Pride and Prejudice in **Progressive Web Apps**: Abusing Native App-like Features in Web Applications

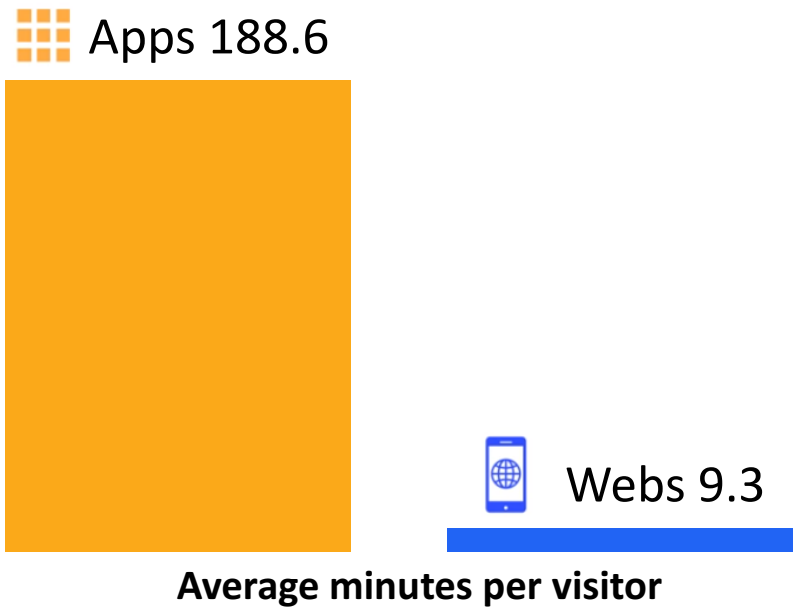
Jiyeon Lee, Hayeon Kim, Junghwan Park,
Insik Shin, Sooel Son

School of Computing,
Graduate School of Information Security

KAIST

Limitations of Web Apps

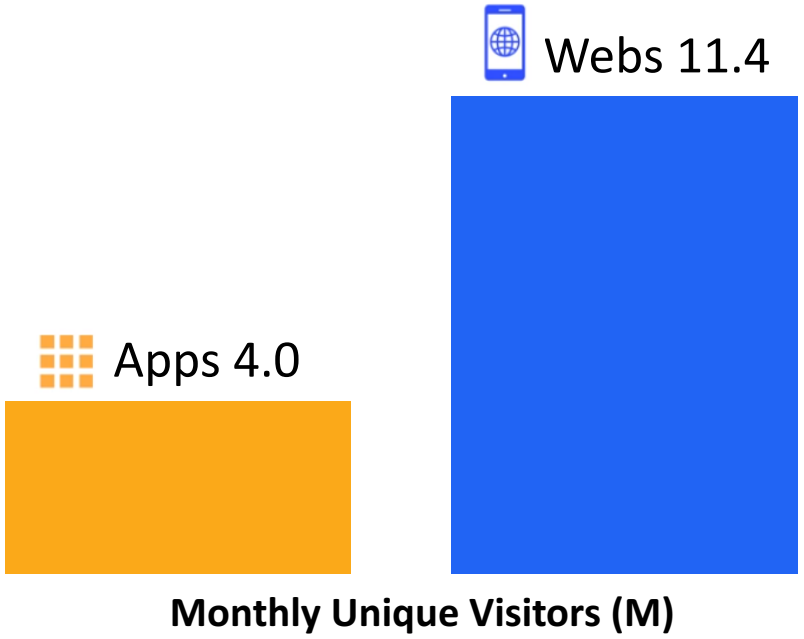
- Users spend most of time in native apps
- Reasons:
  - Heavily depend on network connection
  - Low user engagement



Source: comScore Mobile Metrix, U.S., Age 18+, June 2016

Limitations of Native Apps

- App usage is highly concentrated
- Reasons:
  - High cost
  - Difficult to share



Source: comScore Mobile Metrix, U.S., Age 18+, June 2016

Progressive **W**eb **A**pps (PWAs)

- Introduced by Google in 2015
- Three design goals: *reliable, fast, engaging*
- Success stories
  - Twitter Lite
  - Financial Times
  - Forbes

Add to Home Screen **Offline Browsing** **Push Notifications**

Progressive **W**eb **A**pps (PWAs)

- Introduced by Google in 2015

Core Components:

1) *Service Worker*
2) *Cache*
3) *Push*

SERVICE WORKER

CACHE

PUSH

This Study

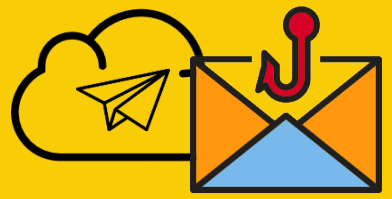
- We addressed the security and privacy risks to PWAs

Vulnerabilities:

1) *Service Worker → Cryptocurrency Mining*
2) *Cache → Inferring User's Browsing History*
3) *Push → Phishing Attack*

SERVICE WORKER

CACHE



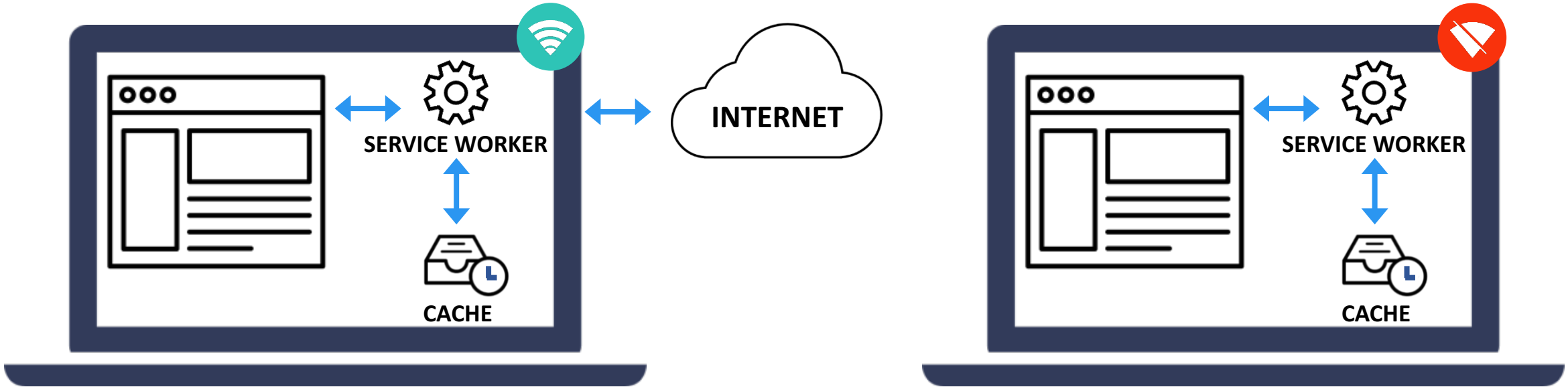
PUSH

Technology behind PWAs: Service Worker

- HTML5 Web standard technology
- Supported by most browsers:
  - Firefox 44+, Chrome 45+, Edge 17+, Opera 32+
- Only usable on HTTPS websites
- Able to run in the background ***<u>even when a user leaves a website</u>***

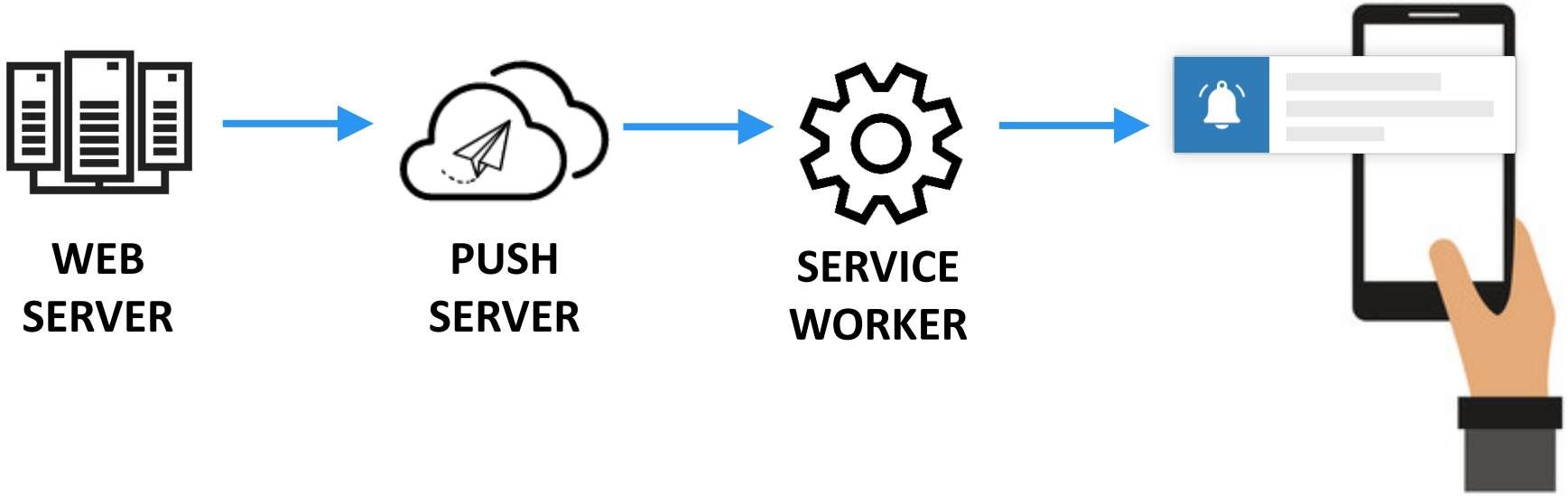
Offline Browsing

- ***Cache*** is an origin-bounded local storage
- Accessible regardless of the network status
- Provides programmable offline interfaces with Service Worker



Web Push Notifications

- Re-engaging users with customized content
- Can be received by Service Worker ***<u>even if the browser is closed</u>***



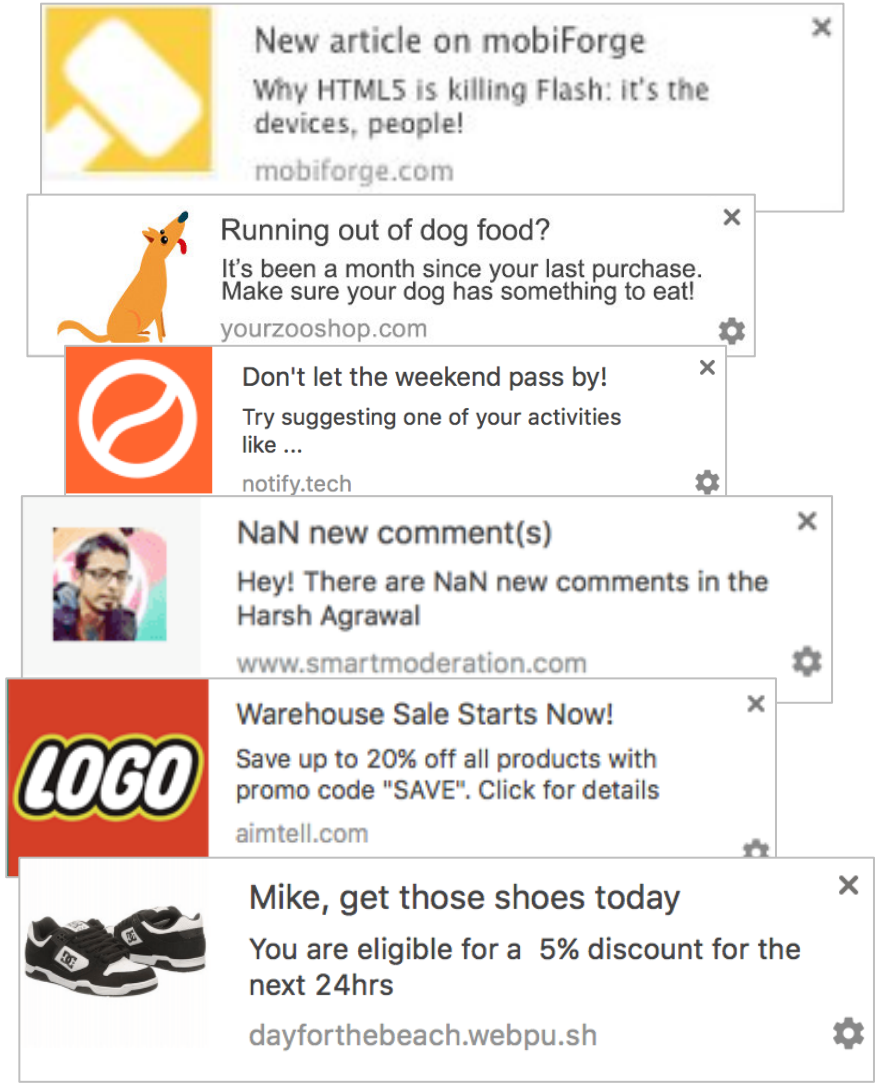
How Many PWAs Exist in the Wild?

