Journal of KIIT. Vol. 24, No. 6, pp. 273-282, Jun. 30, 2026. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2026.24.6.273

CSS-based Exfiltration for Cross-Origin Login State Fingerprinting Without Javascript

Roby Firnando Yusuf*, Abhishek Chaudhary**, and Sunoh Choi***

Abstract

Browser fingerprinting remains a powerful mechanism for identifying users despite increasing restrictions on traditional tracking techniques. While prior work has largely focused on JavaScript-based approaches, recent studies suggest that Cascading Style Sheets (CSS) can also introduce unintended side channels, yet their use for cross-origin login-state inference remains insufficiently explored. In this paper, we investigate the feasibility of CSS-based exfiltration for cross-origin login-state fingerprinting without relying on JavaScript. Experimental results show that, under default browser settings where credentials are included in cross-origin requests, the proposed attack successfully infers login states across all evaluated services and enables the construction of multi-dimensional fingerprints with moderate uniqueness across users.

요 약

브라우저 지문 인식은 기존 추적 기술에 대한 규제가 강화되고 있음에도 불구하고 사용자를 식별하는 강력한 메커니즘으로 남아 있습니다. 기존 연구는 주로 자바스크립트 기반 접근 방식에 초점을 맞추었지만, 최근 연구에 따르면 CSS(Cascading Style Sheets) 또한 의도치 않은 사이드 채널을 유발할 수 있음이 밝혀졌으며, 이를 활용한 교차 출처 로그인 상태 추론은 아직 충분히 연구되지 않았습니다. 본 논문에서는 자바스크립트에 의존하지 않고 CSS 기반 데이터 유출을 통해 교차 출처 로그인 상태 지문 인식을 수행하는 가능성을 조사합니다. 실험 결과는 교차 출처 요청에 자격 증명이 포함된 기본 브라우저 설정에서 제안된 공격이 평가된 모든 서비스에서 로그인 상태를 성공적으로 추론하고, 사용자 간에 적당한 고유성을 갖는 다차원 지문을 구축할 수 있음을 보여줍니다.

Keywords

browser fingerprinting, css-based exfiltration, login-state inference, cross-origin attacks, web privacy

* M.S. Student, Dept. of Software Engineering, Jeonbuk National University
- ORCID: https://orcid.org/0009-0002-9517-3053
** PhD. Student, Dept. of Software Engineering, Jeonbuk National University
- ORCI: https://orcid.org/0009-0004-0119-0579
*** Associate Professor, Dept. of Software Engineering, Jeonbuk National University
- ORCID: https://orcid.org/0000-0002-0654-7109

· Received: Mar. 31, 2026, Revised: Apr. 24, 2026, Accepted: Apr. 27, 2026
· Corresponding Author: Sunoh Choi
Dept. of Software Engineering, Jeonbuk National University, Korea
Tel.: +82-63-270-4784, Email: suno7@jbnu.ac.kr

Ⅰ. Introduction

In modern interconnected world, the Social media has profoundly influenced the formation of virtual identities throughout users everyday activities. Our digital identities are influenced not only by the information we intentionally provide, but also by the vast amount of digital traces produced through our interconnected actions.These digital footprints act as identifiers and can be utilized to monitor our online behavior, resulting in possible breaches of privacy and abuse. Cookies have traditionally been used to track users across the web [1][2]. However, increasing privacy concerns have led to restrictions such as third-party cookie blocking and stricter consent requirements [3][4], pushing the ecosystem toward alternative tracking techniques. Recent work shows that browser fingerprinting remains possible even in restrictive environments where JavaScript is disabled, leveraging modern CSS features such as container queries and conditional rules [5]. These mechanisms expose rendering differences that can form high-entropy fingerprints. In parallel, prior studies have demonstrated that login-state inference can be achieved through indirect resource-loading behaviors, such as redirect-based image probing [6][7]. These approaches exploit differences between authenticated and unauthenticated resource access without violating the Same-Origin Policy. Although CSS cannot directly access authentication data, it can detect rendering differences caused by resource loading outcomes [8]. By embedding URL-based requests within conditional CSS rules, these differences can be transformed into observable network events, enabling script-less login-state inference and forming a binary login-state fingerprint [9][10]. However, the impact of modern browser defenses, such as storage partitioning and credential isolation, remains insufficiently understood. Unlike prior approaches that rely on JavaScript execution or timing side channels, we demonstrate that CSS alone can serve as a cross-origin inference channel for login-state fingerprinting.

