Reversing the Virtual Maze: An Overview of the Technical and Methodological Challenges for Metaverse App Analysis

Alfonso Rodriguez Barredo-Valenzuela[†‡], Sergio Pastrana Portillo[‡], Narseo Vallina-Rodriguez[†]
Guillermo Suarez-Tangil[†]
[†]*IMDEA Networks Institute*, [‡]*Universidad Carlos III de Madrid*

***Abstract*—The Metaverse is a virtual world that is becoming increasingly popular. Recent technological advances, such as head-mounted displays and novel sensors, have allowed for the harmonious integration of the virtual world into our reality. This integration is transforming how we interact with our environment and with each other and offers endless entertainment, social, and business opportunities. Since the technology is in its early stages and far from market consolidation, there has been a notable proliferation of new platforms and devices resulting in a highly heterogeneous ecosystem.**

Much like with Smartphones, popular platforms allow for the installation of third-party applications, accessible through online marketplaces. However, the Metaverse aims to enable seamless coordination between virtual environments. As a result, developers face the pressure of being present on a wide range of platforms and rely on cross-platform development frameworks that add yet another layer of complexity to the stack. This complexity, together with the fact that XR headsets are equipped with an arsenal of disrupting new sensors, has the potential to pose new risks to the security and privacy of their users. Although progress has been made in identifying risks in this ecosystem, there is still a significant gap in methodologies and techniques for studying actionable risks in popular headsets.

In this work, we present a vision — based on experience — for application analysis that considers various ecosystem components and highlights challenges to address emerging threats effectively.

I. Introduction

The Metaverse is the composition of three technologies that comprise the so-called Extended Reality (XR), namely Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR). XR applications extend far beyond the gaming experience for which they were characterized in their early days. Health [61], Professional Training [23], Therapy [38] or Culture [1] are just a few examples. The growing interest in this digital world has an estimated financial impact of $1.5 trillion by 2030 [55]. As a result, major tech companies whose primary business is not directly related to the Metaverse are investing in this domain and developing their own VR applications, e.g., Mozilla (FirefoxVR [14]), Google (YouTubeVR [60]), and Netflix [2].

As a disruptive technology, the potential of the Metaverse to enhance user entertainment experiences is accompanied by important security, privacy, and safety concerns. Yet, the Metaverse does not have a wide literature about such concerns as in other domains (e.g., Smartphones). The technology is in its early stages and so are analysis methodologies, techniques, and tools for conducting appropriate assessments. Furthermore, the absence of market consolidation has led to a lack of widely adopted standards. Consequently, there is a plethora of platforms, markets, and devices utilizing their tailored solutions, resulting in a complex and heterogeneous market and technological landscape. This poses one of the major challenges for achieving a large-scale analysis of applications, exacerbated by the complexity introduced by the unique characteristics of Metaverse technologies formed by sensors, controllers, or interactions in a 3D environment. While some attack surfaces and threats may be similar to those found in other fields, the emergence of new XR-specific operating systems, new hardware (e.g., sensors), and new functionalities leads to new threats that should be addressed carefully. Examples include room fingerprint, virtual asset theft, or virtual harassment.

Garrido et al. [15] conducted recently a longitudinal analysis using existing literature about security and privacy in the Metaverse, summarizing the different threads and possible defenses. They assess potential attacks that users and devices may face, along with suitable protections that could help prevent such disruptive activities. Due to the ongoing evolution of the Metaverse, as prior work shows, we lack mature methodologies to detect and mitigate risks. There are other studies addressing security and privacy in the Metaverse [4], [13], [31], [44], [52], [56], [62], which, besides technical issues, they also discuss ethical, social, or economic problems. However, these works offer an incomplete overview of the threat landscape, which does not cover the key technological and methodological challenges that need to be addressed to procure practical security risks over real-world applications.

Overall, we note a substantial knowledge gap in methods and techniques aimed at analyzing XR applications, worsened by the primitive state of these techniques. Consequently, based on our own experience and a complete literature review, in this paper, we conduct a qualitative study to understand existing methodological and technical challenges preventing the analysis of the Metaverse ecosystem, which would enable researchers to evaluate XR applications at scale. Specifically, we address the following research questions:

- **RQ1:** What are the emerging security and privacy risks that current threat models fail to adequately capture?

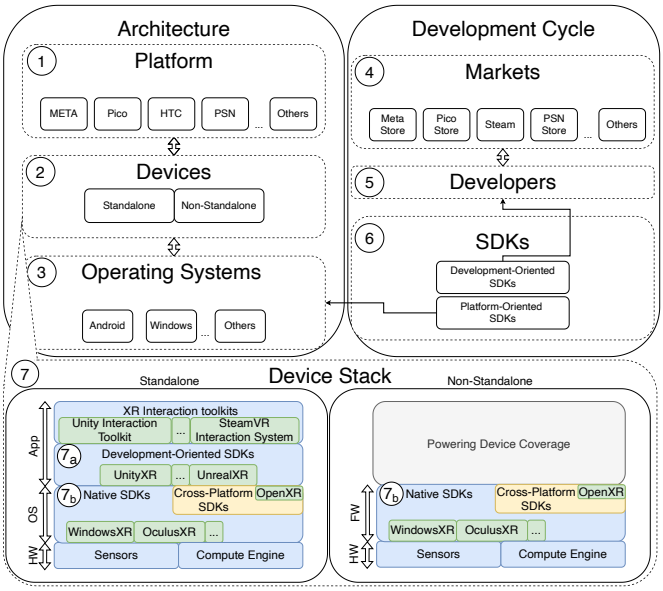

Fig. 1: Metaverse ecosystem.

- **RQ2:** What techniques can be adopted from other domains to automate the analysis of Metaverse applications?
- **RQ3:** What are the key challenges in developing robust and scalable techniques for the automatic analysis of Metaverse applications?

Our study, which builds upon previous work from Garrido et al. [15], goes a step further by not only considering the potential S&P threats but by also delving into concrete issues related to the empirical analysis of applications in the Metaverse. Thus, our objective is to gain a comprehensive understanding of the entire ecosystem, and then pinpoint specific issues and obstacles associated with analyzing XR software, either with dynamic or static analysis methods. Concretely, we address the following points:

> **1st:** We study and describe the current landscape of the Metaverse, elucidating the software ecosystem, application markets, and devices while highlighting differences in application runtime environments (see §II-A & §II-B).
> **2nd:** We analyze the technological gaps (RQ1) in existing literature on Metaverse application analysis (see §II-C).
> **3rd:** We identify and discuss the techniques (RQ2) and challenges (RQ3) that researchers must overcome to analyze XR applications at scale (see §IV, §V, & §III).

Our findings pinpoint the need to provide tools that deal with the heterogeneity of the Metaverse. This is, to a great extent, due to a critical dependency on the OS (mostly Android) and Device Stack (i.e., platform-specific libraries) for the analysis of the whole functionality of XR apps. We also demonstrate the existence of a research gap, whereby previous work has pinpointed potential risks without considering the difficulty of detecting these while analyzing apps. We note the importance of component identification and the need for analysis of native code, among other challenges.

II. Background

As XR (Extended Reality) gains popularity among developers and users, more business opportunities and use cases are emerging. This has led to a plethora of new platforms, tools, stores, and devices, making the whole XR ecosystem (branded as the Metaverse) particularly complex. In this section, we examine the architecture (§II-A) supporting XR technology and its development cycle (§II-B) as summarized in Figure 1. We then identify the knowledge gap (§II-C) required to provide user privacy and security.

A. Architecture

We describe the architecture of the Metaverse by identifying the platforms that underpin XR (see step ① in Figure 1), type of devices ②, and Operating Systems (OSs) they run ③.

① **XR platforms**: The two most popular XR platforms nowadays are Meta and PlayStation [27]. Meta has become the most prominent XR platform by developing the Android-based Meta Quest OS and releasing several popular devices, including its latest headset, the Meta Quest 3 [29]. PlayStation already offers PSVR2 [40] to play VR games over PlayStation 5 and many other platforms are gaining market presence, such as Pico [37], HTC [19], Valve [50], or Apple [51].

② **Devices**: To access the Metaverse, users require an XR headset, also called Head Mounted Display (HMD). At the time of this writing (April'24), at least 244 headsets have already been introduced into the market over the years [54]. But not all platforms follow the same architecture. Depending on where the XR application is executed, we distinguish between two types of headsets. On the one hand, `Non-Standalone` headsets require an external Powering Device (PD), such as a PC, Console, or Smartphone, to execute the XR application. On the other hand, `Standalone` devices have their own OS installed in the HMD, and are capable of locally installing and running apps (although it is also possible for them to be powered by an external device for performance purposes). PSVR2 and Meta Quest 3 are examples of Non-Standalone and Standalone devices respectively. Since 2022, 61% of the handsets on the market are Standalone, making it the predominating architecture.

③ **OSs**: The emergence of a vast array of Standalone devices has led to the proliferation of new dedicated OSes over time, as Figure 2 [54] shows together with their market share. The two key OSes in the Metaverse are:

- **Android:** Approximately 95% of Standalone devices run an Android-based OS. Applications for these devices are compiled into APK (Android Package) format.
- **Windows:** Within the gaming space as PC platforms predominantly operate on Windows, Metaverse gaming applications are primarily distributed as PE (Portable Executable) files through market platforms such as Steam or Epic Games.

Other popular OSs are the PlayStation OS for PSVR2 [40], the Vision OS for Vision Pro [51] headsets from Apple, and the Windows Holographic OS for HoloLens 2 [18] from

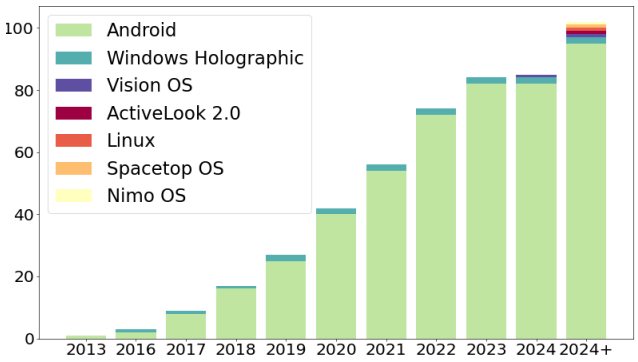

Fig. 2: Market share of OSs in Standalone devices over time. (2024+ is a projection).

Microsoft. However, less popular ones such as Nimo OS, ActiveLook, and Spacetop OS, as well as variants of the widely recognized Linux, are expected to emerge. The heterogeneity of platforms and technical solutions poses a challenge to carrying out application analysis since it demands customized or dedicated tools for each different type of binary and/or API.

B. Development Cycle

We study the development cycle of Metaverse apps by analyzing the way apps are distributed, the tools and apps available to developers, and how they integrate altogether in the stack (steps ④ to ⑦ in Figure 1).

④ **Markets**: As in other ecosystems, applications are accessible to users through Marketplaces. Some platforms operate their own markets such as the Meta Store [30] or Apple's App Store [5]. Others, like Steam, are third-party markets acting as application hubs for multiple platforms. Interestingly, "SideQuest," is a market where developers can publish beta versions of their applications to gather early feedback, allowing beta testers to test potentially buggy apps, despite the risks involved in installing software from unknown sources. Some apps initially released on this market then became popular applications distributed through platforms such as Steam and other platform-specific stores (e.g., Pavlov VR [35]).

⑤ **Developers,** ⑥ **XR SDKs, and** ⑦ **Device Stack**: Metaverse developers can leverage a wide range of third-party tools and libraries to assist their application development process, known as Extended Reality Software Development Kits (XR SDKs). We identify two types of XR SDKs: `Platform SDKs` like native libraries and `Development SDKs`. Native SDKs consist of tools provided by each platform to allow developers to build applications in their specific ecosystems, enabling them to create native applications that interact with the specific hardware of their devices. These run in user-space (see step ⑦$_a$ in Figure 1). Native SDKs are embedded either in the OS for Standalone devices or as part of the firmware in Non-Standalone devices as in step ⑦$_b$. Although creating native applications is possible (e.g., using Android Studio), vendors such as Meta [28], Pico [36], or HTC [20] recommend the use of the same third-party development frameworks. Development-oriented SDKs are used by game engines, such as Unity or Unreal Engine, which embed these

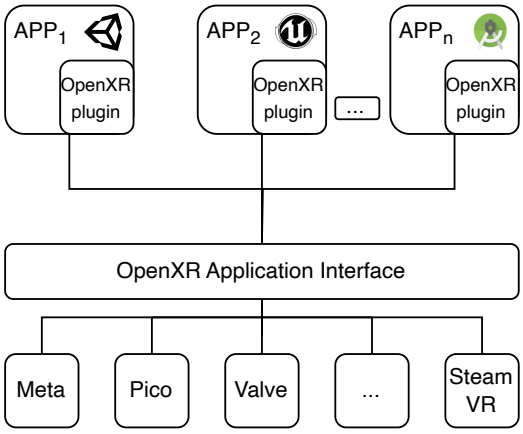

Fig. 3: OpenXR SDK Application Interface. (Engines left to right: Unity, Unreal Engine and Android Studio).

in their Integrated Development Environments (IDEs) to ease the development of games.

Cross-platform XR SDKs like OpenXR (the industry standard by Khronos Group) allow developers to build portable cross-platform apps. Here, XR developers write code once and deploy it across different platforms, similar to React Native in the mobile world. XR developers resort to native libraries to efficiently reuse such code across disparate platforms. Libraries are compiled to be executed in the targeted architecture without additional layers like virtual machines (e.g., Java) or interpreters (e.g., Python). As the main functionality resides in these libraries, the application code primarily acts as a wrapper. Supported by 65 different platforms like Meta or Pico and Game engines like Unity or Unreal Engine, OpenXR abstracts away platform-specific APIs through platform-agnostic SDKs dubbed OpenXR plugins as depicted in Figure 3.

C. Threat Model

Due to the blurred boundaries between real and virtual reality in the Metaverse, this new paradigm introduces new risks related to user-to-user interactions within the XR environment and the cyberphysical environment. In fact, the realism of these ecosystems possesses a greater potential for data and user monitoring by privacy-intrusive practices, but also for concerns about user safety and security.

Prior work has recently identified VR privacy threats [15], categorizing adversaries into four generic groups: hardware, client, server, and user adversaries. This effort also identified five overarching risks in the Metaverse: data collection, user identification, profiling, and virtual harassment. We build on these risks and threats to present a complementary vision based on experience, focusing on technical challenges that hinder the use of state-of-the-art software analysis methods in the Metaverse. We classify such threats as follows:

- **Privacy threats** related to the collection of sensitive user and behavioral data for ⓐ **User fingerprinting and profiling**; comprising the analysis of user data to determine individual's behavior, preferences, and demographics to craft personalized experiences and targeted

content including advertisements; ⓑ **Metaverse and room fingerprinting:** for uniquely identifying or characterizing the physical space of a user based on its specific features (e.g., layout, and other environmental factors) that can be used to infer information about users, including social interactions on the Metaverse; and ⓒ **Data leaks, data deletion, and data theft:** Unauthorized disclosure of data that can be abused to impersonate individuals, or unauthorized data removal.

- **Security threats** introduced by the malicious behavior of both software, malicious actors, and other metaverse users, including: ⓓ **Malicious and deceptive actors**, including third-party SDKs, malicious users compromising the regular operation of the Metaverse app and hardware, or co-located devices and software developers [16]; ⓔ **Device hijacking and asset theft,** which involves unauthorized takeover or control of a computing device or Metaverse asset by an external actor; ⓕ **Lateral movement attacks,** involving progressive movement through the network and the Metaverse environment to gain access to different systems or resources (e.g., PDs); and ⓖ **Virtual extortion, scam, and harassment**, which includes coercion or blackmailing of individuals or organizations in a VR context, or deception of individuals or organizations for financial or other malicious purposes. In fact, the latter category has a wide range of abusive and criminal behaviors such as cyberbullying, fraud, Virtual Stalking, or Virtual Sexual Harassment [41].

Answering RQ1, we perform a literature review to identify existing defenses to the threats above underscoring those aspects lacking in previous approaches. We depart from the 2023 systematization of knowledge done by Garrido et al. [15], which defines four possible layers where attackers could act (Hardware, Client, Server, and User), and perform a snowball identification of other recent works between 2023 and 2024. In total, we review 45 works. Prior work mainly focuses on threats around "user fingerprint" [33], [34], [45] and "data theft" [6], [9], [57] the psychological effects of using XR applications and virtual harassment [43], [59], or law enforcement in the Metaverse [41]. However, most of the proposed attacks and defenses do not validate their theories and implementations in real-world applications limiting the extensibility of their takeaways. We only identify eight studies measuring threats or experimenting with real applications, whereby they only focus on VR settings [6], [22], [25], [32], [46], [46], [49], [53]. Only Trimananda et al. [49] attempt to empirically analyze the privacy threats of VR applications by inspecting the network traffic of 140 real applications, with limited visibility into the traffic payloads and threat coverage.

Knowledge gap: The emerging Metaverse presents novel security challenges that are poorly understood and analyzed. The research literature from the security and privacy research communities is still limited in this domain. Current application analysis techniques are inadequate to empirically detect and mitigate emerging threats in Metaverse software — either from malicious developers or users,— in the face of the rapid development of XR products. While some works have tackled critical aspects like user fingerprint, or data theft, there are important flaws that prevent their application at scale. Other key threats like room fingerprinting have not been addressed yet in real-world applications, mostly due to a lack of appropriate methodologies to conduct the analyses. We next identify and provide an informed analysis of the most significant technical and methodological challenges that the community must address to effectively assess the privacy and security threats of real-world Metaverse applications.

Having answered RQ1, we address RQ2 and RQ3 in what follows. For RQ2, we investigate the applicability of analysis techniques established in other domains such as smartphones or desktop malware analysis, in the context of Metaverse platforms. For RQ3, we leverage the community expertise in successfully applying these techniques to identify privacy and security threats for determining existing limitations when adopting them in the analysis of XR applications. In particular, we set out to study the challenges around market crawling (§III), static analysis (§IV), and dynamic analysis (§V).

III. Market Crawls

As mobile app stores, Metaverse marketplaces are not databases or repositories of apps, but rather user-interface-based platforms for purchasing applications. This makes it difficult to automate the process of gathering apps for testing them at scale; collecting Metaverse applications requires dedicated crawling methods aware of the market and Metaverse platform architectures discussed in the previous section. While this is not a research problem per se, the inability to collect software samples at scale limits our ability to gain an accurate picture of the Metaverse ecosystem, software engineering approaches, and their associated threats.

As such, we identify three main installation paradigms and discuss approaches to address data-gathering limitations.

1) The Metaverse application is installed in a PC or a gaming console (like PS5) and the headset acts as a visor/controller (e.g., Steam) since is the PC the one running the XR application.
2) Standalone PC-based marketplaces that offer a catalog of applications for direct installation into the device (headset), which then runs the app (e.g., SideQuest).
3) The marketplace is installed on the platform headset so the app is directly downloaded, installed, and executed in the headset itself.

To acquire samples, one can automatically interact with market UIs **CH-MC-1**. The challenge here lies in facilitating the technical means so that the UI exerciser can navigate

effectively through the market. This requires analyzing and modeling the variety of marketplaces available, as well as their interface and installation process. An alternative approach is **CH-MC-2 Crawling apps through APIs**. This requires groundwork reversing device-market interactions to identify the network services supporting the installation of apps through the devices. It also requires bypassing checks on the server side to directly tap into these APIs.

IV. Static Analysis Challenges

Static analysis examines compiled apps to identify properties without running the application by inspecting (or reversing) compiled objects. While useful for finding basic and potential security and privacy issues, we see significant limitations when adopting many existing tools and pipelines to the context of Metaverse apps.

CH-SA-1 Understanding platforms' architectures: One of the main challenges in the static analysis of Metaverse applications is the heterogeneity of the existing platforms, many with their own different hardware and software architectures. Metaverse platforms customize popular operating systems like Android to distinguish their XR platform and user experience from competitors. As a result, we believe that there is a great opportunity to adapt and enhance existing Analysis pipelines for mobile app analysis to the Metaverse. However, these OS modifications translate into new APIs to allow programmatic access to new sensors and peripherals, incorporating new modules to render 3D video and sound, or to enable new permission and security controls to prevent unauthorized access to data. Understanding how the device behaves and interacts with the different OS modules and applications requires gaining a good understanding of platform fundamentals, and how these can be incorporated in static analysis tools to assess the security and privacy risks of Metaverse applications across platforms.

CH-SA-2 Modeling the Application Lifecycle: Understanding the abstraction layer implemented by third-party SDKs is important to model the lifecycle of Metaverse apps. Such abstraction enables essential methods to map out the code flow and understand how different components interact. Cross-platform development frameworks like OpenXR have become the de-facto standard for metaverse app development. OpenXR offers a platform-independent abstraction for metaverse development that can ease the process of understanding app development and its lifecycle. However, not every application developer follows this approach, forcing the research community to adapt existing static analysis methods to the increasing number of software development processes, paradigms, and platforms. The large number of Metaverse platforms and cross-platform development SDKs that are available today make this a daunting task until the market becomes stabilized.

CH-SA-3 Code Analysis: Code analysis involves extracting signals from the compiled program to examine the code's structure, syntax, and semantics. These signals are later processed to model program behaviors and infer potential threats such as personal data dissemination, detecting the use of third-party libraries, or insecure software development practices. As we have introduced above, we believe that the arsenal of mobile app analysis pipelines existing today can be enhanced and adapted to the Metaverse. For instance, tools and methodologies such as LibRadar [26] can be enriched with new signatures to identify both native and third-party SDKs in the Metaverse. Similarly, taint analysis can detect data leakage in apps by tracing and identifying potentially vulnerable data flows. However, existing taint analysis techniques can not be directly adopted to analyze Metaverse apps due to their failure to model software component lifecycles effectively in different environments. For example, FlowDroid [7] — one of the most consolidated frameworks to perform taint analysis in Android — has proven to fail when applied to other Android-based OS such as in WearOS where the lifecycle of software components is not comprehensively modeled [47]. Yet, one main limitation is achieving code coverage, as existing mechanisms might not effectively handle the diversity of platforms with proprietary features and the full scope of Metaverse apps formed by native languages, high-level programming languages such as Java, and cross-platform SDKs. In fact, understanding machine code in native libraries is a complex process as the analysis must be conducted at the assembly code level using tools such as IDA Pro. The failure stems from the lack of understanding of how platform-oriented asynchronous parts work together as per CH-SA-2, as well as handling developers and SDKs relying on code obfuscation methods to protect their intellectual property or business interests. As of today, no scalable static analysis pipeline for the Metaverse exists.

CH-SA-4 Vulnerability Identification and Supply Chain Analysis: As a consequence of the heterogeneity discussed in CH-SA-1, issues may emerge in apps created using SDKs or configurations that, as part of their dependencies, use a specific vulnerable component on a targeted platform. For instance, for certificate validation in VR applications for Meta Quest 2, Unity utilizes *mbed TLS*. Recently, an Integer Overflow vulnerability has been discovered for TLS 2.x before 2.28.7 and 3.x before 3.5.2, identified as CVE-2024-23775 [11]. In contrast, Unreal Engine apps would not be vulnerable in this particular case since they use *OpenSSL* for this functionality. Therefore, it is not enough to understand a single technology or architecture, and obtaining a comprehensive Software Bill of Materials (SBOM) of the Metaverse is challenging as it is necessary to create high-quality knowledge bases containing SDK signals for their detection, preferably at the version level. This is, as of today, an open research challenge in the general field of software analysis as extensively discussed by Jean Camp and Vafa Andalibi [10].

V. Dynamic Analysis Challenges

Dynamic analysis runs the software and monitors its actions (memory usage, method invocation in runtime, network traffic, etc.), thus validating identified issues from static analysis and reducing false positives. However, Metaverse's device owners have limited access and control over certain settings and

functionalities compared to the administrator. The lack of permissions may lead to a limitation when trying to access some resources of the OS during the runtime, preventing researchers from being able to monitor the state of the application. Dynamic analysis may also miss issues due to constraints in simulating all possible scenarios within an interactive XR environment. This is because certain functionalities only manifest under specific conditions, and the number of states in such apps is extremely high. Furthermore, some events are hard to emulate or monitor, like the interaction with third users which also raises ethical concerns. Next, we delve into these limitations by addressing the following challenges in detail.

CH-DA-1 Instrumentation of the Metaverse: Instrumentation requires having tackled CH-SA-1 to understand the specific architecture. It can be done at three different layers, at the network, OS, and app level.

The instrumentation of the operating system could provide a comprehensive view of application behavior, including low-level (privileged) events, system calls, and data interactions during execution. An additional challenge would be to establish an instrumentation approach that generalizes for the different types of OSs specific to Metaverse platforms.

Instrumenting the application itself provides an alternative to instrumenting the OS. This is a widely used practice in smartphones [12], dubbed in-app reference monitor, where a common approach is to inject a library directly into the program and subsequently repackage the app with enhanced functionality (e.g., using Frida Gadgets). In the Metaverse, however, this option would modify the signature. Some Metaverse applications have signature-based app-to-device authentication, in addition to app-to-server authentication mechanisms, which prevent modified applications from connecting to the backend. While these signature checks could be bypassed, they require ad-hoc efforts that do not scale. Another approach is to run an external program that injects code at runtime (e.g., using Frida Agents). However, this requires rooting Metaverse devices, which is still an open issue. Root permissions provide access to protected system files, enable modifications to system settings, facilitate the installation and execution of custom software, and grant access to system resources such as system services or network interfaces.

Unfortunately, most modern games include anti-cheat software that prevents other programs from debugging, patching instructions, or instrumenting the program [24]. Anti-cheat engines are ported to the Metaverse and have the risk of thwarting any instrumentation introduced by security researchers to monitor the behavior of apps. This brings tension between building anti-instrumentation techniques to prevent malicious actors and allowing researchers to introduce in-app reference monitors. Consequently, the security industry and research community need a solution for the instrumentation of the Metaverse, to enable an appropriate and confident analysis of the behavior of applications at scale.

At the network level, a key challenge arises from the ability to collect Metavese apps' traffic with cloud endpoints, particularly for encrypted TLS flows. This challenge resembles the ones faced when analyzing Amazon Alexa and Google Echo voice apps [17], but without informative cloud-based test consoles or well-defined interaction APIs as those exposed by Amazon or Google that enable the capture of traffic traces [8]. Due to the aforementioned instrumentation challenges arising from the closed nature of Metaverse platforms, using Person-In-the-Middle proxies is not always possible as the trusted root store of these devices can not be modified [39]. This is aggravated by the challenges of modifying applications' execution with dynamic analysis techniques such as Frida. If this challenge is solved, TLS traffic analysis serves as a compelling smoking gun for revealing PII leaks on apps [42] — with ramifications for advertising and software engineering practices, — to unveil program dependencies on cloud services and even to assess developers' adherence to the correct use of security protocols like TLS.

As discussed in §II-C, the only existing research paper on the presence of third-party endpoints in the Metaverse to date [49] is limited in scope due to these challenges. The study primarily focused on endpoint analysis owing to the inherent difficulty in accessing the payload of encrypted traffic.

CH-DA-2 Automation of XR applications: Without automation software to interact with Metaverse applications, testing requires manual controls. This significantly limits large-scale analysis, as exploring every possible path within a dynamic Metaverse app becomes nearly impossible through manual navigation. Manual interaction impedes the adoption of a multi-device approach and limits the parallelization of tasks, thus undermining the scalability of the analysis. Automating UI interactions within the Metaverse is way harder than for desktop and mobile applications, which are mostly in 2D graphics, have limited states, and their environments typically remain static over time. Analyzing XR experiences, however, requires a different approach. Some games offer content that can last for 20, 40, or even more than 70 hours of gameplay. These games have a dynamic environment that can trigger different functionalities based on multiple external inputs (e.g., sensors or other remote players). In XR experiences, the movement range is continuous, and interactions with objects include not only hit-based actions but also pressing, grabbing, dragging, and rotating. Additionally, interactions with the environment may vary depending on factors such as user height, speed, acceleration, angle, distance, direction, gesture, or spatial movement. Finally, XR applications inherently come with functionalities derived from their novel sensors and capabilities, often implemented in cross-platform libraries (e.g., body tracing) that may have different performance or observable behaviors depending on the platform they are running on. Altogether, we require smart and scalable approaches to fuzz Metaverse applications that consider these challenges.

CH-DA-3 Detecting and Moderating Harmful Behaviors on the Metaverse: Metaverse applications extract critical information from sensor data. This information might be used for legitimate purposes like user authentication or health

monitoring, and also for secondary purposes such as user profiling for advertising. This information can also be used maliciously, e.g., by malware. A key challenge is to detect and distinguish between such malicious behavior from that of legitimate applications since relying solely on dynamic traces to determine whether an application is misbehaving can be difficult.

Monitoring the activity of XR applications is also needed to detect harmful behaviors from users (e.g., cyber-stalking or sextortion) who do not abide by the accepted policy or terms of use of the app. Understanding such violations is critical, so as to enforce appropriate actions (e.g., banning users). Different from automatic content moderation in traditional social networks, which often rely on advanced natural language understanding techniques, monitoring XR apps at scale is challenging. Indeed, there is a plethora of wearable devices and sensors that can be added to headsets to enhance VR experiences (e.g., VR haptic feedback vests or VR locomotion platforms), each of them providing sensorial data in different formats, which require dedicated monitoring techniques.

Consequently, auditing the behavior of apps and users to look for harmful practices requires dedicated tools and techniques that, unfortunately, are still lacking in the Metaverse.

VI. Discussions and Future Work

After defining the challenges that must be overcome to analyze Metaverse applications at scale, we present our work's most relevant takeaways and limitations, and discuss the roadmap to enable research on security and privacy aspects of the Metaverse.

A. Key Observations

Gap Analysis: The Metaverse is presenting new security challenges that are not well understood or adequately analyzed, and we see that the existing research on security and privacy in this area is still limited. Current techniques for analyzing applications are not sufficient for identifying and addressing emerging threats in Metaverse software, whether they originate from malicious developers or users, due to the fast pace of development. While some XR studies have looked into important issues such as user fingerprinting [33], [34], [45] and data theft [6], [9], [57], there are significant limitations that make them difficult to apply on a large scale. For instance, several studies have developed mock applications to validate their proof of concept (PoC) [33], [34], [48], [58]. While PoCs are useful for positioning the applicability of threats, they conduct assessments within non-realistic environments, potentially limiting the extendability of findings to real scenarios. This is the case of major threats like room fingerprinting, which have not been addressed in practical situations. While room fingerprinting may pose a threat in lab conditions, it remains unclear if attackers can harm privacy when tracking users at scale.

Limited work has validated their threat models in real-world applications leveraging static and dynamic analysis of the Metaverse applications. Trimananda et al. [49] evaluated data from 140 real applications. However, their analysis is restricted to a handful of apps all from a single platform. More importantly, the interactions with applications to gather network traffic were manual, and they did not deal with traffic encryption or server authentication, steps required for some applications to cover the complete network scenario.

Overall, we see a lack of methods and tools for systematic code analysis of Metaverse apps. Our work is the first to distill the challenges that need to be overcome to produce such methods and tools.

The diverse metaverse ecosystem challenges the security analysis: Our work highlights the challenge of analyzing security in the metaverse due to the vast array of VR, AR, and MR solutions. Standalone and non-standalone (i.e., dependant on third-party PDs such as PCs, consoles, or smartphones) devices, coupled with the multitude of OSs (mostly Android-based), marketplaces, and development frameworks, create a fragmented landscape. This fragmentation hinders comprehensive security analysis, difficulting in the discovery of vulnerabilities and bad practices, or the analysis of supply chain integrity efforts (CH-SA-4). Therefore, the OS ③ and the software stack ⑦ are critical steps in the application analysis since it is dependent on the software architecture and App-OS interactions. We also lack a dedicated security certification for such devices, akin to what exists for IoT devices (such as ioXt [21]) since it could aid vendors and developers in effectively mitigating the inherent risks of the Metaverse, and also to comply with regulations such as the EU Cyber-Resilience Act, which will soon come into force.[1]

Sample collection lacks of automated approaches: Prior work use manual approaches to collect Metaverse applications. This approach is not scalable and hinders the reproducibility of results. We propose two different approaches to address this limitation. First, we suggest using UI automators, such as PythonAutoGui [3]. Second, we discuss the possibility of relying on the APIs of various marketplaces, despite the additional challenges of reversing network protocols and dealing with server authentication or encryption of communications. Both methodologies face a common problem: marketplaces are independent solutions. Therefore, each store may require tailored tools to retrieve their data, although the process of developing such tools is similar, much like in Android but with a more fragmented market.

Several technical challenges must be overcome to carry out a comprehensive analysis of the Metaverse: In this paper, we have identified challenges and barriers to the adoption of existing methods to perform static and dynamic analysis of third-party software in the Metaverse. One prevalent issue we find is that proprietary solutions hinder the adoption of well-established scientific methods and tools that have flourished in other technologies, such as in the analysis of smartphone apps. On the one hand, the emergence of new sensors and functionalities, along with the importance of native languages in the development of cross-platform applications (CH-SA-

[1] https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act

3), has emphasized the need to understand new architectures, identify different components (CH-SA-1), and comprehend the lifecycle of XR applications (CH-SA-2). This understanding enables static analysis tools to accurately model when and how data is processed and passed between components. This is critical for identifying potential vulnerabilities and data leaks (CH-SA-4) and allows dynamic analysis tools to provide a comprehensive view of the application's behavior (CH-DA-3).

On the other hand, the inability to gain root access limits the visibility into the device's operating system, restricts access to protected resources, and prevents the installation and execution of custom software, making it more difficult to monitor applications (CH-DA-1). Another significant issue is the difficulty of covering every possible state in XR applications (CH-DA-2). The automation of interactions in these applications is particularly challenging due to the extensive range of states, but necessary to trigger evasive behaviors.

We identify that some challenges are contingent on others: By reflecting on the dependencies across challenges, we can identify cornerstone issues. For instance, understanding the architectures (CH-SA-1), analyzing external components (CH-SA-4), and automatically downloading applications (CH-MC-1) are three methodological and technological challenges of analysis that are required to overcome all other challenges. Likewise, an effective instrumentation mechanism (CH-DA-1), and monkey testers for Metaverse apps are needed to enable dynamic testing approaches (CH-DA-2) and data gathering tools (CH-MC-2).

B. Limitations

While this study serves to provide valuable insights into the challenges behind Metaverse app analysis, our work has also limitations. This being an experienced paper, our insights and knowledge are biased toward the platforms we have been using the most, i.e., the Meta Quest 2 ecosystem. To mitigate the impact of this bias, we have complemented our understanding of the ecosystem with an exhaustive literature review, as well as with the research expertise we have in the Android environments.

Another limitation of our work is the lack of comprehensiveness. Therefore, certain precautions should be considered. We based our study on previous work by Munilla et al. and other literature representing the current state of the art, identifying the existing gap (RQ1) at the time of writing. However, new works may emerge during the publication process. Due to time and space constraints, we have discussed analysis techniques (RQ2) in a broad manner, focusing on the details researchers find most relevant. Similarly, mapping all possible challenges (RQ3) is a daunting task, and our work is not exhaustive. Nevertheless, we have concentrated on the challenges we consider most important based on our experience.

Our plan for future work involves a broader spectrum of VR, AR, and MR devices. This will offer us a more holistic understanding of existing challenges. Furthermore, we aim to confront several challenges outlined in this study, providing researchers with valuable methodologies to undertake analysis within the "Virtual Maze". The colocation of the different challenges poses a logical priority for addressing them. Therefore, we derive the following roadmap.

VII. Conclusions

We have provided an overview of the Metaverse ecosystem, discussing the core elements of its architecture and the development cycle. We observed that the heterogeneity of the ecosystem could pose important challenges for the scrutiny of the security (or lack of) of these devices and the privacy of their users. We then reviewed the state-of-the-art and extensively discussed threat models and user risks. We identified a gap in the literature covering key threats and presented a thorough review of the challenges that need to be overcome to bridge the existing gap. Our work is the first to distill these challenges and offer valuable insights into which of these challenges sit on the critical path.

Acknowledgments

This paper has been funded by project PID2022-143304OB-I00 (PARASITE) funded by MICIU/AEI/10.13039/501100011033/ and by the ERDF, EU. A. Rodriguez is a grantee of the "Programa Investigo" (2022-C23.I01.P03.S0020-0000038), funded by the European Union NextGeneration-EU/PRTR and MITES/SEPE. S. Pastrana was supported by Grant TED2021-132170A-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by the "EU NextGenerationEU/PRTR". G. Suarez-Tangil and N. Vallina-Rodriguez have been appointed as 2019 Ramon y Cajal fellows (RYC-2020-029401-I and RYC2020-030316-I, respectively) funded by MICIU/AEI/10.13039/501100011033 and the ESF Investing in your future.

References

[1] "National Geographic VR Experiences." [Online]. Available: https://www.meta.com/en-gb/experiences/2046607608728563/

[2] "Netflix VR." [Online]. Available: https://www.meta.com/en-gb/experiences/2184912004923042/

[3] "Pyautogui · pypi," https://pypi.org/project/PyAutoGUI/, 2024, accessed: 2024-07-08.

[4] D. Adams, A. Bah, C. Barwulor, N. Musaby, K. Pitkin, and E. M. Redmiles, "Ethics emerging: the story of privacy and security perceptions in virtual reality," 2018.

[5] "App Store." [Online]. Available: https://www.apple.com/app-store/

[6] A. A. Arafat, Z. Guo, and A. Awad, "VR-Spy: A Side-Channel Attack on Virtual Key-Logging in VR Headsets," 2021.

[7] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps," *ACM sigplan notices*, 2014.

[8] R. Barceló-Armada, I. Castell-Uroz, and P. Barlet-Ros, "Amazon alexa traffic traces," *Computer Networks*, vol. 205, p. 108782, 2022.

[9] E. Bozkir, O. Günlü, W. Fuhl, R. F. Schaefer, and E. Kasneci, "Differential privacy for eye tracking with temporal correlations," 2021.

[10] L. J. Camp and V. Andalibi, "Sbom vulnerability assessment & corresponding requirements," *NTIA Response to Notice and Request for Comments on Software Bill of Materials Elements and Considerations*, 2021.

[11] "CVE-2024-23775 : Integer Overflow vulnerability in Mbed," https://www.cvedetails.com/cve/CVE-2024-23775/, 2024.

[12] B. Davis, B. Sanders, A. Khodaverdian, and H. Chen, "I-arm-droid: A rewriting framework for in-app reference monitors for android applications," 2012.

[13] R. Di Pietro and S. Cresci, "Metaverse: security and privacy issues." IEEE, 2021.

[14] "Firefox Reality," https://www.meta.com/en-gb/experiences/firefox-reality/2180252408763702/, 2024.

[15] G. M. Garrido, V. Nair, and D. Song, "SoK: Data Privacy in Virtual Reality," 2023.

[16] A. Girish, T. Hu, V. Prakash, D. J. Dubois, S. Matic, D. Y. Huang, S. Egelman, J. Reardon, J. Tapiador, D. Choffnes *et al.*, "In the room where it happens: Characterizing local communication and threats in smart homes," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, 2023, pp. 437–456.

[17] Z. Guo, Z. Lin, P. Li, and K. Chen, "{SkillExplorer}: Understanding the behavior of skills in large scale," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2649–2666.

[18] "HoloLens 2." [Online]. Available: https://www.microsoft.com/en-us/hololens/buy

[19] "HTC." [Online]. Available: https://www.htc.com/

[20] "HTC Developer," https://developer.vive.com/resources/vive-wave/download/archive/531/, 2024.

[21] "ioXt," https://ioxtalliance.org/, 2024.

[22] S. Ishaque, A. Rueda, B. Nguyen, N. Khan, and S. Krishnan, "Physiological signal analysis and classification of stress from virtual reality video game." IEEE, 2020.

[23] V. Jayaram, S. Ananthakrishnan, S. Mohan, V. Ganesh, N. Ravi, and K. L. N. Rao, "Virtual Reality Simulation for Surgical Training: A Review of the Literature," 2019.

[24] P. Karkallis, J. Blasco, G. Suarez-Tangil, and S. Pastrana, "Detecting video-game injectors exchanged in game cheating communities," 2021.

[25] Z. Ling, Z. Li, C. Chen, J. Luo, W. Yu, and X. Fu, "I Know What You Enter on Gear VR," 2019.

[26] Z. Ma, H. Wang, Y. Guo, and X. Chen, "Libradar: Fast and accurate detection of third-party libraries in android apps," in *Proceedings of the 38th international conference on software engineering companion*, 2016, pp. 653–656.

[27] "XR headset market share by quarter 2023 — Statista," https://www.statista.com/statistics/1222146/xr-headset-shipment-share-worldwide-by-brand/, 2024.

[28] "Oculus Developers," https://developer.oculus.com/get-started-platform/, 2024.

[29] "Meta Quest 3." [Online]. Available: https://www.meta.com/quest/quest-3

[30] "Meta Quest VR Games, Apps, Deals and More." [Online]. Available: https://www.meta.com/experiences/

[31] V. Nair, G. M. Garrido, D. Song, and J. O'Brien, "Exploring the privacy risks of adversarial vr game design," *Proceedings on Privacy Enhancing Technologies*, 2023.

[32] V. Nair, W. Guo, J. Mattern, R. Wang, J. F. O'Brien, L. Rosenberg, and D. Song, "Unique Identification of 50,000+ Virtual Reality Users from Head & Hand Motion Data," 2023.

[33] V. C. Nair, G. Munilla-Garrido, and D. Song, "Going Incognito in the Metaverse: Achieving Theoretically Optimal Privacy-Usability Tradeoffs in VR," ser. UIST '23, 2023.

[34] I. Olade, C. Fleming, and H.-N. Liang, "Biomove: Biometric user identification from human kinesiological movements for virtual reality systems," 2020.

[35] "Pavlov VR — Steam." [Online]. Available: https://store.steampowered.com/app/555160/Pavlov_VR/

[36] "PICO Developer," https://developer.picoxr.com/, 2024.

[37] "VR PICO," https://www.picoxr.com, 2024.

[38] M. B. Powers, S. J. Kim, and D. F. Tolin, "Virtual Reality Exposure Therapy for the Treatment of Height Phobia: A Meta-Analysis," 2021.

[39] A. Pradeep, M. T. Paracha, P. Bhowmick, A. Davanian, A. Razaghpanah, T. Chung, M. Lindorfer, N. Vallina-Rodriguez, D. Levin, and D. Choffnes, "A comparative analysis of certificate pinning in android & ios," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 605–618.

[40] "PS VR2," https://www.playstation.com/en-us/ps-vr2/, 2024.

[41] H. X. Qin, Y. Wang, and P. Hui, "Identity, crimes, and law enforcement in the metaverse," *arXiv preprint arXiv:2210.06134*, 2022.

[42] I. Reyes, P. Wijesekera, J. Reardon, A. Elazari Bar On, A. Razaghpanah, N. Vallina-Rodriguez, S. Egelman *et al.*, ""won't somebody think of the children?" examining coppa compliance at scale," in *The 18th Privacy Enhancing Technologies Symposium (PETS 2018)*, 2018.

[43] J. Šalkevicius, R. Damaševičius, R. Maskeliunas, and I. Laukienė, "Anxiety level recognition for virtual reality therapy system using physiological signals," 2019.

[44] C. Sandeepa, S. Wang, and M. Liyanage, "Privacy of the Metaverse: Current Issues, AI Attacks, and Possible Solutions." IEEE, 2023.

[45] Y. Shen, H. Wen, C. Luo, W. Xu, T. Zhang, W. Hu, and D. Rus, "GaitLock: Protect Virtual and Augmented Reality Headsets Using Gait," 2019.

[46] C. Slocum, Y. Zhang, N. Abu-Ghazaleh, and J. Chen, "Going through the motions:{AR/VR} keylogging from user head motions," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 159–174.

[47] M. Tileria, J. Blasco, and G. Suarez-Tangil, "{WearFlow}: Expanding information flow analysis to companion apps in wear {OS}," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020, pp. 63–75.

[48] P. P. Tricomi, F. Nenna, L. Pajola, M. Conti, and L. Gamberini, "You Can't Hide Behind Your Headset: User Profiling in Augmented and Virtual Reality," 2023.

[49] R. Trimananda, H. Le, H. Cui, J. T. Ho, A. Shuba, and A. Markopoulou, "{OVRseen}: Auditing network traffic and privacy policies in oculus {VR}," in *31st USENIX security symposium (USENIX security 22)*, 2022.

[50] "Valve Index," https://store.steampowered.com/valveindex, 2024.

[51] "Vision Pro of Apple." [Online]. Available: https://www.apple.com/apple-vision-pro/

[52] M. Vondráček, I. Baggili, P. Casey, and M. Mekni, "Rise of the metaverse's immersive virtual reality malware and the man-in-the-room attack & defenses," 2023.

[53] M. Vondráček, I. Baggili, P. Casey, and M. Mekni, "Rise of the Metaverse's Immersive Virtual Reality Malware and the Man-in-the-Room Attack & Defenses," 2023.

[54] "VRcompare." [Online]. Available: https://vr-compare.com/

[55] "PWC - Economic impact of VR and AR." [Online]. Available: https://www.pwccn.com/en/tmt/economic-impact-of-vr-ar.pdf

[56] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," 2022.

[57] X. Wei and C. Yang, "FoV Privacy-aware VR Streaming," 2022.

[58] A. Winkler, J. Won, and Y. Ye, "QuestSim: Human Motion Tracking from Sparse Sensors with Simulated Avatars."

[59] D. Wu, C. G. Courtney, B. J. Lance, S. S. Narayanan, M. E. Dawson, K. S. Oie, and T. D. Parsons, "Optimal Arousal Identification and Classification for Affective Computing Using Physiological Signals: Virtual Reality Stroop Task," 2010.

[60] "YouTube VR," https://www.meta.com/en-gb/experiences/2002317119880945/, 2024.

[61] T. Zhang, C. Shi, P. Walker, Z. Ye, Y. Wang, N. Saxena, and Y. Chen, "Passive Vital Sign Monitoring via Facial Vibrations Leveraging AR/VR Headsets," 2023.

[62] T. Zhang, Z. Ye, A. T. Mahdad, M. M. R. R. Akanda, C. Shi, Y. Wang, N. Saxena, and Y. Chen, "FaceReader: Unobtrusively Mining Vital Signs and Vital Sign Embedded Sensitive Info via AR/VR Motion Sensors," 2023.