- A PWA is a website that registers Service Worker
- Collected from the Alexa top 100,000 websites

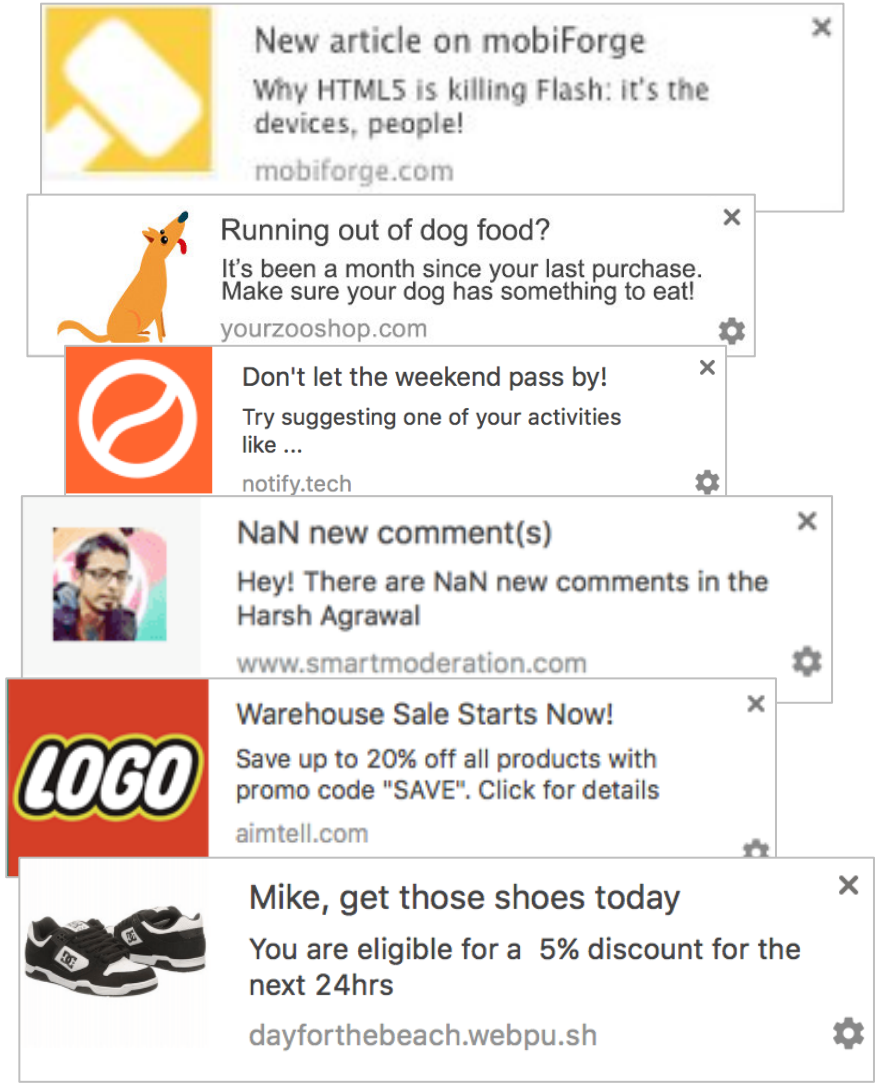
Features Used	Number of websites
Push	3,351 (80.5%)
Cache	513 (12.3%)
Both	196 (4.7%)
Others	495 (11.9%)
Total	**4,163 (100%)**

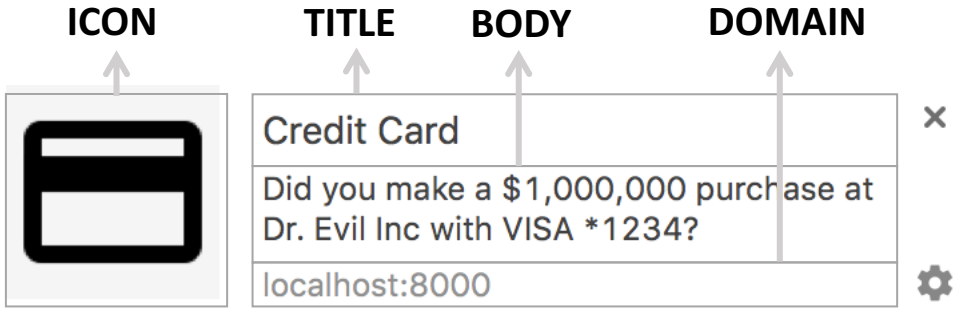
I-I. Phishing Risks of **Web Push**

General Appearance of Web Push

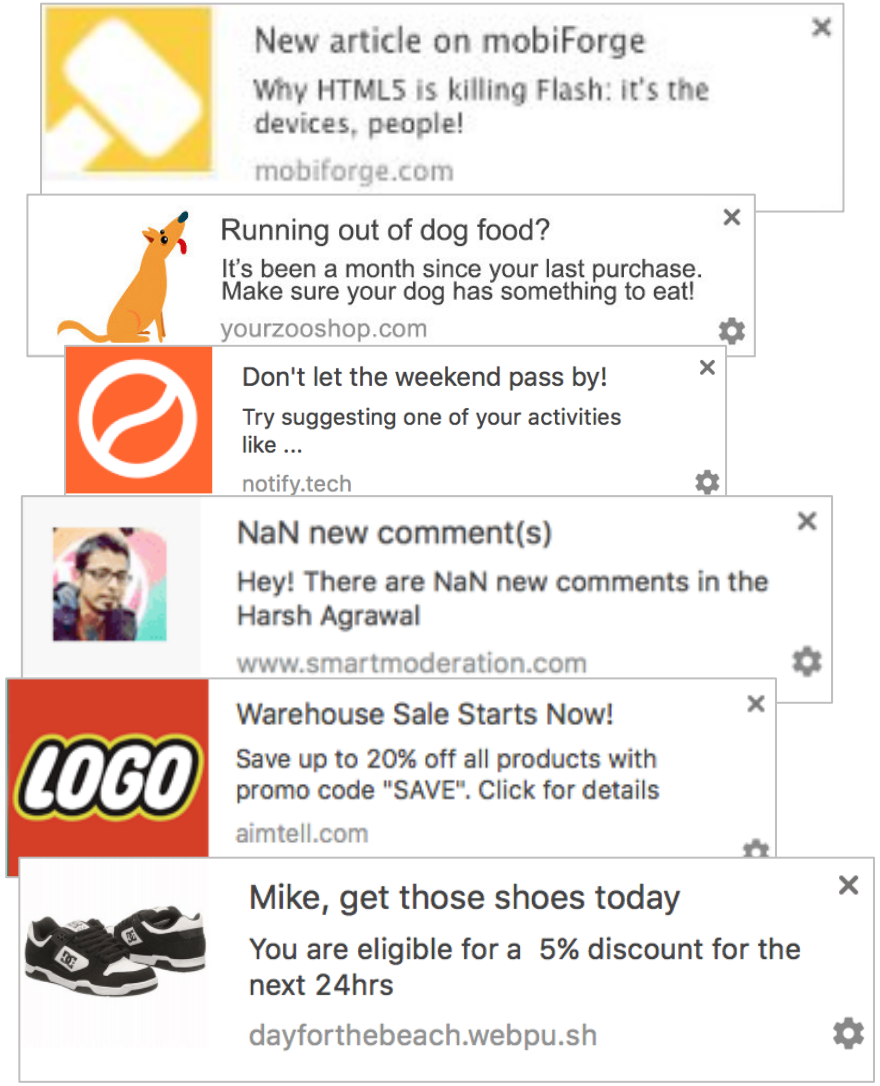


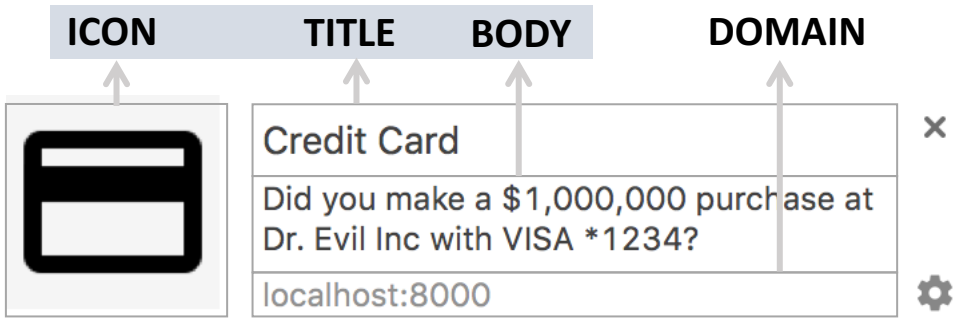
General Appearance of Web Push





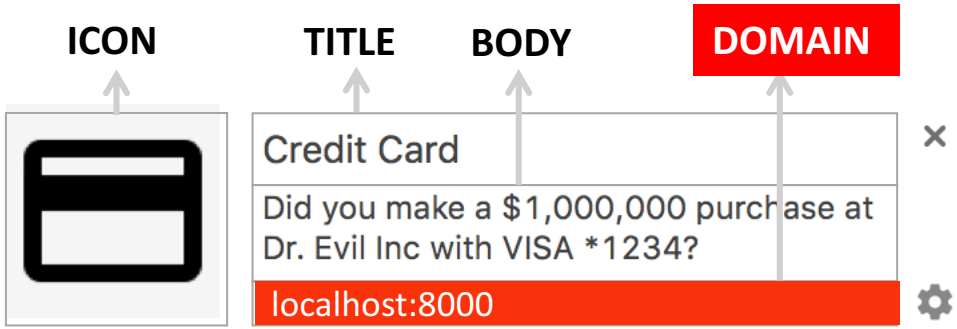
Sender Can Customize,





Sender Can Not Customize,

- **A domain name is the only element** representing the source of a push message



Vulnerabilities We Found

- The environments that **do not display domains**
  - *Firefox* on GNOME, Ubuntu MATE, Cinnamon, Budgie, and Pantheon
  - *Samsung Internet, Firefox* on Android
- Causes phishing risks

Push without domain	Push with domain
Firefox Gmail Account Manager 11:12 AM Your Gmail account needs to be validated. **Samsung Internet** Gmail Account Manager 11:10 AM Your Gmail account needs to be validated.	**Chrome** Gmail Account Manager 11:06 AM Your Gmail account needs to be validated. test.kaist.ac.kr

I-II. Phishing risks of **Third-Party Push Libraries**

Emerging Third-party Push Services

- Enable website owners to use push features
- Provide useful features:
  - Scheduling push notifications, Reporting the statistics of subscribers, ***<u>Supporting HTTP websites</u>***

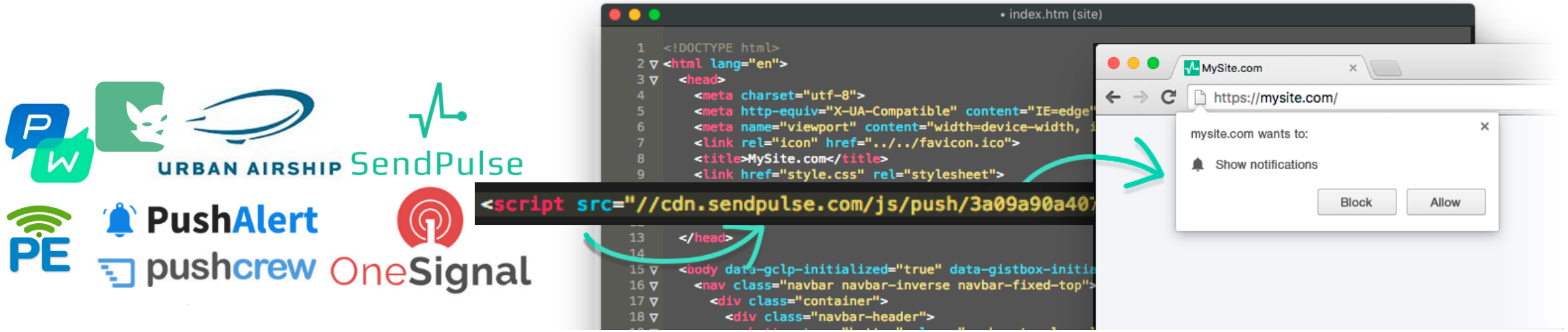
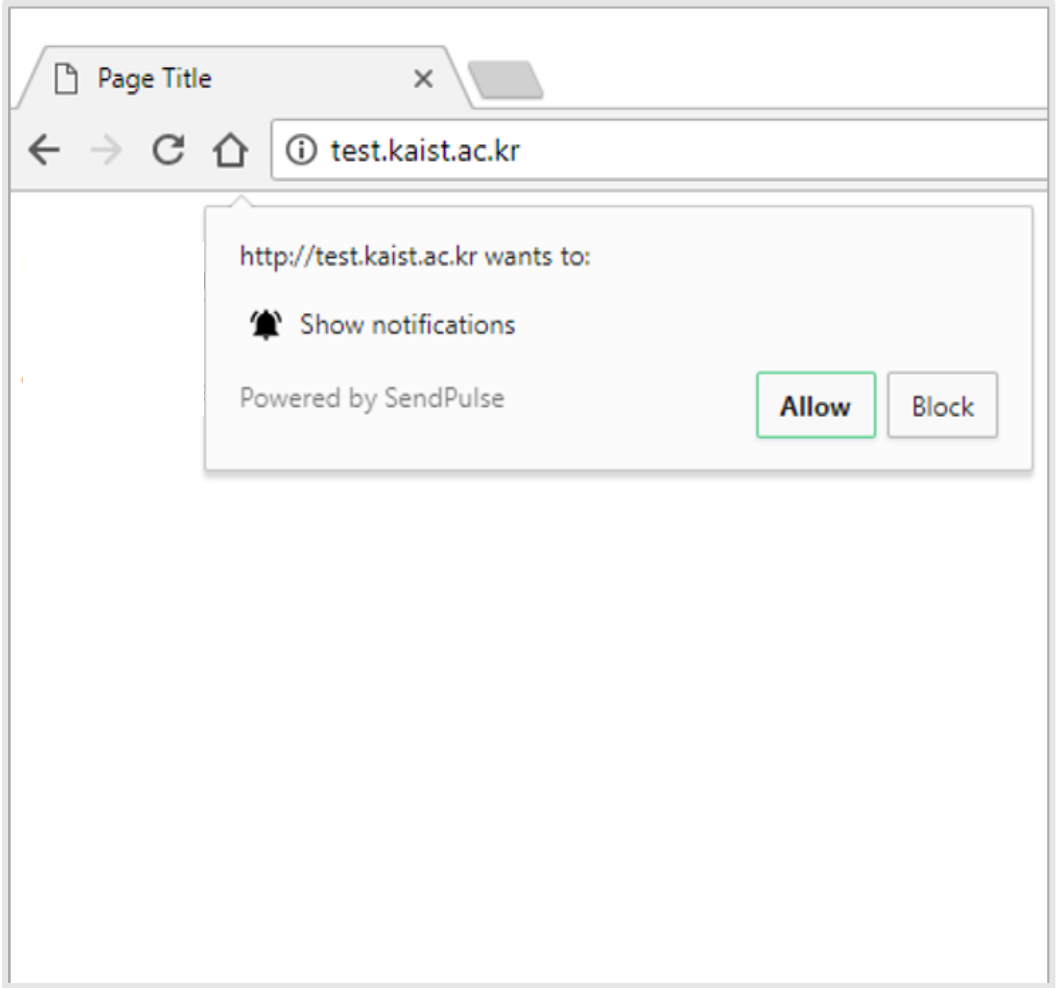
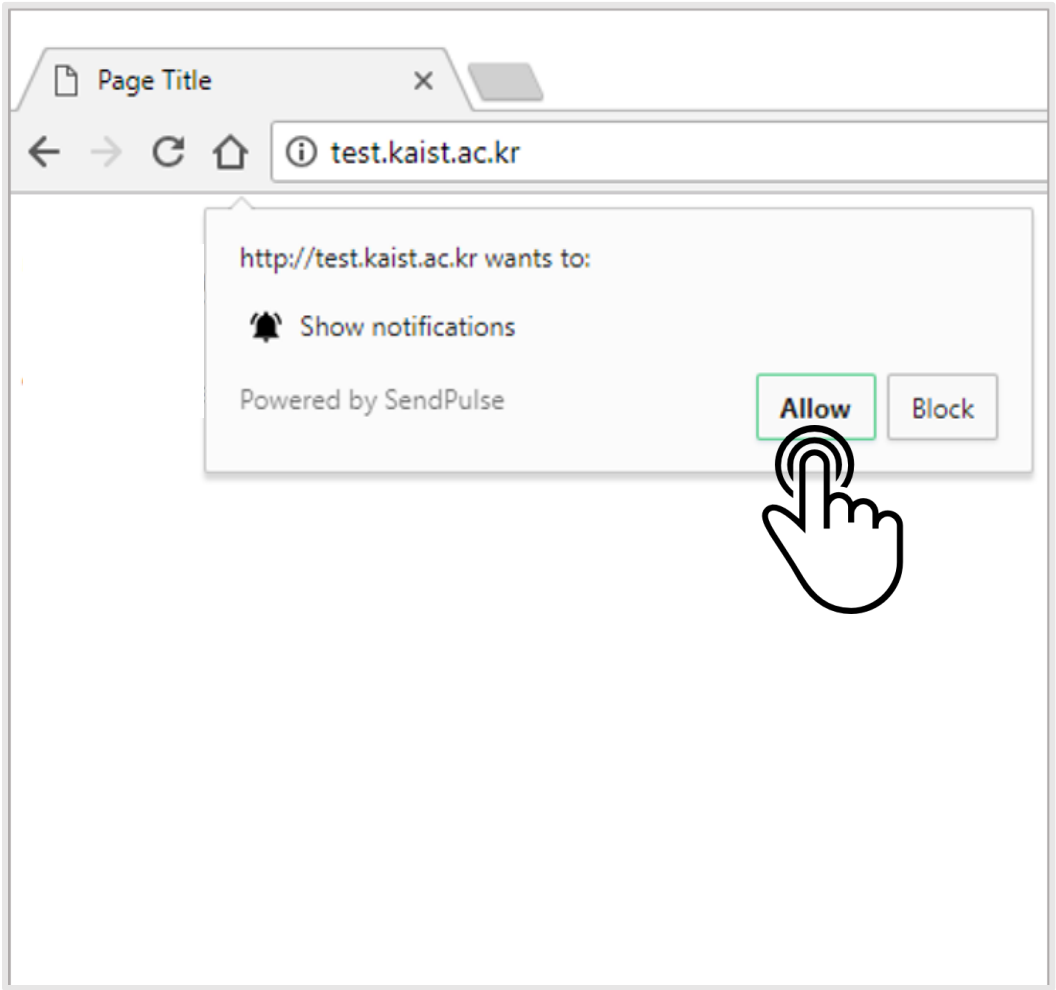


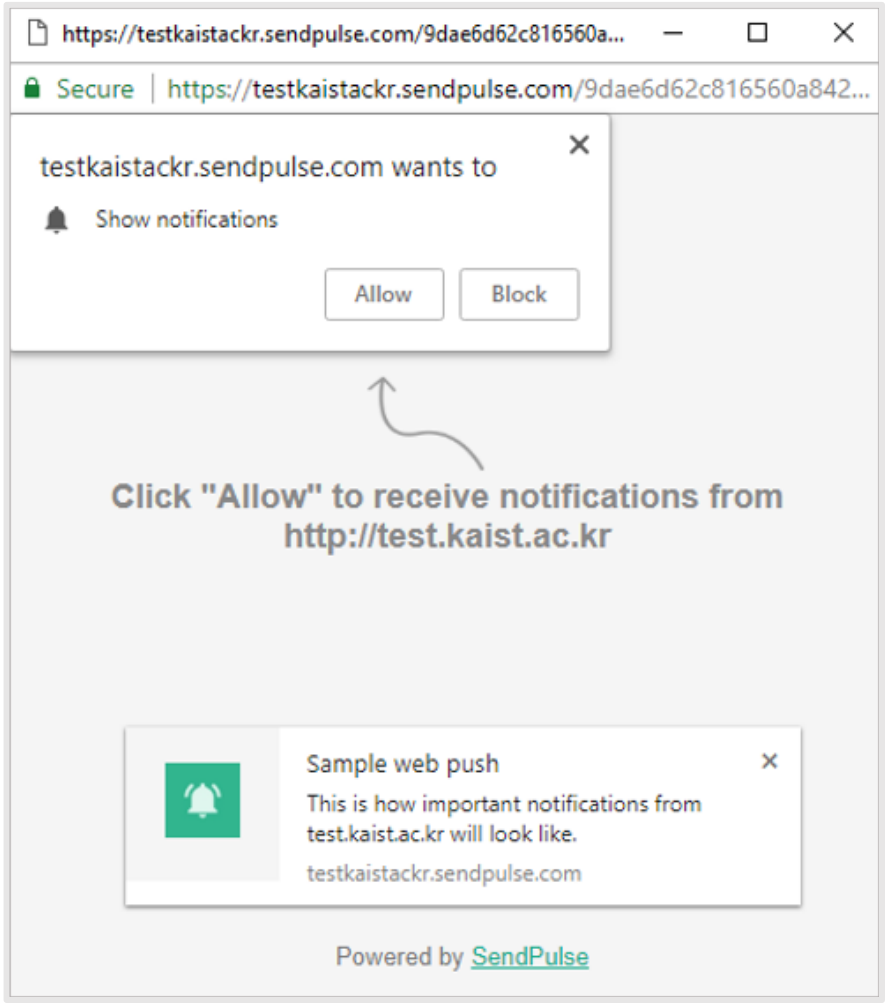
Image Source: https://sendpulse.com/features/webpush