In this paper, we formalize login-state fingerprints as binary vectors representing authentication across multiple services and analyze their identifying capability. We implement a proof-of-concept attack using CSS-triggered resource requests to infer authentication states across multiple platforms.

Our evaluation shows that login states can be inferred under default browser configurations where credentials are included in cross-origin requests, producing distinguishable patterns under certain conditions. However, modern defenses such as third-party cookie blocking and storage partitioning significantly constrain this attack surface. These findings highlight that rendering-layer side channels remain a viable source of cross-origin information leakage even in the absence of JavaScript.

Ⅱ. Related Work

Browser fingerprinting has been widely studied as a mechanism for identifying users without explicit identifiers such as cookies [11][12]. Prior work shows that browser attributes, including rendering behavior, installed extensions, and system configurations, can be combined to produce high-entropy fingerprints that uniquely distinguish users.

Recent research demonstrates that Cascading Style Sheets (CSS) can also serve as a fingerprinting vector, even when JavaScript is restricted or disabled [5]. By exploiting conditional rules and layout-dependent behaviors, these approaches show that CSS can act as a side channel for information leakage.

In parallel, cross-origin login-state inference has been explored through indirect observation of resource-loading behavior, such as redirect-based image probing [12]. These techniques enable attackers to infer authentication states without accessing protected content, but they typically rely on JavaScript execution, timing side channels, or specialized probing mechanisms.

However, the use of CSS as a structured, script-less inference channel for cross-origin login-state fingerprinting remains underexplored, particularly under modern browser defenses such as storage partitioning and credential isolation. In this work, we address this gap by demonstrating that CSS alone can infer cross-origin authentication states and construct multi-dimensional login-state fingerprints without relying on JavaScript or timing measurements.

III. Threat Model and Attack Design

3.1 Threat model

We assume an attacker controls a webpage capable of embedding arbitrary CSS rules but cannot execute JavaScript or bypass the Same-Origin Policy. The victim may be authenticated to multiple platforms, and the attacker aims to infer these login states through CSS-based rendering behavior

3.2 Conditional rendering as an observation channel

The proposed attack leverages rendering-dependent resource loading in modern browsers. CSS properties that accept URL values, such as background-image, list-style-image, content, cursor, border-image, and font sources (e.g., @font-face), can trigger external network requests during style evaluation [13], [14]. When these properties are used within conditional contexts (e.g., selectors, @media, or @container), requests are issued only if specific rendering conditions are satisfied. As a result, observable network behavior becomes dependent on the internal rendering state of the browser. Authentication state may introduce structural or layout differences in rendered content, which influence CSS evaluation without exposing cross-origin data directly. These differences can propagate into conditional rules, allowing authentication states to be mapped into observable network requests.

