ACSAC 2022

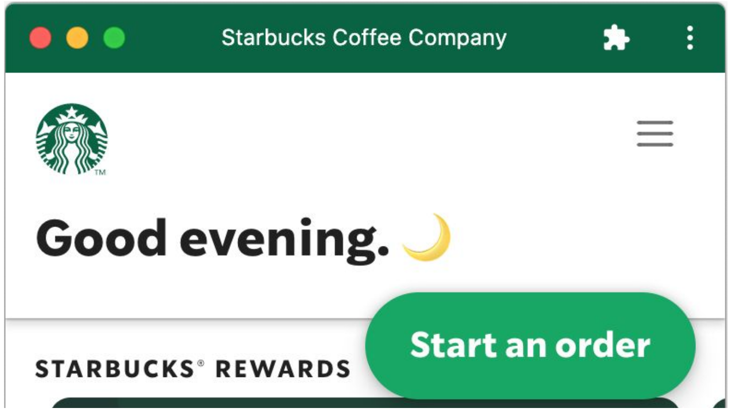
DeView: Confining Progressive Web Applications by Debloating Web APIs

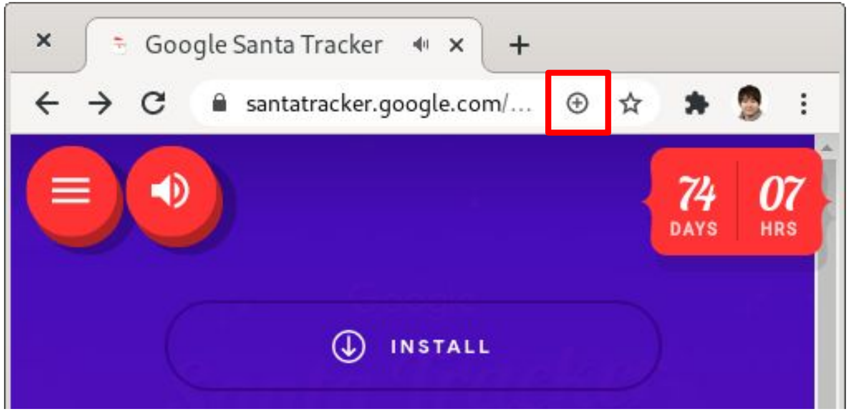
ChangSeok Oh, Sangho Lee[†], Chenxiong Qian[‡], Hyungjoon Koo[*], and Wenke Lee

Georgia Institute of Technology, [†]Microsoft Research,
[‡]University of Hong Kong, [*]Sungkyunkwan University

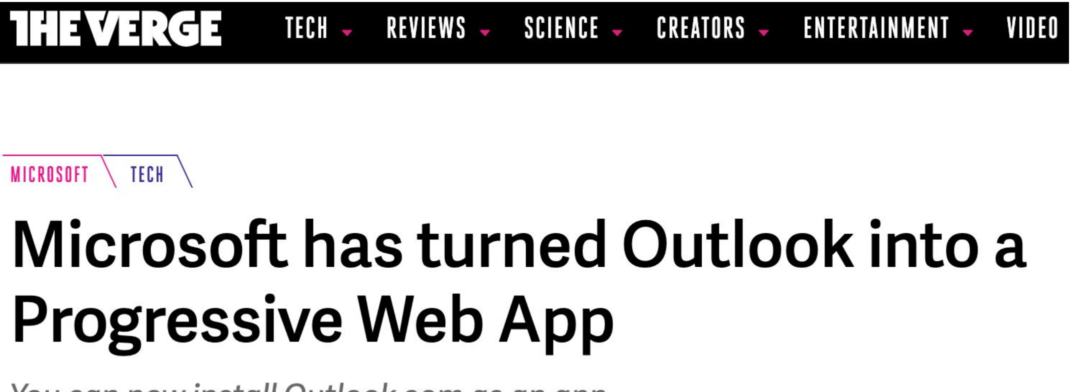
Progressive Web Application

- A stand-alone web app running outside the web browser
- Similar to the native app's look-and-feel
- Installable via the ⊕ button in the address bar or one in the content





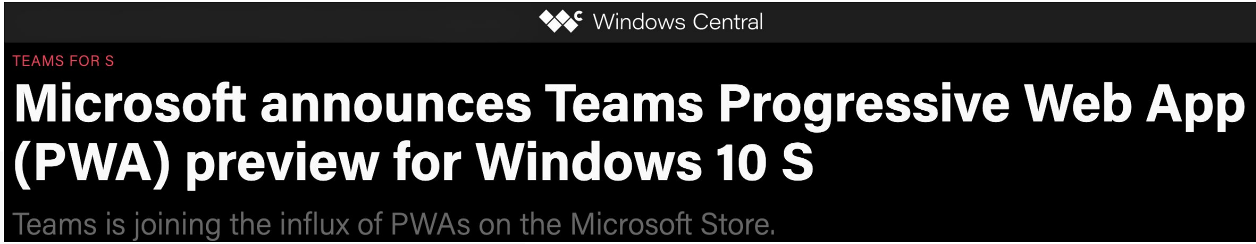
Progressive Web Application



Microsoft has turned Outlook into a Progressive Web App

You can now install Outlook.com as an app

By Tom Warren | @tomwarren | Nov 26, 2019, 12:05pm EST

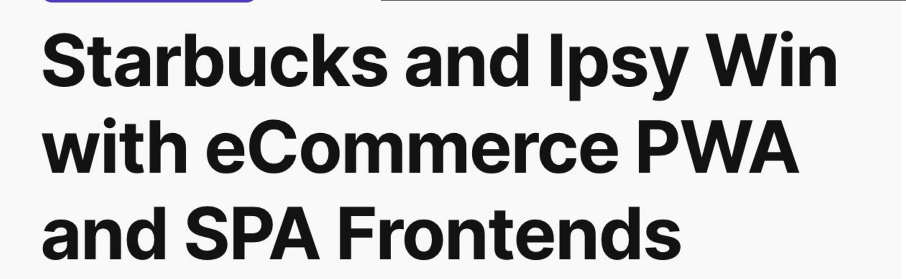


Microsoft announces Teams Progressive Web App (PWA) preview for Windows 10 S

Teams is joining the influx of PWAs on the Microsoft Store.

MOOVWEB

Progressive Web Apps



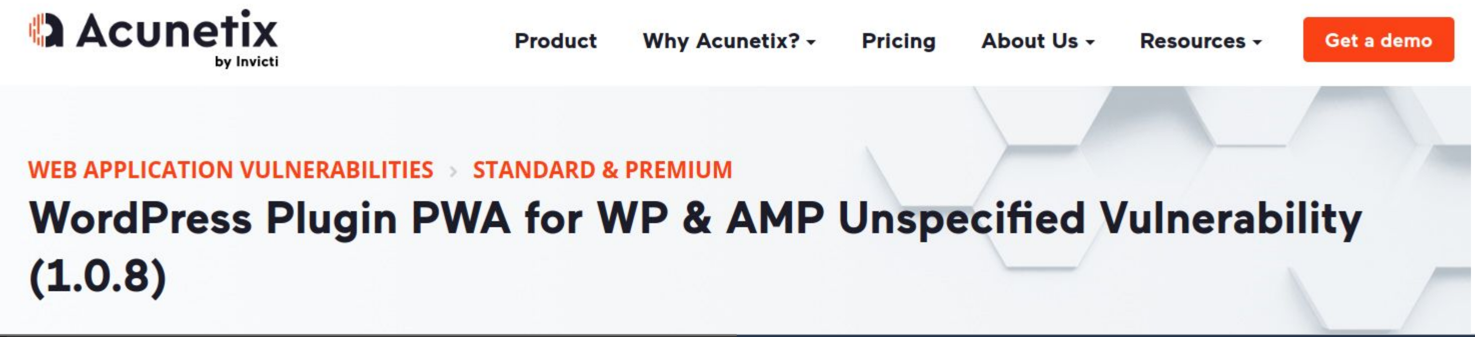
Starbucks and Ipsy Win with eCommerce PWA and SPA Frontends

The Large and Identical Attack Surface of PWAs

Rich Web APIs inevitably result in a large attack surface of PWAs.

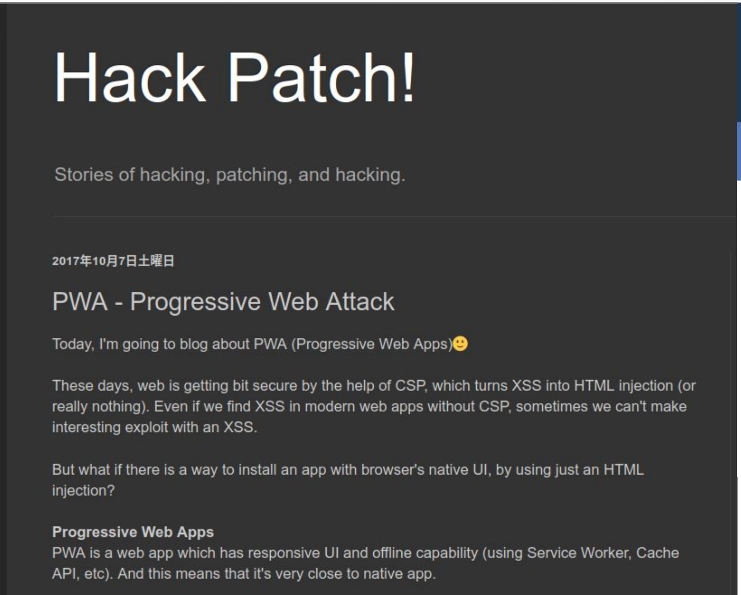
- Having more attack vectors than native applications
- Sharing the same vulnerability incurred by unwanted Web API across PWAs
- Suffering from traditional web attacks (XSS/UXSS, spoofing) and supply chain attacks

The Large and Identical Attack Surface of PWAs



WEB APPLICATION VULNERABILITIES › STANDARD & PREMIUM

WordPress Plugin PWA for WP & AMP Unspecified Vulnerability (1.0.8)



Hack Patch!

Stories of hacking, patching, and hacking.

2017年10月7日土曜日

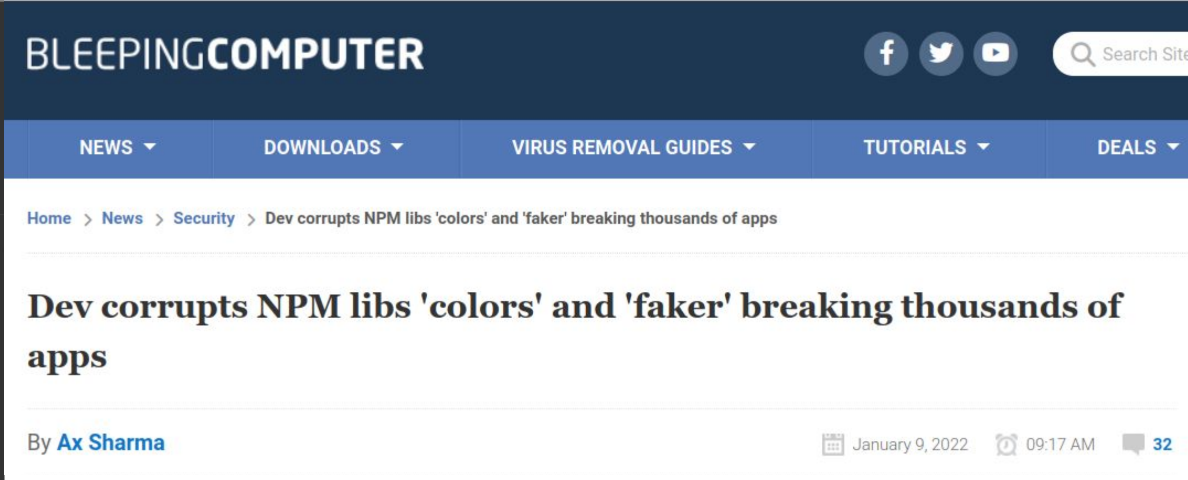
PWA - Progressive Web Attack

Today, I'm going to blog about PWA (Progressive Web Apps)🙂

These days, web is getting bit secure by the help of CSP, which turns XSS into HTML injection (or really nothing). Even if we find XSS in modern web apps without CSP, sometimes we can't make interesting exploit with an XSS.