How push is Supported on HTTP Sites

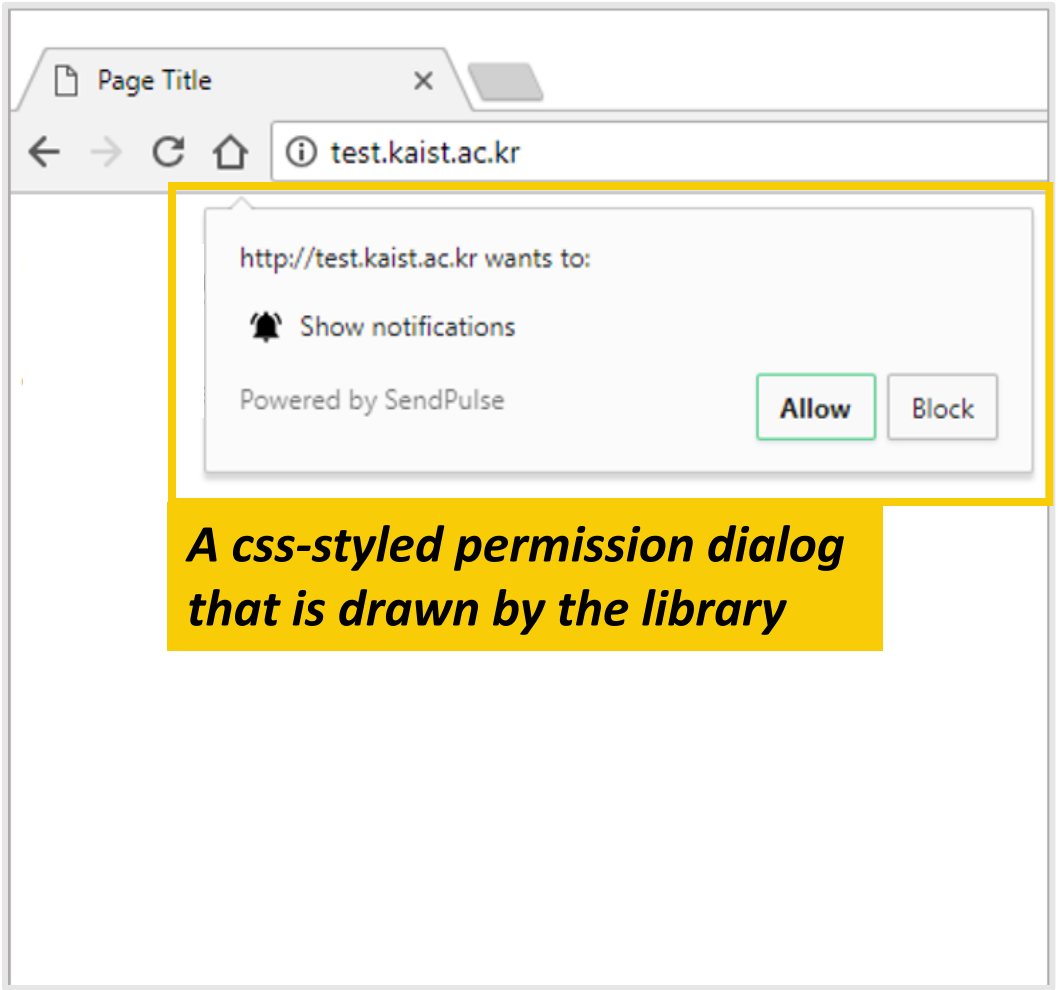


How push is Supported on HTTP Sites

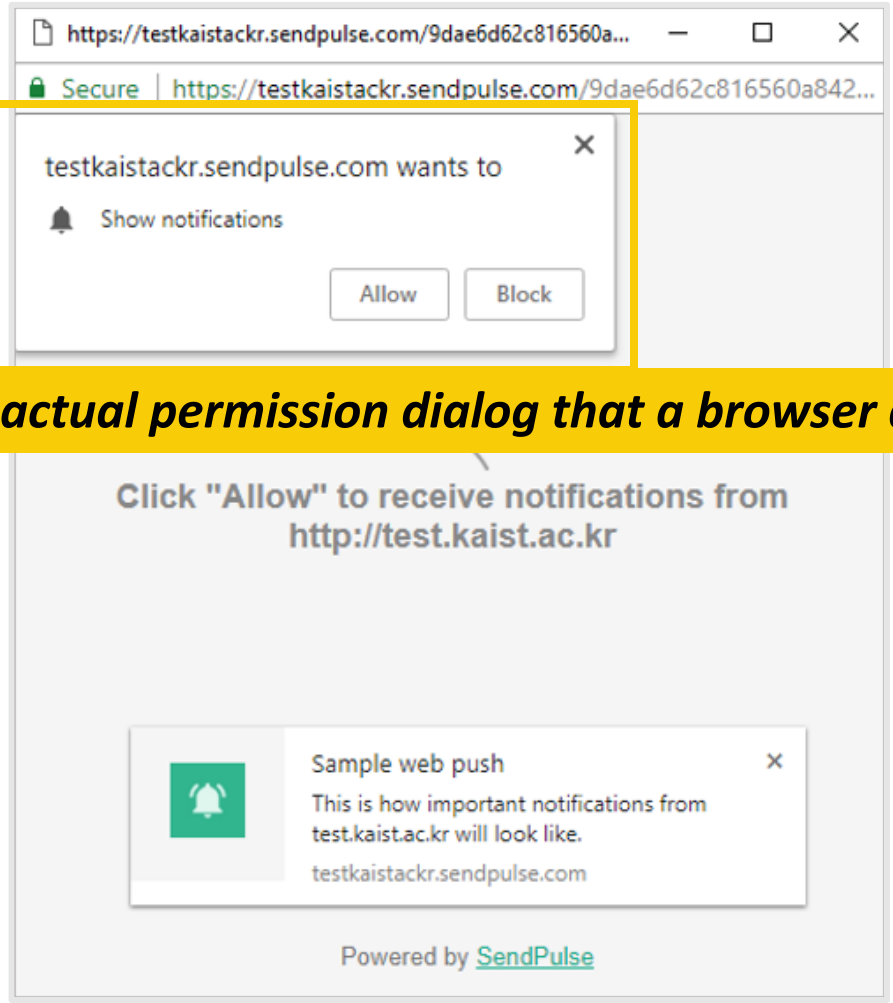




How push is Supported on HTTP Sites

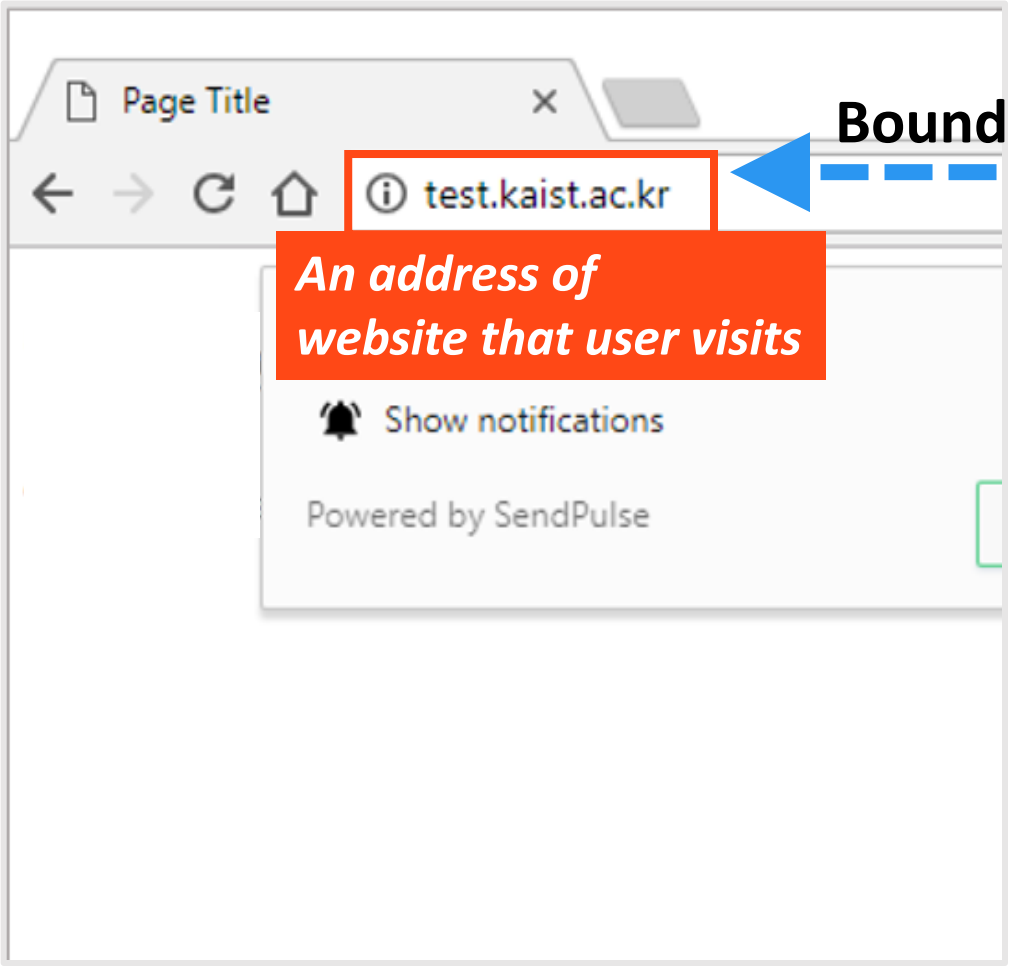


A css-styled permission dialog that is drawn by the library

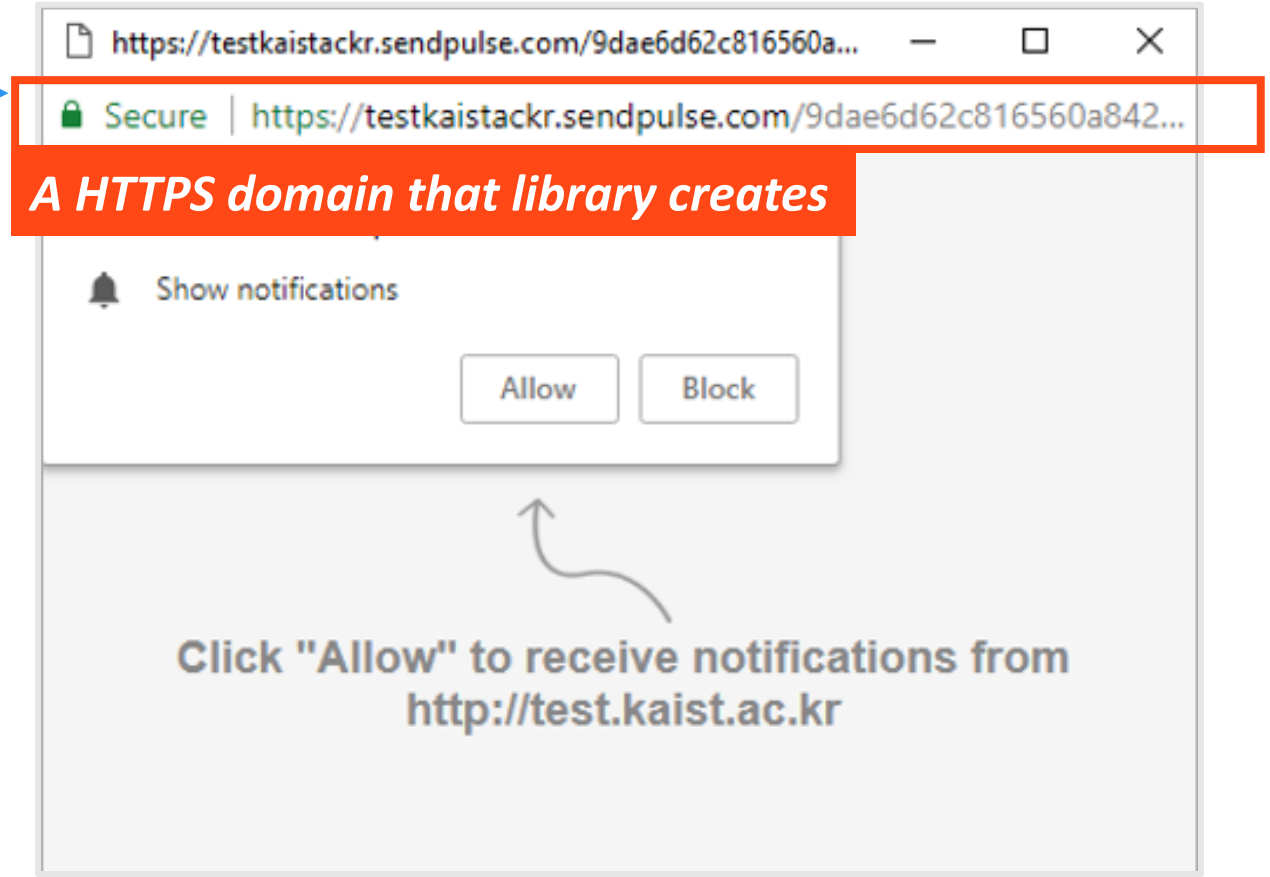


An actual permission dialog that a browser asks

How push is Supported on HTTP Sites

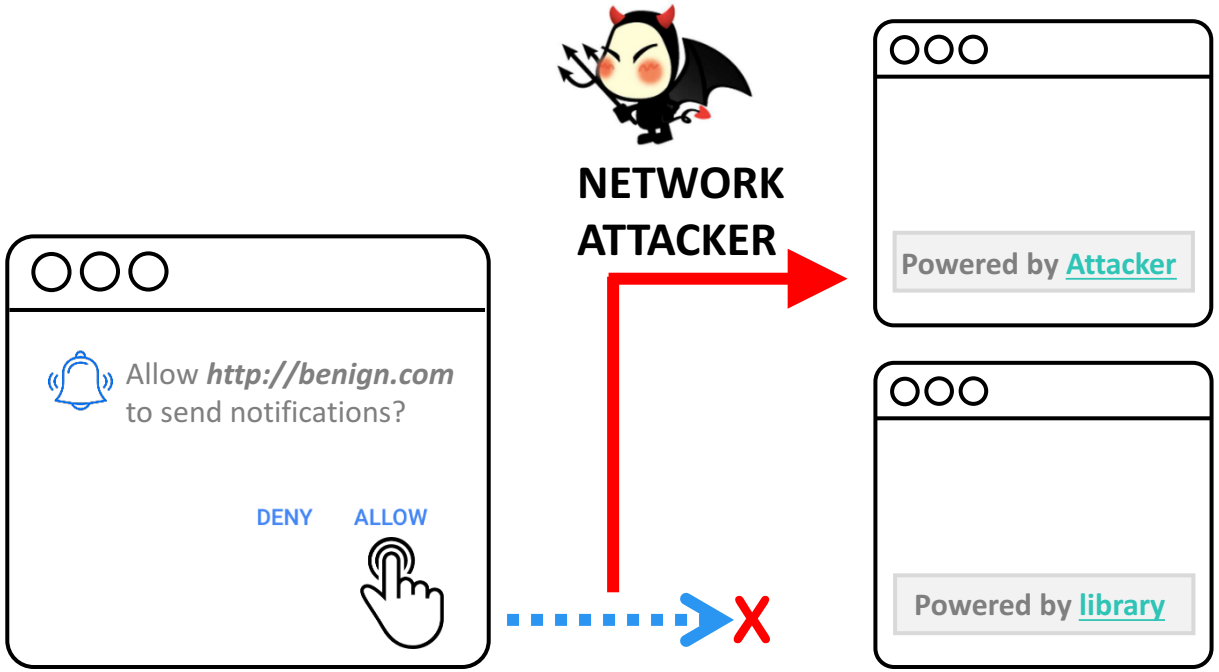


Bounding



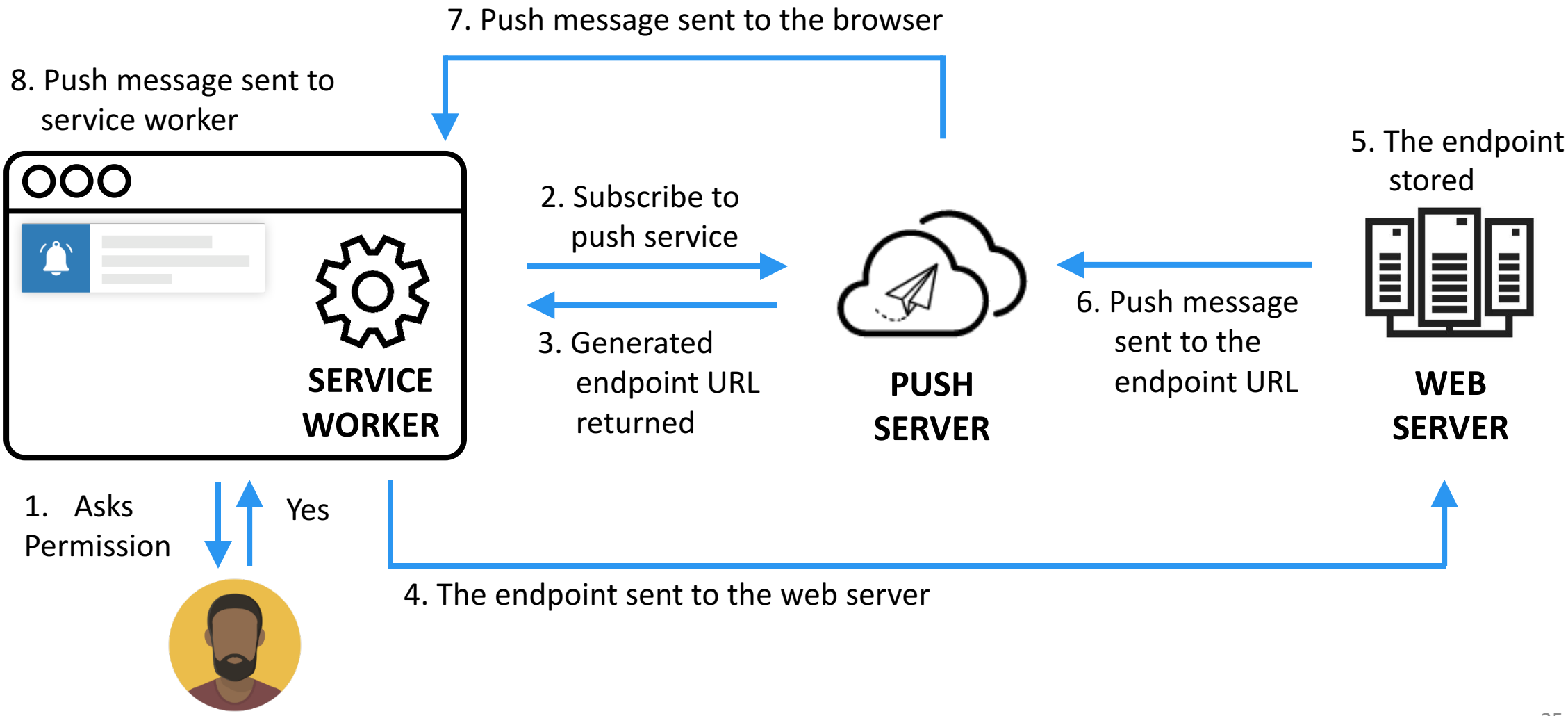
Permission Delegation Attack

- A network attacker can redirect users to an attacker-controlled website
- A visitor has no clue why she is redirected to a different domain



I-III. Domain Name Spoofing Attack of **Web Push Notifications**

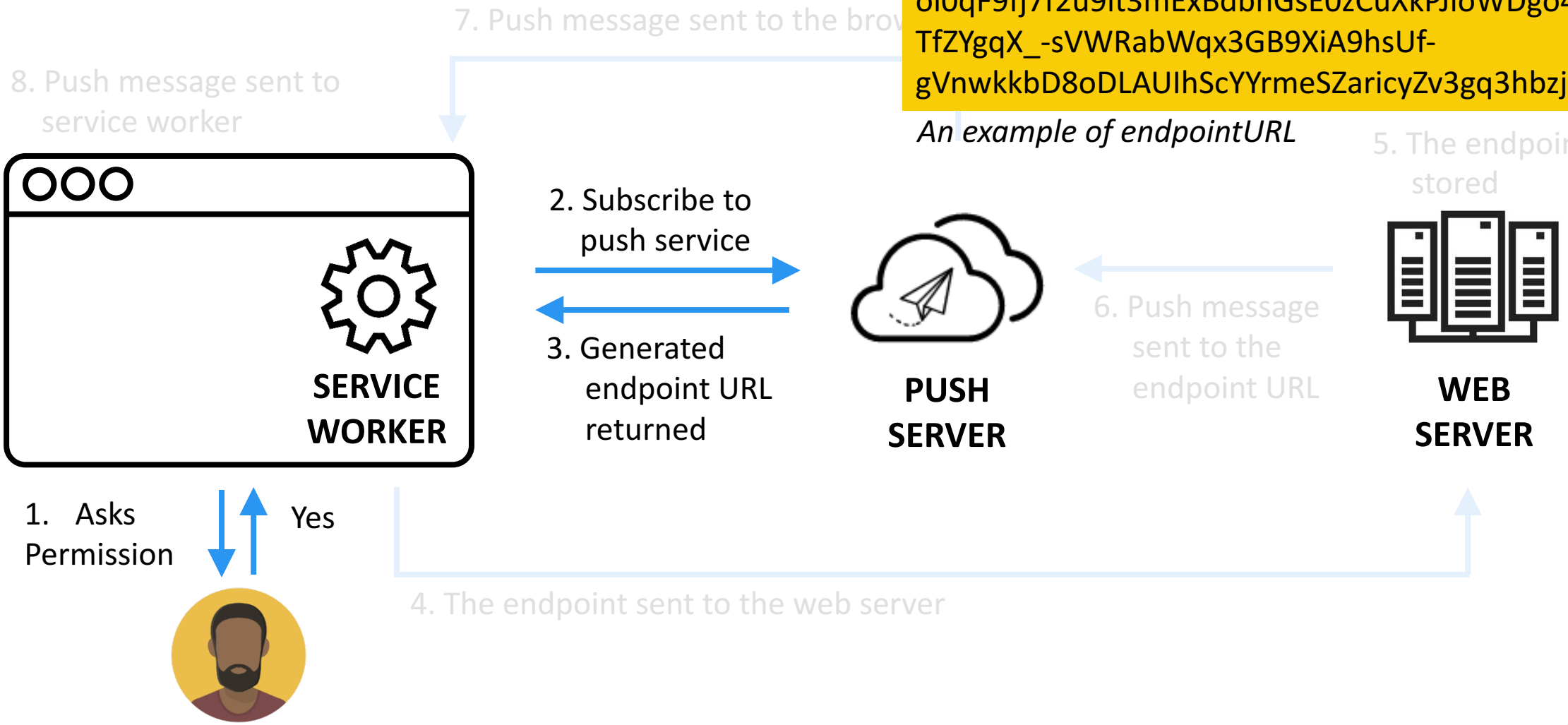
Web Push in Detail



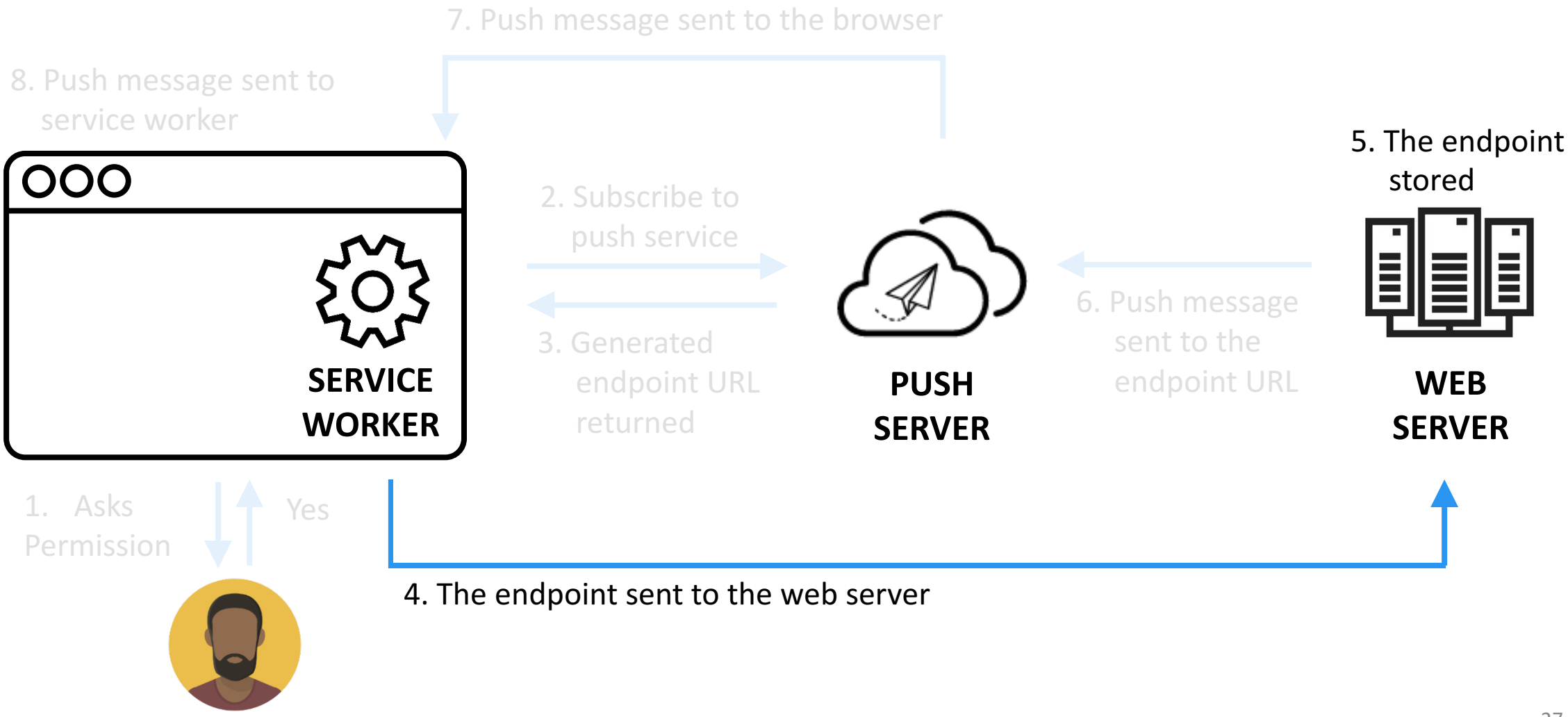
Web Push in Detail

https://fcm.googleapis.com/fcm/send/dTb6ILBpUYs:APA91bGX_Xa91bizHC-ol0qF9fj7f2u9lt3mExBdbhGsE0zCuXkPJioWDgo4wf1mTfZYgqX_-sVWRabWqx3GB9XiA9hsUf-gVnwkkbD8oDLAUIhScYYrmeSZaricyZv3gq3hbzjh48Ad

