Detecting framerate-oriented cyber attacks on user experience in virtual reality

Blessing Odeleye
University of Greenwich

George Loukas
University of Greenwich

Ryan Heartfield
University of Greenwich

Fotios Spyridonis
University of Greenwich

Abstract

Virtual Reality (VR) is expected to become an enabling technology for training in realistic conditions, data visualisation, education and many other applications. However, there is still limited research on cyber threats to VR environments and even less on technical protections against them. We are currently developing a VR testbed specifically designed for assessing different cyber threats, their impact to user experience and corresponding defences. In this work in progress, we demonstrate two novel approaches by which a cyber attack can potentially cause VR sickness on demand based on frame rate manipulation by taking advantage of GPU and network vulnerabilities. We further show that a simple unsupervised machine learning method using Isolation Forest can provide early warning of such attacks likely before they have significant impact on the VR system and its user.

1 Introduction

Virtual Reality (VR) is being adopted in a rapidly increasing number of application domains. It is estimated that by 2025 the VR market will reach USD 20.9 billion [9] and the technology will be on the way to become an important part of modern digital infrastructure. Yet, unlike other digital environments that have been scrutinised extensively in terms of the cyber security risks they introduce (consider the Internet of Things, Cloud computing and 5G), research in this space is still limited. We argue that this can become a considerable blind spot in the protection of digital environments, especially as the use of Head Mounted Displays (HMDs) reduces drastically users' own ability to observe cues of malicious manipulation, such as network state, CPU usage, physical devices attached or web redirections.

To understand the nature of VR cyber threats, it is important to view it against the two fundamental concepts of immersion and presence. VR environments are designed for immersion by presenting the human brain with artificially generated stimuli, which is the sum total of sensory feedback based on the hardware and software VR components [14], isolating the user from the real world [3]. Presence is the subjective experience of being there or the psychological response of the user to the VR world, which in turn is dependent on immersion and engagement [16]. With presence, the user is aware that they are in a VR world, but respond to virtual entities like they would in the real world, allowing for spatial and social engagement similar to human behaviour in the real world.

These two aspects may be targets or facilitators of cyber attacks. Some excellent examples of such attacks were demonstrated by Casey et al. in [4] who exploited the OpenVR API to disorient users, turn their HMD camera on without their knowledge, overlay unwanted 2D images in their field of vision, and modify VR environmental factors that forced users into hitting physical objects and walls. They coined a proof of concept attack the "human joystick" where the user was deceived into moving to a target physical location without their knowledge. Immersion and the HMD's suppression of visual cues from the real world makes the human vulnerable to such an attack in the same way a GPS spoofing attack has been shown to remotely control a drone or a ship as if it were a joystick [2]. A VR user relies on the integrity of the artificially generated stimuli in largely the same manner. Along the same lines of deception, Rafique and Sen-ching [13] developed a device which uses an infrared LED to jam and manipulate an HMD's tracking system, as well as an attack that manipulates the pose estimation by generating fake sync pulse.

As is the case in most emerging digital environments, work on cyber protection mechanisms for VR environments has mainly focused on risk assessment [5] and on preventing a security or privacy breach altogether through authentication. For

USENIX Symposium on Usable Privacy and Security (SOUPS) 2021.
August 8–10, 2021, Vancouver, B.C., Canada.

example, Mathis, Fawaz, and Khamis [10] developed an authentication method based on Guiard's kinematic chain model to address the issue of bystanders inferring user input. In their method, the non-dominant handheld controller interacts with a cube attached to it in VR, while the dominant controller enters a set of PIN numbers displayed on the cube while in VR. This two way handed interaction technique makes it difficult for a bystander to infer user input. They further added a deep learning architecture for time series classification to recognise a user's legitimate behavioural biometric PIN input. Lu et al. [8] proposed the use of hand motion authentication with eye-free interaction in VR systems, while Li et al. [6] proposed the use of brain signal biometric authentication.

Risk assessment and authentication-based prevention constitute a good start in VR cyber security, but do not address the need to detect whether a security breach has occurred when prevention fails, especially for new threats. Here, we take the first step towards intrusion detection in VR cyber security. We recognise that it would be counterproductive to develop a method that would detect an attack after it has caused impact on the user, or that would generate erroneous warnings of attacks (false positives) that would adversely affect a user's immersive experience. So, we evaluate our solution in terms of the time it takes to detect an attack and provide a simple suggestion on how to minimise false positives. To conduct our experimental evaluations, we have developed a virtual environment designed to carry out a variety of cyber attacks and observe their impact in different conditions and locomotion techniques. Specifically, the contributions of this paper are:

- We devise a GPU-based attack involving a malware that uses the OpenGL application programming interface (API) to overwhelm the GPU resources.
- We devise a network-based attack with the aim of causing visual, interaction, locomotion and auditory disruptions within a collaborative VR environment.
- We propose the first VR intrusion detection system which utilises machine learning to detect an ongoing attack and warn the user before its impact escalates.

2 Frame rate-oriented cyber attacks on user experience

From a system perspective, attacks in VR may relate to the GPU, sensors and displays, which jointly determine the output, input and computing efficiency of a VR system, i.e. its depth of information [14]. In this paper, we specifically focus on GPU as it directly determines the frame rate and through it the user experience. As proof of concept, we also target the frame rate through a network denial of service attack, especially as collaborative VR environments require network connection to function and typically network disruptions affect the frame rate. Both attacks aim at maliciously causing visual discomfort through disruption of user locomotion, interaction, audio and visuals, as framerate disruption is a known major factor leading to VR sickness [17].

2.1 The VR test-bed

We have developed out testbed in the Unity 3D game engine, using the XR Interaction Toolkit for the mechanics and various interaction techniques. We chose to use a Room-Scale XR Rig, which allows for six degrees of freedom movement. Locomotion was used as our main form of movement technique enabling the user to move around using the Joystick on the controllers. The environment was designed as an office space with objects placed at random locations, allowing the user to freely interact with them. To run our experiments, we used two VR-ready laptops (Alienware 51M: GeForce RTX 2060 and MSI GE66 Raider 10SGS: GeForce RTX 2070) and two VR headsets (Oculus Rift S and Oculus Quest 2).

2.2 Cyber Attack 1: GPU-based attack

Cyber attacks on GPUs have been previously shown, but not in relation to their impact on VR experience [12] [11] [15]. To implement our GPU-based attack, we created a malware using the OpenGL API, designed to assign long running tasks to the GPU, thereby affecting its availability to compute other graphics tasks. We created our malware in C++, delivered in the form of an executable file format (exe). Next, we created a series of images to load into the GPU via the malware. In our attack model, we assume that the target machine is already compromised and our payload is delivered using a post-exploitation tool, such as Meterpreter [1]. We configure the intensity of the malware's attack based on the size of the image (in our case, 542 KB, 1.06 MB, 1.82 MB and 3.13 MB). A simple batch script was developed to automate the attack. The operation of the malware was hidden from the user using lines of code;

```
HWND hWnd = GetConsoleWindow()
ShowWindow(hWnd, SW_HIDE)
GLCall(glfwHideWindow(windowone))
```

Figure 2 shows the environment in normal conditions where there is no visual discomfort (top) and during an attack (bottom), where the frame rate has dropped and extreme screen tearing is observed. Figure 1 further shows the mechanics of the impact of the missed and dropped frames on frame processing during an attack, which eventually leads to the VR system crashing. The default time window for a VR device to display images on the screen is 11 ms. VR uses reprojection, a technique that warps the rendered image before sending it to the display to correct for the head movement occurred after the rendering. This can reduce latency and help maintain frame rate. During the attack, frames take longer to render as a result of the GPU being overwhelmed. This results initially

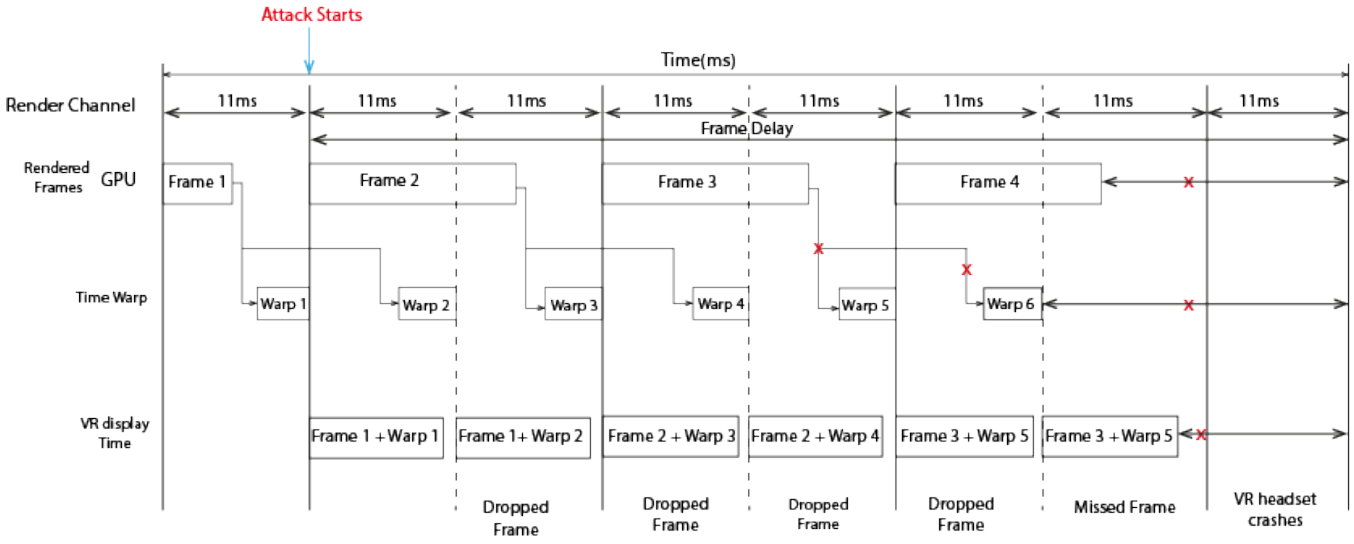


Figure 1: Missed and dropped frames' impact on frame processing during an attack

to dropped frames (which is experienced as judders and visual distortion). After a while, when warps also take too long, missed frames start occurring, which eventually lead to the VR system crashing.

2.3 Cyber attack 2: Network-based attack

This attack involves a script of continuous Internet Control Message Protocol (ICMP) echo requests to flood the target machine in a simple denial of service manner (Figure 3). The goal is to cause disruptions in a collaborative VR environment. To explore its impact, we created a VR collaborative environment which allows for two users to remotely connect using a server, set up with the Unity 3D package Photon (PUN2). For remote communication, we used Photon Voice 2. We created a batch script to initialize our attack in the form of a distributed ping flood. Our experiment shows disruption in engagement and interaction between the users as the frame rate drops dramatically. Figure 3 shows our VR environment where two users are actively interacting in normal conditions (top) and then during the attack (bottom), with the impact of the attack on the network traffic manifested as screen tearing and a drop in frame rate.

3 Intrusion detection for GPU-based attack in VR

Here, we propose a data-driven intrusion detection approach using unsupervised machine learning to learn what is normal for a particular VR system and warn the user of an attack when our chosen set of parameters monitored show departure

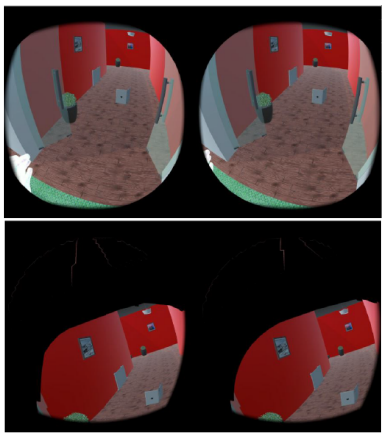


Figure 2: Extreme tearing during GPU-based attack

from normal behaviour. We have used Isolation Forest implemented in Python with the Scikit Learn library. Isolation Forest requires no prior assumptions regarding the distribution of feature values [7], which makes it attractive in new areas where there is limited knowledge of the parameters monitored. In our experiments shown here, we perform the GPU-based attack for different image sizes and using a two-second sliding window (generating a feature vector with a time series sequence transition between system indicators), where we monitor: average framerate, framerate standard deviation, average frametime, frametime standard deviation, and framerate entropy change.

Figure 4 shows the machine learning based classification versus the actual state (whether there is an attack or not). Note that shortly after the attack starts, the classification correctly

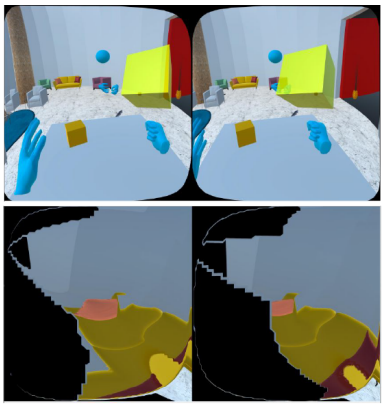


Figure 3: Extreme tearing during network-based attack

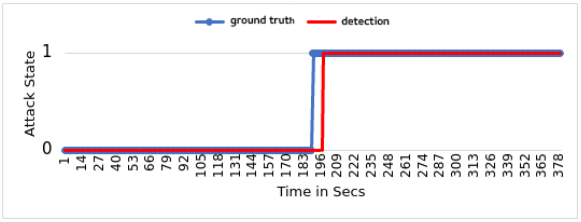
turns from normal to attack too. We also see that there are a small number of isolated false positives (attack states detected while the groundtruth is a normal state). This would be unacceptable for VR use, as triggering any warning to the user would affect adversely their immersive experience. For this reason, we propose to raise a flag and warn the user only when there are a number of consecutive attack data points detected. Empirically, we set this to two consecutive points (which represents three seconds of sliding window system state analysed by the machine learning process), which removed all false positives at a cost of one extra second of delay before raising a warning of a detected attack. Specifically, following the addition of the three consecutive detection rules, the detection latency is calculated as shown in Tables 1 and 2.

Table 1: Detection latency

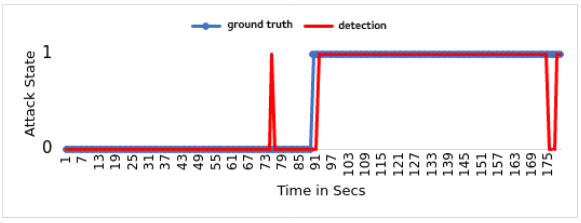
Attack 1	**Attack 2**	**Attack 3**	**Attack 4**
542 KB	1.06 MB	1.82 MB	3.13 MB
9 s	3 s	2 s	4 s

Table 2: Detection Accuracy

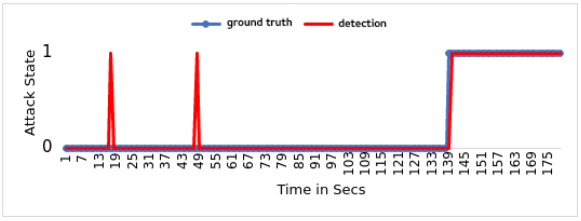
Attack 1	**Attack 2**	**Attack 3**	**Attack 4**
542 KB	1.06 MB	1.82 MB	3.13 MB
0.98	0.97	0.99	0.98



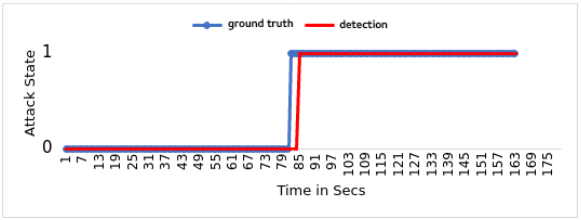
(a) Attack 1 (Image size: 542KB)



(b) Attack 2 (Image size: 1.06 MB)



(c) Attack 3 (Image size: 1.82 MB)



(d) Attack 4 (Image size: 3.13 MB)

Figure 4: Machine learning classification results for all attack intensities

4 Conclusions

We have described our work in progress towards studying the impact of cyber attacks in VR environments and developing intrusion detection mechanisms for warning users. The rationale is that the earlier the users are warned about a particular ongoing security breach, the more likely they will take action in time to protect themselves. In the particular example of a framerate-oriented attack, a reasonable action would be to simply pause the session and take off the HMD, so as to prevent VR sickness from developing. However, this is an assumption and research is needed specifically to establish how a user would and should behave upon recognition of an attack. In addition, different attacks leave different cyber traces and affect different system parameters. We have provided a first example of an intrusion detection mechanism to serve as proof of concept for further mechanisms, exploring different attacks and the associated parameters to be monitored. In our future work, we plan to carry out a large-scale user evaluation of the proposed intrusion detection mechanism as the logical next step to investigate its objective usefulness and effectiveness in user experience.

References

[1] Monika Agarwal and Abhinav Singh. *Metasploit penetration testing cookbook*. Packt Publishing Ltd, 2013.

[2] Jahshan Bhatti and Todd E Humphreys. Hostile control of ships via false gps signals: Demonstration and detection. *NAVIGATION, Journal of the Institute of Navigation*, 64(1):51–66, 2017.

[3] Doug A Bowman and Ryan P McMahan. Virtual reality: how much immersion is enough? *Computer*, 40(7):36–43, 2007.

[4] Peter Casey, Ibrahim Baggili, and Ananya Yarramreddy. Immersive virtual reality attacks and the human joystick. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[5] Aniket Gulhane, Akhil Vyas, Reshmi Mitra, Roland Oruche, Gabriela Hoefer, Samaikya Valluripally, Prasad Calyam, and Khaza Anuarul Hoque. Security, privacy and safety risk assessment for virtual reality learning environment applications. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–9. IEEE, 2019.

[6] Sukun Li, Sonal Savaliya, Leonard Marino, Avery M Leider, and Charles C Tappert. Brain signal authentication for human-computer interaction in virtual reality. In *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 115–120. IEEE, 2019.

[7] F.T. Liu, K.M. Ting, and Z. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[8] Yujun Lu, BoYu Gao, Jinyi Long, and Jian Weng. Hand motion with eyes-free interaction for authentication in virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 715–716. IEEE, 2020.

[9] V Market. Virtual Reality Market with COVID-19 Impact Analysis by Offering (Hardware and Software), Technology, Device Type (Head-Mounted Display, Gesture-Tracking Device), Application (Consumer, Commercial, Enterprise, Healthcare) and Geography - Global Forecast to 2025, 2020.

[10] Florian Mathis, Hassan Ismail Fawaz, and Mohamed Khamis. Knowledge-driven biometric authentication in virtual reality. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2020.

[11] Sparsh Mittal, SB Abhinaya, Manish Reddy, and Irfan Ali. A survey of techniques for improving security of gpus. *Journal of Hardware and Systems Security*, 2(3):266–285, 2018.

[12] Hoda Naghibijouybari, Ajaya Neupane, Zhiyun Qian, and Nael Abu-Ghazaleh. Rendered insecure: Gpu side channel attacks are practical. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2139–2153, 2018.

[13] Muhammad Usman Rafique and S Cheung Sen-ching. Tracking attacks on virtual reality systems. *IEEE Consumer Electronics Magazine*, 9(2):41–46, 2020.

[14] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992.

[15] Giorgos Vasiliadis, Michalis Polychronakis, and Sotiris Ioannidis. GPU-assisted malware. *International Journal of Information Security*, 14(3):289–297, 2015.

[16] Séamas Weech, Sophie Kenny, and Michael Barnett-Cowan. Presence and cybersickness in virtual reality are negatively related: a review. *Frontiers in psychology*, 10:158, 2019.

[17] David J Zielinski, Hrishikesh M Rao, Mark A Sommer, and Regis Kopper. Exploring the effects of image persistence in low frame rate virtual environments. In *2015 IEEE Virtual Reality (VR)*, pages 19–26. IEEE, 2015.