LETTER

Spy in Your Eye: Spycam Attack via Open-Sided Mobile VR Device

Please confirm each of your IEICE memberships.

Jiyeon LEE[†] ***and*** **Kilho LEE**[†a)], ***Nonmembers***

SUMMARY Privacy violations via spy cameras are becoming increasingly serious. With the recent advent of various smart home IoT devices, such as smart TVs and robot vacuum cleaners, spycam attacks that steal users' information are being carried out in more unpredictable ways. In this paper, we introduce a new spycam attack on a mobile WebVR environment. It is performed by a web attacker who maliciously accesses the back-facing cameras of victims' mobile devices while they are browsing the attacker's WebVR site. This has the power to allow the attacker to capture victims' surroundings even at the desired field of view through sophisticated content placement in VR scenes, resulting in serious privacy breaches for mobile VR users. In this letter, we introduce a new threat facing mobile VR and show that it practically works with major browsers in a stealthy manner.

key words: *virtual reality, mobile computing, hidden camera, spycam, webvr, privacy leak*

1. Introduction

Spycam crime, which illegally films people with a tiny or camouflage camera, is a growing social issue [1]. Previous studies have shown that unauthorized parties can eavesdrop on cameras mounted on IT devices (laptops, smartphones, etc.) to perform spycam crimes [2], [3]. This can lead to illegal filming of people's private spaces and appearances, and seriously, these films are traded on the black market, causing serious social crimes. Recently, as the popularity of the Internet of Things (IoT) has increased and cameras have begun to be installed in home appliances such as smart TVs and robot vacuum cleaners, spycam crimes are expanding in more unpredictable ways [4], [5]. As a result, our society is turning into a dangerous environment where anyone could be a victim of crime.

In this paper, we introduce a new spycam attack, named VRSpycam, in a mobile WebVR environment. The proposed attack uses a back-facing camera on a user's mobile device to capture their surroundings while the user is visiting an attacker's WebVR site. With the rapid improvement in the computing power of mobile devices, mobile platforms have the adequate processing capacity to support WebVR without any additional modifications. Such mobile VR is getting popular because it does not require wires and is not limited to a single location. IT companies such as Samsung and Google have released VR headsets for mobile devices (Fig. 1) that enable VR by placing mobile devices to the head mount.

In addition, with the advent of open VR platforms such as WebVR, VR can be viable more easily in mobile environments. WebVR [6] is an open specification that allows websites to provide a VR environment by means of browser compatibility. WebVR makes it easier for anyone to participate in immersive experiences, regardless of their device. It provides JavaScript (JS) interfaces for managing VR peripherals such as Head Mounted Displays (HMDs) and controllers, making it easy for web developers to build VR applications.

As an opposite side of these advantages, the mobile WebVR introduces a new attack vector which can be exploited for spycam crimes. The problem is that mobile devices are equipped with cameras and mobile web browsers typically allow the back-facing camera to be turned on while running WebVR. Although the back-facing camera is not an essential sensor for VR applications, it can be turned on even in the VR mode, due to the insecure design of web browsers. This allows an attacker to lure the victims to his/her WebVR site, making them believe they are running VR, while simultaneously filming the surroundings with the back-facing camera mounted on the victim's mobile device.

The most critical part of VRSpycam is that the victims do not recognize that they are subjects of illegal filming on their own. Once the WebVR site runs in the VR mode which renders the VR scene in full-screen, it is very hard for the victims to recognize whether the back-facing camera is turned on or not. Moreover, as the camera is placed to the head mount which moves according to the user's movement, the attacker can induce exposures at more various view angles than conventional spycams through a WebVR site that fully utilizes 360-degree spaces. This threat imposes that large amounts of accidental sensitive information, such as valuable items, documents, and children's faces, can be found in the camera streams.

In this paper, we demonstrate the feasibility of the proposed attack through extensive experiments with the proof-of-concept implementation on 12 popular mobile browsers. As a result, 8 browsers (66.7%), including Firefox and Edge, were able to exploit a back-facing camera in the VR mode. We also observed meaningful results that some browsers are applying additional security policies, such as displaying warning signs for camera use during the VR mode. In the performance evaluation, our attacks have been demonstrated as real threats with negligible FPS reduction.

Manuscript received May 12, 2022.
Manuscript publicized July 22, 2022.

[†] The authors are with School of AI Convergence, Soongsil University, Seoul 06978, South Korea.
a) E-mail: khlee.cs@ssu.ac.kr

DOI: 10.1587/transinf.2022EDL8042

Please confirm E-mail and the owner.

2. Threat Model

We assume our threat model is a classic web attacker [7], who serves his or her own website and successfully makes users visit this website using well-known techniques such as phishing [8]. We also assume that the attacker's website has permission to access the camera sensors on the victim's device. An attacker can gain camera permission under the guise of a legitimate website (e.g., a site that provides both AR and VR capabilities) or hack victims' devices to access the camera without user permission [3]. We note that it is a common situation for users to perceive when the camera is in use, apart from permission approval. The spycam attack presented in this paper is threatening in that users do not know that they are filming through mobile devices.

3. VRSpycam Attack

The VRSpycam attack is a new hidden camera attack in the mobile WebVR environment. This attack takes advantage of the fact that WebVR sites with camera permission can turn on the camera even in the VR mode. With the recent improvement in the computing performance of mobile devices, heavy usage of 3D rendering computations has become possible on commercial smartphones. Accordingly, WebVR is receiving keen attention. WebVR [6] is an open specification for providing VR on the Internet. Due to the nature of the web, it does not require any software installation. Instead, it uses web browsers to access VR contents as if accessing regular websites.

WebVR can be used on both desktop and mobile devices. Unlike desktop WebVR, which requires standalone HMD devices, mobile WebVR utilizes a mobile device itself as an HMD along with a mobile VR headset (See Fig. 1). When a user starts VR (so-called VR mode), a VR scene is expanded to fill the entire screen and produced on a display that dynamically reflects the user's movement. This rendering output is then refracted by the two convex lenses linked to the mobile VR headset, providing the user with the sense of being in a virtual world.

Mobile VR tracks user head movements using motion sensors installed on mobile devices. That is, camera sensors are not used to support VR functionality. Unfortunately, some browsers allow WebVR sites with camera privileges to simultaneously activate cameras in the VR mode. If the back-facing camera is turned on while VR users play in a 360-degree virtual world, it causes serious privacy violations not only for VR users themselves but also for individuals standing nearby.

(a) Samsung GearVR (open-typed VR headset)

(b) Google Daydream (close-typed VR headset)

Fig. 1 Commercial mobile VR headsets.

The whole process of the proposed attack is depicted in Fig. 2. To conduct the attack, an attacker entices a victim to visit a malicious WebVR site and lures the victim to grant a camera permission. After the camera access is granted, the malicious site starts to serve the VR application in the VR mode; the victim then mounts his/her smartphone on a mobile VR headset and enters the VR world. Meanwhile, the attacker records the victim's play area with the smartphone's back-facing camera and receives it on their web server to obtain sensitive information about the victim.

This attack is dangerous in that the user who gave the camera permission does not recognize that he or she is filming the surroundings. Due to the VR scene in full screen, it is very hard for users in the VR mode to recognize whether the camera is currently in use or not. It is also hard for people around the user to recognize because no changes (e.g., flashlights) are made to the mobile device. In addition, compared to conventional spycam attacks performed in static environments [2], [4], [5], our VRSpycam attack can induce shooting in various directions by fully exploiting the 3D space. For example, an attacker can place interactive objects to attract users' attention in various directions. It simultaneously causes the mobile device to move, enabling more dynamic recording around users.

The proposed attack employing a smartphone's back-facing camera does not work with VR headsets that completely envelop the smartphone (see Fig. 1(b)). This attack is also difficult to conduct in a desktop VR environment. Some desktop VR devices have camera sensors (e.g., HTC VIVE [9]), but these sensors are difficult to utilize for spycam attacks because they are not accessible at the application level. On the other hand, mobile VR, which utilizes the mobile device itself as an HMD, is easy to attack as the application can access camera resources with the user's permission. Moreover, due to the high applicability, WebVR enables a wide range of attacks that are not limited to any specific individual platforms provided by mobile VR headset manufacturers (e.g., Oculus Gear VR [10]).

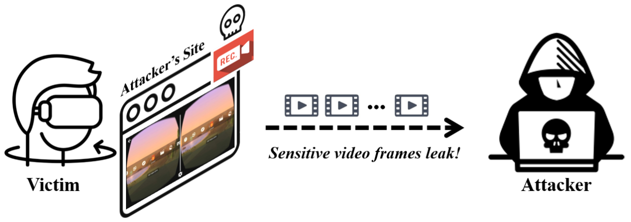
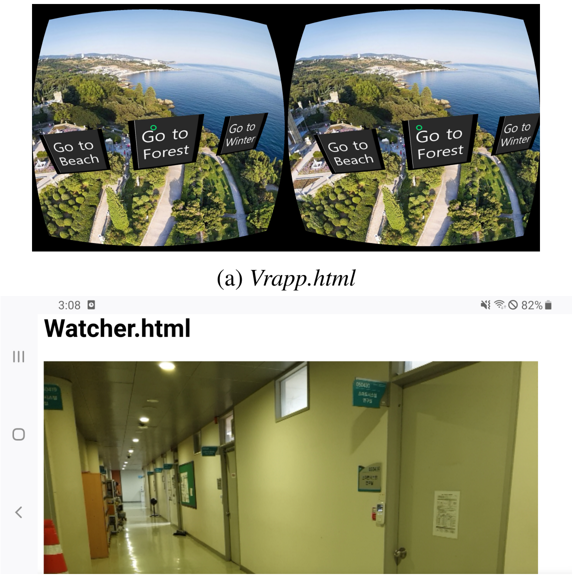


Fig. 2 The entire process of a VRSpycam attack. While the victim visits an attacker's WebVR site through a mobile device, the back-facing camera mounted on the device captures the victim's play area and transmits video streams to the attacker.



(a) *Vrapp.html*

(b) *Watcher.html*

Fig. 3 Two web pages that enable VRSpycam attack.

4. Implementation

To demonstrate the VRSpycam attack, we implemented two web pages, named *vrapp.html* and *watcher.html*. As shown in Fig. 3(a), vrapp.html is a WebVR site containing VR content created to attract victims. For vrapp.html, we simply implemented a gallery app that displays 360° photographs. When a user clicks one of the UI buttons shown in Fig. 3(a), a 360° picture is displayed in the VR scene. It is worth noting that this WebVR site can be designed more finely. For example, an attacker may analyze the video stream being transmitted in real-time and induce the user's movement in a direction that is desired to be filmed in more detail.

When vrapp.html starts, it entices the victim to grant the camera permission to properly run the VR application. With this access permission, a video stream which contains the victim's surroundings is delivered to the watcher.html (Fig. 3(b)). Therefore, an attacker can get sensitive personal information (e.g., room structure, documents on the desk, etc.). To establish peer connections between vrapp.html and watcher.html, we used WebRTC [11], which allows web applications to exchange data such as audio and video without the need for plug-ins or third-party software. Data transfer continues as long as the WebRTC peers' connection is valid.

5. Evaluation

In this section, we evaluate the VRSpycam attack against two questions: 1) the feasibility of the VRSpycam attack on commercial mobile browsers, and 2) the performance overhead caused by the VRSpycam attack.

For the first question, we investigated design differences in terms of security by conducting a VRSpycam attack on 12 popular mobile browsers. In the experiment, we used a Galaxy S9 device with Android 10 and a Samsung GearVR headset.[†] We also used the latest versions of the browsers. Note that iOS does not fully support WebVR yet, so we excluded it from the experiment [12].

Table 1 Summary of attack possibilities and security policies for 12 commercial browsers

Browser	Camera enabled in VR mode	Additional security policy
Firefox	O	X
Chrome	O	DP[1]
Edge	O	X
Opera	O	X
Samsung Internet	O	X
Whale	O	X
Brave	O	DP[1], ASK[2]
Vivaldi	O	X
DuckDuckGo	X	-
Mint Browser	X	-
Dolphin	X	-
UC Browser	X	-

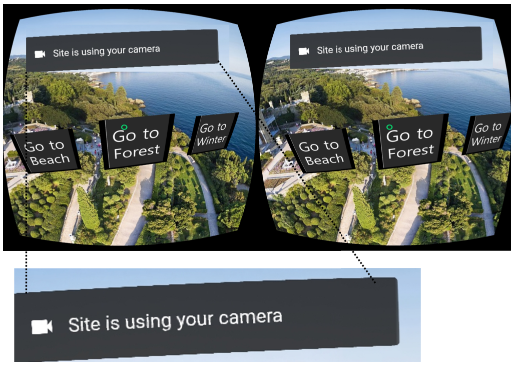
[1] DP: Display warning message (Fig. 4)
[2] ASK: Ask camera permission again when entering VR mode

Table 1 shows the results of the VRSpycam attack on 12 popular browsers. We confirmed that the proposed attack works with 8 out of 12 browsers, including Firefox, Chrome, and Edge. This result indicates that mobile VR users who visit WebVR sites with these browsers are vulnerable to our attack. On the other hand, UC Browser, Mint, and Dolphin did not allow WebRTC connections on the WebVR site and therefore could not access the camera in the VR mode. A little differently, DuckDuckGo [13] is a browser that provides limited functionalities to enhance security, which does not allow the camera access even on regular websites.

While testing the feasibility, we observed a notable difference in the behavior of browsers. Unlike other browsers, Brave asks for camera permission once again when a user enters the VR mode, making it more clear whether the camera is allowed while playing VR. Furthermore, Chrome and Brave automatically display a warning message in the VR scene regarding using the camera. Figure 4 shows the message that says, "Site is using your camera" displayed in vrapp.html. This message makes sense, given that users cannot see any status information on the system bar while in the VR mode. As a consequence, our spycam attack may become less stealthy.