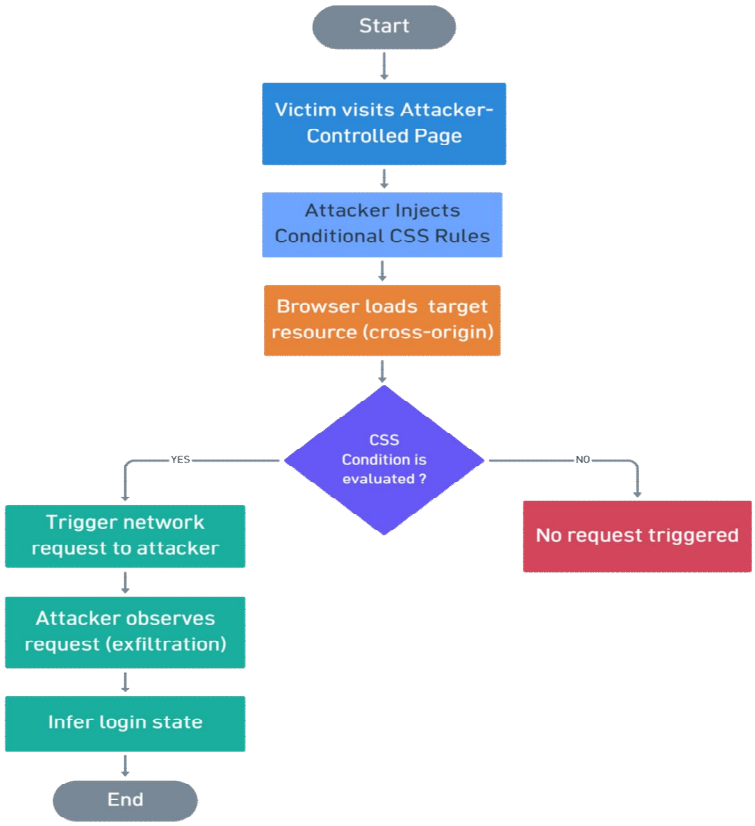

Fig. 1. CSS-based cross-origin login-state inference via conditional resource loading and observable network requests

To provide a illustration of this process, Fig. 1 presents the step by step flow of CSS-based conditional rendering and request-based inference.

The flow illustrated in Figure 1 highlights that the attack does not rely on direct access to cross-origin data, but instead exploits rendering-dependent side effects. Specifically, authentication state influences the rendering state, which determines whether conditional CSS rules are satisfied. This, in turn, controls whether network requests are triggered toward attacker-controlled endpoints.

As a result, internal browser states are indirectly exposed through observable request behavior, establishing a side channel that enables login-state inference without JavaScript or explicit data access.

IV. Experimental Evaluation

4.1 Experimental setup

We implemented a proof-of-concept attack consisting of an attacker-controlled webpage and a logging server. Twenty participants accessed the page while logged into different combinations of services, each represented as an 8-dimensional binary login-state vector.

Experiments were conducted under multiple browser configurations, including default settings, third-party cookie blocking, and private browsing modes. An inference was considered successful when a CSS-triggered request was consistently observed across repeated trials.

To ensure reliability, each inference was repeated multiple times, and only consistent outcomes were considered valid. The observed signals were then aggregated per participant to construct binary login-state fingerprint vectors, which are used for subsequent analysis.