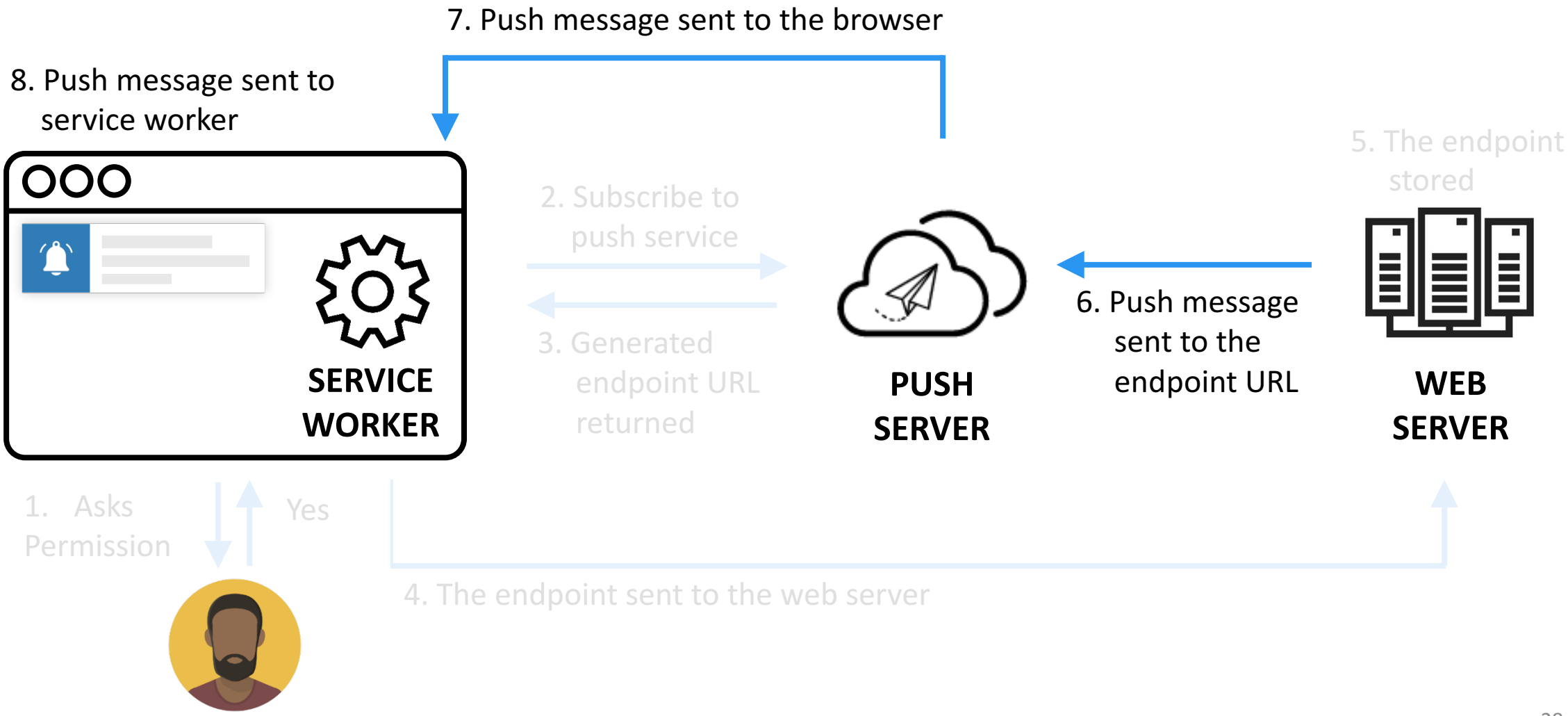
An example of endpointURL



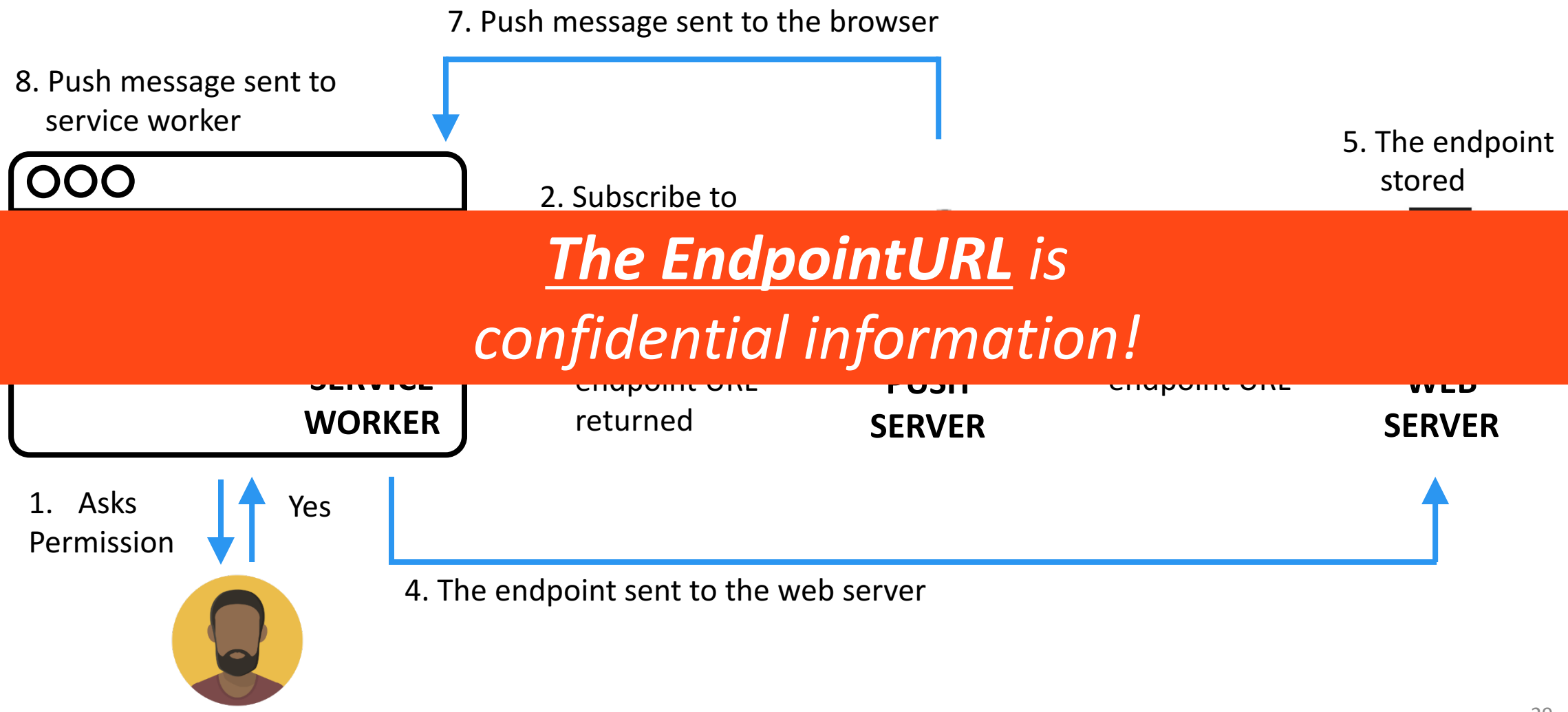
Web Push in Detail



Web Push in Detail

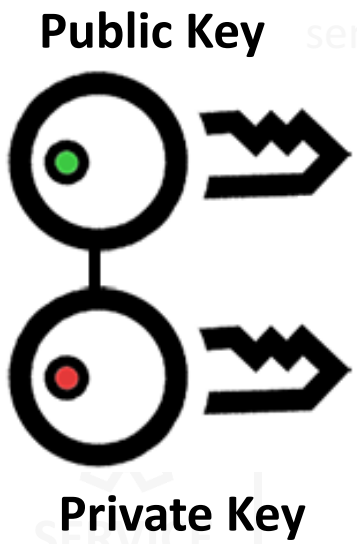


Web Push in Detail



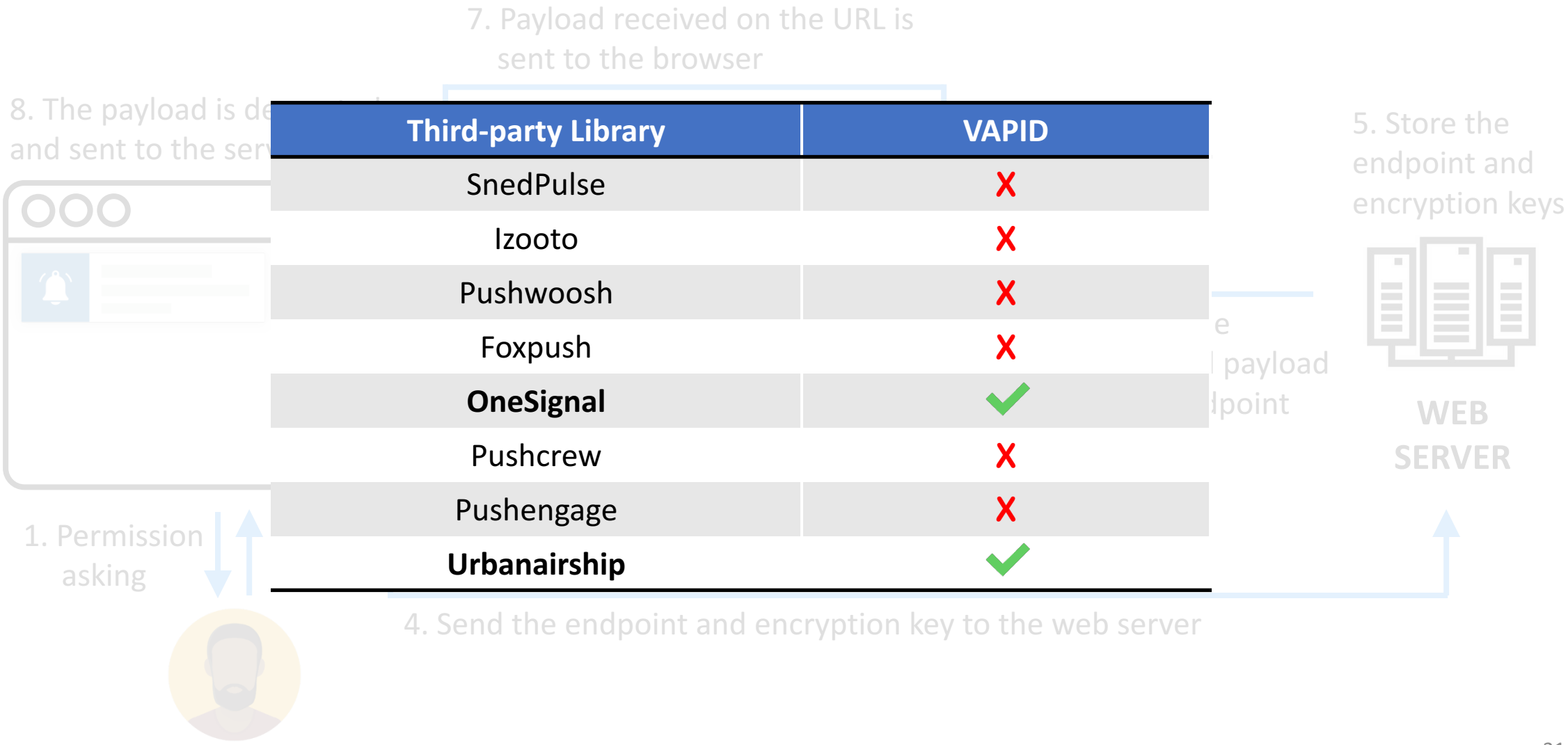
<u>The EndpointURL</u> is confidential information!

Web Push Protocol: **VAPID**



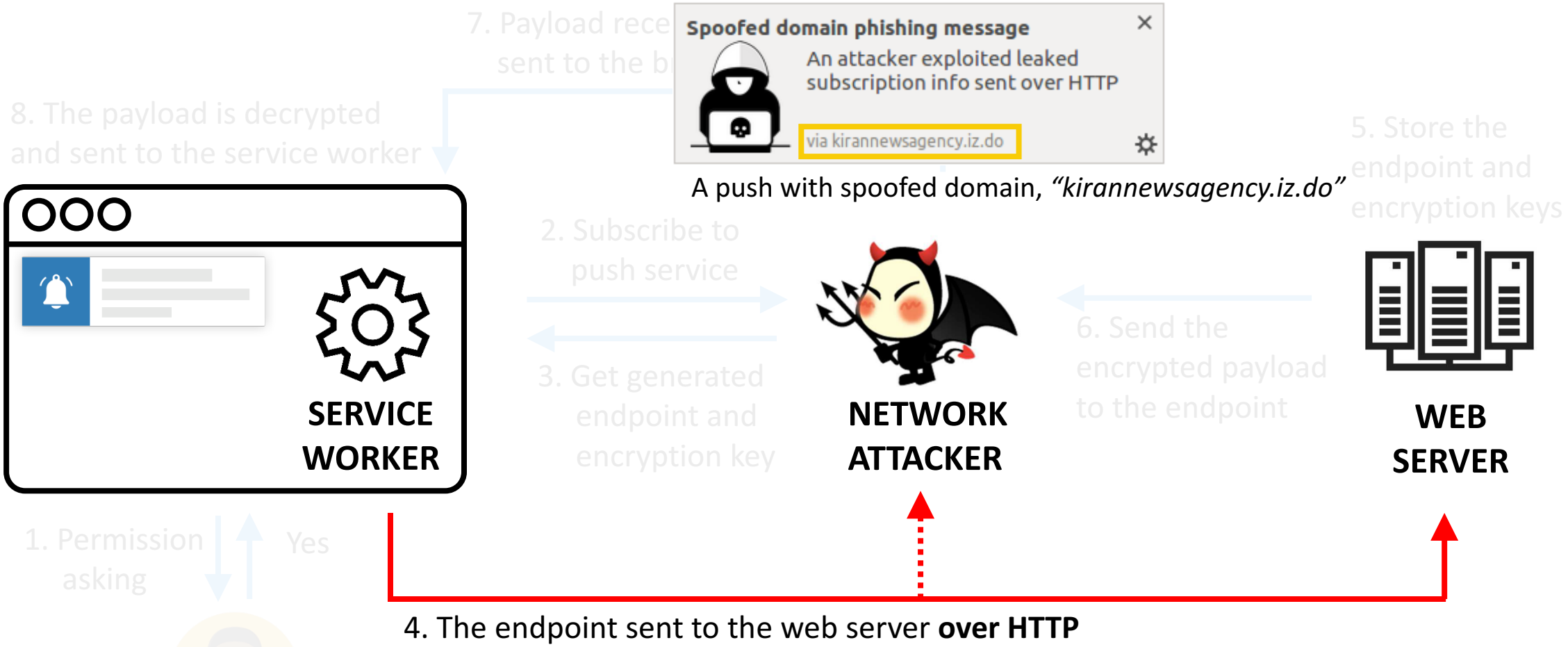
- Designed to authenticate web servers
- Utilizes asymmetrical key pairs
  - Without a private key, cannot send push messages

VAPID in the Wild



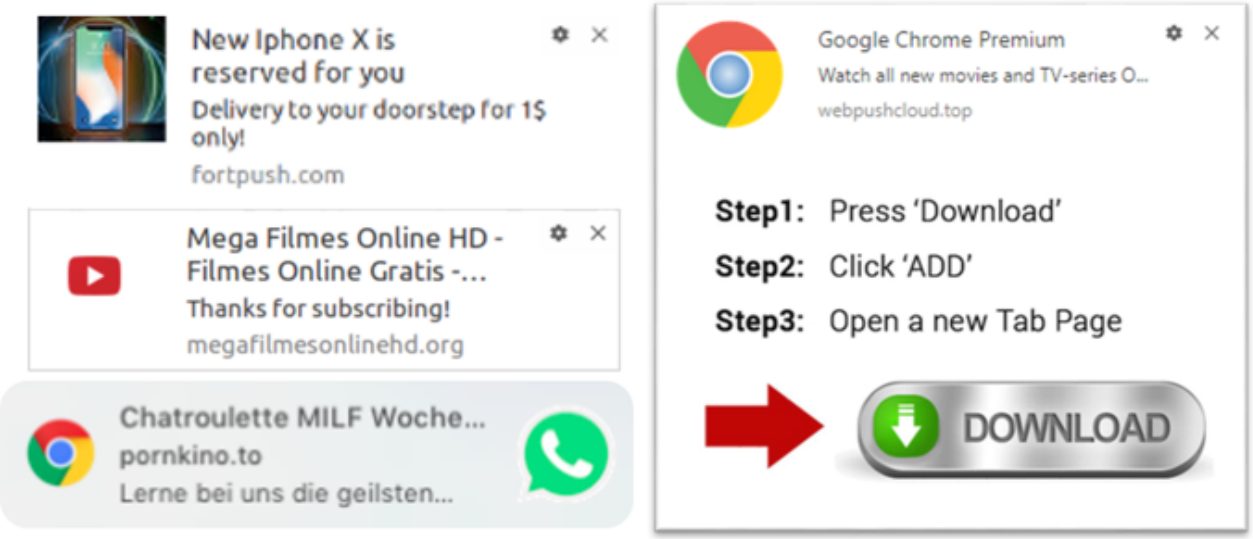
Third-party Library	VAPID
SnedPulse	✗
Izooto	✗
Pushwoosh	✗
Foxpush	✗
OneSignal	✓
Pushcrew	✗
Pushengage	✗
Urbanairship	✓

Domain Spoofing Attack



Why **Phishing via Web Push** Matters?

- Difficult to determine the origin of messages
- An attacker can send push messages at any time



Real-world phishing

II. User Privacy Leak via **Offline Usage**

History Sniffing Attack

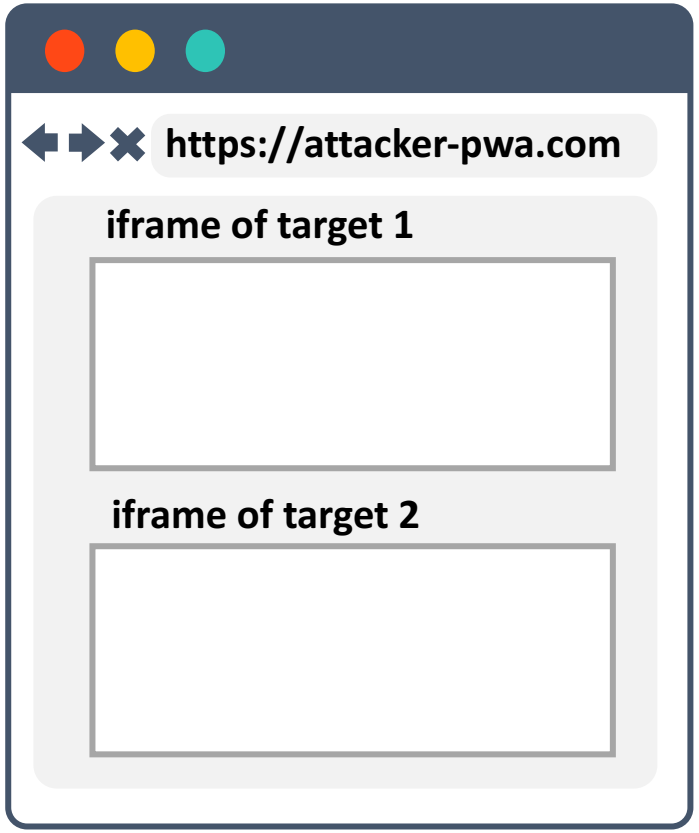
- Critical privacy threat
  - E. Felten at al., Timing Attacks on Web Privacy [CCS 2000]
  - Z. Weinberg at al., I Still Know What You Visited Last Summer: Leaking Browsing History via User Interaction and Side Channel Attacks [S&P 2011]
  - S. Son at al., What Mobile Ads Know About Mobile Users [NDSS 2016]
- Can leak personal information

History Sniffing Attack on PWAs

- A new side channel attack that exploits ***Cache***

History Sniffing Attack on PWAs

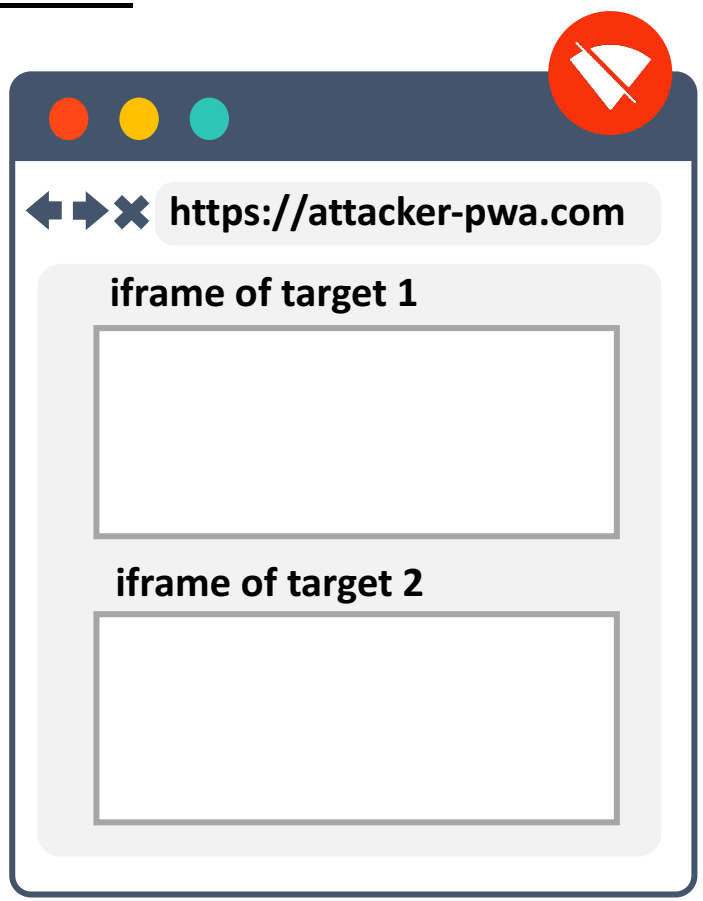
- A new side channel attack that exploits ***Cache***
- How it works:



History Sniffing Attack on PWAs

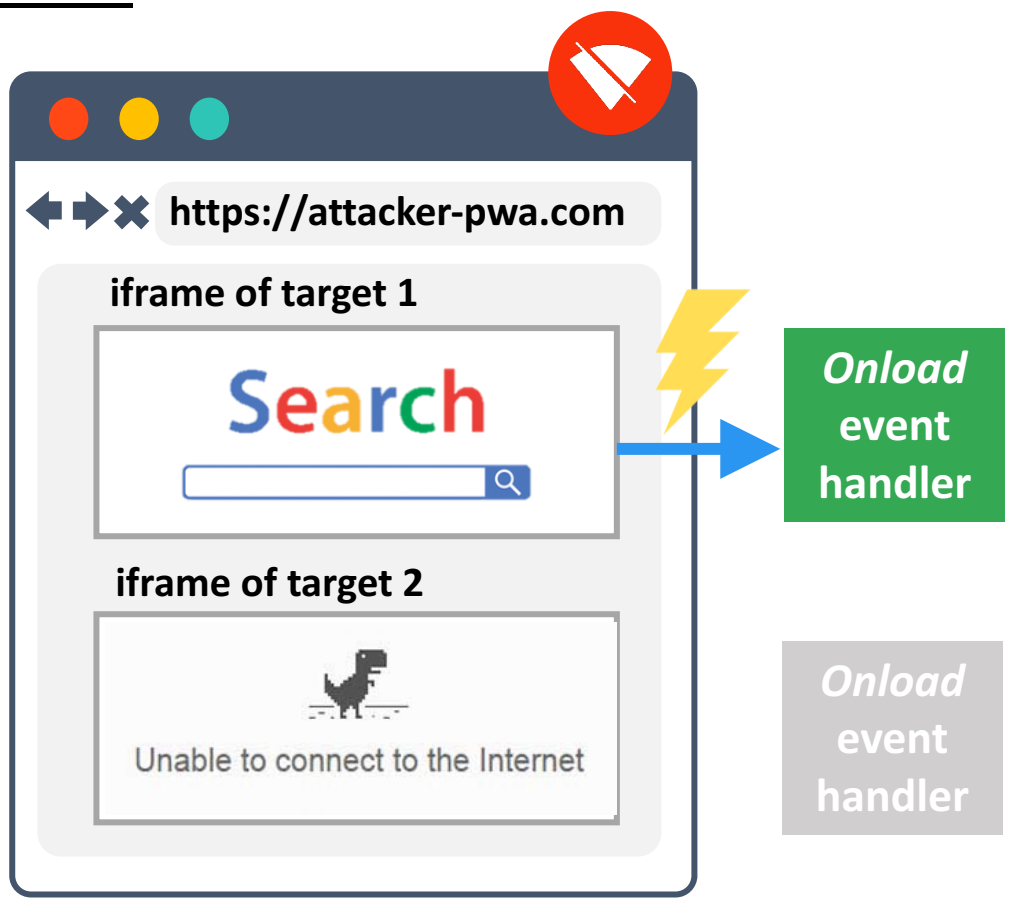
- A new side channel attack that exploits ***Cache***
- How it works:

  1. A victim opens the attacking PWA offline



History Sniffing Attack on PWAs

- A new side channel attack that exploits ***Cache***
- How it works:
  1. A victim opens the attacking PWA offline
  2. An *onload* event will only be triggered if victims have visited target PWAs



History Sniffing Attack on PWAs

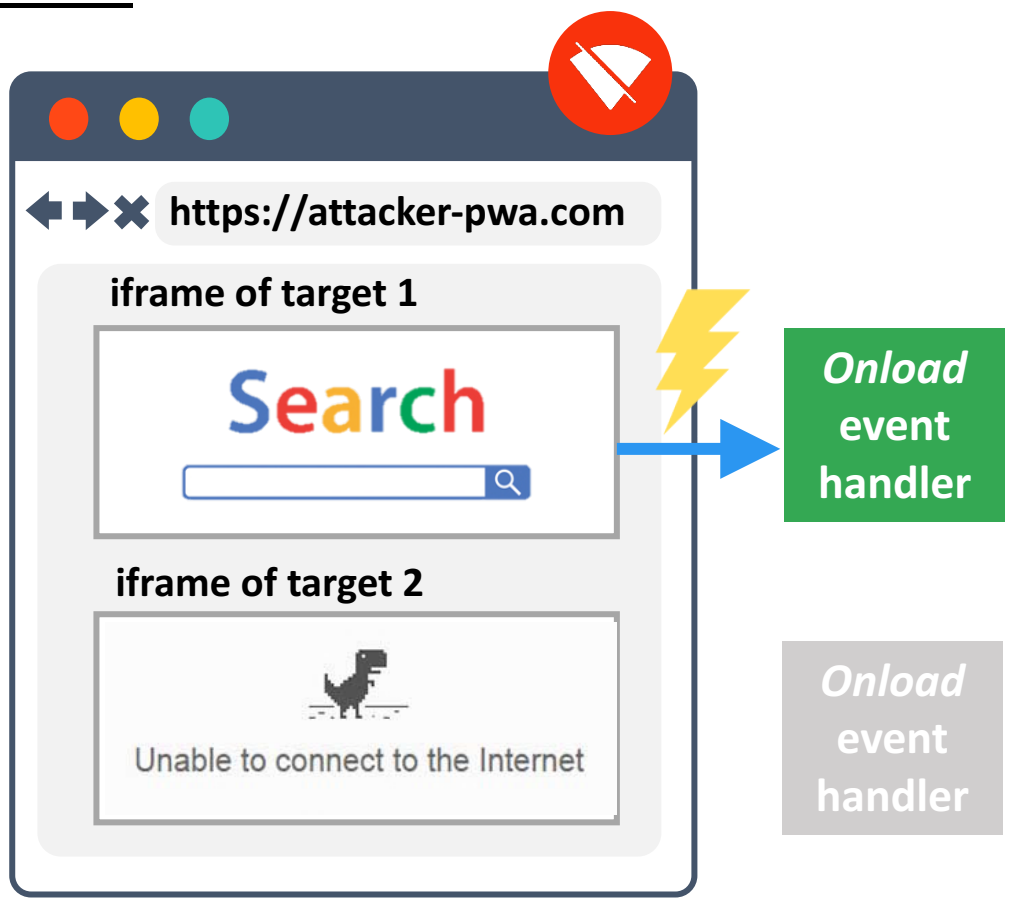
- A new side channel attack that exploits ***Cache***
- How it works:

  1. A victim opens the attacking PWA offline
  2. An *onload* event will only be triggered if victims have visited target PWAs

Advantages:

1) Accuracy

2) No outgoing requests



Consequences of History Sniffing Attack

- Vulnerable Browser: Firefox 59.0.2
- X-Frame-Options, CSP, and Frame Busting are effective to defense

Offline Cache Attack		# of Websites
Vulnerable		**187 (36.5%)**
Not Vulnerable	X-Frame-Options	132 (25.7%)
	CSP	22 (4.3 %)
	Frame Busting	10 (1.9%)
	Others	162 (31.6%)
Total		513 (100%)

- *Safari manages cache separately from the first-party*

III. Cryptocurrency Mining Attack **Using Service Worker**

Cryptocurrency Mining in the Web

- *CoinHive* is a popular JavaScript cryptocurrency mining service
- Main Limitation:
  - Stops when user leaves

COINHIVE

Cryptocurrency Mining Attack

- *CoinHive* is a popular JavaScript cryptocurrency mining service
- Main Limitation:
  - Stops when user leaves

- Introducing cryptocurrency mining attack ***using Service Worker***

Advantages:

1) *Stealthy*
2) *Lasting Longer*

COINHIVE

SERVICE WORKER

Cryptocurrency Mining Attack

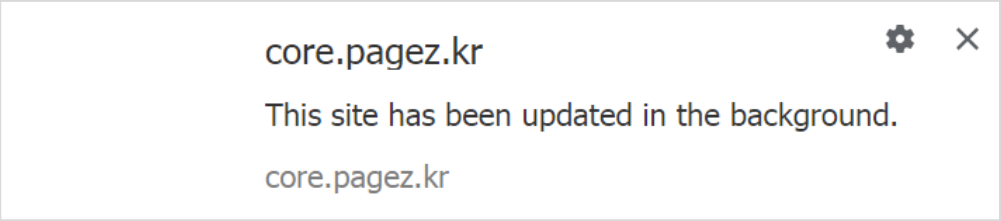
- Technical challenges:
  - Service Worker becomes idle
  - Service Worker cannot use *WebSocket*

Cryptocurrency Mining Attack

- Technical challenges:
  - Service Worker becomes idle
  - Service Worker cannot use *WebSocket*
- Solution:
  - *Push notifications*

Cryptocurrency Mining Attack

- Two tricks:
  - Non-visible push
  - Re-subscription
- Different browsers have different policies:



A warning sign if push API is not called

	Whale	Brave	Samsung Internet	Opera	Chrome	Edge	Firefox
Non-visible push	X	X	X	X	X	✓	✓
Re-subscription in the background	-	-	-	-	-	X	✓

Most stealthy!

Cryptocurrency Mining Results

- Mined *Monero* coins for 24 hours using a single service worker

Browser	Environment	Number of Solved Hashes (24h)	Amount of Monero (24h)
Chrome 65	Window 10 Desktop (3.6GHz Intel Core i7, 16GB)	225,024	0.00001266
Firefox 69	Window 10 Desktop (3.6GHz Intel Core i7, 16GB)	195,840	0.00001119
Chrome 65	Android 8.0 Google Pixel	50,176	0.00000282
Chrome 65	macOS High Sierra 10.13.4 (1.3GHz Intel Core i5, 8GB)	138,496	0.00000778

- ***The more victims, the more lucrative this attack is***

Lessons Learned

- Web Push requires careful use
  - adopt VAPID
  - treat EndpointURL as confidential information
- Well known defenses are helpful
- Better design for supporting web push for HTTP websites is Required

Conclusion

- The first in-depth study of PWAs
- Proposed novel attacks that abuse fundamental features of PWAs
- Provided mitigating recommendations
- Reported findings to corresponding vendors
- All demonstrations can be found at https://github.com/spostman/ppp-ccs2018

Thank You!

