Stealth Tracking: The Hidden Threats of Browser Fingerprinting and Phishing*

Oleksii Chalyi[1,*], Laurynas Pivoras[1] and Kęstutis Driaunys[1]

[1] *Institute of Social Sciences and Applied Informatics, Vilnius University, Muitines St 8, LT-44280 Kaunas, Lithuania*

Abstract

This research examines browser-based information gathering techniques, distinguishing between passive browser fingerprinting, which collects data without user interaction, and active browser phishing techniques that require minimal user engagement. The study evaluates the types of information that browsers expose, their security implications, and the effectiveness of countermeasures. A risk-based information value estimation model is introduced to assess the sensitivity of collected data. Experiments using AmIUnique and BrowserScan demonstrate the extent of fingerprintable attributes, while phishing simulations highlight the risks of unauthorized access to user credentials, clipboard data, and hardware resources. The findings indicate that passive browser fingerprinting poses a moderate privacy risk, whereas active browser phishing techniques present a more severe threat. The research underscores the need for improved browser security measures, user awareness, and proactive countermeasures to mitigate these threats.

Keywords

Information Security, Information Gathering, Browser Fingerprinting, Risks, Information Value Evaluation, Browser Phishing, Social Engineering

1. Introduction

As of early 2025, over 67% of the global population actively uses the Internet [1]. Web applications have become the primary method for accessing information, conducting transactions, and engaging in digital communications [2]. However, this widespread adoption has also increased the attack surface for cybercriminals, making web browsers a critical vector for both information gathering and security exploitation.

Previous studies have demonstrated that browser security varies significantly due to differences in architecture, update policies, and patching frequency, leading to varying levels of vulnerabilities recorded in the CVE database [3]. This research builds upon prior work by shifting the focus from analyzing browser vulnerabilities to exploring how browsers collect and expose user information through Browser Fingerprinting and Phishing techniques.

Browser Fingerprinting refers to data collection that occurs without direct user interaction, leveraging elements such as HTTP headers, User-Agent strings, WebRTC leaks, IP addresses, and cookies. In contrast, Browser Phishing involves executing scripts and requiring user engagement—for example, recording audio and camera, copying clipboard and stealing credentials, or prompting the user to interact with specific elements that reveal additional device attributes.

This study provides a comprehensive analysis of Browser Fingerprinting and Browser Phishing from a security perspective, investigating the scope of data collected by browsers and evaluating its security implications using CVSS scoring. Unlike other works, it systematically assesses the potential

*IVUS2025: Information Society and University Studies 2024, May 15, Kaunas, Lithuania

[1] Corresponding author.

oleksii.chalyi@knf.vu.lt (O. Chalyi); laurynas.pivoras@knf.stud.vu.lt (L. Pivoras); kestutis.driaunys@knf.vu.lt (K. Driaunys);

0009-0006-3536-9715 (O. Chalyi); 0009-0008-9176-9021 (L. Pivoras); 0000-0002-8456-123X (K. Driaunys);

risks associated with Browser Fingerprinting methods, estimating collected information value, CVSS scores and comparing fingerprinting results across different browsers and operating systems to highlight variations in security exposure. The findings contribute to a better understanding of browser-based information gathering, its role in cybersecurity, and possible countermeasures against unauthorized data collection, ultimately improving cybersecurity awareness by identifying vulnerabilities that could be exploited through browser information leakage.

2. Related Works

D. Zhang et al., in their work [4], provided a survey of Browser Fingerprinting research and applications. They investigated different fingerprinting techniques based on various technologies, such as JavaScript-based fingerprinting, CSS-based fingerprinting, Canvas-based fingerprinting, Hardware- and Software-based fingerprinting, Plugin-based fingerprinting, and fingerprinting based on the Audio API. As countermeasures, they proposed a browser protection plugin that returns random values when querying certain properties, a randomization method that can disguise itself as a desired feature while protecting user privacy, and uniform methods that construct a set of fake fingerprinting information to counteract browser fingerprint tracking. Other protective measures include hiding browser fingerprints through browsers or modifying network agents. In conclusion, D. Zhang et al. state that browser fingerprints, when used alone, serve as indicators of user identity but still pose many challenges. However, the combination of browser fingerprints with traditional user identity tracking technologies can be applied in various fields, such as identity tracking, user authentication, and security defense.

A. Berke et al. investigated the role of demographics in Browser Fingerprinting among U.S. users [5]. They created a dataset that includes browser attributes along with users' demographics and survey responses, collected with informed consent from 8,400 U.S. study participants. Their study demonstrates how fingerprinting risks vary across demographic groups. They found that lower-income users are more at risk and observed that as users' age increases, they are both more likely to be concerned about fingerprinting and at greater actual risk of being fingerprinted. Furthermore, they highlighted an overlooked risk: user demographics—such as gender, age, income level, and race—can be inferred from browser attributes commonly used for fingerprinting. In conclusion, A. Berke et al. state that fingerprinting risks differ across demographic groups due to variations in device attributes and user configurations.

D. Moad et al., in their research, investigated defense methods against browser-based user tracking [6]. As the main countermeasures, they highlighted browser extensions designed to prevent fingerprinting and compared three of them. Their results show that after applying these extensions, the visibility of some attributes changed; however, the most important one in their study, the user agent, remained visible. This indicates that extensions cannot fully block crucial information collected by browsers. They also reviewed the randomization method, but found that most information still could not be effectively randomized. In conclusion, D. Moad et al. state that while various methods can help prevent Browser Fingerprinting, they can only do so partially and cannot completely eliminate the risk.

A. Lawall, in his work, explores the development and impact of browser fingerprinting on digital privacy [7]. This paper provides an overview by examining various fingerprinting techniques and analyzing the entropy and uniqueness of the collected data. The analysis highlights that browser fingerprinting poses a complex challenge from both technical and privacy perspectives, as users often have no control over the collection and use of their data. In addition, it raises significant privacy concerns, as users are often tracked without their knowledge or consent. He reviews different fingerprinting methods and also compares them according to their uniqueness, stability, entropy, and type of impact. In conclusion, the author states that despite stricter privacy laws like the GDPR in the EU, browser fingerprinting remains a grey area, and anti-fingerprinting techniques are limited and continually evolving to keep up with new tracking methods.

P. Laperdix et al., in their work, explore the validity of browser fingerprinting in today's environment [8]. For this, they collected 118,934 fingerprints composed of 17 attributes gathered through the most recent web technologies, such as HTML5, which provides access to highly discriminating attributes. They also show that browser fingerprinting is as effective on mobile devices as it is on desktops. They also provide countermeasures against browser fingerprinting, such as disabling JavaScript, removing browser plugins, reducing the surface of HTML APIs, and increasing common default content. Additionally, they find that 81% of mobile fingerprints in their collected database are unique. They also found that having generic HTTP headers and removing browser plugins could reduce fingerprint uniqueness on desktops by a significant 36%.

3. Methodology

3.1. Collecting Data

This research examines data collected through user interaction with a browser and passive information gathered via Browser Fingerprinting.

For passive data collection via Browser Fingerprinting, the BrowserScan [9] and AmIUnique [10] websites were used, as they provide extensive information about users visiting them. Two data collection scenarios were analyzed: one with default website permissions and another with all permissions forbidden. Since users can modify these settings in their browsers, this comparison helps determine which information can be hidden and whether additional actions are necessary for users to minimize the data collected during website visits.

For active data collection via browser Phishing, the focus was on information gathered with minimal user interaction, such as clicking a button or downloading and running a file. For checking the availability of data collection, the simple JavaScript codes was developed with using navigator function.

3.2. Estimating the Value of Collected Data

In this research, the passive information collected from Browser Fingerprinting was estimated. To evaluate it, the equation (1), based on risk calculation [11], was proposed:

$$IV = S * T, \tag{1}$$

where IV – information value, S – similarity, T – threat level.

The similarity value, which represents the uniqueness of the data, was collected from the AmIUnique website, which already provides a percentage of similarity. For data collected from the BrowserScan website, the similarity was determined based on the general percentage of similarity for this data, which can vary.

In the context of this research, more unique data holds greater informational value, as it provides a rare user fingerprint that may be exploited in future attacks, such as phishing or social engineering. Therefore, additional calculations are required to convert the similarity ratio into a 10-point scale.

Formula (2), based on a previously developed formula [12] for estimating such data (where a lower result corresponds to a better score), was proposed:

$$S = \frac{10 * S_b}{S_c}, \tag{2}$$

where S – similarity, S_b – best similarity ratio, S_c - current similarity ratio.

The Threat Level, which in this research refers to the extent to which the information can be used to identify the user's identity or facilitate more complex attacks, was also estimated on a 10-point scale to maintain consistency with the Similarity score. To ensure objectivity, a group of cybersecurity experts was assembled. The evaluation involved three cybersecurity specialists with varied practical and academic backgrounds. To minimize bias, individual profiles were not revealed during scoring. The evaluation was conducted independently, and the results of other experts

remained anonymous, as the Expert Scoring [13] method was used. The average score across experts' responses was calculated for each parameter. Detailed calculations can be provided upon request.

4. Results

4.1. Browser Fingerprinting Data

To detect Browser Fingerprinting, the data collected from the BrowserScan and AmIUnique websites, which provide passive information that a browser can gather without direct user interaction, were used.

Additionally, a comparison of Browser Fingerprinting across different browsers and platforms (Windows 10 22H2 and Kali Linux 2024.3) was performed to analyze the dependence between the platform and the data collected by the browser. The results showed that the collected information remained the same. However, platform differences resulted in variations in the vendor and renderer, likely due to running the Linux machine in Virtual Box. It was also observed that BrowserScan could not accurately determine the precise Linux distribution; instead, it displayed the Linux kernel version (Linux 6.8.11).

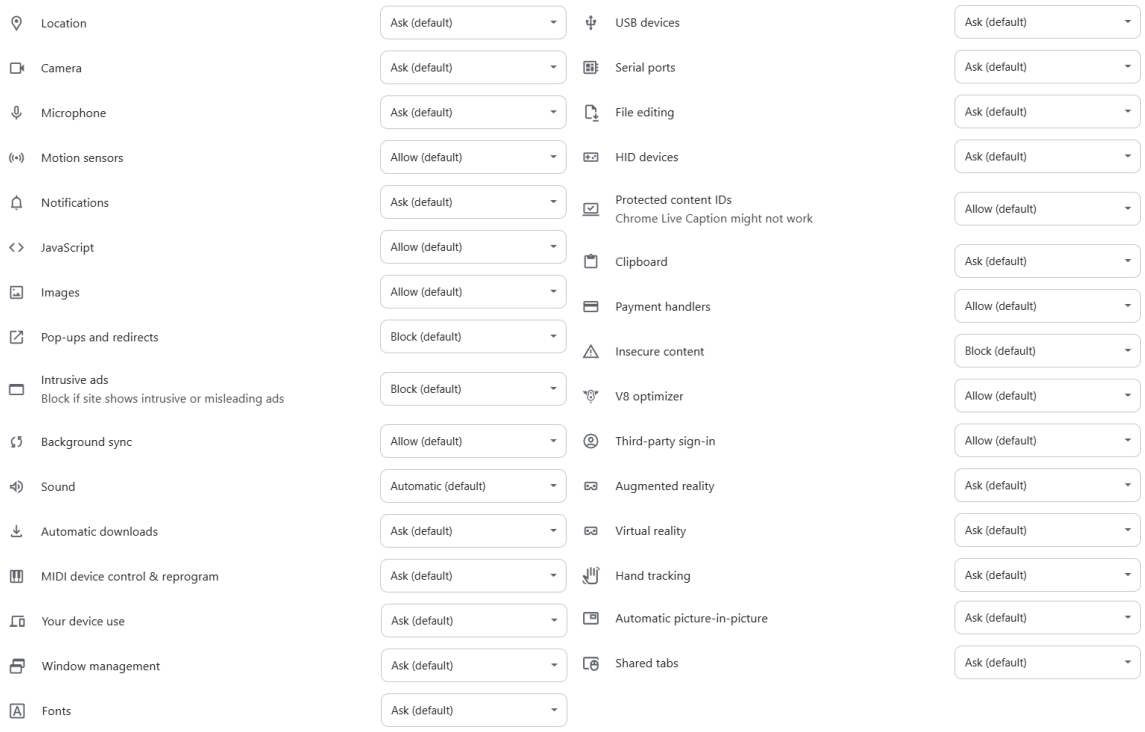

Figure 1: Default permissions in Google Chrome v.133.1907

This figure shows the different permissions that users can set. In the context of the experiments, all permissions were set to the "forbid" security level to check the dependence between Browser Fingerprinting and these permissions. Because the default permissions already provide the required information, there was no point in testing all permitted settings. Figure 2 shows the results of changing permissions for data gathered by the AmIUnique website.

Collected data	Default Permissions	Forbid Permissions
Browser	✔	✔
Platform	✔	✔
IP Time Zone	✔	✔
Language	✔	✔
Proxy	✔	✔
Flash availability	✔	✔
Cookies	✔	✔
Fonts	✔	✘
Canvas	✔	✘
WebGL	✔	✘
GPU Renderer	✔	✘
Device memory	✔	✘
User-Agent	✔	✔
Content Encoding	✔	✔
Use of Adblock	✔	✔
List of Plugins	✔	✔
Permissions	✔	✔
Battery	✔	✔
Connection	✔	✔

Figure 2: Permissions' Influence on Data Collected by AmIUnique.

Additionally, based on the formulas in the previous section, the information value and other parameters were calculated. Figure 3 shows the visualization of the results for data collected by AmIUnique.

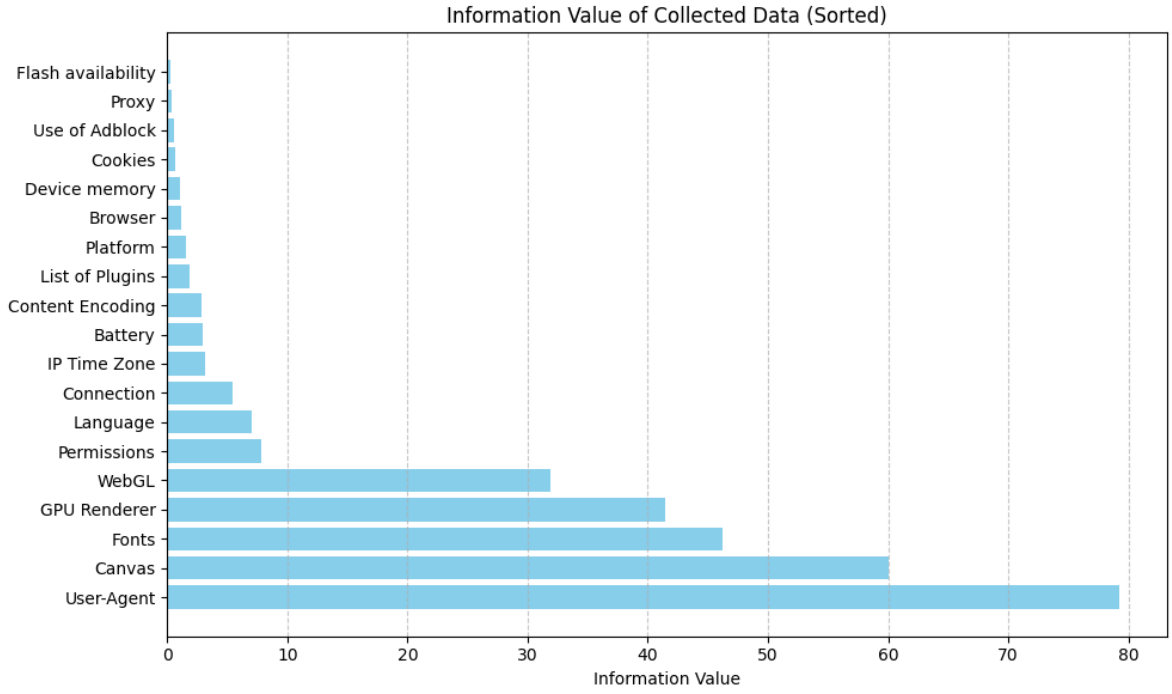

Figure 3: Information Value Scores for Data Collected by AmIUnique.

According to this figure, the User-Agent is the most valuable data that a browser can obtain through fingerprinting. Unlike Canvas, this data cannot be hidden either through browser settings or by using browser extensions. Flash availability, Proxy, and Cookies have the lowest information value, as the AmIUnique website only provides information about their usage. Parameters such as Platform and Browser have high T scores, but due to their high similarity,

their IV scores were reduced—unlike the User-Agent, which is more unique and contains a combination of this data.

Figure 4 shows the results of changing permissions for data gathered by the BrowserScan website.

Collected data	Default Permissions	Forbid Permissions
Browser Version	✔	✔
IP address	✔	✔
Location	✔	✔
Postal Code	✔	✔
ISP (Provider)	✔	✔
DNS Leak	✔	✔
WebRTC	✔	✔
Incognito Mode	✔	✔
Header	✔	✔
Country	✔	✔
Region	✔	✔
City	✔	✔
Javascript availability	✔	✔
Screen Resolution	✔	✘
Screen Size	✔	✘
Color Depth	✔	✘
Touch Support	✔	✘
PDF viewer	✔	✘
CPU Architecture	✔	✔

Figure 4: Permissions' Influence on Data Collected by BrowserScan.

Additionally, based on the formulas in the previous section, the information value and other parameters were calculated. Figure 5 shows the visualization of the results for data collected by BrowserScan.

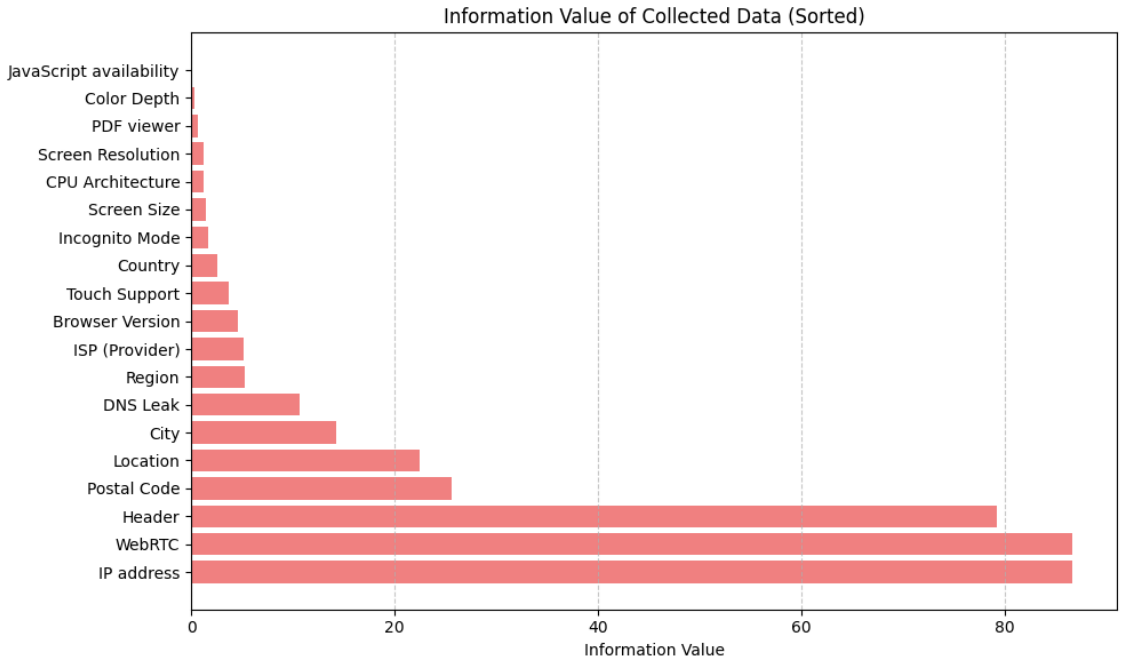

Figure 5: Information Value Scores for Data Collected by BrowserScan.

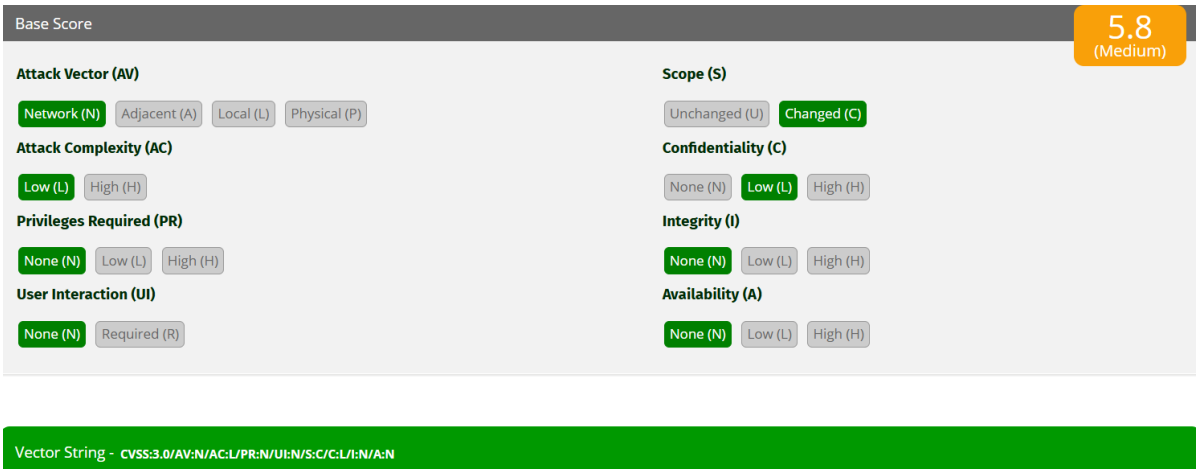

Figure 6: Browser Fingerprinting CVSS 3.0 Score.

This figure shows that the CVSS score for the Browser Fingerprinting threat is medium (5.8) due to its low attack complexity, lack of required user interaction, network-based attack vector, and absence of privilege requirements.

4.2. Browser Phishing Data

Browser phishing data refers to information that requires minimal user interaction to be accessed via the browser. The amount of data collected via browser phishing is not limited, but the most crucial pieces of information include the following:

- User Credentials
- Copy Clipboard
- Audio Recording
- Camera Recording
- Loading the CPU Usage up to 100%

The most important aspect is that collecting such data does not require extensive programming knowledge. Figure 7 shows a snippet of JavaScript code that enables copying clipboard contents and loading the CPU up to 100% usage.

```
Copy Clipboard
<script>
        function copyText() {
            navigator.clipboard.writeText(document.getElementById("textInput").value);
        }
</script>

100% CPU Usage
<script>
        function stressCPU() {
            while (true) {}
        }
</script>
```

Figure 7: Example of JavaScript Code for Copying Clipboard Data and Causing 100% CPU Load.

This figure proves that collecting critical data such as clipboard content does not require extensive programming and can be easily integrated into a website. Figure 8 shows an example of JavaScript code required for camera recording.

```
Camera Recording
    <script>
        let mediaRecorder, videoChunks = [];

        async function startRecording() {
            let stream = await navigator.mediaDevices.getUserMedia({ video: true });
            document.getElementById("video").srcObject = stream;
            mediaRecorder = new MediaRecorder(stream);
            mediaRecorder.ondataavailable = e => videoChunks.push(e.data);
            mediaRecorder.onstop = () => {
                let videoBlob = new Blob(videoChunks, { type: 'video/webm' });
                document.getElementById("recorded").src = URL.createObjectURL(videoBlob);
            };
            mediaRecorder.start();
        }

        function stopRecording() {
            mediaRecorder.stop();
        }
    </script>
```

Figure 8: Example of JavaScript Code for Camera Recording.

This figure shows that even for collecting critical user data through camera recording, only general knowledge of JavaScript is required. The main drawback of this method is the necessity of user interaction with the website. However, social engineering techniques or hiding such buttons can mitigate this issue.

To validate the feasibility of browser phishing techniques, a simple experimental simulation was conducted. A basic web page implementing JavaScript-based clipboard reading and camera access were tested on Chrome (v135) and Edge (v135). Clipboard reading succeeded in both browsers after minimal user interaction, while camera recording required explicit permission. These findings confirm that even simple phishing scripts can extract sensitive data with limited user involvement. Using the CVSS Calculator 3.0, the score for browser phishing data was determined and is presented in Figure 9.

The CVSS score for Browser Phishing is high (7.4) and is higher than that of Browser Fingerprinting due to its greater impact on user confidentiality, even considering the required user actions. Recorded audio or video provides more valuable data for social engineering compared to information such as a user's Canvas fingerprint or IP address, making Browser Phishing a more dangerous cyberattack.

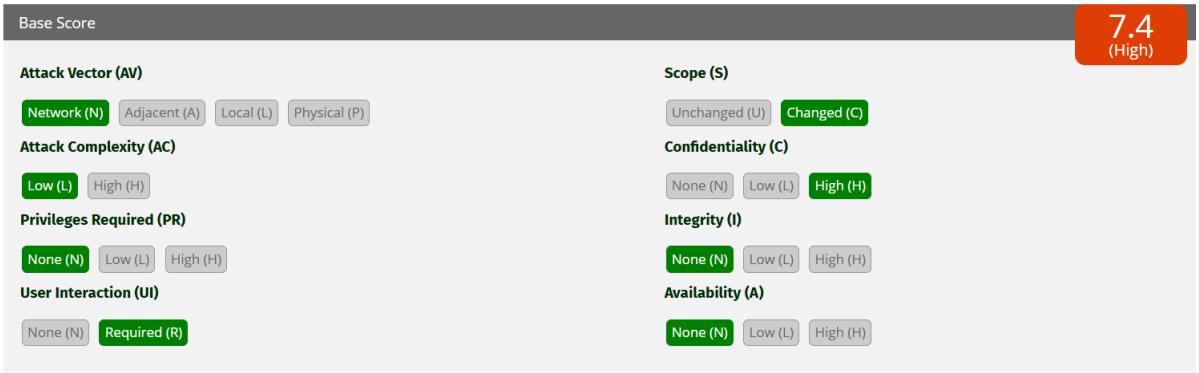

Figure 9: Browser Phishing CVSS 3.0 Score.

The CVSS score for Browser Phishing is high (7.4) and is higher than that of Browser Fingerprinting due to its greater impact on user confidentiality, even considering the required user actions. Recorded audio or video provides more valuable data for social engineering compared to information such as a user's Canvas fingerprint or IP address, making Browser Phishing a more dangerous cyberattack.

5. Discussion

Browser phishing represents a particularly dangerous threat because it often combines technical attacks with social engineering techniques to deceive users into compromising their own devices or credentials. Unlike passive browser fingerprinting, which collects less sensitive information without interaction, browser phishing can lead to the unauthorized capture of user credentials, clipboard contents, audio, video, and more.

For mitigating Browser Fingerprinting, previous research [4-6] suggests that using privacy-focused browsers like Brave, enabling anti-fingerprinting features, blocking third-party cookies, and disabling JavaScript when possible can help reduce data exposure. It is also recommended to remove browser plugins [8]. Additionally, using a VPN or Tor to mask the IP address and regularly clearing cookies and cache can further limit tracking. Restricting permissions in the web browser can also help hide some Canvas-related data.

To protect against browser phishing, enabling multi-factor authentication (MFA), verifying URLs before clicking, and educating users on recognizing suspicious UI elements are effective measures. Installing anti-phishing browser extensions and keeping software updated to block known phishing sites and vulnerabilities can also enhance security.

The results of this research provide a foundation for future work on developing a browser-based penetration testing system. The primary goal of such a system would be to assess the information and resources that could be extracted from a user during phishing attacks and Browser Fingerprinting. By simulating scenarios where users are directed to spoofed websites, this system could identify vulnerabilities specific to the user's browser, shedding light on potential risks at both individual and organizational levels.

6. Conclusions

This work presents research on the information collected through Browser Fingerprinting and browser phishing threats. A list of collected data was created, and the information value (IV) for Browser Fingerprinting was determined. Additionally, using the CVSS calculator, threat scores were estimated.

Regarding the information gathered by Browser Fingerprinting, the most critical data points are the User-Agent, IP address, and WebRTC, as they have the highest IV scores in this research. Based on AmIUnique data, the uniqueness of the HTTP header is not low (1.08% similarity), and it also provides a high information value score, according to this research, unlike previous research [7].

The CVSS assessment categorized Browser Fingerprinting as a medium threat with a score of 5.8 due to its low complexity and lack of user interaction. However, it provides less critical information compared to data obtained through browser phishing techniques.

The study on browser phishing data demonstrated that JavaScript code can be used to integrate UI elements, such as buttons, to extract sensitive user information, posing a threat to confidentiality. The high CVSS score of 7.4 indicates that this type of cyberattack can cause significant harm, despite requiring user interaction. Since browser phishing can be part of social engineering and phishing attacks, it is considered a particularly dangerous threat.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] S. Kemp, "Digital 2025: Global Overview Report," DataReportal – Global Digital Insights, Feb. 2025. URL: https://datareportal.com/reports/digital-2025-global-overview-report

[2] R. R. Ling, D. C. Yen, and D. C. Chou, “From Database to Web Browser: The Solutions to Data Access,” Journal of Computer Information Systems, vol. 41, no. 2, pp. 58–63, Jan. 2001, doi: https://doi.org/10.1080/08874417.2002.11646993
[3] O. Chalyi, K. Driaunys, V. Rudžionis, ”Assessing Browser Security: A Detailed Study Based on CVE Metrics,” Future Internet, vol. 17, no. 3, Feb. 2025, doi: https://doi.org/10.3390/fi17030104
[4] D. Zhang, J. Zhang, Y. Bu, B. Chen, C. Sun, and T. Wang, “A Survey of Browser Fingerprint Research and Application,” Wireless Communications and Mobile Computing, vol. 2022, pp. 1–14, Nov. 2022, doi: https://doi.org/10.1155/2022/3363335
[5] A. Berke et al., “How Unique is Whose Web Browser? The role of demographics in Browser Fingerprinting among US users,” Proceedings on Privacy Enhancing Technologies, vol. 2025, no. 1, pp. 720–758, Nov. 2024, doi: https://doi.org/10.56553/popets-2025-0038
[6] D. Moad, Vikas Sihag, G. Choudhary, Daniel Gerbi Duguma, and I. You, “Fingerprint Defender: Defense Against Browser-Based User Tracking,” Springer eBooks, pp. 236–247, Jan. 2022, doi: https://doi.org/10.1007/978-981-16-9576-6_17
[7] A. Lawall, “Fingerprinting and Tracing Shadows: The Development and Impact of Browser Fingerprinting on Digital Privacy,” arXiv (Cornell University), Nov. 2024, doi: https://doi.org/10.48550/arxiv.2411.12045
[8] P. 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), May 2016, doi: https://doi.org/10.1109/sp.2016.57
[9] BrowserScan, “Best BrowserScan Fingerprint Detection Tool - Improve your online privacy security,” BrowserScan, 2025. URL: https://www.browserscan.net/
[10] AmIUnique.org, “My Fingerprint - web-am-i-unique,” amiunique.org, 2024. URL: https://amiunique.org/fingerprint
[11] G. Apostolakis and S. Kaplan, “Pitfalls in risk calculations,” Reliability Engineering, vol. 2, no. 2, pp. 135–145, Apr. 1981, doi: https://doi.org/10.1016/0143-8174(81)90019-6
[12] O. Chalyi and M. Kolomytsev, “Comparison of Tools for Web-Application Brute Forcing,” Theoretical and Applied Cybersecurity, vol. 4, no. 1, Feb. 2023, doi: https://doi.org/10.20535/tacs.2664-29132022.1.274117
[13] H. Li and Y. Yu, “Research on the evaluation of expert scoring method in the competitiveness of high colleges and universities of Jiangxi province,” pp. 448–450, Nov. 2013, doi: https://doi.org/10.1109/iciii.2013.6702970
[14] First.org, “Common Vulnerability Scoring System Version 3.0 Calculator,” FIRST — Forum of Incident Response and Security Teams. URL: https://www.first.org/cvss/calculator/3.0