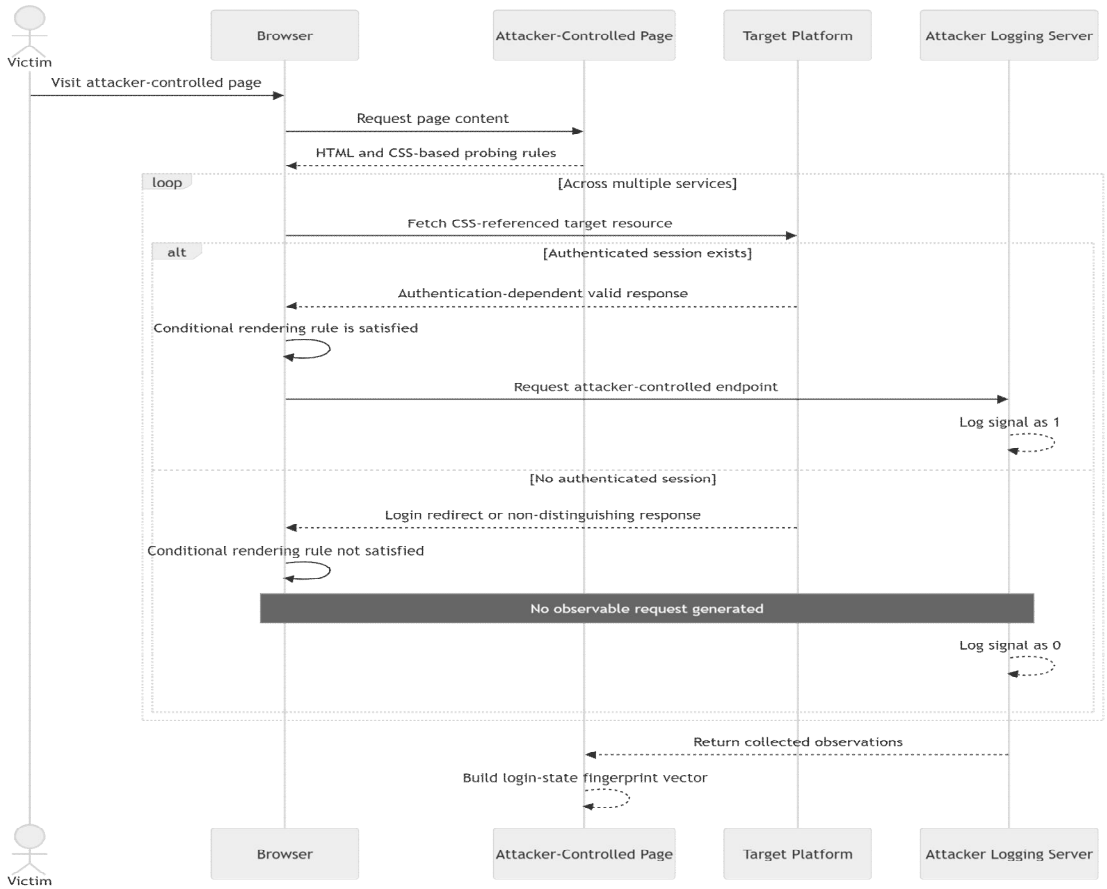

Fig. 2. Sequence diagram of the experimental procedure for CSS-based login-state inference

To clarify the experimental procedure, Fig. 2 presents the sequence of interactions among the participant, browser, attacker-controlled page, target service, and logging server. Unlike the attack overview shown earlier, this diagram focuses on the evaluation workflow used in Section IV. Specifically, it illustrates how authentication- dependent resource responses influence CSS evaluation and how the resulting observable requests are recorded as binary signals for fingerprint construction.

4.2 Target platforms and security variability

We evaluate the proposed attack across a diverse set of web platforms with different authentication and resource-handling mechanisms. Our analysis includes both vulnerable and non-vulnerable services in order to understand the conditions under which CSS-based login-state inference becomes feasible. First, we select eight widely used platforms, including Gmail (GM), YouTube (YT), Blogger (BL), Steam (ST), Medium (MD), Disqus (DS), LinkedIn (LI), and NotebookLM (NB). These services exhibit authentication-dependent resource-loading behavior, such as redirect-based responses that differ between authenticated and unauthenticated sessions. When authentication credentials are included in cross-origin requests, the server follows a redirect chain that resolves to a valid resource; otherwise, it redirects to a login page. These differences result in distinct rendering behaviors that can be captured by conditional CSS rules and observed through attacker-controlled endpoints, enabling the construction of a multi-dimensional login-state fingerprint. In addition, we evaluate several widely used platforms that are not vulnerable to the proposed attack, including GitHub, Bitbucket, Kakao, Naver, Tistory, AfreecaTV, Ruliweb, and Clien. Our results show that these platforms do not expose observable differences in resource-loading behavior based on authentication state.

The key distinction lies in how authentication-gated resources are handled. Unlike vulnerable services, these platforms restrict direct access to login-dependent endpoints or introduce additional validation mechanisms, such as intermediate confirmation steps or controlled redirection flows. As a result, both authenticated and unauthenticated requests produce equivalent observable behavior at the rendering layer.

This eliminates the necessary condition for the attack, where authentication state must influence observable rendering-dependent resource loading without user interaction. Without such differences, conditional CSS rules cannot generate distinguishable network requests, and login-state inference becomes infeasible. Overall, this comparison demonstrates that the feasibility of CSS-based login-state fingerprinting is highly dependent on platform-specific design choices, particularly the exposure of authentication-dependent resources to cross-origin requests.

4.3 Attack success rate

We evaluate whether authentication states can be inferred across different browser configurations. Under default browser settings, the attack successfully infers login states across all evaluated platforms, as authentication credentials are included in cross-origin requests. However, when third-party cookie blocking or private browsing mode is enabled, the attack becomes ineffective due to credential exclusion caused by storage isolation. Notably, disabling JavaScript does not affect the attack, as it relies solely on CSS-triggered resource requests.

These results indicate that the feasibility of CSS-based login-state inference is primarily determined by browser credential inclusion policies.

4.4 Comparative analysis

To clearly position our contribution, we compare the proposed CSS-based login-state inference approach

with prior work on browser fingerprinting and login detection techniques.

As shown in Table 1, prior fingerprinting approaches such as Eckersley et al [11] and Laperdrix et al [12]. primarily focus on collecting browser and device attributes to construct high-entropy identifiers. These methods rely on JavaScript execution and aim to measure uniqueness across users at scale. In contrast, CSS-based techniques such as CSS Spy Sheets demonstrate that rendering behavior can also be used for fingerprinting without JavaScript, but they still target device-level characteristics.

Table 1. Comparative analysis with prior fingerprinting and login detection techniques

Aspect	Eckersley et al.[11]	Gulyás et al.[7]	Trampert et al.[5]	This work
Goal	Uniqueness	FP+login	CSS FP	Login inference
JS	Yes	Yes	No	No
No-JS	No	No	Yes	Yes
Technique	Attributes	JS probing	CSS render	CSS Infer.
Target	Device	Device+login	Device	Login state
Login role	None	Auxiliary	None	Primary
Cross-origin	No	Yes	Limited	Indirect
Dependency	Config	JS+resource	Render	Credentials
Robustness	High	JS-only	Context	Conditional

Our approach differs in both objective and methodology. Instead of using login information as an auxiliary signal, as in Gulyás et al. [7], we treat login-state information as the primary target of inference. This shifts the focus from device-level fingerprinting to user-specific authentication states across multiple services, which are represented as structured binary vectors.

Furthermore, while prior login detection techniques depend on JavaScript-based probing mechanisms, our method demonstrates that login-state inference can be achieved purely through CSS-based resource loading behavior. This shows that disabling JavaScript alone is not sufficient to prevent cross-origin information leakage. Although the attack operates across origins without violating the Same-Origin Policy, its effectiveness depends on authentication dependent resource loading. Therefore, the attack is conditional on credential inclusion policies. When third-party cookies are blocked or storage partitioning is enforced, the attack becomes ineffective due to the absence of distinguishable signals.

Finally, unlike prior large-scale fingerprinting studies that aim to quantify uniqueness across large user populations, our objective is to demonstrate the feasibility of this attack model. Since the inference mechanism is deterministic and depends on browser behavior under specific configurations, the number of participants does not affect the feasibility of the attack, but it influences the observed uniqueness of fingerprint patterns

4.5 Fingerprint uniqueness analysis

When the attack succeeds, authentication states across multiple services can be represented as a binary login-state fingerprint.

Table 2. summarizes the observed login-state fingerprint vectors across participants

User	YT	BL	GM	ST	MD	DS	LI	NB
U1	1	1	1	0	0	0	1	1
U2	1	0	1	0	0	0	0	1
U3	1	1	1	0	1	1	0	1
U4	1	1	1	1	0	0	1	1
U5	0	0	0	0	0	0	0	0
U6	1	1	1	0	1	0	0	1
U7	0	0	0	0	0	0	0	0
U8	1	1	1	0	0	0	0	1
U9	0	0	0	0	0	0	0	0
U10	1	0	1	1	0	0	1	1
U11	1	1	1	0	0	0	0	1
U12	1	0	1	0	0	0	1	1
U13	1	0	1	0	0	0	0	1
U14	1	0	1	0	0	0	1	1
U15	1	0	1	0	0	0	0	1
U16	0	0	0	0	0	0	0	0
U17	1	0	1	0	0	0	0	1
U18	1	1	1	0	1	0	0	1
U19	1	0	1	0	0	0	0	1
U20	0	0	0	0	0	0	0	0
U20	0	0	0	0	0	0	0	0

As shown in Equation (2), the uniqueness rate U is defined as the ratio of the number of distinct fingerprint vectors F_{unique} to the total number of participants N.

$$(U = \frac{|F_{unique}|}{N}) \quad (1)$$

Using Equation (1), in our experiment with 20 participants, we observed 10 distinct fingerprint vectors, yielding a uniqueness rate of 0.5.

$$U = \frac{10}{20} = 0.5 \quad (2)$$

This result indicates that only half of the participants exhibit unique fingerprint patterns, while the remaining participants share identical login-state combinations. Compared to the initial smaller-scale experiment, the uniqueness rate decreases as the number of participants increases, revealing the presence of fingerprint collisions.

A primary factor contributing to these collisions is the dependency among several evaluated services. Gmail, YouTube, Blogger, and NotebookLM rely on a shared Google account (SSO), which introduces correlation among their authentication states. However, this correlation is not always consistently observable in practice, as different platforms expose authentication-dependent behaviors in different ways. As a result, some services may not produce detectable signals despite sharing the same underlying authentication session.

In particular, we observe that login-state inference can depend on service-specific interaction states. For example, in the case of Blogger, the detectable login-state signal is influenced by whether the user has previously initiated a login action on the platform. When a user is authenticated with a Google account but has never interacted with Blogger's login flow, the service may not expose authentication-dependent resource differences, resulting in a detectable signal of 0. However, once the user explicitly completes the Blogger login process, the service establishes an active session context, and subsequent observations consistently produce a detectable signal corresponding to a logged-in state.

This behavior indicates that login-state signals are not solely determined by SSO authentication, but also by service-level activation and prior interaction history. Therefore, the inferred fingerprint reflects observable authentication behavior rather than the ground-truth authentication state.

Consequently, although eight services are evaluated, the number of independent and consistently observable features is lower in practice. This limits the number of distinguishable fingerprint combinations and explains the repeated patterns observed across multiple participants.

In contrast, services such as LinkedIn, Medium, Disqus, and Steam rely on independent authentication mechanisms and contribute additional variability to the fingerprint space. However, their contribution alone is insufficient to eliminate collisions when combined with correlated or partially observable SSO-based services.

Although the theoretical fingerprint space contains up to (2^8 = 256) possible combinations, the effective fingerprint space is significantly smaller due to inter-service dependencies and differences in signal observability. This highlights that simply increasing the number of services does not necessarily improve distinguishing capability unless those services provide independent and consistently observable authentication signals.

Importantly, the purpose of this analysis is not to demonstrate high-entropy fingerprinting performance comparable to large-scale JavaScript-based approaches. Instead, our goal is to evaluate the feasibility of inferring cross-origin login states using CSS-only mechanisms.

Even with moderate uniqueness, the ability to infer authentication states across multiple services without JavaScript represents a meaningful privacy risk.

Furthermore, the results suggest that incorporating additional independent platforms could improve fingerprint diversity.

In addition to login-state signals, we observe that certain Chrome extensions expose internal resources via the web_accessible_resources field, allowing them to be accessed without scripting. By referencing extension-specific resource paths, an attacker can infer the presence of installed extensions based on rendering-dependent effects. Although this behavior was not systematically measured in our experiment, it indicates that extension-level signals can provide additional observable features. Such extension-based signals may further expand the effective fingerprint space when combined with login-state inference, suggesting that the identifying capability can increase when multiple independent sources of information are aggregated.

Overall, these findings demonstrate that CSS-based login-state inference is feasible under default browser configurations, while its distinguishing capability is constrained by the independence and observability of the underlying services.

4.6 Limitations and discussion under modern browser defenses

Our results show that the proposed CSS-based login-state inference remains effective even when JavaScript is disabled, indicating that rendering-layer mechanisms alone are sufficient for cross-origin inference. However, the attack becomes ineffective in privacy-enhanced browser configurations, such as Brave and Chrome incognito mode, where third-party cookie blocking and storage isolation prevent authentication credentials from being included in cross-origin requests.

The attack relies on authentication-dependent differences in server responses that produce distinguishable resource loading outcomes. When credentials are included, authenticated users receive valid resources, while unauthenticated users are redirected to login pages, resulting in observable rendering differences detectable via CSS. When credentials are excluded, both cases yield identical responses, eliminating the signal required for inference.

These findings indicate that the feasibility of CSS-based inference depends primarily on credential inclusion policies rather than the availability of JavaScript. While disabling JavaScript does not mitigate the attack, strict credential isolation effectively neutralizes it.

Despite this limitation, the approach remains relevant in practice, as credential inclusion may still occur due to legacy systems, misconfigured SameSite attributes, or incomplete deployment of isolation mechanisms. In such cases, CSS-based inference remains viable. Furthermore, combining CSS-based techniques with other side channels (e.g., timing or cache-based signals) or considering same-site contexts may restore partial inference capability.

From a defensive perspective, our results confirm that strict credential isolation is effective in mitigating this attack class. However, its effectiveness depends on consistent enforcement across all resource types; any inconsistency may reintroduce exploitable differences. Overall, the results highlight that rendering-based side channels remain viable without JavaScript but are fundamentally constrained by browser credential handling policies.

V. Conclusion

This paper demonstrates the feasibility of cross-origin login-state fingerprinting using CSS-only techniques without JavaScript. We show that modern CSS features can induce conditional resource-loading behavior, enabling side-channel inference of authentication states across multiple platforms.

Through a proof-of-concept implementation

evaluated with 20 participants, we observed that login-state fingerprints can be constructed as binary vectors, yielding 10 distinct patterns and a uniqueness rate of 0.5. This result indicates that while the approach can distinguish a subset of users, fingerprint collisions occur due to dependencies among services.

In particular, we find that services relying on shared authentication infrastructures, such as Google SSO, introduce correlation among login-state signals. Our results also show that observability of login states is influenced not only by authentication status but also by service-specific interaction conditions. For example, in the case of Blogger, detectable signals depend on whether the user has previously initiated a login process, highlighting that inferred signals reflect observable behavior rather than guaranteed ground-truth authentication states.

Despite these limitations, the results confirm that CSS alone can act as a cross-origin inference channel capable of leaking authentication-related information without JavaScript execution. The attack is effective under default browser configurations where credentials are included in cross-origin requests, but it is significantly mitigated by modern defenses such as third-party cookie blocking and storage partitioning.

These findings indicate that rendering-layer side channels remain a viable and previously underexplored source of cross-origin information leakage, with effectiveness determined by the independence and observability of authentication signals.

As future work, we plan to extend this analysis to a broader range of web services to better understand how platform-specific behaviors influence the observability of authentication signals. In particular, identifying services that consistently expose rendering-dependent differences remains an important direction. We also aim to explore additional CSS-based mechanisms that may introduce new observable side channels. For example, rendering-related features such as color blending and compositing effects could amplify subtle differences in resource loading outcomes, enabling more robust inference signals.

References

[1] A. Cahn, S. Alfeld, P. Barford, and S. Muthukrishnan, "An empirical study of web cookies", WWW '16: Proceedings of the 25th International Conference on World Wide Web, Montréal, Québec, Canada, pp. 891-901, Apr. 2016. https://doi.org/10.1145/2872427.2882991.

[2] M. W. Docs, "Using HTTP cookies", 2023. https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies. [accessed: Apr. 01, 2026]

[3] Google, "Third-party cookie deprecation", 2025. https://developers.google.com/privacy-sandbox/3pcd. [accessed: Apr. 01, 2026]

[4] J. Schuh, "Building a more private web: A path towards making third party cookies obsolete", 2020. https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html. [accessed: Apr. 01, 2026]

[5] L. Trampert, D. Weber, L. Gerlach, C. Rossow, and M. Schwarz, "Cascading Spy Sheets: Exploiting the Complexity of Modern CSS for Email and Browser Fingerprinting", Internet Society, pp. 24-28, Feb. 2025 https://dx.doi.org/10.14722/ndss.2025.230238

[6] XS-Leaks Wiki, "Cross-site leaks (XS-Leaks)", 2023. https://xsleaks.dev. [accessed: Apr. 01, 2026]

[7] G. Gulyás, D. Somé, N. Bielova, and C. Castelluccia, "To Extend or not to Extend: on the Uniqueness of Browser Extensions and Web Logins", WPES'18: Proceedings of the 2018 Workshop on Privacy in the Electronic Society, Toronto Canada, pp. 14-27, Oct. 2018. https://doi.org/10.1145/3267323.3268959.

[8] W3C, "CSS Containment Module Level 3", 2022. https://www.w3.org/TR/css-contain-3/. [accessed: Apr. 01, 2026]

[9] G. Heyes, "Blind CSS Exfiltration: Exfiltrate

unknown web pages", PortSwigger Research, 2023. https://portswigger.net/research/blind-css-exfiltration. [accessed: Apr. 01, 2026]

[10] OWASP Foundation, "Testing for CSS Injection", WSTG v4.1, 2023. https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/11-Client_Side_Testing/05-Testing_for_CSS_Injection. [accessed: Apr. 01, 2026]

[11] P. Eckersley, "How unique is your web browser?" Privacy Enhancing Technologies (PETS 2010), Berlin, Germany, Vol. 6205, pp. 1-18, Jul. 2010. https://doi.org/10.1007/978-3-642-14527-8_1.

[12] G. Laperdrix, W. Rudametkin, and B. Baudry, "Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints", 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, May 2016. https://doi.org/10.1109/SP.2016.57.

[13] World Wide Web Consortium (W3C), "CSS Cascading and Inheritance Level 4", https://www.w3.org/TR/css-cascade-4/. [accessed: Apr. 01, 2026]

[14] MDN Web Docs, "Pseudo-classes", https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes. [accessed: Apr. 01, 2026]

[15] MDN Web Docs, "Attribute selectors", https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors. [accessed: Apr. 01, 2026]

[16] W3C, "Same Origin Policy", https://www.w3.org/Security/wiki/Same_Origin_Policy. [accessed: Apr. 01, 2026]

[17] MDN Web Docs, "CSS: Cascading Style Sheets", https://developer.mozilla.org/en-US/docs/Web/CSS. [accessed: Apr. 01, 2026]

[18] M. V. Goethem, W. Joosen, and N. Nikiforakis, "The Clock is Still Ticking: Timing Attacks in the Modern Web", CCS '15: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, pp. 1382-1393, Oct. 2015. https://doi.org/10.1145/2810103.2813632.

저자소개

Roby Firnando Yusuf

2023. 9 : B.S. degree, Dept. of Computer Science, Universitas Bhinneka Nusantara(UBHINUS)

2025. 9 ~ present : M.S. candidate, Dept. of Software Engineering, Jeonbuk National University

Research interests : Web Security, Blockchain, Digital Forensics

Abhishek Chaudhary

2022. 8. : B.S. degree, Dept. of Computer Science, Coventry University, Nepal

2025. 8 : M.S. degree, Dept. of Software Engineering, Jeonbuk National University

2025. 9 ~ Present : Ph.D. candidate, Dept. of Software Engineering, Jeonbuk National University

Research interests : cyber-physical systems, intrusion detection, anomaly detection, AI-driven security

Sunoh Choi

2005. 2 : B.S. degree, Dept. of Computer Science, Korea University, Republic of Korea

2014. 5 : Ph.D. degree, Dept. of Computer Engineering, Purdue University, USA

2014. 8 ~ 2019. 2 : Senior Researcher, Information Security Research Division, Electronics and Telecommunications Research Institute (ETRI), Republic of Korea

2021. 3 ~ Present : Associate Professor, Dept. of Software Engineering, Jeonbuk National University, Republic of Korea

Research Interests : Intelligent Security, Physical AI Security, Cyber-Physical Systems (CPS) Security