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

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

***Abstract*— The Metaverse is an increasingly popular virtual environment. Recent technological advances — such as head-mounted displays and novel sensors — have enabled a more seamless integration of virtual experiences into our daily lives. This integration is reshaping how we interact with both our environment and each other, offering boundless opportunities across entertainment, social engagement, and business. As the technology remains in its early stages and far from market consolidation, a proliferation of new platforms and devices has resulted in a highly heterogeneous ecosystem. Much like the smartphone landscape, leading platforms in the Metaverse allow users to install third-party applications via online marketplaces. However, the Metaverse introduces the additional ambition of enabling seamless interaction across multiple virtual environments. Consequently, developers face mounting pressure to support a wide range of platforms and often depend on cross-platform development frameworks — adding yet another layer of complexity to the application stack.**

This complexity, combined with the presence of disruptive new sensors in XR headsets, introduces novel risks to user security and privacy. Although progress has been made in identifying potential threats within this ecosystem, there remains a substantial gap in methodologies and tools for analyzing actionable risks on popular headsets. In this work, we present a vision for application analysis grounded in practical experience. We consider the various components of the XR ecosystem that influence security assessments and highlight key challenges that must be addressed to effectively mitigate emerging threats.

I. Introduction

The Metaverse is founded on three key technologies that constitute Extended Reality (XR): Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR). VR immerses users in entirely synthetic environments; AR enhances the physical world by overlaying digital elements; and MR encompasses a continuum between VR and AR, enabling dynamic interaction between virtual and real-world components.

XR applications extend far beyond the gaming experience for which they were characterized in their early days. Health [68], Professional Training [26], Therapy [45] 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 [62]. 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 [17]), Google (YouTubeVR [67]), 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. [18] 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], [16], [38], [51], [60], [63], [69], which, besides technical issues, they also discuss ethical, social, or economic problems. However, most of them explore the possibilities of user identification [14], [41], [52] and data collection [10], [15], [30] authentication and identification of users [27], [28], [32], [54] offering 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.

II. Methodology and Contribution

Overall, there remains a significant knowledge gap in the methods and techniques available for analyzing XR applications—one that is further exacerbated by the underdeveloped and fragmented state of current analytical approaches. In response to this gap, and informed by both our direct experience with XR systems and a comprehensive review of the relevant literature, this paper presents a qualitative study aimed at identifying the methodological and technical challenges that currently hinder effective analysis of the broader Metaverse ecosystem. Addressing these challenges is essential for enabling researchers to evaluate XR applications systematically and at scale.

To structure our investigation and guide the development of our qualitative study, we articulate a focused set of research questions. These questions are designed to uncover the methodological and technical barriers currently hindering rigorous analysis within the Metaverse ecosystem. To ensure a comprehensive understanding, our study draws from two complementary sources: an extensive review of the existing academic and industry literature on XR evaluation, and our own first-hand experience developing, deploying, and assessing XR applications in varied contexts. The literature review enables us to trace prevailing methodological trends, identify gaps in analytical practices, and map the evolution of XR-related evaluation strategies. In parallel, our experiential insights ground the study in practical realities, allowing us to critically reflect on current limitations and recurring challenges observed in real-world XR implementations. Together, these methodological foundations inform our analysis and support the formulation of research questions that aim to identify foundational requirements for advancing scalable evaluation techniques for XR applications. Our inquiry is therefore driven by the following core research questions:

- **RQ1:** What are the structural and technical characteristics of the Metaverse ecosystem — including platforms, marketplaces, devices, and runtime environments — and how do they impact the feasibility of analyzing XR applications?
- **RQ2:** What are the emerging security and privacy risks that current threat models fail to adequately capture?
- **RQ3:** What techniques can be adopted from other domains to automate the analysis of Metaverse applications?
- **RQ4:** 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. [18], 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. This report is an extended version of [9]. 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:

1^{st}: We conduct a systematic examination of the current Metaverse landscape, providing a detailed overview of the software ecosystem, including key platforms, application marketplaces, and hardware devices (RQ1). This includes an analysis of the fragmentation across vendors and the diversity of runtime environments in which XR applications operate. We also highlight differences in development practices, distribution models, and execution contexts that complicate standard analysis approaches (see §III-A & §III-B).