To see how the VRSpycam attack affects the VR performance, we measured the FPS (frames per second) changes when vrapp.html is running. It is important to check because the FPS reduction produces physical symptoms such as motion sickness. We deliberately placed events every 10 seconds into vrapp.html to generate an additional computing workload. Each event loads a sequence of a 3D object, an image, and a video object. As shown in Fig. 5, vrapp.html shows almost similar FPS compared to the base-

[†] Although we tested a single device for evaluation, we expect that other Android devices also show the same result as they have identical web browsers used in our evaluation.



Site is using your camera

Fig. 4 Brave's warning message displayed in the VR scene when accessing the device camera in the VR mode

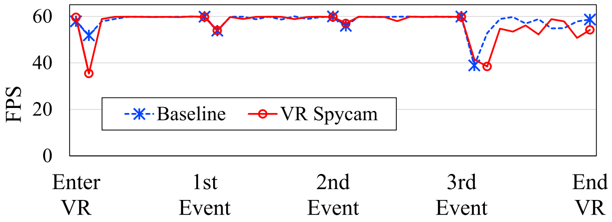


Fig. 5 FPS drops in response to interaction events.

line that does not transmit video streams, except at the "Enter VR" point where an additional task is performed for preparing video transmission. It indicates that our attack can be carried out practically with a negligible performance overhead.

6. Discussion

In this section, we discuss the defense against the proposed spycam threat. As we mentioned in Sect. 3, close-typed mobile VR headsets such as Google's Daydream inherently defend against our attack. These headsets are safe from our attack because they completely cover the camera regardless of the browser design. We also confirmed that browsers' security policies can protect against turning on the back-facing camera during VR use. This restriction makes sense because, unlike AR, VR does not require real-world information. We believe that a same level of security scheme can be implemented with browser extensions. For example, extensions could monitor the getUserMedia API [14], which retrieves camera resources, and alert users when malicious webVR sites call this API in the VR mode. In addition, it will be able to provide a control button to turn off the camera for the user in the VR scene. We leave them as future work.

7. Conclusion

Spycam attacks, which eavesdrop on people's private lives, are becoming more unpredictable by the day. In this paper, we present a new spycam attack that exploits WebVR. WebVR overcomes the constraints imposed by the native VR and allows web users to enjoy VR more easily. However, as a dark point in WebVR, the tremendous features of the web and the characteristics of VR are merged, resulting in undesired scenarios. We've demonstrated how an open-typed mobile VR headset in a sensor-rich environment may be used to launch a powerful and covert spycam attack. For safer WebVR use, we advocate a better browser design that restricts sensitive resource access in the VR mode.

References

[1] 'I was humiliated': The continuing trauma of South Korea's spy cam victims, BBC News, Retrieved May 4, 2022, from https://www.bbc.com/news/world-asia-57493020.
[2] Edward Snowden: the whistleblower behind the NSA surveillance revelations, The Guardian, Retrieved May 4, 2022, from https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance.
[3] How Attackers Could Hijack Your Android Camera To Spy On You, Checkmarx, Retrieved May 4, 2022, from https://checkmarx.com/blog/how-attackers-could-hijack-your-android-camera/.
[4] Smart vacuum flaws could give hackers access to camera feed, say security researchers, ZDNet, Retrieved May 4, 2022, from https://www.zdnet.com/article/smart-vacuum-flaws-could-give-hackers-access-to-camera-feed-say-security-researchers/.
[5] Who's watching whom? Camera-equipped TV can be hacked, says researcher, NBCNews, Retrieved May 4, 2022, from https://www.nbcnews.com/technolog/whos-watching-whom-camera-equipped-tv-can-be-hacked-says-1c7596675.
[6] WebXR Device API, W3C, Retrieved May 4, 2022, from https://www.w3.org/TR/webxr/.
[7] A. Barth, C. Jackson, and J. Mitchell, "Securing Frame Communication in Browsers," USENIX Security Symposium, 2008.
[8] K. Thomas, F. Li, A. Zand, J. Barrett, J. Ranieri, L. Invernizzi, Y. Markov, O. Comanescu, V. Eranti, A. Moscicki, D. Margolis, V. Paxson, and E. Bursztein, "Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials," Proc. ACM Conf. Computer and Communications Security, 2017.
[9] VIVE Pro, HTC, Retrieved May 4, 2022, from https://www.vive.com/us/product/vive-pro/.
[10] Gear VR, Samsung, Retrieved May 4, 2022, from https://www.samsung.com/global/galaxy/gear-vr/.
[11] WebRTC 1.0: Real-time Communication Between Browsers, W3C Groups (2021), Retrieved May 4, 2022, from https://www.w3.org/TR/webrtc/.
[12] WebXR Device API Support Info, Can I Use, Retrieved May 4, 2022, from https://caniuse.com/?search=webxr.
[13] DuckDuckGo Browser, DuckDuckGo, Retrieved May 4, 2022, from https://duckduckgo.com/.
[14] MediaDevices.getUserMedia(), Mozilla Developer Network (2021), Retrieved May 4, 2022, from https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia.