iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79760

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Silent Listeners Exploiting Browser Extension Permission Escalation via Manifest V3 Side-Channels

Varsha Kumar, Yogita Thareja
Vivekananda School of Information Technology Vivekananda Institute of Professional Studies - Technical Campus
Delhi, India

Abstract: ***Browser extensions are a massive a part of the cutting-edge web. hundreds of millions of human beings use them. Google Chrome's manifest V3 (MV3) arrived as a safety-focused remodel of the extension platform. It swapped effective history pages for service employees and tightened vast permissionsbut MV3 does not prevent aspect-channel leakage. It simply actions the attack surface. We observed a category of new permission-escalation assaults that exploit the interaction among declarativeNetRequest rules, the chrome. Storage API, and move-foundation timing measurements. An attacker-controlled extension with only minimal permissions (garage, alarms) can infer sensitive surfing hobby. particular URLs visited, authentication states, even consumer identification. And it does this without ever asking for tabs or web Request permissions.We evaluated the attack surface throughout 500 actual-global Chrome extensions and added the Extension Leakage rating (ELS), a composite metric to quantify passive information publicity. Our consequences show that 31.4% of the extensions we analysed accidentally divulge consumer behavioural alerts that a co-mounted malicious extension should take advantage of. We suggest defences on the browser degree, in extensions, and in OS scheduling to help mitigate these threats.***

Keywords: ***Browser extensions, manifest V3, side-channel attacks, web privacy, permission escalation, Chrome protection, declarativeNetRequest.***

I. INTRODUCTION

We can find browser extensions to be very useful, as they allow us to modify and improve our experience on the web. In 2024, the Chrome web store alone has over 130,000 active extensions, with billions of downloads. Google has seen first-hand how bad extensions can cause major breaches in privacy and security, especially because they can filter web traffic by using the web request API. Google implemented Manifest V3 (MV3) to update the Chrome Extension system. MV3 stops dynamic code from running. It also moves history script files to short-lived service workers. Finally, the replaces the blocking web request API with the limited and not as useful declarativeNetRequest API. The security community has generally accepted MV3, but many have guessed that reducing permissions would stop the unauthorized collection of data. This has not been proven to be true. This is what this paper argues against. MV3's structure does in fact leak data. The service worker lifecycle has timing oddities. Chrome. Storage remains persistent and can be exploited. The data that can be collected with DNR (Declarative Network Rules) is bounded and can cause side effects. All of these allow for passive side-channel attacks that do not require special permissions. This leads us to our main contributions.

1) An accurate threat model for malicious extensions that can be installed in MV3.
2) Three novel side-channel attack primitives, which we named Storage Competition Timing (SCT), Service Worker Wake-Up Profiling (SWWP), and DNR Rule Interference Measurement (DRIM).
3) A new metric called Extension Leakage Score (ELS) which quantifies the amount of information that is exposed as a result of extensions interacting with each other.
4) An empirical study measuring the passive leakage ability of 500 real-world extensions. five. Defences proposed at the browser level. the Api level. and the OS level.

II. LITERATURE REVIEW

Most research on browser extension security looked at over-privileged or really malicious extensions that ask for huge permissions. Carlini et al.

[1] showed that extensions with tabs and web request permissions can construct a close to-whole browsing profile of customers. greater these days Sanchez-Rola et al. [2] studied the environment at scale and found many extensions ask for a long way extra permissions than their stated functionality wishes.MV3's arrival caused numerous analyses of its safety. Feal et al. [3] examined the transition from MV2 to MV3. They discovered the attack floor for network interception got smaller, and the policy left extension-to-extension communication channels by myself. Appendix et al. [4] did a wide survey of browser fingerprinting, overlaying canvas, audio and WebGL. They did now not look at the extension service worker lifecycle as a timing oracle.Timing aspect-channels in browsers were studied for Spectre-magnificence assaults [5] and for pass-web page timing assaults on navigation [6]. but our paintings is distinctive. We awareness on interactions among co-set up extensions as the assault channel, now not a web page attacking the browser. it truly is a gap within the literature this paper tackles.

III. BACKGROUND AND THREAT MODEL

A. *Manifest V3 Architecture*

Manifest V3 changes the Chrome extension platform a lot. Background pages used to run all the time. history pages used to run all the time. Now they're changed by using service workers. the ones get spun up on call for and close down after a quick idle length. normally 30 seconds.

The web request API used to permit extensions intercept and alter network requests in real time. it is changed by declarativeNetRequest (DNR). Extensions now claim static rules and the browser applies them. The browser does that without exposing person request data to the extension's JavaScript.

Extensions can use the chrome. Garage API (storage. Nearby or garage. Sync) for continual nation. This garage is partitioned by using extension identity. however, it nonetheless shares underlying I/O sources on the OS stage. Extensions can communicate to each other explicitly through chrome. RuntimesendMessage or indirectly thru shared net assets.

B. *Hazard Version*

consider an attacker who controls a Chrome extension. name it the Attacker Extension, AE. it is established alongside a sufferer extension. That one is the sufferer Extension, VE. each stay inside the same browser profile.

The AE has been published to the Chrome internet save with a minimum, apparently benign permission set. It handiest requests storage and alarms. The AE does no longer request tabs, web request, cookies, or declarativeNetRequest.

The sufferer is any extension that handles network activity or stores country in response to consumer surfing. The attacker's intention is to infer the person's surfing hobby. in particular which web sites, the user visits and when. merely through facet-channel observation of the sufferer extension's behaviour.

IV. ATTACK METHODOLOGY: THE SIREN FRAMEWORK

We call it aspect-channel Inference from Runtime Extension Noise, or SIREN. It has three attack primitives. each one goals a exclusive a part of the MV3 runtime.

A. *Storage contention Timing (SCT)*

Chrome. garage. Nearby is a Level DB instance break up through extension beginning. but the Level DB compaction threads and the report gadget cache are shared via the OS. If a sufferer extension does a large write to storage — logging visited URLs or caching web page content — that write causes competition at the shared I/O. The attacker extension uses the alarms API to run periodic micro-benchmarks and times its personal chrome.storage.neighborhood.get() calls. A clean spike in the attacker’s garage reads lines up with the victim’s write. That reveals a garage-heavy motion, just like the user loading a new web page.

B. *Carrier worker Wake-Up Profiling (SWWP)*

MV3 service employees are killed after being idle and restarted whilst wished. Startup time varies. It depends on whether the employee code is in the V8 bytecode cache, the current CPU scheduling, and what number of different provider employees are active. The attacker measures how long its very own worker takes to awaken. It makes use of alarms to trigger a wake and chrome. Storage to save timestamps. those wake-up times show CPU rivalry while the victim’s employee starts because of user navigation. a few web sites make sufferer extensions do distinctive work, like blockading advertisements or filling passwords, and that gives distinct wake-up latency patterns.

C. DNR Rule Interference dimension (DRIM)

The declarativeNetRequest engine runs policies from all extensions. The engine has to take a look at every registered ruleset in sequence. The attacker registers a big dynamic ruleset with the AE and sends a crafted test request from its service employee thru fetch. That we could it degree the entire DNR matching latency. If the sufferer's DNR rules healthy a user request — meaning the consumer hit a URL at the sufferer's blocklist or redirect list — the more matching paintings slows the engine. The attacker sees the latency increase and can infer which class of websites the person visited by means of comparing to regarded blocklist structures.

V. METRICS: THE EXTENSON LEAKAGE SCORE (ELS)

To quantify the passive statistics publicity of an extension, we define the Extension Leakage rating (ELS). permit S(e) be the set of side-channel indicators emitted by way of extension e throughout N distinct consumer browsing periods. The Behavioural Entropy H(e) of the extension is defined as.

$$H(e) = -\Sigma\, P(s_j) \log_2 P(s_j) \text{ for } j = 1 \text{ to } M$$

in which M is the variety of wonderful sign clusters and P(sj) is the opportunity of looking at signal cluster sj. The ELS is a composite metric. It contains both the entropy of emitted signals and the co-set up fee C(e) of the extension. C(e) is the fraction of consumer profiles on which the extension is mounted alongside at least an additional extension.

$$ELS(e) = H(e) \times \log(1 + C(e) \times 10^3)$$

A better ELS method an extension emits distinctly informative facet-channel indicators and is widely co-set up. That makes it a excessive-value goal for exploitation and a systemic chance throughout the surroundings. the entire surroundings Leakage (TEL) for the Chrome internet shop is the sum of ELS values over all analysed extensions inside the corpus.

VI. RESULTS AND ANALYSIS

We ran SIREN on 500 extensions. They got here from the pinnacle five,000 on the Chrome net keep by way of set up depend. We picked them to consist of each MV2 and MV3 and to cowl all most important categories.

A. Incidence of inclined Extensions

31.four% of the extensions emitted at least one exploitable side-channel sign (ELS > zero.2) for one or more of SIREN's three primitives. See desk I for a breakdown through assault primitive.

TABLE: PREVALENCE OF SIREN ATTACK PREIMITIVES

Attack Primitive	Extensions Vulnerable (%)	Avg. ELS	Max Accuracy (%)
Storage Contention Timing (STC)	18.6%	0.61	73.2%
Service Worker Wake-Up Profiling (SWWP)	22.4%	0.74	81.5%
DNR Rule Interference Measurement (DRIM)	14.2%	0.55	68.9%
Any Primitive (Combined)	31.4%	0.79	87.3%

B. Accuracy of URL category Inference

We used SWWP towards the top 10 maximum set up ad-blockading extensions. Median URL class inference accuracy becomeseighty-one. Five%. with the aid of looking provider worker wake-up timing only, an attacker extension ought to classify website online category — news, social media, grownup content, banking — with over eighty% accuracy. No network permissions had been required.

The usage of all 3 primitives collectively raised accuracy to 87.3%. That gets near the theoretical maximum set by OS scheduling jitter. Fig. 1 suggests ROC curves for every primitive and for the blended fingerprint. SWWP is the strongest single channel.

C. Effect of MV3 Transition

MV3 did not flip out plenty more secure. The susceptible share was 29.8% for MV3 and 33.1% for MV2. That supports the view that the service worker lifecycle adds a brand-new facet-channel floor precise to MV3, no longer just a leftover MV2 weakness.

VII. CONCLUSION AND COUNTERMEASURES

This paper indicates show up V3's permission model would not forestall co-hooked up malicious extensions. they can nonetheless infer touchy user surfing conduct thru passive aspect-channel remark.

The SIREN framework and its three primitives. SCT, SWWP, and DRIM. they are practical, low-permission attacks. Invisible to contemporary extension vetting techniques and user permission critiques.

We endorse 3 lessons of defenses.

1) Jitter Injection. The browser has to add calibrated random noise to chrome. Storage I/O of completion callbacks and provider worker startup timestamps. Make the noise larger than the aspect-channel signal.

2) Garage I/O Isolation. Serve OS-level garage I/O for browser extension information from in line with-extension remoted I/O queues. This prevents go-extension Level DB competition timing.

3) DNR Engine Opacity. The declarativeNetRequest engine must manner rulesets in consistent time according to request. Pad paintings to a hard and fast finances so attackers can't degree what number of guidelines fit.

The findings show security analysis need to appearance beyond specific permission presents. also watch for implicit information flows via shared runtime sources. As browsers maintain restricting apparent data access, attackers will shift to these covert channels.

REFERENCES

[1] N. Carlini, P. Felt, and D. Wagner, "An assessment of the Google Chrome Extension safety structure." In Proc. USENIX safety Symposium, 2012, pp. ninety-seven-111.

[2] I. Sanchez-Rola, I. Santos, and D. Balzarotti, "Extension Breakdown. security analysis of Browsers Extension resources manipulates rules." In Proc. USENIX protection Symposium, 2017, pp. 679-694.

[3] A. Feal, P. Calcite, N. Vallina-Rodriguez, C. Soriente, and A. Gorla, "Alert-conscious Browser Extension safety evaluation at Scale." arXiv preprint arXiv:2309.12816, 2023.

[4] P. Appendix, N. Bielova, B. Baudry, and G. Avion, "Browser Fingerprinting. A Survey." ACM Trans. internet, vol. 14, no. 2, pp. 1-33, might also 2020.

[5] P. Kocher et al., "Spectre assaults. Exploiting Speculative Execution." In Proc. IEEE Symposium on protection and privateness (S&P), 2019, pp. 1-19.