2^{nd}: We investigate and characterize the core technological gaps (RQ2) found in existing literature related to the analysis of Metaverse and XR applications. This includes identifying assumptions, blind spots, and limitations in current methodologies, particularly around threat modeling, data access, and environmental constraints. Our goal is to surface the areas where analytical capabilities are lacking or incompatible with emerging XR paradigms (see §IV).

3^{rd}: We identify, categorize, and critically discuss both the techniques (RQ3) that have been proposed or could be adapted to analyze XR applications, as well as the persistent challenges (RQ4) that hinder their practical implementation. These challenges include, but are not limited to, issues in performing static code analysis, obstacles in achieving realistic and scalable dynamic testing, and complexities in automating the discovery and download of XR applications for research purposes. Our discussion emphasizes the need for methodological innovation to move beyond superficial testing or mock-up environments and towards more robust, scalable analysis pipelines (see §VI, §VII, & §V).

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.

III. Fragmentation in the Metaverse

As XR (Extended Reality) continues to gain traction among both developers and end-users, an increasing number of business opportunities and application scenarios are emerging. This rapid growth has resulted in a proliferation of new platforms, development tools, application stores, and devices, rendering the XR ecosystem — often branded as the Metaverse — particularly complex. To address RQ1, in this section, we examine the underlying architecture (§III-A) that supports XR technologies, as well as the corresponding development cycle (§III-B), as summarized in Figure 1.

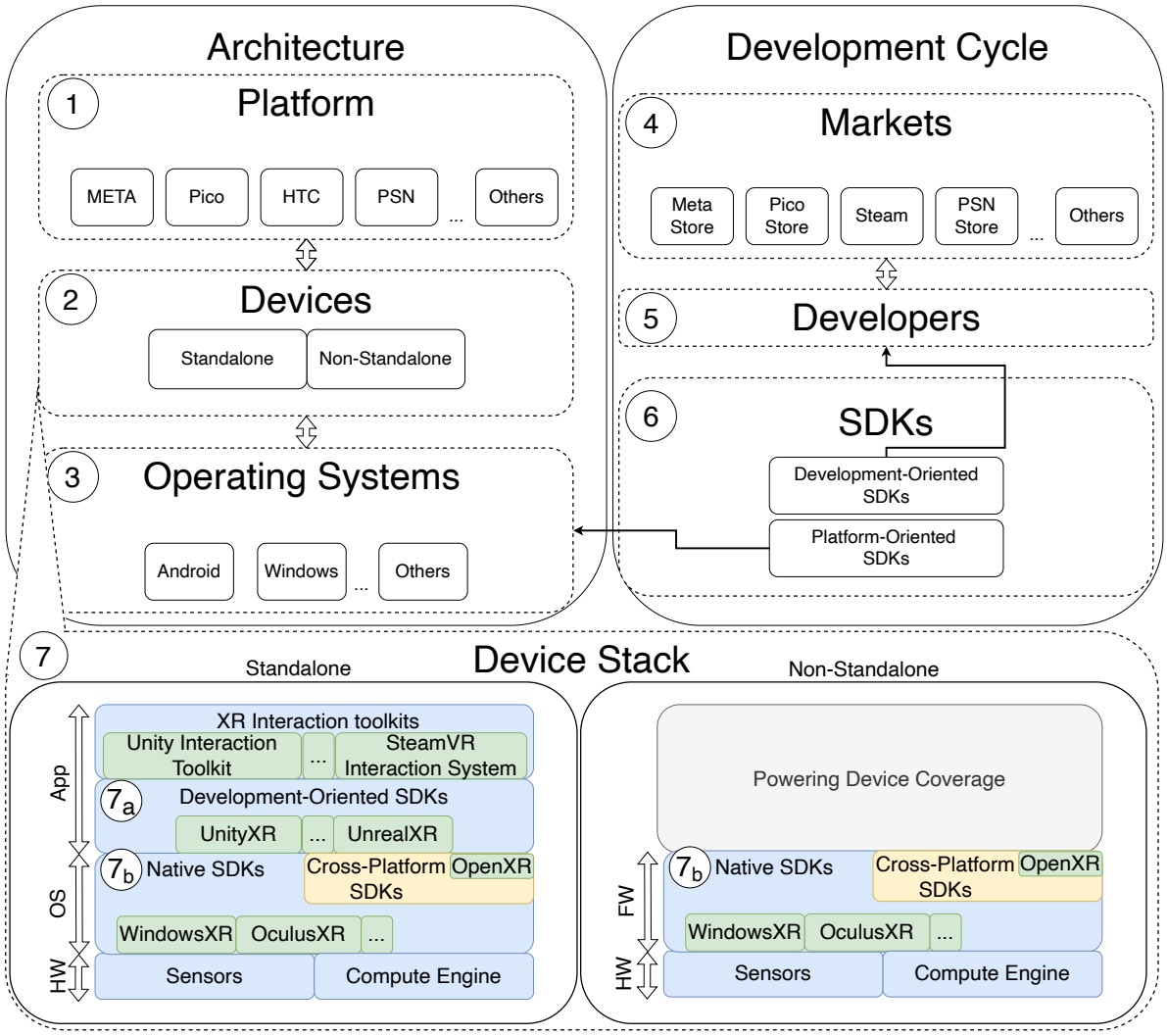

Fig. 1: Metaverse ecosystem.

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 [34]. 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 [36]. PlayStation already offers PSVR2 [47] to play VR games over PlayStation 5 and many other platforms are gaining market presence, such as Pico [44], HTC [22], Valve [58], or Apple [59].

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 [61]. 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 [61] 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

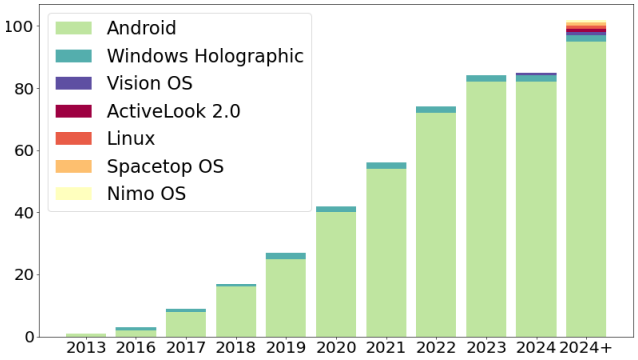

Fig. 2: Market share of OSs in Standalone devices over time. (2024+ is a projection).

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 [47], the Vision OS for Vision Pro [59] headsets from Apple, and the Windows Holographic OS for HoloLens 2 [21] from 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 [37] 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 [42]).

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 [35], Pico [43], or HTC [23] 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 in their Integrated Development Environments (IDEs) to ease the development of games.

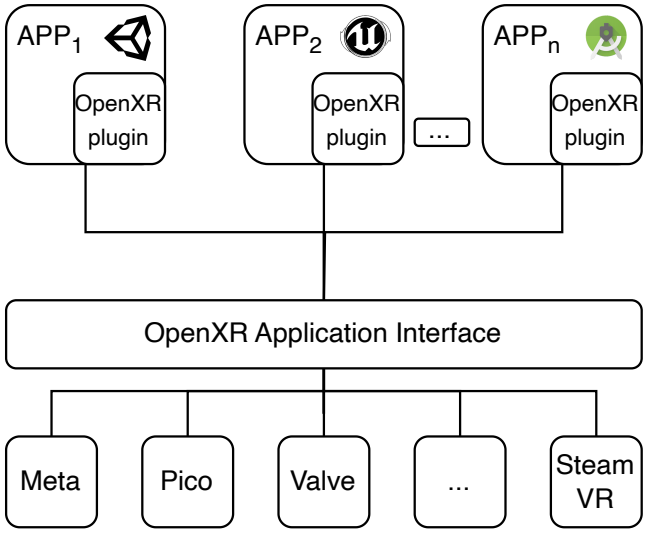

Fig. 3: OpenXR SDK Application Interface. (Engines left to right: Unity, Unreal Engine and Android Studio).

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.

The growing complexity of XR systems—driven by new sensors, platform-specific functionalities, and the growing importance of native languages in cross-platform development—demands a deep understanding of their underlying architectures and component interactions. At the same time, the heterogeneity of platforms and proprietary technologies complicates analysis efforts, often requiring custom tools tailored to specific binaries, APIs, or execution environments. Addressing these challenges is critical for evolving existing techniques to reliably assess the security and behavior of XR applications within the diverse Metaverse ecosystem.

IV. Threat Model

We then identify the existing knowledge gap (RQ2) that

must be addressed to ensure user privacy and security. The Metaverse blurs the boundaries between physical and virtual realities, introducing novel risks associated with user-to-user interactions within both XR environments and the broader cyber-physical context. The heightened realism and immersive nature of these ecosystems amplify the potential for privacy-invasive practices, enabling more extensive data collection and user monitoring. Additionally, this realism raises pressing concerns around user safety and security, as the consequences of virtual interactions increasingly mirror those in the real world.

Prior work has recently identified VR privacy threats [18], 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 [19]; ⓔ **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 [48].

Answering RQ2, 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. [18], 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" [40], [41], [52] and "data theft" [6], [10], [64] the psychological effects of using XR applications and virtual harassment [50], [66], or law enforcement in the Metaverse [48]. 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 seven studies measuring threats or experimenting with real applications, whereby they only focus on VR settings [6], [25], [31], [39], [53], [57], [60]. Only Trimananda et al. [57] 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. Table I illustrates the identified knowledge gap within this context by presenting only those studies that were conducted using real-world XR applications. Notably, the sparse coverage across the table — evidenced by the substantial number of empty rows and columns — highlights the lack of prior work addressing key dimensions of XR application analysis. This absence underscores both the limited scope of existing research and the need for broader, more systematic approaches to studying the Metaverse ecosystem.

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 detect and mitigate emerging threats in Metaverse software empirically — 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 fingerprinting or data theft, important flaws prevent their application at scale. Other key threats, such as room fingerprinting, have not yet been addressed, 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 (§VI, §VII, and §V) that the community must address to effectively assess the privacy and security threats of real-world Metaverse applications.

Having answered RQ1 and RQ2, we address RQ3 and RQ4 in what follows. For RQ3, 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 RQ4, we leverage the community expertise in successfully applying these techniques to identify

Threats\Adversaries	Hardware			Client			Server			User		
	VR	AR	MR	VR	AR	MR	VR	AR	MR	VR	AR	MR
ⓐ Data collection				[6], [31], [57]								
ⓑ Identification				[39], [53]								
ⓒ Profiling												
ⓓ Impersonating												
ⓔ Data leaks												
ⓕ Data deletion												
ⓖ Virtual asset theft												
ⓗ Room fingerprint												
ⓘ Malicious design of hardware												
ⓙ Malicious software												
ⓚ Device hijacking												
ⓛ Lateral movement												
ⓜ Virtual extortion												
ⓝ Virtual scam												
ⓞ Virtual harassment				[60]						[25]		

TABLE I: Knowledge Gap in Application Analysis.

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 (§V), static analysis (§VI), and dynamic analysis (§VII).

V. Market Crawls

Similar to mobile app stores, Metaverse marketplaces are not traditional databases or repositories of applications, but rather user-interface-driven platforms designed for browsing and purchasing software. This characteristic poses challenges for automating the collection of applications at scale, as it complicates efforts to systematically gather samples for testing. Collecting Metaverse applications thus requires specialized crawling techniques that are aware of the marketplace structures and platform architectures discussed in the previous section.

While this challenge may not constitute a research problem in itself, the inability to collect software samples at scale significantly limits our capacity to develop a comprehensive understanding of the Metaverse ecosystem, the underlying software engineering practices, and the security and privacy threats associated with them.

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 the PC is 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.

VI. 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. Additionally, static analysis only reveals potential threats as it can not be used to identify actual evidence of harm. Without fully modeling the Metaverse ecosystem, existing static methods (e.g., control and data flow analysis) are unfortunately less effective.

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: The specific characteristics of the operating system running an XR application, along with the SDKs or development frameworks used by developers, can significantly influence how the application interacts with its environment. In particular, understanding the abstraction layer implemented by third-party SDKs is essential for modeling the lifecycle of Metaverse applications and, consequently, for designing appropriate methodologies for the analysis of these applications. 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 [33] 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 [55]. 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. Since having the source code is rarely possible in real-world applications where we may rely on approximations given by decompilers or disassemblers. The former provides researchers with a closer representation of the source code than the latter which is only capable of retrieving the assembly code of a binary. Unfortunately, cross-platform development used to rely on native libraries, distributed in binary format, due to performance optimization and interoperability, among other reasons. Understanding bytecode-based native libraries is a complex process as the analysis must be conducted at the assembly code level, which depends on the processor architecture of the device. Furthermore, anti-analysis techniques can be performed by developers to make more difficult the analysis of their products such as code encapsulation or obfuscation. Researchers need tools to perform static code analysis over both decompiled and disassembled XR Applications that are capable of getting rid of those anti-analysis practices and supporting the different programming languages and architectures.

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 [12]. 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 [11].

VII. Dynamic Analysis Challenges

Dynamic analysis involves executing software and monitoring its behavior—such as memory usage, runtime method invocations, and network traffic—to validate findings from static analysis and reduce false positives. However, in the context of the Metaverse, device owners typically have limited access to system settings and functionalities compared to administrators. This lack of permissions can hinder access to certain operating system resources at runtime, thereby preventing researchers from fully observing an application's internal state.

Moreover, dynamic analysis may fail to detect certain issues due to the inherent difficulty of simulating all possible execution paths within an interactive XR environment. Many functionalities are only triggered under specific conditions, and the number of potential application states is exceedingly large. In addition, some events—such as interactions involving other users—are particularly challenging to emulate or monitor and may also raise ethical concerns.

In the following section, we examine these limitations in greater detail by addressing the key challenges associated with dynamic analysis in the XR ecosystem.

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 [13], 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 [29]. 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 [20], 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 [46]. 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 [49] — 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 §IV, the only existing research paper on the presence of third-party endpoints in the Metaverse to date [57] 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.

VIII. 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: For this work, we exclusively consider studies that validate their theories and implementations in real-world applications. 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 [40], [41], [52] and data theft [6], [10], [64], 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) [40], [41], [56], [65]. 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 the context of XR applications.

While some prior work has proposed threat models for Metaverse applications, limited efforts have validated these models through static and dynamic analysis of real-world applications. For example, Trimananda et al. [57] analyzed data from 140 applications; however, their evaluation was confined to a small subset of apps from a single platform. Additionally, the process of interacting with applications to collect network traffic was conducted manually, and their approach did not address key challenges such as encrypted communication or server authentication—factors that are essential for capturing a complete view of the network behavior in many applications

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 significant challenges in analyzing security within the Metaverse, largely due to the broad and diverse range of VR, AR, and MR solutions. The ecosystem includes both standalone and non-standalone devices (i.e., those dependent on third-party platforms such as PCs, consoles, or smartphones), each introducing different architectural considerations. This diversity, combined with the wide variety of operating systems—most of which are Android-based—multiple app marketplaces, and a growing set of development frameworks, results in a highly fragmented landscape. Such fragmentation presents substantial obstacles to comprehensive security assessments. It complicates the identification of vulnerabilities, detection of insecure development practices, and evaluation of supply chain integrity (CH-SA-4). The complexity of the environment makes it difficult to apply uniform analysis methods or reuse established tools across platforms. As a result, understanding the underlying OS ③ and software stack ⑦ becomes a critical prerequisite for any meaningful application analysis. These elements shape the overall software architecture and define how applications interact with the operating system, which in turn influences attack surfaces and security guarantees. Furthermore, the absence of a dedicated security certification framework for XR devices—comparable to existing standards for IoT, such as ioXt [24]—creates a regulatory and assurance gap. A standardized certification process could assist both vendors and developers in identifying and mitigating security risks specific to XR technologies. It would also provide a pathway to meet emerging compliance requirements, such as those outlined in the EU Cyber Resilience Act, which is expected to come into force soon.[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

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

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-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) which is critical for identifying potential vulnerabilities and data leaks. 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. Exploring as many execution paths as possible—ideally all—is essential, as some vulnerabilities or malicious behaviors may reside in functionalities that are only triggered under specific conditions. **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 offers valuable insights into the challenges of analyzing Metaverse applications, it is not without limitations. As an experience-based paper, our observations and conclusions are inherently shaped by the platforms with which we are most familiar, primarily the Meta Quest 2 ecosystem. To mitigate this bias, we supplemented our platform-specific insights with a comprehensive review of existing literature and drew upon our broader research expertise in Android environments. However, the security features and vulnerabilities of applications and environments can differ significantly across platforms, and our study does not provide an exhaustive analysis of these variations. As such, the generalizability of our findings to other platforms should be interpreted with caution.

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 (RQ2) 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 (RQ3) in a broad manner, focusing on the details researchers find most relevant. Similarly, mapping all possible challenges (RQ4) 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 to enable us to adopt a more diverse and expansive approach, thereby fostering a comprehensive understanding of the cybersecurity landscape within the Metaverse. This will offer us a more holistic understanding of the 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.

IX. Conclusions

In this work, we presented an overview of the Metaverse ecosystem, outlining its core architectural components and development cycle. Our analysis highlights how the ecosystem's heterogeneity introduces significant challenges in evaluating the security of XR devices and protecting user privacy. We reviewed the state of the art, examined relevant threat models, and explored user risks in depth. Through this analysis, we identified a notable gap in the literature regarding key security and privacy threats, and we provided a detailed discussion of the challenges that must be addressed to bridge this gap. To the best of our knowledge, this is the first work to systematically distill these challenges and offer actionable insights into which ones lie on the critical path toward securing the Metaverse ecosystem.

Acknowledgments

This extended review was supported by the Spanish National Cybersecurity Institute (INCIBE) under Proyectos Estratégicos de Ciberseguridad – CIBERSEGURIDAD EINA UNIZAR and by the Recovery, Transformation and Resilience Plan funds, financed by the European Union (Next Generation). A. Rodriguez was a grantee of the "Programa Investigo" (2022-C23.I01.P03.S0020-0000038) when this project started, 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. Preliminary work was funded by project PID2022-143304OB-I00 (PARASITE), funded by MICIU/AEI/10.13039/501100011033/ and by the ERDF, EU.

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] A. R. Barredo-Valenzuela, S. P. Portillo, N. Vallina-Rodriguez, and G. Suarez-Tangil, "Reversing the virtual maze: An overview of the technical and methodological challenges for metaverse app analysis," in *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*. IEEE, 2024, pp. 173–181.

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

[11] 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.

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

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

[14] J. A. de Guzman, K. Thilakarathna, and A. Seneviratne, "Conservative plane releasing for spatial privacy protection in mixed reality," 2020.

[15] ——, "SafeMR: Privacy-aware Visual Information Protection for Mobile Mixed Reality," 2019.

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

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

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

[19] 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.

[20] 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.

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

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

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

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

[25] 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.

[26] 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.

[27] B. John, S. Jörg, S. Koppal, and E. Jain, "The security-utility trade-off for iris authentication and eye animation for social virtual avatars," 2020.

[28] B. John, S. Koppal, and E. Jain, "EyeVEIL: degrading iris authentication in eye tracking headsets," 2019.

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

[30] J. Li, A. R. Chowdhury, K. Fawaz, and Y. Kim, "Kalido: Real-Time Privacy Control for Eye-Tracking Systems," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 1793–1810. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/li-jingjie

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

[32] S. Luo, A. Nguyen, C. Song, F. Lin, W. Xu, and Z. Yan, "OcuLock: Exploring human visual system for authentication in virtual reality head-mounted display," 2020.

[33] 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.

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

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

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

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

[38] 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.

[39] 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.

[40] 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.

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

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

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

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

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

[46] 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.

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

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

[49] 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.

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

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

[52] 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.

[53] 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.

[54] S. Stephenson, B. Pal, S. Fan, E. Fernandes, Y. Zhao, and R. Chatterjee, "Sok: Authentication in augmented and virtual reality." IEEE, 2022.

[55] 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.

[56] 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.

[57] 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.
[58] "Valve Index," https://store.steampowered.com/valveindex, 2024.
[59] "Vision Pro of Apple." [Online]. Available: https://www.apple.com/apple-vision-pro/
[60] 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.
[61] "VRcompare." [Online]. Available: https://vr-compare.com/
[62] "PWC - Economic impact of VR and AR." [Online]. Available: https://www.pwccn.com/en/tmt/economic-impact-of-vr-ar.pdf
[63] 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.
[64] X. Wei and C. Yang, "FoV Privacy-aware VR Streaming," 2022.
[65] A. Winkler, J. Won, and Y. Ye, "QuestSim: Human Motion Tracking from Sparse Sensors with Simulated Avatars."
[66] 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.
[67] "YouTube VR," https://www.meta.com/en-gb/experiences/2002317119880945/, 2024.
[68] 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.
[69] 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.