But what if there is a way to install an app with browser's native UI, by using just an HTML injection?

Progressive Web Apps
PWA is a web app which has responsive UI and offline capability (using Service Worker, Cache API, etc). And this means that it's very close to native app.



Home › News › Security › Dev corrupts NPM libs 'colors' and 'faker' breaking thousands of apps

Dev corrupts NPM libs 'colors' and 'faker' breaking thousands of apps

By Ax Sharma   January 9, 2022   09:17 AM   32

Preliminary Research on Web API Usage of PWAs

PWAs often use unpopular Web APIs. Thus, cost-benefit-based approaches do not work for debloating web APIs.

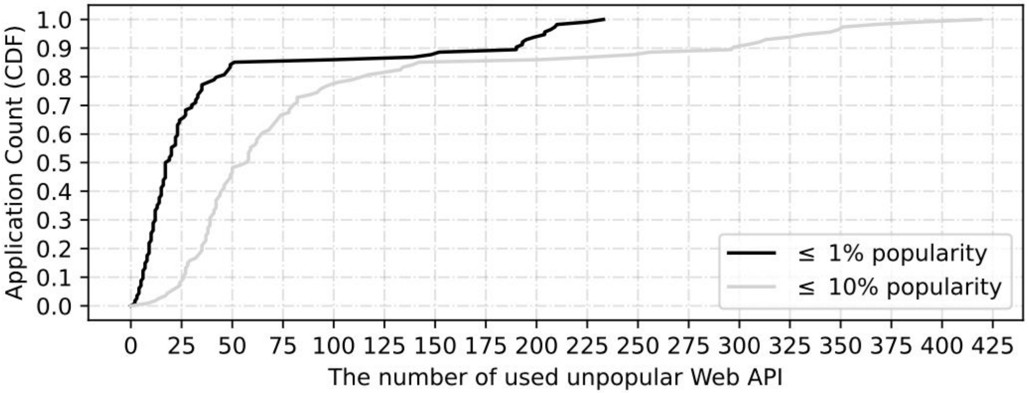


Figure 2: Cumulative distribution of PWAs according to the popularity of required Web APIs. PWAs frequently use unpopular Web APIs.

Preliminary Research on Web API Usage of PWAs

A different PWA shows different Web API usage. PWA pairs do not have many Web APIs in common.

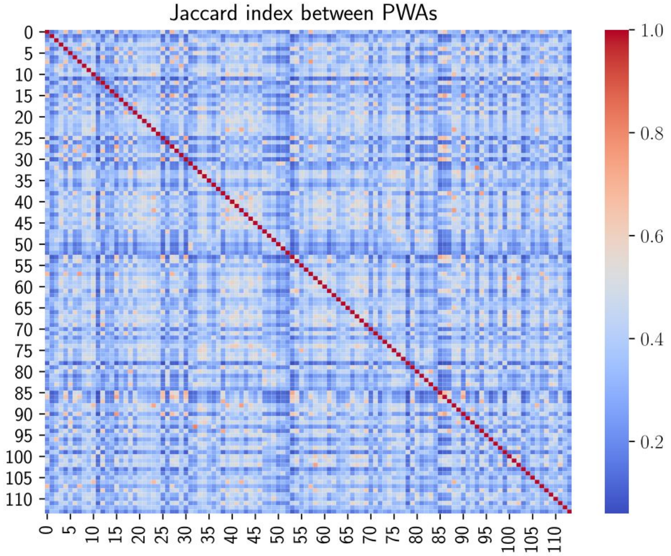


Figure 4: Jaccard indexes of PWA pairs. The blue color dominates the heatmap, meaning most PWA pairs in our dataset do not have similar Web APIs in common.

Preliminary Research on Web API Usage of PWAs

Each PWA uses a small portion of common Web APIs. Thus, a single debloated browser engine that covers all PWAs is still bloated in the view of each PWA.

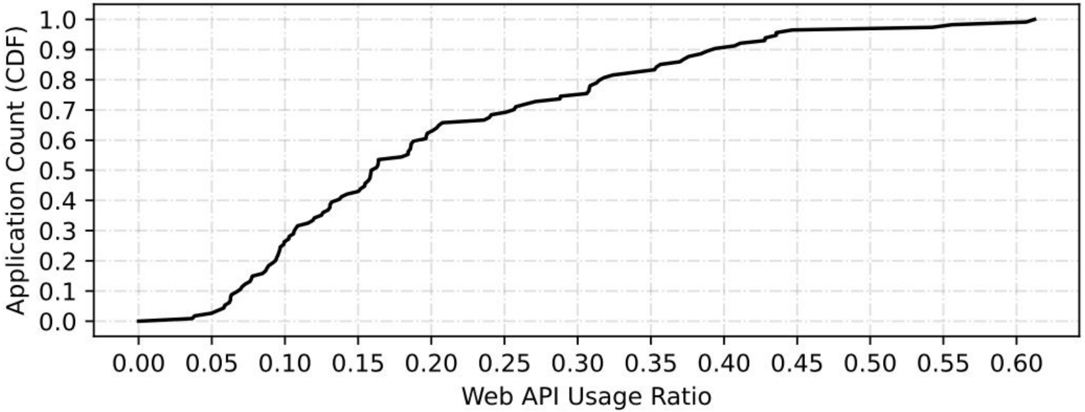
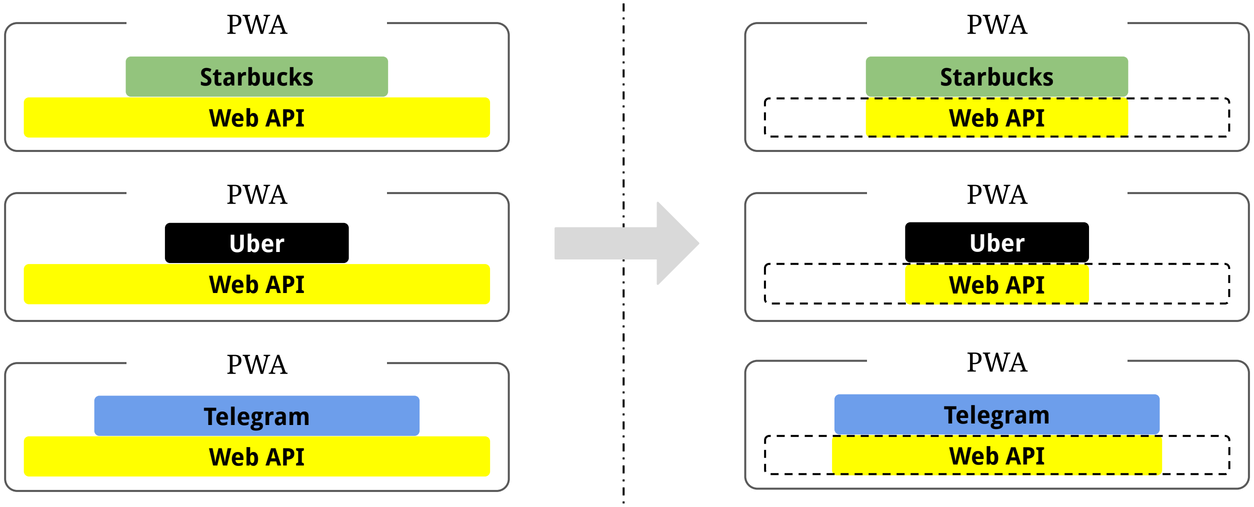


Figure 5: Cumulative distribution of PWA Web API usage ratios over total Web APIs used by at least one PWA.

Research Goal

Can we reduce an attack surface and customize it for each PWA?



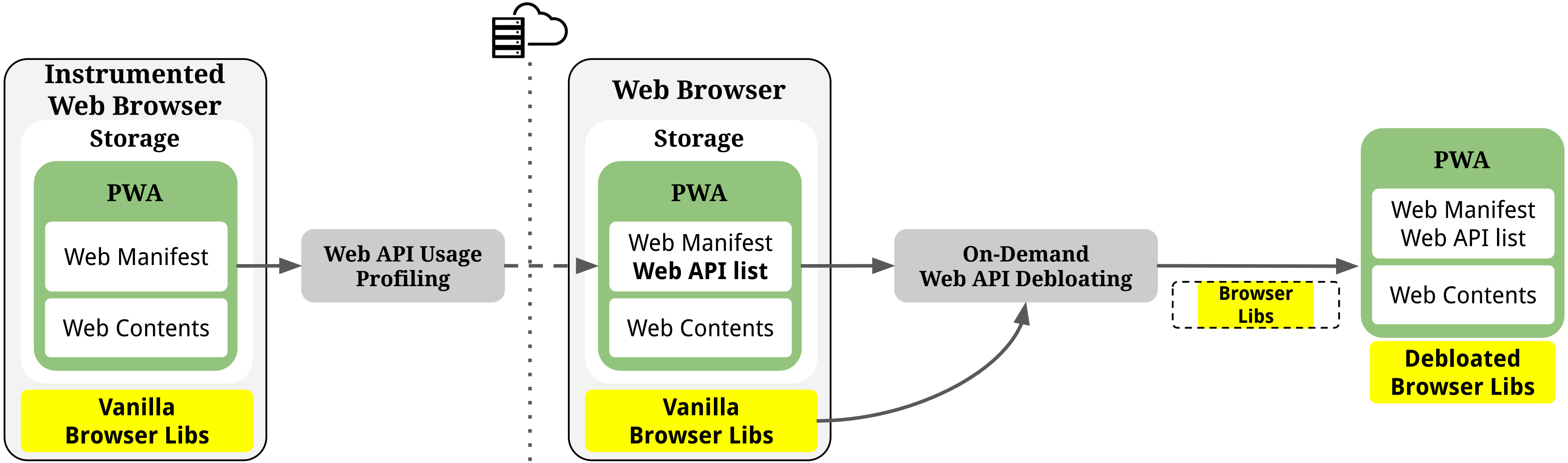
DeView

A Web API debloating system for PWAs. DeView confines the web API each PWA can access by removing unwanted ones from browser engine libraries.

To this end, DeView introduces two techniques:

- Record-and-replay-based Web API Profiling on the server-side
- Compiler-assisted on-demand browser binary debloating on the client-side

DeView: System Overview



Evaluation

RQ1. Removable Web APIs: How many Web APIs can DeView remove in a debloating browser engine?

RQ2. Security Benefit: How effectively does DeView prevent possible attacks?

RQ3. Code Coverage: How much code coverage can DeView achieve in finding exercised web APIs?

RQ4. Costs: What are the performance overheads of DeView?

Evaluation: Removable Web APIs

For 114 real-world PWAs, DeView removes 91.8% of 8,249 Web APIs from the Chromium browser engine on average.

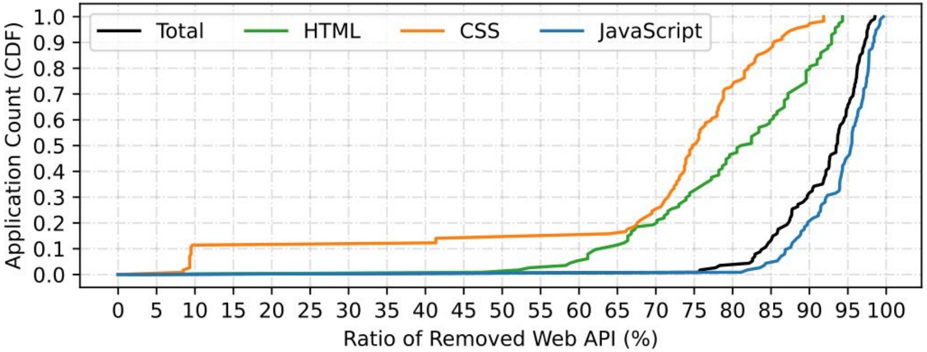


Figure 9: CDF of removable web APIs ratio in our dataset. On average, 91.8% of web APIs are removable.

Evaluation: Security Benefit

DeView prevents 76.3% out of 478 CVEs on average. It is the most effective in defeating XSS and bypass attacks.

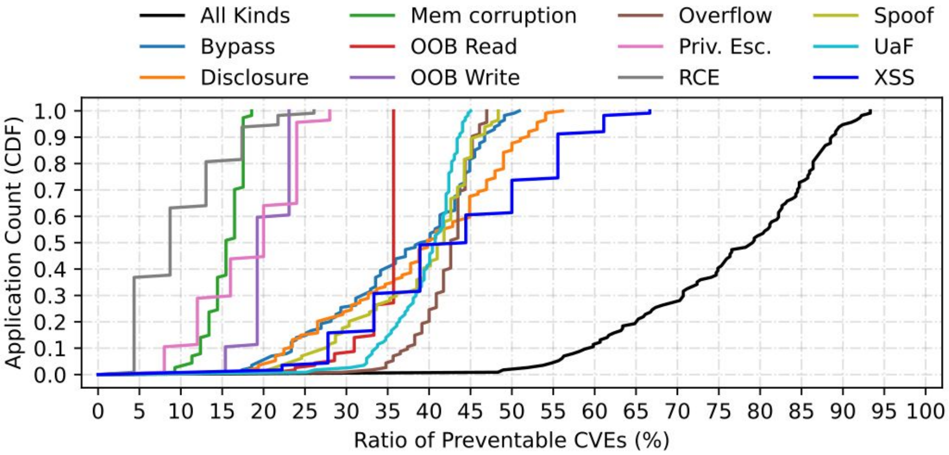
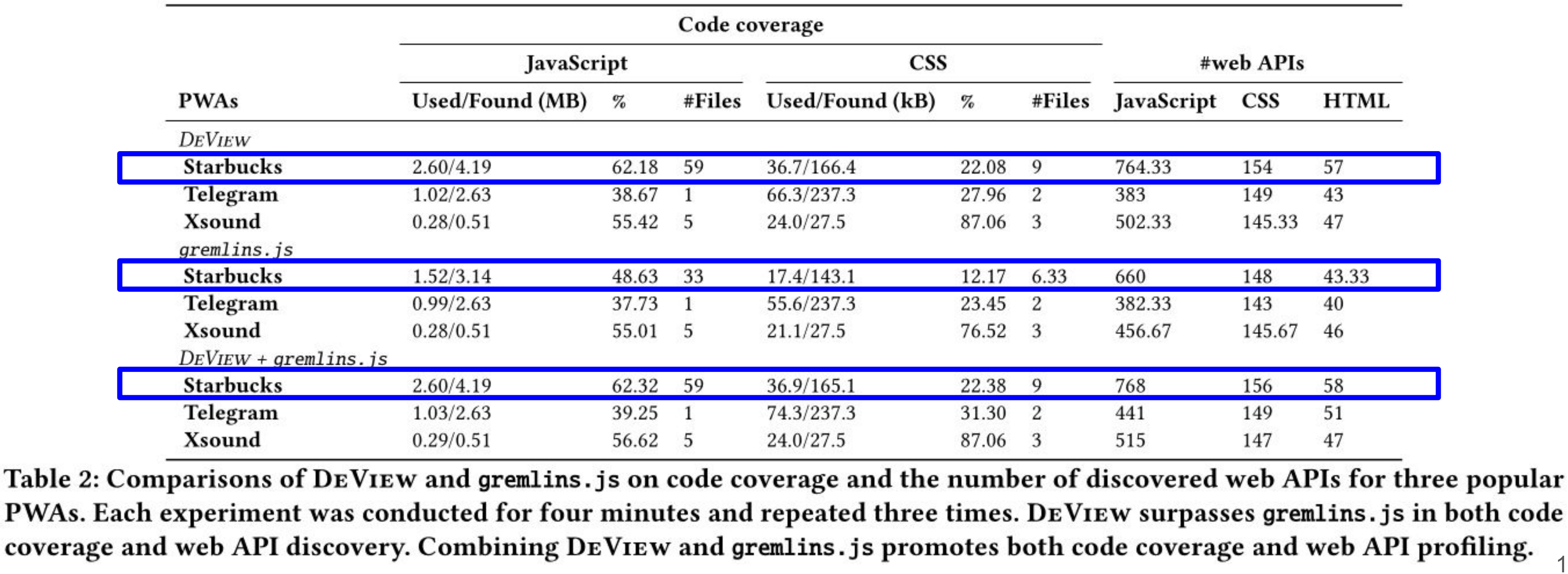


Figure 10: CDF of the ratio for CVEs preventable by DeView. It can prevent 76.3% of 478 CVEs on average (Table 5).

Evaluation: Code Coverage

DeView outperforms a monkey test regarding code coverage and Web API finding.
Combining the two approaches can improve both results.

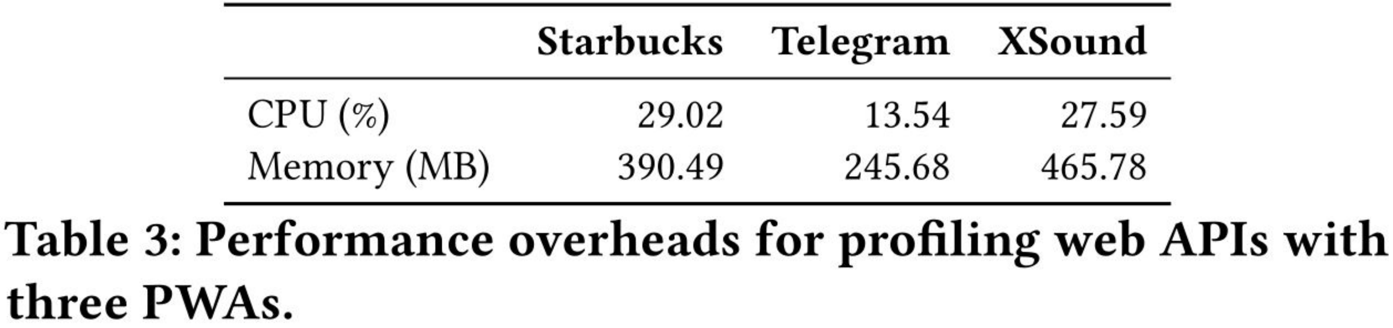


	Code coverage								
	JavaScript			CSS			#web APIs		
PWAs	Used/Found (MB)	%	#Files	Used/Found (kB)	%	#Files	JavaScript	CSS	HTML
DeView									
Starbucks	2.60/4.19	62.18	59	36.7/166.4	22.08	9	764.33	154	57
Telegram	1.02/2.63	38.67	1	66.3/237.3	27.96	2	383	149	43
Xsound	0.28/0.51	55.42	5	24.0/27.5	87.06	3	502.33	145.33	47
gremlins.js									
Starbucks	1.52/3.14	48.63	33	17.4/143.1	12.17	6.33	660	148	43.33
Telegram	0.99/2.63	37.73	1	55.6/237.3	23.45	2	382.33	143	40
Xsound	0.28/0.51	55.01	5	21.1/27.5	76.52	3	456.67	145.67	46
DeView + gremlins.js									
Starbucks	2.60/4.19	62.32	59	36.9/165.1	22.38	9	768	156	58
Telegram	1.03/2.63	39.25	1	74.3/237.3	31.30	2	441	149	51
Xsound	0.29/0.51	56.62	5	24.0/27.5	87.06	3	515	147	47

Table 2: Comparisons of DeView and gremlins.js on code coverage and the number of discovered web APIs for three popular PWAs. Each experiment was conducted for four minutes and repeated three times. DeView surpasses gremlins.js in both code coverage and web API discovery. Combining DeView and gremlins.js promotes both code coverage and web API profiling.

Evaluation: Costs for Debloating Web APIs

- The CPU and memory overheads of Deview's profiling arise running from a web page.
- Debloating slightly slows down launching a PWA (0.24s).
- 68 MB of disk space per PWA is needed to save the debloated binaries.



	Starbucks	Telegram	XSound
CPU (%)	29.02	13.54	27.59
Memory (MB)	390.49	245.68	465.78

Table 3: Performance overheads for profiling web APIs with three PWAs.

Takeaways

- Each PWA uses Web API differently, so a PWA doesn’t need all Web APIs.
- DeView eliminates 91% of the whole Web APIs per application on average.
- DeView prevents 76% of 478 CVEs related to Web API exploits on average.
- DeView significantly reduces the attack surface of a PWA with negligible costs.
- DeView is open-sourced.

https://github.com/shivamidow/deview

Thank You!