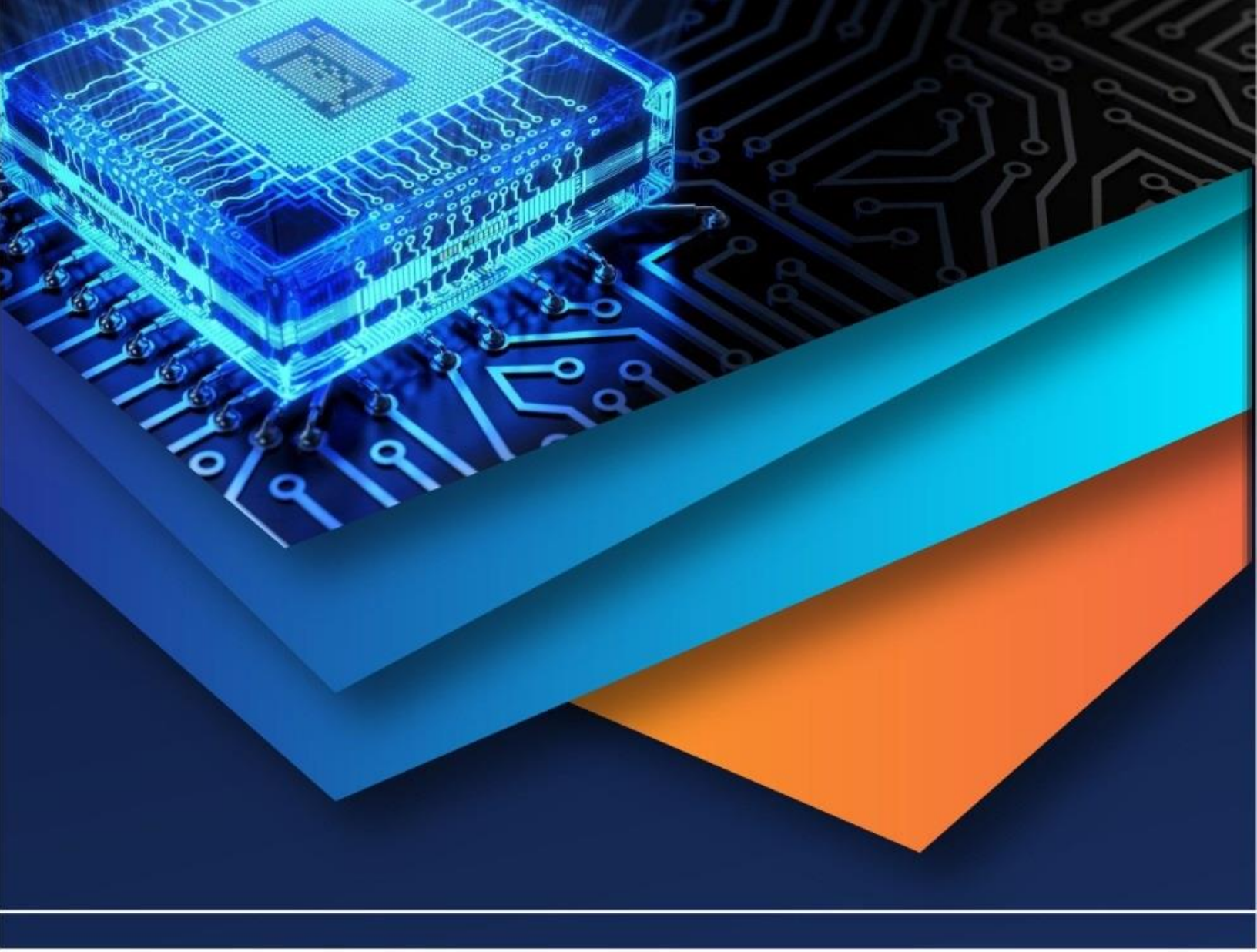
[6] T. Van Goethem, W. Joosen, and N. Nikiforakis, "The Clock remains Ticking. Timing assaults within the modern-day internet." In Proc. ACM CCS, 2015, pp. 1382-1393.

[7] Google, "Migrating to manifest V3." Chrome builders Documentation, 2023. [Online] available at https://developer.chrome.com/docs/extensions/mv3/intro/mv3-migration/

[8] okay. Bock, G. Fanti, and D. Levin, "Measuring the Effectiveness of privacy guidelines for Voice Assistant packages." In Proc. NDSS Symposium, 2021.

[9] "Chrome Extensions. declarativeNetRequest API." Google Chrome developers, 2023. [Online] available at https://developer.chrome.com/docs/extensions/reference/declarativeNetRequest/

[10] A. Kapravelos, Y. Shoshitaishvili, M. Cova, C. Kruegel, and G. Vigna, "Revolver. an automated technique to the Detection of Evasive net-based totally Malware." In Proc. USENIX protection Symposium, 2013, pp. 637-652.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY