LETTER

Tracking WebVR User Activities through Hand Motions: An Attack Perspective*

Jiyeon LEE[†a)], *Nonmember*

SUMMARY With the rapid advancement of graphics processing units (GPUs), Virtual Reality (VR) experiences have significantly improved, enhancing immersion and realism. However, these advancements also raise security concerns in VR. In this paper, I introduce a new attack leveraging known WebVR vulnerabilities to track the activities of VR users. The proposed attack leverages the user's hand motion information exposed to web attackers, demonstrating the capability to identify consumed content, such as 3D images and videos, and pilfer private drawings created in a 3D drawing app. To achieve this, I employed a machine learning approach to process controller sensor data and devised techniques to extract sensitive activities during the use of target apps. The experimental results demonstrate that the viewed content in the targeted content viewer can be identified with 90% accuracy. Furthermore, I successfully obtained drawing outlines that precisely match the user's original drawings without performance degradation, validating the effectiveness of the attack.

key words: *virtual reality, WebVR, side-channel attacks, hand motion tracking, privacy violation*

1. Introduction

The rapid advancement of graphics processing units (GPUs) is hastening the era of Virtual Reality (VR). Simultaneously, security concerns in VR remain an ongoing topic of discussion [10], [11], [13]–[15]. While using VR applications, users typically engage in head and hand movements, and these actions may unintentionally generate additional tracking logs, posing potential risks to users' privacy. For instance, malicious entities can exploit these motion-tracking logs to identify users and discern their app usage, including determining typed inputs, purchased items, consumed content, and more.

In this paper, I present a concerning scenario wherein users' hand motions within a WebVR environment may be exposed to web attackers, giving rise to a new attack that captures user activities in virtual spaces. WebVR [1] is an open specification that enables websites to provide VR experiences through browser support. With WebVR, users can easily engage with VR apps (referred to as WebVR sites), irrespective of the specific VR devices they possess. However, WebVR is susceptible to privacy threats due to the inherent openness of the web. Specifically, malicious websites can access hand motion inputs from benign WebVR sites open in multiple browser tabs without any permission, introducing new privacy risks.

Exploiting the vulnerability present in WebVR, I introduce a novel attack aimed at capturing VR users' activities within virtual environments. Focusing on two privacy-sensitive VR apps—the content viewer app and the drawing utility app—I demonstrate the capability to recognize the content users consume in virtual scenes and monitor their private drawing activities. The proposed approach involves implementing an attacker's website leveraging the security defect on WebVR to collect features for activity tracking. Furthermore, I employ machine learning techniques to discern the target app and the viewed content.

To evaluate the impact of the attack, I collected traces of VR controller motions from seven users as test data. The experimental results indicate that the proposed attack can accurately identify target apps with an 92.4% success rate and extract user activities of the targeted content viewer with 90% accuracy. In addition, I successfully obtained drawing outlines that closely matched users' drawings during the operation of the 3D drawing app, without any noticeable performance degradation. Overall, this paper demonstrates the significant potential of VR controller motion in inferring VR users' activities and highlights the need for robust security measures in VR environments.

2. Background

2.1 Background on WebVR

WebVR [1] is a powerful HTML5 technology that facilitates the seamless integration of VR experiences in web browsers. It offers developers and users a convenient way to create and share VR content. Combined with WebGL, WebVR opens up new possibilities for immersive 3D experiences by providing a collection of VR-enabled interfaces that handle VR devices. Among the essential components of WebVR is the Gamepad API [2], which plays a key role in interacting with VR controllers. Calling this API enables developers to read the positions and orientations of the VR controller in the global coordinate system (e.g., always fixed to the world) in 3D space. To ensure security, the Gamepad API can only be executed in secure contexts, meaning that it requires the use of HTTPS protocol.

Manuscript received January 22, 2024.
Manuscript revised March 19, 2024.
Manuscript publicized April 16, 2024.

[†]School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, South Korea.

[*]This study was supported by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4120240214871).

a) E-mail: jiyeon@knu.ac.kr

DOI: 10.1587/transinf.2024EDL8009

2.2 Problem and Motivation

Although the Gamepad API is maintained in secure contexts, a previous research [10] introduced a critical vulnerability where the Gamepad objects running in the browser context with lack appropriate protection. As a result, multiple websites from different origins can access the Gamepad object simultaneously, presenting a significant security risk. This vulnerability allows an untrusted website to access and gather all controller inputs from a benign WebVR site while both websites are open in the same browser. In [10], Lee et al. proposed a new attack that exploits the security defect in WebVR to infer user inputs. They demonstrated the ability to extract input values entered through a virtual keyboard with up to 96.8% accuracy using only controller sensor data. Given the recent trend that VR has enabled a wide range of applications, including immersive news, 360-degree video players, and content for adults, identifying VR users' activities can also be considered severe privacy-infringing. Therefore, this study aims to assess the accuracy of identifying VR users' activities in virtual spaces using hand motion information.

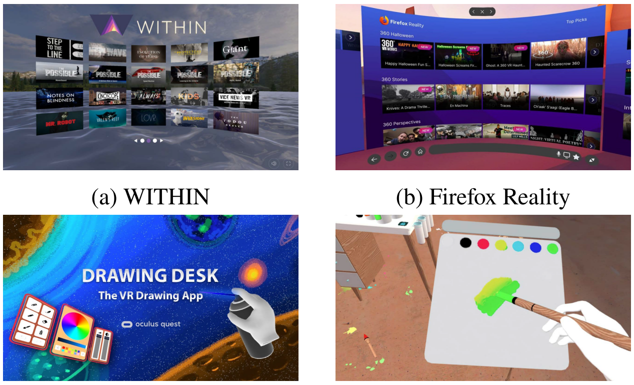
3. Related Work

Recently, there has been a growing number of studies introducing new threats in the VR environment [10]–[12], [15], [16]. The majority of research introduces side-channel attacks that steal users' sensitive information based on tracking information from VR devices [12]–[14]. One of the intuitive methods for capturing VR motions involves bystander observation or vision-based approaches that track VR devices using stereo cameras [12]. More recently, [15] and [16] revealed that tracking VR motions is possible through side-channel information (i.g., performance counters) generated in multi-app scenarios of VR systems. Through this, they introduced new attack vectors, including the extraction of VR users' activities and learning about their surrounding environments. This paper stands out from these research by exploiting security vulnerabilities arising from the combination of VR with web technologies.

4. VR Activity Extraction Attack

4.1 Target Apps

In this section, I introduce a new attack that captures VR users' activities using hand motion tracking information. To achieve this, I examined 16 WebVR showcases offered by A-Frame [8], a representative WebVR framework. From these, I singled out five apps that are using controllers and working at the time of writing, considering them as potential targets for this research. I note that all target apps allow the expression of the controller's movement in a VR environment to be free in a three-dimensional space. After a thorough investigation, I identified two types of VR apps dealing with sensitive information as the primary targets for the attack:



(a) WITHIN (b) Firefox Reality

(c) DRAWING DESK (d) Painting-VR

Fig. 1 Example of target apps: content viewer and 3D drawing utility

- **3D content viewer utility:** 3D content viewers (shown in Fig. 1 (a) and 1 (b)) allow users to view 3D images or videos in an immersive environment. Identifying a content viewed by users is considered a significant breach of privacy. I demonstrate that the disclosure of VR users' hand motions can serve as a means to identify the specific content they are viewing.
- **3D drawing utility:** A 3D drawing app is a representative VR app that enables drawing in a three-dimensional virtual space (see Fig. 1 (c) and 1 (d)). When the app is launched, most 3D drawing apps render the controller as a drawing tool (such as a brush), and the user draws by moving his or her hand while pressing the controller's button. I show that leakage of controller motion can be used to infer 3D outcomes generated by the target drawing app.

4.2 Approach

Figure 2 illustrates the overall attack process. It is composed of two parts: *a client-side website* and *a server-side machine*. First, an attacker's website is set up to collect sensor data from the controller, and a server is prepared to receive the data and extract sensitive information. I assume that the attacker's site is open simultaneously with the user entering a VR mode on the target WebVR site[†]. The attack site can determine the moment when a user is playing a WebVR through the `enter-vr` and `exit-vr` events. Using this information, it records real-time positions and orientations from the controller device using the Gamepad API (mentioned in Sect. 2.1) while the target app is running. In addition, the attack site registers click event handlers on the acquired controller objects. This information proves to be very useful as it allows us to determine all click actions generated by the target app. In order to collect more detailed features, I further refine the click event handler to read `down-click` (button

[†] This scenario commonly occurs during web browsing, where multiple websites are open in different browser windows, or when multiple webpages from various origins are loaded into iframe elements within a single webpage.

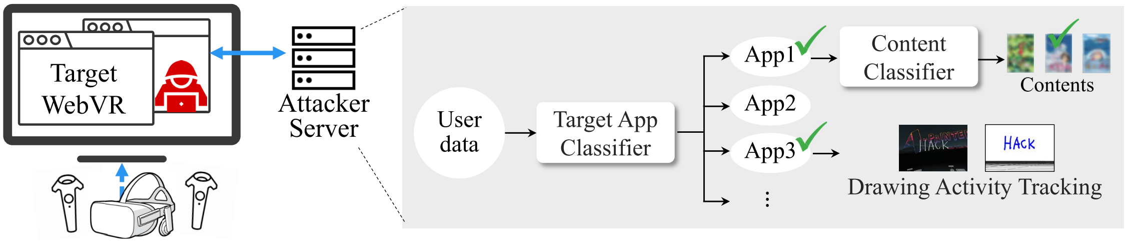


Fig. 2 The overall process of tracking VR users' activities

press) and up-click (button release) events, enabling us to infer user activities by logging frames during these periods.

After the server receives the collected data from the attack site, the attacker first identifies which app the user is running. The main idea for identifying the running app is to leverage the controller motion data. This is valid because different apps have different controller usage patterns (e.g., the drawing app involves long pressing the controller button for dynamic movement, while the content viewer statically clicks on a fixed layout), so the controller's motion data itself serves as a powerful clue for identifying the target app. To achieve this, we extract features for app identification from the received data and use an ML model to train them. This model is trained on the controller's poses, rotations, and click intervals generated when running tested VR apps, enabling it to distinguish between target and non-target apps.

After the target app is identified, I further attempt to extract sensitive activity according to the target app. For the 3D drawing app, reading the controller's poses is sufficient for conducting the attack. To extract the content that the user sees in the targeted viewer app, I utilize an SVM model with the C-SVC multi-class categorization scheme. The ground truth dataset consists of the author clicking content thumbnails 100 times, collecting 400 instances with labels for each click. A detailed description of the performance evaluation is presented in the next section.

5. Evaluation

5.1 Experimental Setup

In this section, I evaluate the attack performance in terms of 1) target app identification accuracy, 2) viewed content identification accuracy, and 3) performance degradation due to logging. As target apps, I used Response UI [6] as a content viewer app and A Painter [7] as a 3D drawing app. Responsive UI currently displays three image contents; I labeled each of the three contents as A, B, and C from left to right. In the case of the 3D drawing app, the attack can track the user's drawing activities with 100% accuracy. To further understand its stealthiness, I measure the performance decrease of A Painter as the attack is carried out. To investigate the accuracy of app and content identification, I recruited seven participants consisting of four males and three females (the average age is 24). I prepared a spacious room for VR with a VR device, HTC VIVE Pro [9]. All experiments were conducted on Firefox (version 116.0) on Windows 10 hosts with a machine capable of running VIVE Pro (Intel Core i9-9900K, GeForce GTX 1070, and 16 GB RAM).

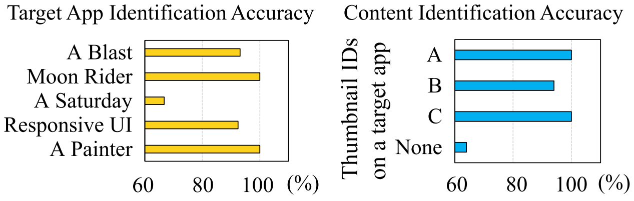


Fig. 3 Accuracy in identifying targeted apps and content. The accuracy is derived as the number of correctly classified instances (the sum of true-positives and true-negatives) over the total number of instances.

For the first experiment, I instructed the participants to randomly select one of the five tested WebVRs[†] and play for five minutes. While they are playing the VR app, I open the attack site in another window of the browser to collect the controller motion data and use this data as test data for the target app identification model. For the second experiment, I instructed the participants to randomly click on the thumbnail of the content they want to view while the content viewer (Responsive UI [6]) is running. If a participant did not accurately click on the thumbnail (e.g., clicked on background), I labeled this instance as 'none'. In conclusion, to assess the performance impact of the attack, I varied the data logging intervals to 16ms, 100ms, and 1000ms, and measured the average Frames Per Second (FPS) of the target app.

5.2 Results

Target app identification accuracy. The graph on the left of Fig. 3 shows the results of measuring how accurately the proposed attack can identify the target apps among the five test apps. The results demonstrate that the target apps, Responsive UI and A Painter, can be identified with accuracies of 92.4% and 100%, respectively. Moreover, non-target apps, A Blast, Moon Rider, A Saturday Night, can also be identified with an accuracy of over 66.7%. The reason for such good performance is that each app has unique characteristics in controller motion. For example, A Painter has

[†] It includes A Blast [3], Moon Rider [4], A Saturday Night [5], Responsive UI [6], and A Painter [7], where the first three are 3D Games and the other two are targets of the proposed attack.

Table 1 An average FPS of a target VR app according to controller logging intervals

Average FPS on A Painter			
normal	1000ms	100ms	16ms
54.5	54.4(-0.1)	51.6(-2.9)	50.2(-4.3)

a long interval between `down-click` and `up-click` events for drawing actions, while other apps that perform simple clicks exhibit relatively short click durations. Additionally, 3D game apps such as A Blast, Moon Rider, and A Saturday Night require a large number of clicks in a short period (e.g., shooting) and have a wide range of movements. On the other hand, Responsive UI has features of low interaction frequency after clicking on the thumbnail of the desired content. As a result, the proposed attack demonstrates the ability to accurately identify the target apps, making the attack more feasible.

Viewed content identification accuracy. The graph on the right of Fig. 3 depicts the performance of content identification accuracy based on 60 collected click instances from experiment participants when the Responsive UI is running. The result demonstrates an overall classification accuracy of 90%, indicating that the attack is highly robust. Upon closer examination, the SVM achieved 100% accuracy in classifying click inputs for contents A and C, while achieving 90% accuracy for content B. This is because the thumbnail of content B is placed between content A and C, so its surrounding margins are relatively narrow, which can be considered to be misclassified as a 'None' click.

Performance degradation. Lastly, to assess the impact of the attack on the target app's performance, I measured the average FPS of the A Painter while varying the data logging intervals. Table 1 illustrates the performance degradation of the target app caused by the attack. The results indicate that the proposed attack has a minimal impact on the reduction of the target app's performance. With a 16ms logging interval, the most frequent recording, I observed a 4.3% decrease in average FPS compared to the scenario where the attack was not executed. The attack site is designed to perform lightweight operations such as reading controller objects, resulting in no noticeable impact on the target's performance during the attack.

6. Conclusion

With the increasing interest in VR, there is a steady resurgence of security threats to VR environments. This paper proposes a new attack that exploits the sensor data about controller motions from real-world WebVR apps to infer VR users' activities. The attack achieved 90% classification accuracy in identifying 3D content viewed by users. Moreover, it perfectly extracted the overall sketch of drawings made in the VR drawing utility with low performance degradation. As the WebVR market expands into areas like education, e-commerce, military, and so on., the proposed attack could lead to more significant damages. To defend against this, future research should focus on strengthening access controls for VR devices to minimize exposure to motion-tracking.

References

[1] Mozilla Developer Network (MDN), "WebVR — Virtual Reality for the Web," https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/WebVR, accessed Aug. 2, 2023.

[2] Mozilla Developer Network (MDN), "GamePad API," https://developer.mozilla.org/en-US/docs/Web/API/Gamepad_API, accessed Aug. 2, 2023.

[3] A-Frame, "An example WebVR site for 3D game: A Blast," https://aframe.io/examples/showcase/a-blast/, accessed Aug. 2, 2023.

[4] A-Frame, "An example WebVR site for 3D game: Moon Rider," https://aframe.io/examples/showcase/moonrider/, accessed Aug. 2, 2023.

[5] A-Frame, "An example WebVR site for 3D game: A Saturday Night," https://aframe.io/examples/showcase/a-saturday-night/, accessed Aug. 2, 2023.

[6] A-Frame, "An example WebVR site for 3D content viewer: Responsive UI," https://aframe.io/examples/showcase/responsiveui/, accessed Aug. 2, 2023.

[7] A-Frame, "An example WebVR site for 3D painter utility: A Painter," https://aframe.io/examples/showcase/a-painter/, accessed Aug. 2, 2023.

[8] A-Frame, "A web framework for building virtual reality experiences," https://aframe.io, accessed Aug. 2, 2023.

[9] HTC, "VIVE Pro Specs," https://www.vive.com/us/product/vive-pro/, accessed Aug. 2, 2023.

[10] J. Lee, H. Kim, and K. Lee, "VRKeyLogger: Virtual Keystroke Inference Attack via Eavesdropping Controller Usage Pattern in WebVR," Elsevier Computers & Security, vol.134, 2023.

[11] H. Lee, J. Lee, D. Kim, S. Jana, I. Shin, and S. Son, "AdCube: WebVR Ad fraud and practical confinement of third-party ads," The 32nd USENIX Security Symposium, 2021.

[12] Z. Ling, Z. Li, C. Chen, J. Luo, W. Yu, and X. Fu, "I Know What You Enter on Gear VR," The 7th IEEE Conference on Communications and Network Security, 2019.

[13] S. Reddy K. Gopal, D. Shukla, J.D. Wheelock, and N. Saxena, "Hidden reality: Caution, your hand gesture inputs in the immersive virtual world are visible to all!," The 32nd USENIX Security Symposium, 2023.

[14] Ü. Meteriz-Yıldıran, N.F. Yıldıran, A. Awad, and D. Mohaisen, "A Keylogging Inference Attack on Air-Tapping Keyboards in Virtual Environments," IEEE Conference on Virtual Reality and 3D User Interfaces, 2022.

[15] C. Slocum, Y. Zhang, N. Abu-Ghazaleh, and J. Chen, "Going through the motions: AR/VR keylogging from user head motions," The 32nd USENIX Security Symposium, 2023.

[16] Y. Zhang, C. Slocum, J. Chen, and N. Abu-Ghazaleh, "It's all in your head(set): Side-channel attacks on AR/VR systems" The 32nd USENIX Security Symposium, 2023.