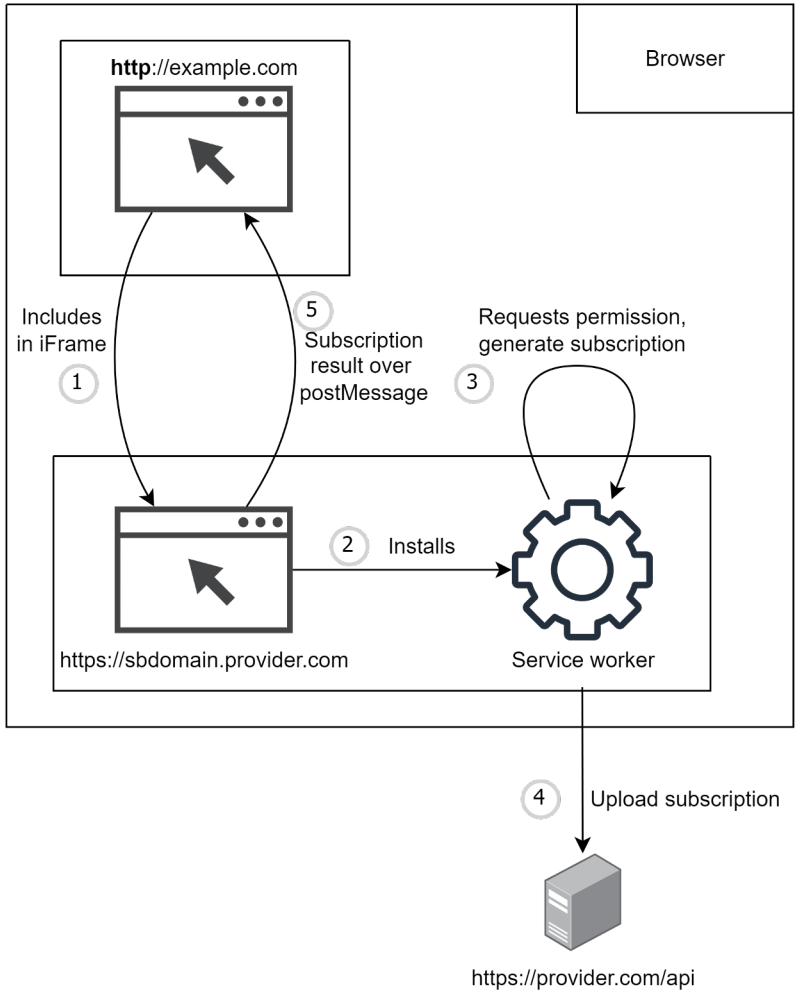


Web Push Notifications (3rd-party Providers)

- Shift complexity to a third-party
  - No need for backend
- Often implemented by script inclusion
- Only option for websites without HTTPS

Web Push Notifications (3rd-party Providers)

- Shift complexity to a third-party
  - No need for backend
- Often implemented by script inclusion
- **Only option for websites without HTTPS**



What are the potential pitfalls of Web Push Notifications?

Design Challenges and Vulnerabilities

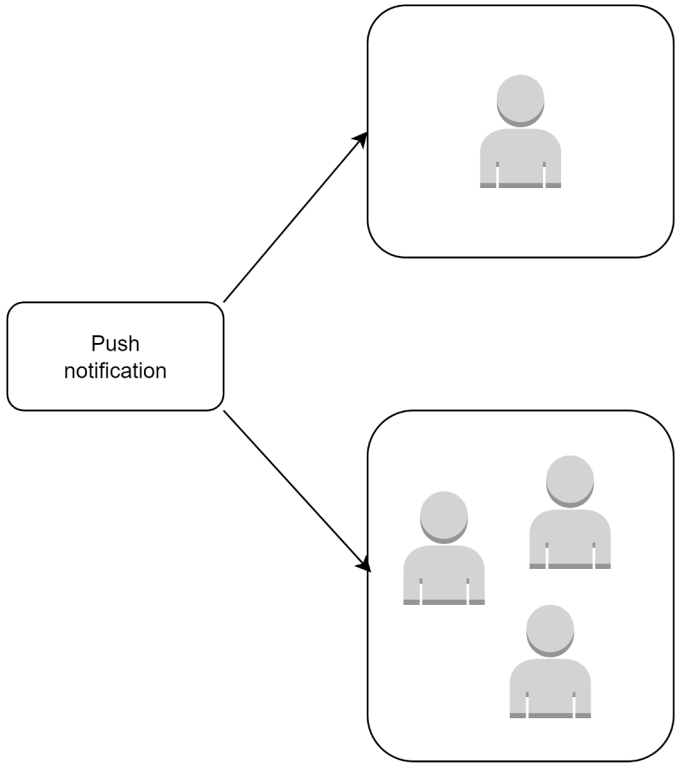
- Design challenges
  - Personalized web push