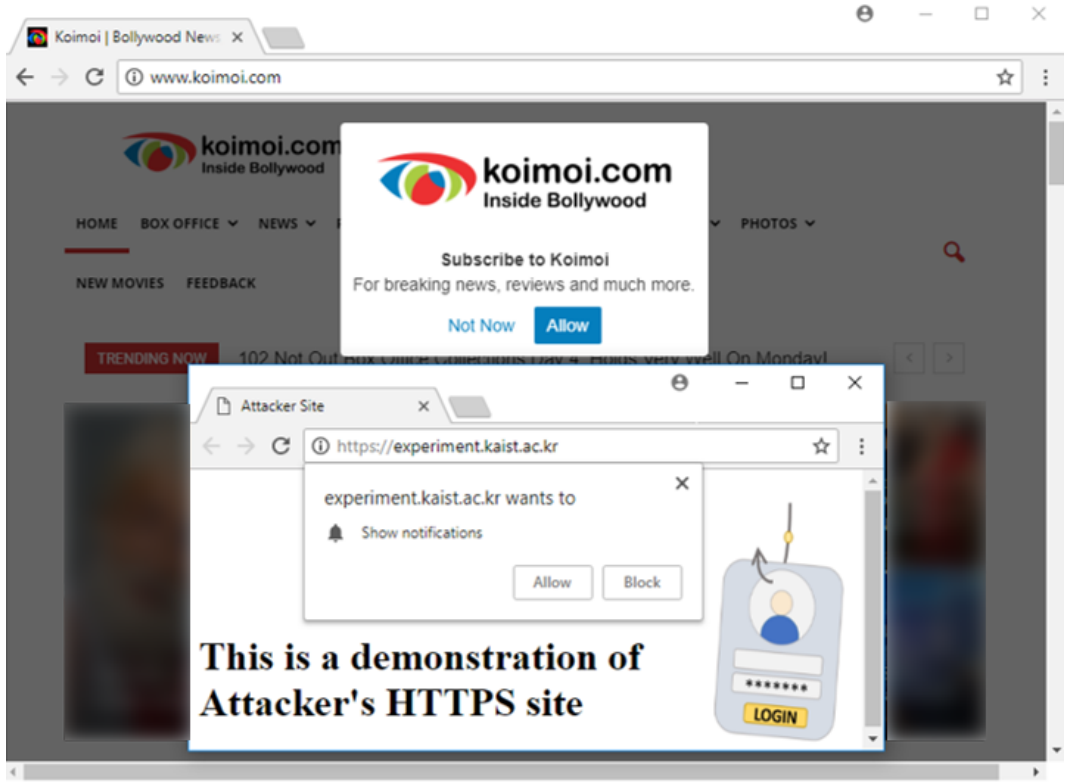
Q&A

KAIST

Web Security & Privacy Lab

Consequences of Permission Delegation Attack

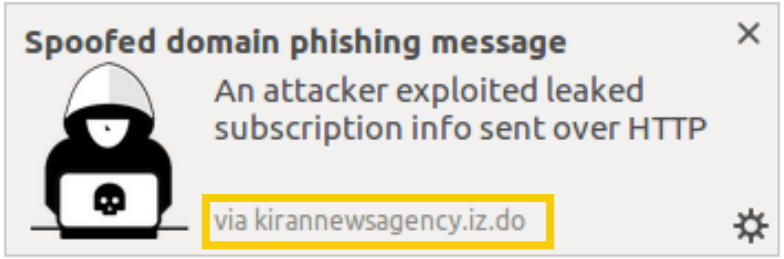
Third-party Library	Attack Success	Number of affected HTTP sites
SnedPulse	✓	**93**
Izooto	✓	**18**
Pushwoosh	✓	**4**
Foxpush	✓	**1**
OneSignal	X	528
Pushcrew	X	31
Pushengage	X	19
Urbanairship	X	2



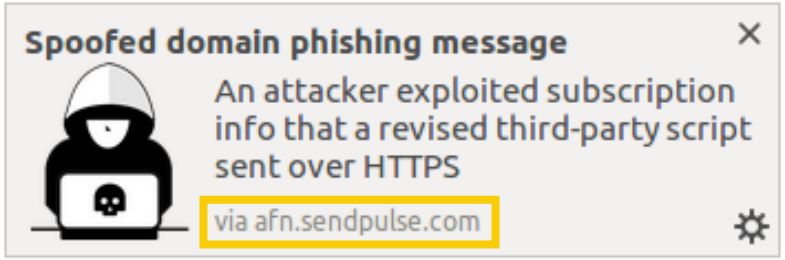
A permission delegation attack against *http://www.koimoi.com*

Domain Spoofing Attack Implication

Third-party Library	Attack Success	Number of affected HTTP sites
SnedPulse	✓	93
Izooto	✓	18
Pushwoosh	✓	4
Foxpush	✗	1
OneSignal	✗	528
Pushcrew	✗	31
Pushengage	✗	19
Urbanairship	✗	2

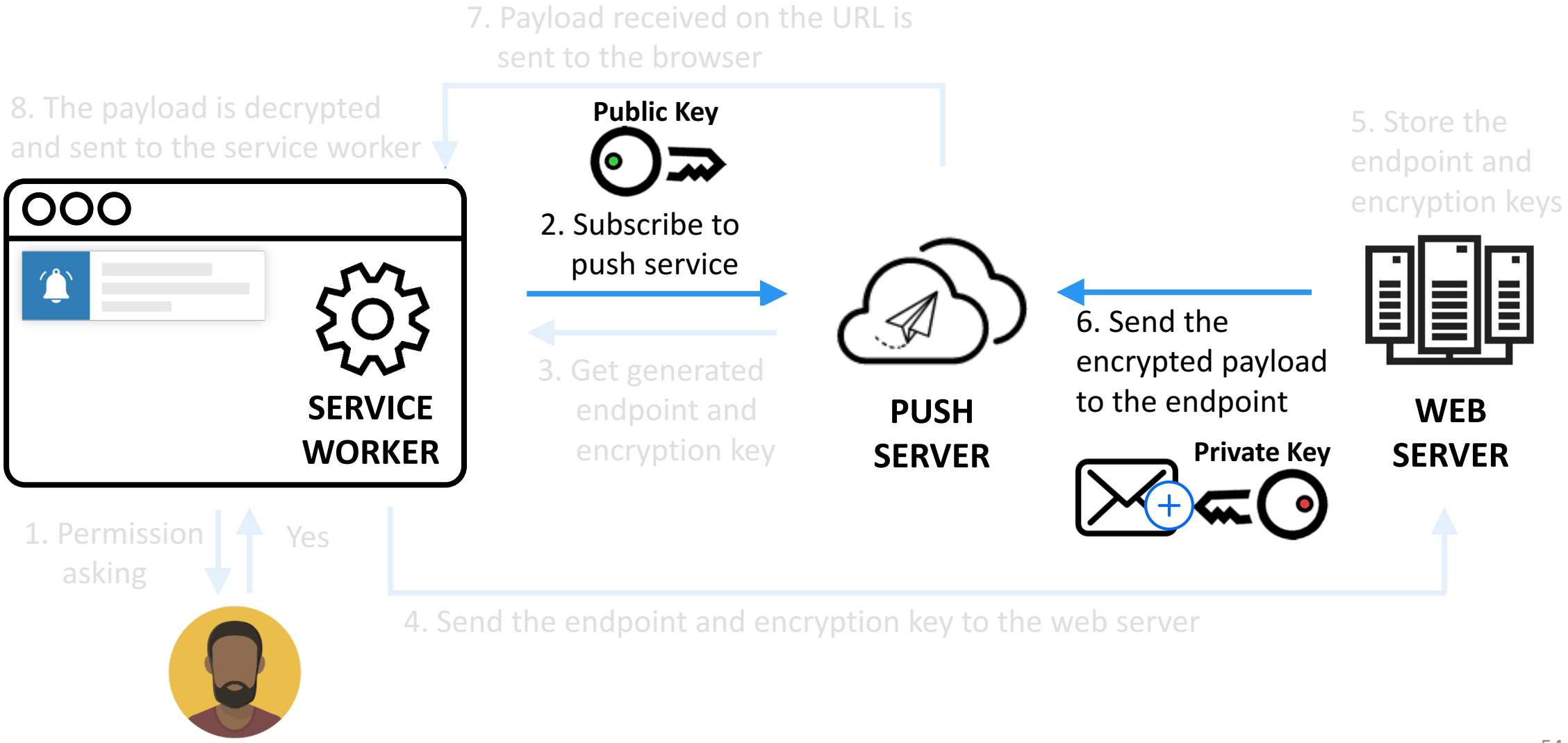


A push with spoofed domain, *"kirannewsagency.iz.do"*

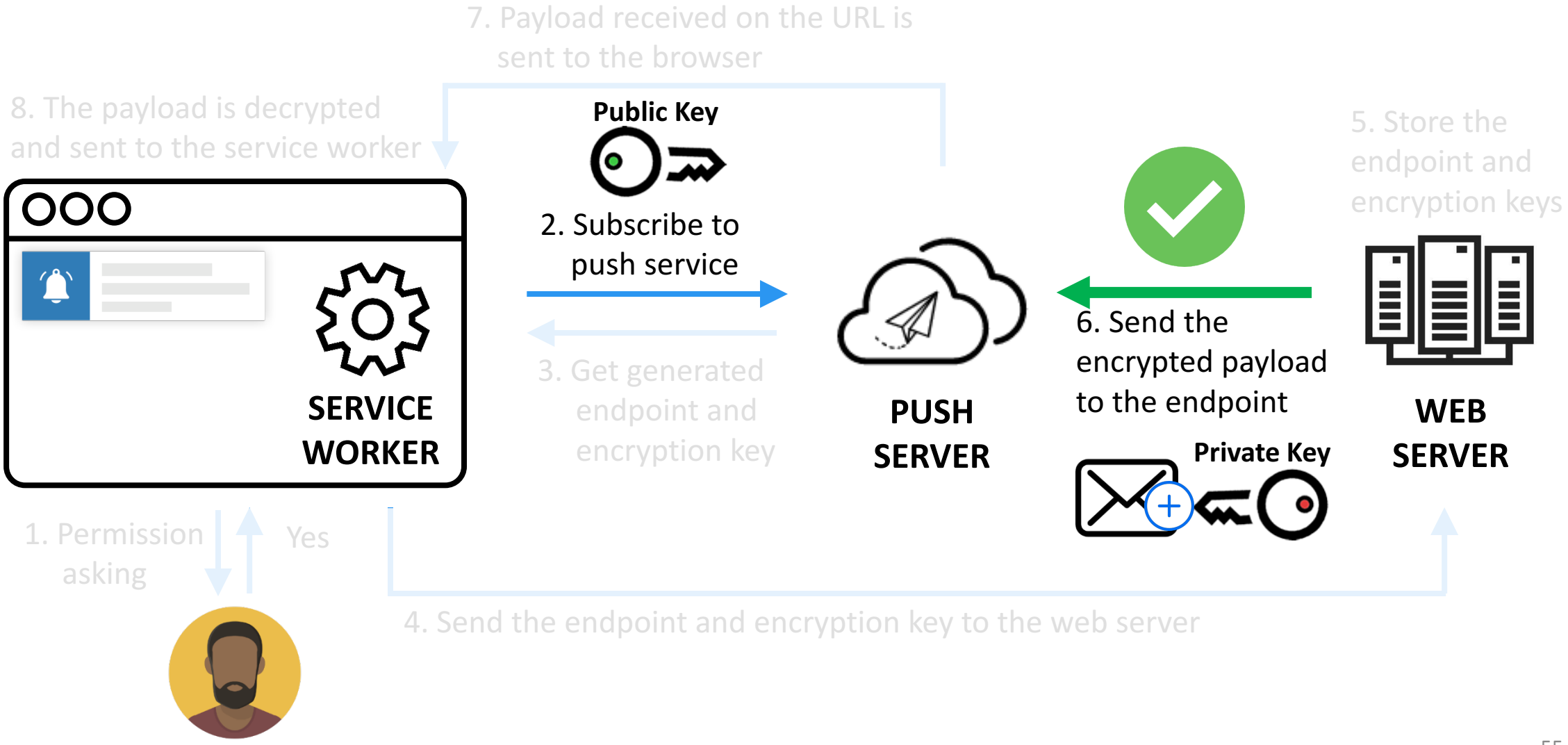


A push with spoofed domain, *"afn.sendpulse.com"*

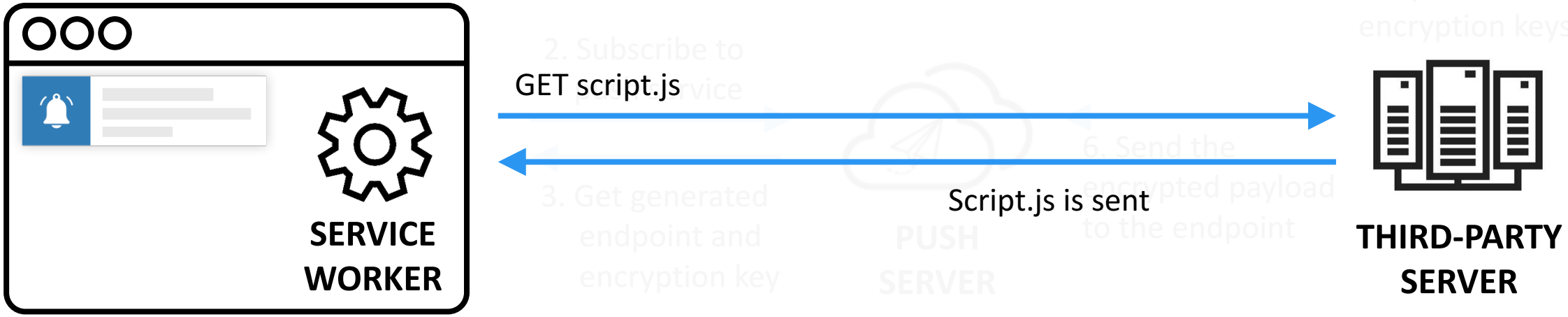
Web Push Protocol: **VAPID**



Web Push Protocol: **VAPID**



Domain Spoofing Attack



Domain Spoofing Attack