- Vulnerabilities
  - Third-party providers
  - CSRF

Design Challenges

- Push notifications are commonly used to send an identical message to large audiences

- Personalized notifications are also supported



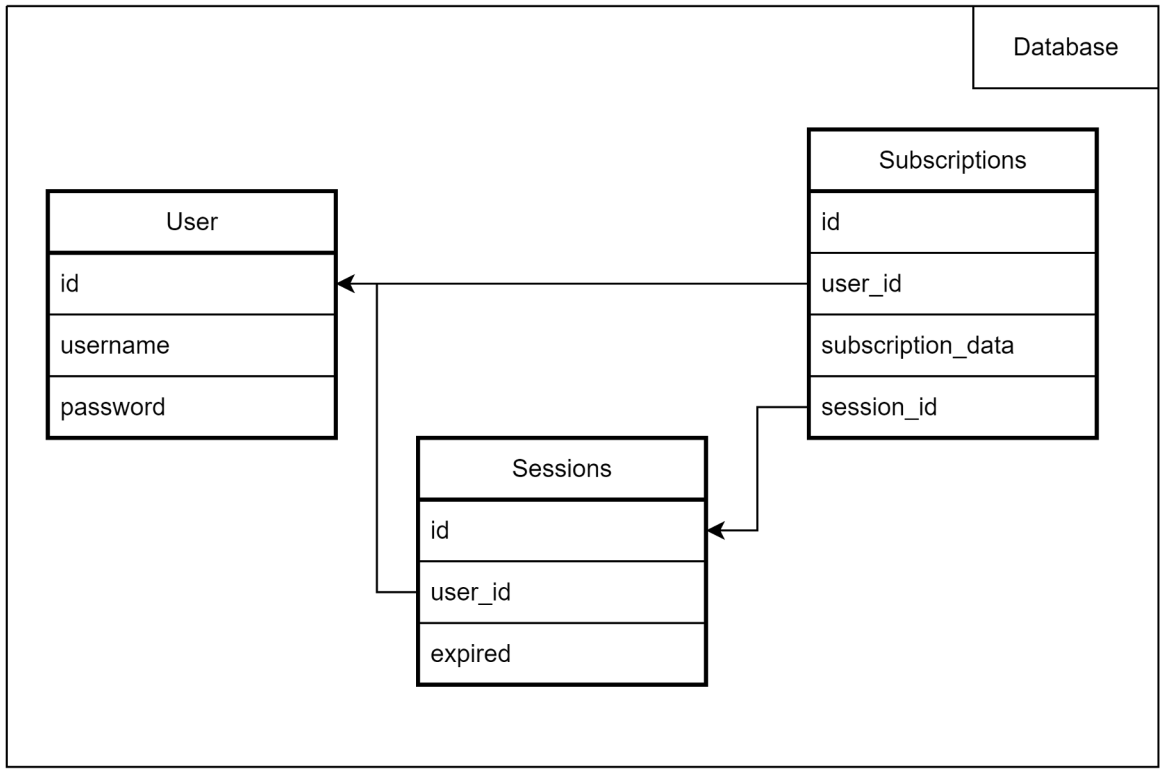
Personalized Notifications

- Subscription data must be linked to user's account
  - Multiple sessions
  - Complex session behavior
  - Complex notification delivery

userid	username	password	subscription
1	Mark	spFRgWTNLjfx . . . nkH4Gh4DeJKOJM	endpoint,keys
2	David	DpRRgSDVjfd . . . nkGTDh4De68VMP	endpoint,keys
3	Maria	DURgSEWNLHfx . . . nk4EW3FxrV4M	endpoint,keys

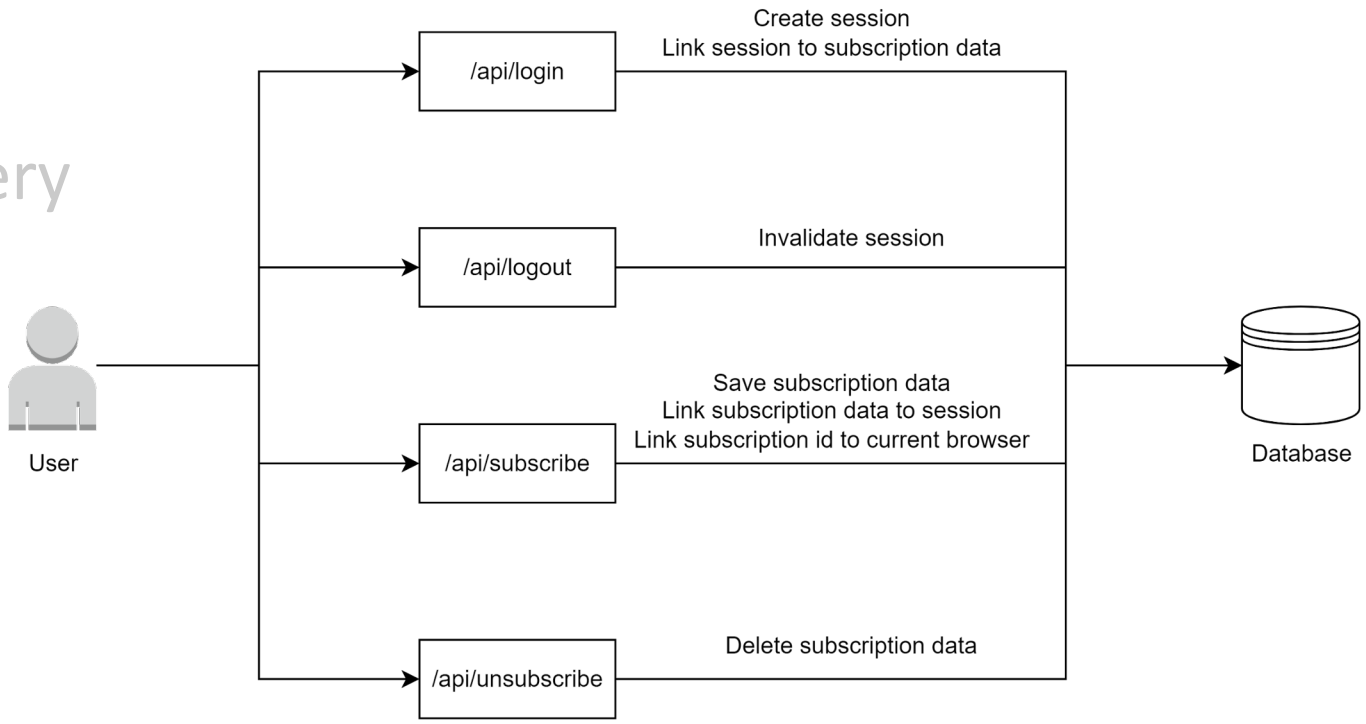
Personalized Notifications

- Subscription data must be linked to user's account
  - Multiple sessions
  - Complex session behavior
  - Complex notification delivery



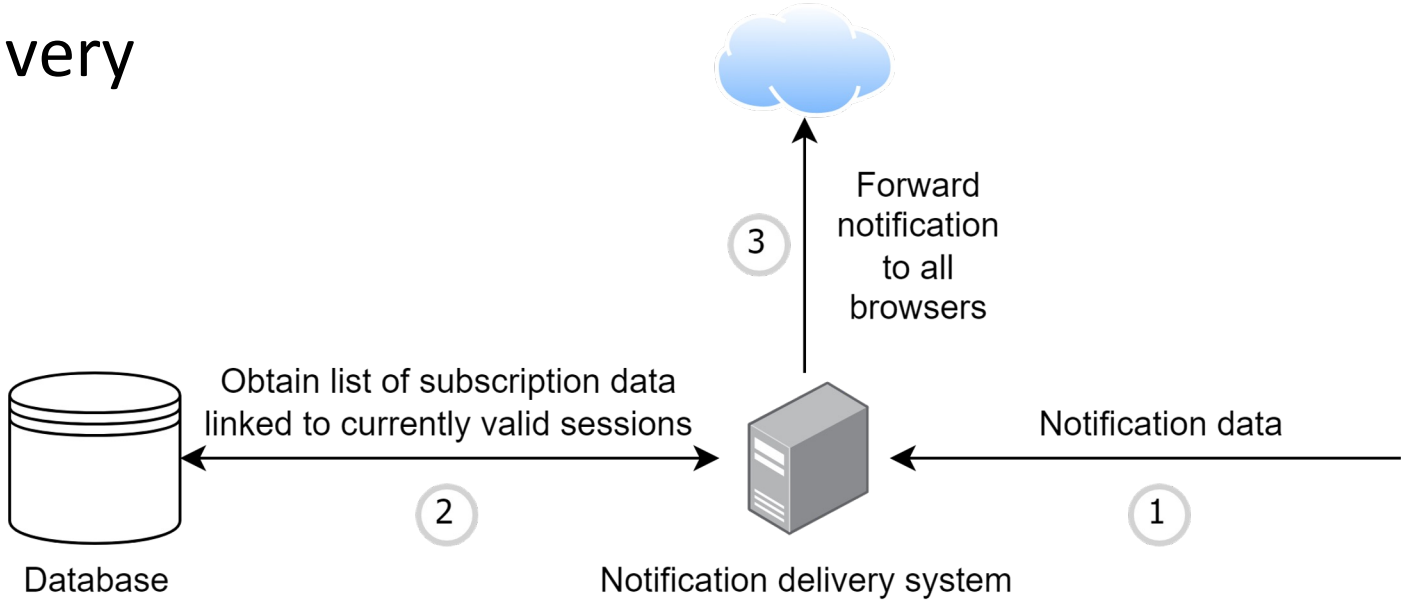
Personalized Notifications

- Subscription data must be linked to user's account
  - Multiple sessions
  - Complex session behavior
  - Complex notification delivery



Personalized Notifications

- Subscription data must be linked to user's account
  - Multiple sessions
  - Complex session behavior
  - Complex notification delivery



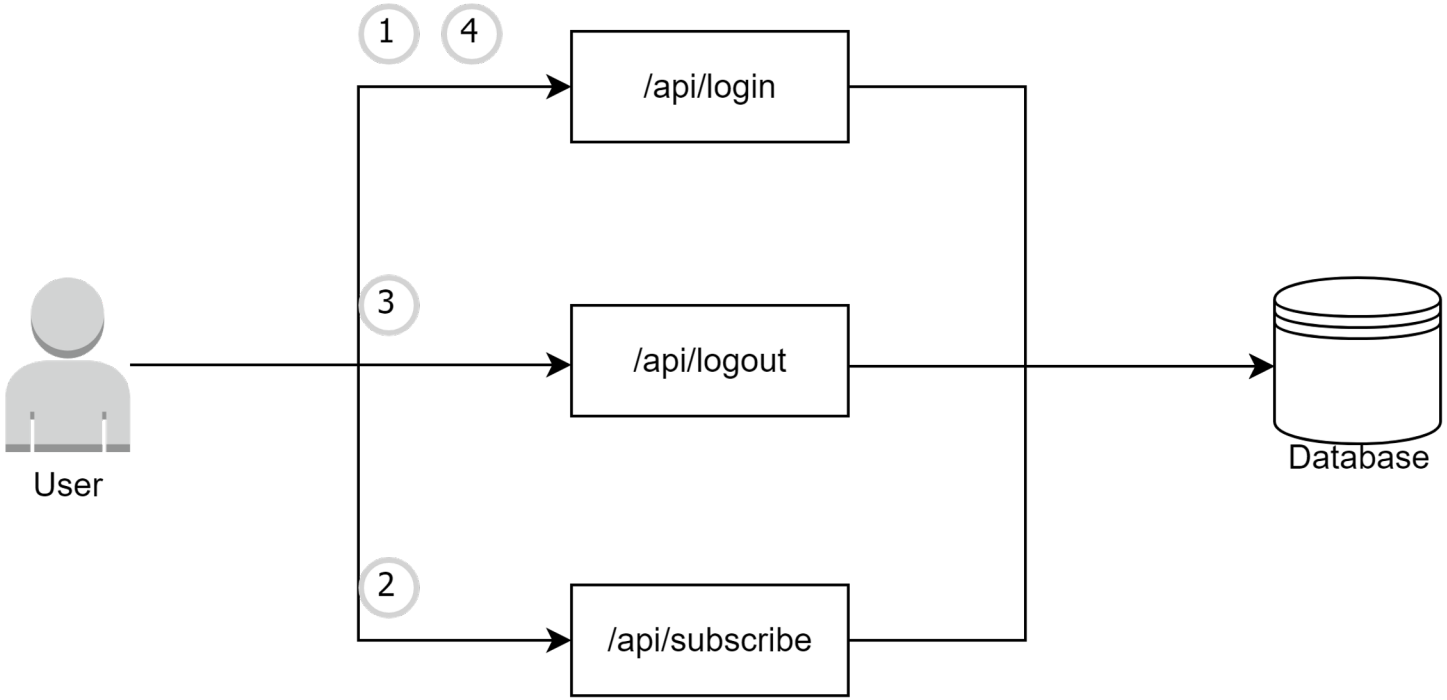
Examples of Personalized Notifications Issues

- Not handling logout or session expiration (*poshmark.com*)
- The subscription data and session are not linking correctly (*twitter*)
- The subscription data and browser are not linking correctly (*twitter*)

userid	session	OS	browser	subscription
user1	nJW3mn3Su9E . . . BHQY4u7HjKBwY	MAC OS	Firefox	endpoint,keys
user2	spRRgWNLjfx . . . nk4Eh4DexrVMM	Ubuntu	Chrome	endpoint,keys
user3	2VrU7RTr4pU . . . eU8cCh4qDBccT	Windows	Brave	endpoint,keys

Possible structure of subscription table in twitter's database

Examples of Personalized Notifications Issues

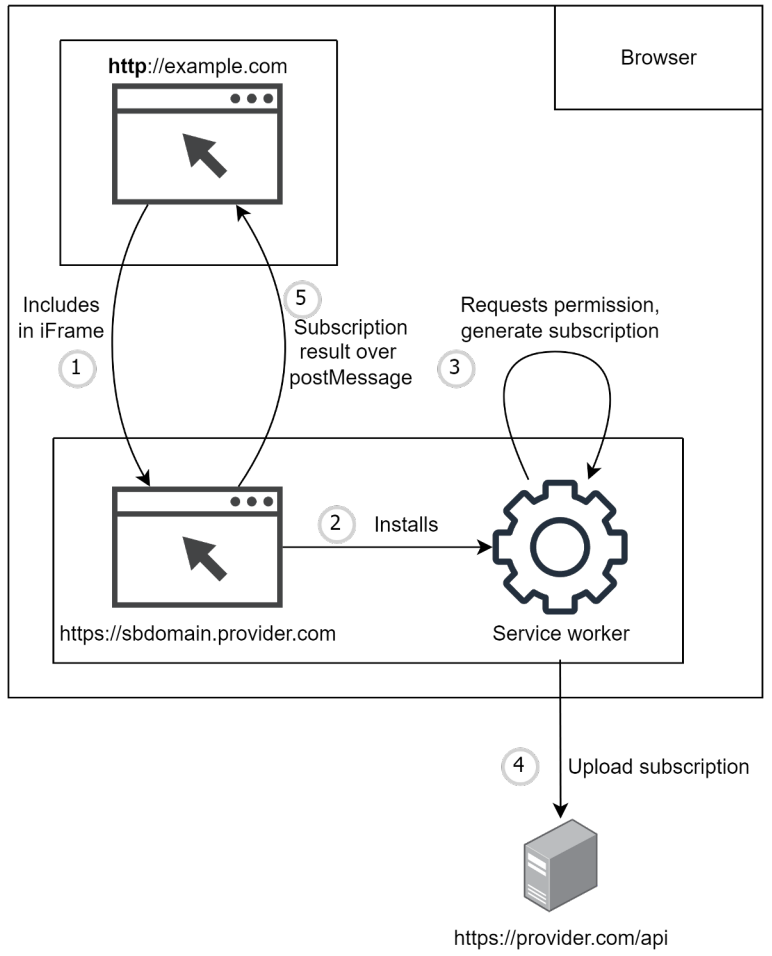


Sequence of actions breaking twitter's system, leading to no new notifications being sent to the user until subscription is refreshed.

What are the potential vulnerabilities of 3rd party providers?

Vulnerabilities (3rd-party Providers)

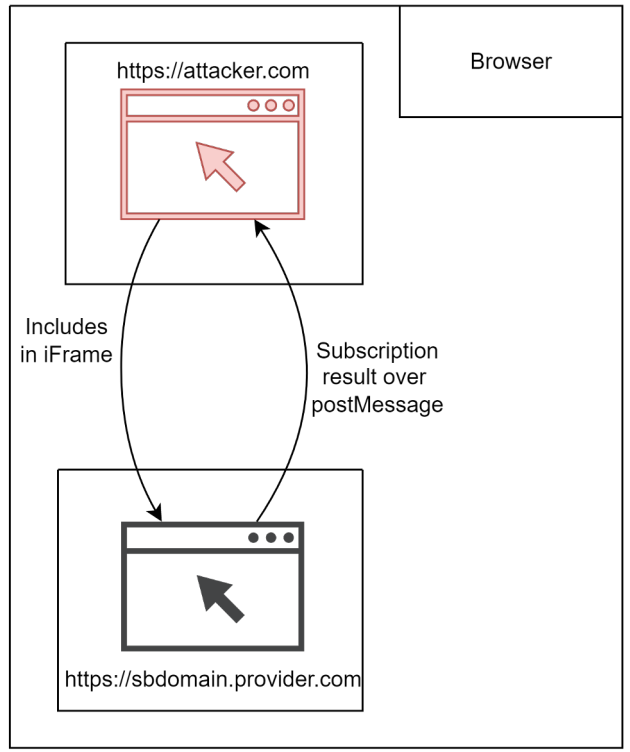
- Websites without HTTPS (iFrame inclusion)
- Subscription result sent over postMessage
  - *postMessage(message, targetOrigin)*
  - *postMessage(message, targetOrigin, transfer)*



Vulnerabilities (3rd-party Providers)

- Attacker receives messages if *targetOrigin* is set to “*”
  - History/subscription sniffing
  - Target a specific third-party customer

- Example: webpushr (third most used provider)



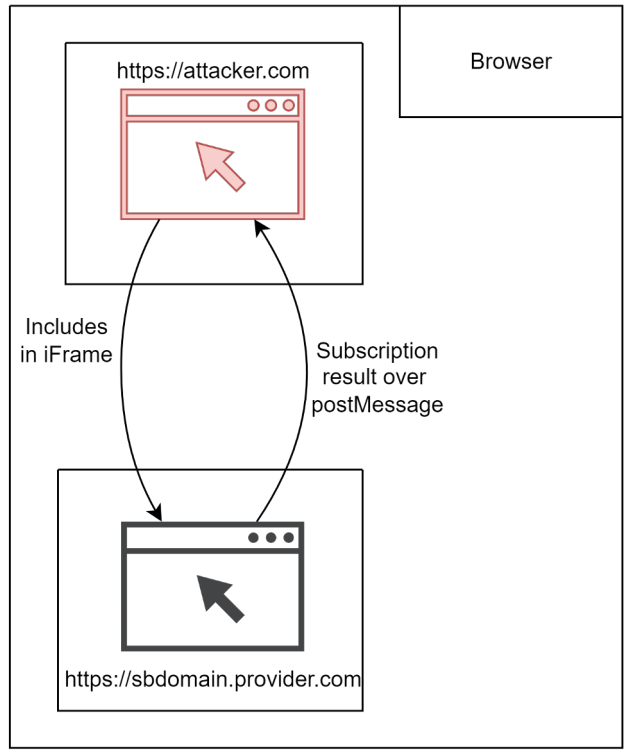
Vulnerabilities (3rd-party Providers)

- If *targetOrigin* is taken from URL parameter website is still vulnerable

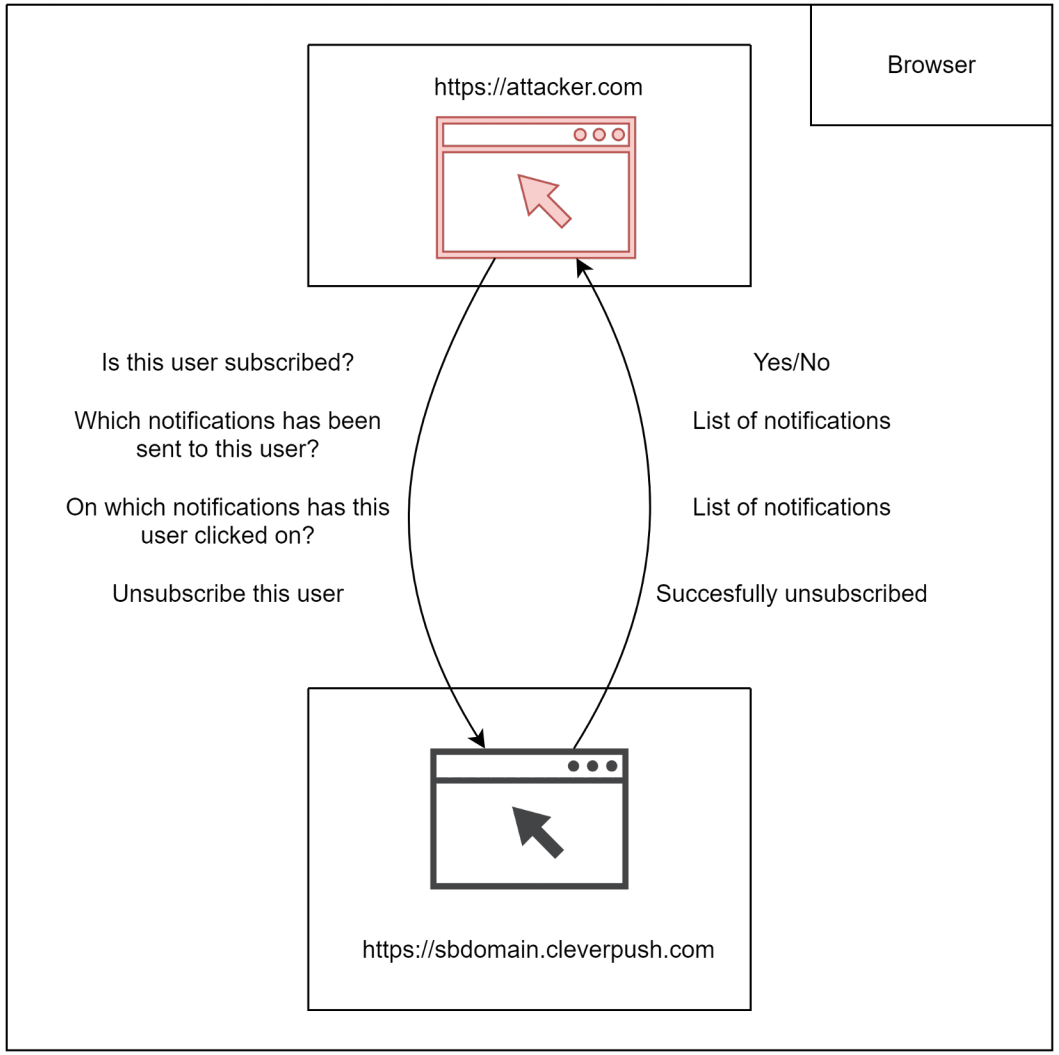
  https://sbdomain.provider.com/iframe?origin=https%3A%2F%2Fattacker.com%3A8089

- Almost all affected providers send unique identifiers

- Cleverpush exposes dangerous APIs
  - Obtain more sensitive data
  - Unsubscribe the user



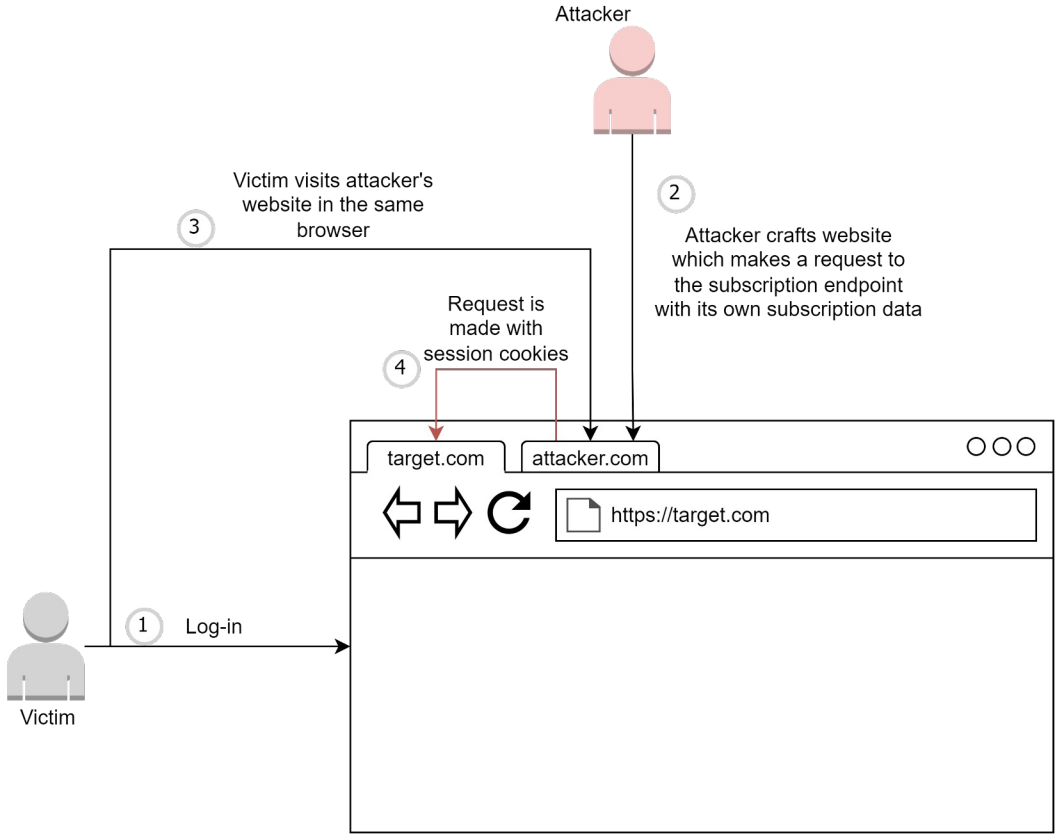
Example of Vulnerability (cleverpush.com)



Vulnerabilities (CSRF)

- In the absence of CSRF protection, attackers can subscribe themselves to the user's notifications

- Example: *gama.ir*
  - Education website
  - Exam question/answer leakage



Some of Our Findings

- Measurement of top 500K websites:
  - 18,566 (4.68%) unique websites uses push notifications
  - 3,117 websites aggressively request for the permission
  - 1187 websites are potentially vulnerable to our iFrame inclusion vulnerability
  - 479 are using Cleverpush as a provider (vulnerable)

Conclusions

- Wrong or bad usage of Web Push may lead to privacy issues or loss of subscriptions
- Aggressive pattern still present
- Need for more security tests, both on feature itself and its implementations

Conclusions

- Wrong or bad usage of Web Push may lead to privacy issues or loss of subscriptions
- Aggressive pattern still present
- Need for more security tests, both on feature itself and its implementations

Moe Ghasemisharif
mghas2@uic.edu

Q&A