Journal of Information Processing Vol.32 779–788 (Sep. 2024)

[DOI: 10.2197/ipsjjip.32.779]

Regular Paper

An Investigation of Privacy and Security in VR APPs through URL String Analysis

Huang Shu-Pei[1] Takuya Watanabe[2] Mitsuaki Akiyama[2] Tatsuya Mori[1,3,a)]

Received: December 5, 2023, Accepted: June 10, 2024

Abstract: This study aims to explore the privacy concerns associated with static hard-coded URLs used by virtual reality (VR) applications. Among these URLs, we identified destinations leading to advertising and analytics services, which may raise user privacy concerns. Our primary focus was in external libraries or sources that developers import, including privacy-related URLs. Through our measurement study of the VR device Oculus Go, we used a proposed categorization methodology to disclose popular advertising and analytics sources. The results revealed the presence of potential privacy threats and their implications for user rights. Consequently, we discuss the possible improvements for VR manufacturers to enhance future VR experiences.

Keywords: apps, measurement, privacy, VR

1. Introduction

The advertising and analytics ecosystems play a crucial role in generating significant profits for service providers and application publishers. To implement advertising and analytics in an application, collecting client-generated data is a common approach for achieving effective functionality. However, from the perspective of the user, privacy concerns have always been a topic of discussion. To mitigate invasive actions related to privacy, consensus or agreement between users and advertising and analytics service providers is necessary. In the real world, various regulations have been enacted by different countries and organizations, such as GDPR in the European Union and PIPEDA in Canada, to legally ensure user privacy.

Metaverse, a massively trending topic, commonly employs virtual reality (VR) devices as a medium to create a virtual-world society connected by the Internet. VR technology is the backbone of Metaverse, which enables users to immerse themselves in virtual environments using VR devices. The most popular VR device is the helmet-mounted display (HMD), which is often paired with controllers to support user input. These VR devices are equipped with sensors to enhance the VR experience. In comparison with mobile devices, VR devices generate sensory data that are relatively closely linked to the personal information of the users, such as head and eye tracking and current room area status. Leakage of VR sensory data raises concerns for users and privacy researchers. Furthermore, the unique user interfaces in VR environments pose significant challenges for users to access and understand privacy policies compared to traditional platforms like PCs and smartphones. The lack of explicit advertising displays in VR can obscure the fact that tracking is occurring, making it difficult for users to realize their privacy may be compromised, even if they have consented to the terms of use. This lack of transparency raises substantial privacy concerns when advertising and analytics are employed in VR applications without clear user awareness.

The advertising and analytics ecosystems of VR devices have been the subject of research for several years. In recent studies, Nair et al. [1] demonstrated the collection of various types of personal data using their designed VR application during short gameplay. Personal data ranging from anthropometrics to demographics are accessible to the application, highlighting the importance of transparent data collection practices from the perspective of the user. Another study by Trimananda et al. [2] presented an approach to link the raw traffic of VR applications to their corresponding privacy policies.

Building upon these findings, our study investigates the state of the advertising and analytics ecosystems of VR devices, focusing on the unique challenges and privacy implications within VR environments. We conducted experiments using Oculus GO as the target device. Oculus Go runs on the Android 7 operating system, which allows the use of popular Android analysis tools. We applied both static and dynamic analyses to VR applications. We collected all available free applications from the Oculus Store to perform a static analysis targeting hard-coded URLs in APKs. By analyzing hard-coded URLs and their sources, we sort to discover possible privacy issues and gain insights into the current state of advertising and analytics ecosystems on VR platforms.

We note that our study highlights several key differences that set it apart from existing research on Android app analysis. We highlight the unique privacy risks in VR environments due to the collection of user interaction data that is more invisible than what is typically observed in traditional Android apps. In addition,

[1] Waseda University, Shinjuku, Tokyo 169–8555, Japan
[2] NTT Social Informatics Laboratories, Musashino, Tokyo 180–8585, Japan
[3] NICT, Koganei, Tokyo 184–8795, Japan
[a)] mori@nsl.cs.waseda.ac.jp

the integration of advertising in VR applications requires special attention to cross-platform compatibility with game engines and packages, a challenge not present in traditional Android app development. Our detailed statistical analysis of the Java packages and native libraries within our dataset underscores the technical intricacies specific to VR advertising, further distinguishing our work from standard Android app analysis studies. By focusing on these VR-specific challenges and opportunities, our findings lay the groundwork for future comparative analyses between different VR ecosystems, marking a significant departure from the scope of existing Android app analysis research.

To shed light on the privacy issues related to VR devices, we propose a methodology that combines static and dynamic analyses to examine VR applications using APK files.

We formulated the following research questions:

Q1: What are the hard-coded URLs contained in VR applications, and how can we categorize them effectively?

Q2: Can we establish connections between hard-coded URLs and external library sources, allowing us to gain insights into the usage and characteristics of these sources?

Q3: Are there viable solutions to address privacy concerns arising from VR applications?

The contributions of this study are as follows:

- We identified the presence of advertising and analytics on the VR platform and demonstrated potential privacy concerns arising from these practices.
- This study analyzed 6,422 hard-coded URL strings among the 376 free VR applications available in the Oculus Store for Oculus Go. We further investigated the external library sources of the identified URLs.
- We proposed a novel measurement methodology to analyze privacy-related connections by inspecting VR application APK files. Through the results from our measurement study, we discussed possible solutions to mitigate privacy issues from VR applications.

2. Background

2.1 Virtual Reality Devices and Applications

VR applications are now widely adopted in various fields, including entertainment, gaming, education, training, healthcare, socialization, and other domains related to computer science. Depending on the nature of the VR application, various sensory data may be used to enable functionality, which raises significant concerns about user privacy owing to the collection of personal information.

Several VR device manufacturers and brands exist in the market, such as Meta Oculus with the Quest series, HTC with HTC Vive, Sony with PlayStation VR, Valve Corporation with Valve Index, Samsung with Samsung Gear VR, and Google with Google Cardboard. Market data and reports indicate that the Oculus Quest series currently holds the position of the best-selling VR device worldwide [3]. Featuring built-in processors and memory, Oculus Quest series devices eliminate the need for connections to computers or gaming consoles that function as mobile phones. Portability and stand-alone functionality contribute to its popularity and global sales success.

Because Oculus Go serves as a predecessor to the Oculus Quest series and shares similar functionalities, we propose conducting our experiments on Oculus Go using its Android 7 operating system. The root-privileged boot images of Oculus Go released by Oculus after they ceased accepting new apps or app updates in the Oculus Store present an opportunity for our research.

2.2 Advertising and Analytics Ecosystem in the Wild

Advertising and analytics play crucial roles in driving revenue, shaping consumer behavior, and enabling data-driven decision-making in modern digital ecosystems. These components provide substantial benefits to businesses by understanding user behavior, delivering targeted advertisements, and optimizing online experiences.

From the perspective of privacy researchers, advertising and analytics practices raise concerns regarding the extensive collection of user data for targeted advertising and analytical purposes without explicit user consent or awareness.

Whereas similar tracking activities have been prevalent on personal computers and mobile phones for decades, wearable devices, including VR devices, have introduced new dimensions of privacy concerns. Unlike past devices, which mainly collect user information related to activities and behaviors on the device itself, wearable devices, such as VR devices, focus on gathering personal data related to head and body movements, eye gaze, hand gestures, and potentially biometric data such as heart rate and emotional responses. Such user information can be considered as personally identifiable information (PII), posing challenges in tracking intentions. Our research aims to uncover the advertising and analytics ecosystem of VR devices, specifically Oculus Go, and proposes measures to address privacy concerns.

2.3 Related Works

This research is motivated by the interest generated by previous studies on VR devices. Latest studies have demonstrated that users of VR applications are confronted with novel privacy risks, including fingerprinting [4], virtual keyboard logging [5], [6], and location inference [7]. Trimananda et al. [2] proposed a comprehensive analysis methodology that focuses on network traffic to assess the consistency of privacy policies. Their study revealed the existence of unrelated and undisclosed data flows in network traffic. Additionally, Nair et al. [8] explored privacy-related personal information collected by their VR application and "escape room" game. Through geospatial telemetry, device specifications, network observations, and behavioral observations, user information such as wingspan, room area, gender, wealth, ethnicity, and biometric data were collected. This observation underscores the significance of addressing the advertising and analytics ecosystems of VR devices and calls for stringent privacy regulations.

Extensive research has been conducted on analyzing Android apps. Automated dynamic analysis techniques [9], [10], [11] have been proposed for analyzing Android applications. However, considering the fundamental differences in UI design, it is challenging to apply these techniques to VR applications. In contrast, for an approach of statically analyzing APKs, we can leverage insights from prior studies. Specifically, by extracting URLs con-

tained within APKs, Pariwono et al. [12] identified communication to abandoned resources, Luoshi et al. [13] detected Android malware, and Calciati et al. [14] observed application update behavior. We utilize URL analysis in the novel context of identifying privacy threats in VR applications. To further comprehend the advertising and analytics ecosystem of Android-based devices, Pourali et al. [15] proposed a design and implementation for detecting privately invasive application behaviors on Android. Their findings on application behaviors provide valuable insights for our research. Similarly, Kuzuno et al. [16] investigated Android permissions related to privacy and provided crucial sensitive information about Android.

3. Methodology

Thus far, we have discussed the importance of privacy from various perspectives. To gain insight into VR applications in practice, a measurement study was conducted using both static and dynamic analyses. An overview of the proposed methodology is presented in **Fig. 1**. In this study, we attempted to understand the possible privacy issues related to the content of VR application APKs by observing the included libraries.

3.1 Experiment Platform

To initiate the measurement study, we used Oculus Go [17], an early product developed by Oculus, in collaboration with Qualcomm and Xiaomi. Oculus Go consists of a HMD and a single-handed controller for user input. Although Oculus Go has only one controller, we were able to find most of the categories of VR applications available on its platform.

There are three notable advantages of using Oculus Go in our study. First, Oculus Go provides a rooted version of the boot image from Oculus/Meta official [18], facilitating our experimentation by granting easy access to the operating system. Second, Oculus Go runs on an Android-based operating system. Specifically, it uses Android 7 with Android SDK v25. Third, application updates for Oculus Go were ceased on December 4, 2020. Although the VR applications that we collected may have been outdated, we benefited from having access to earlier versions. It is highly likely that a newer version of the same application will exhibit a structure similar to that of its APK file. We also observed that most of the APKs we collected from Oculus Go were not obfuscated.

3.2 Dataset

We found no website or service that collects a large amount of APK data, such as APKPure [19] or APKMirror [20] for Oculus Go. We observed that there are 503 available free applications; therefore, it was possible for us to manually download them from Oculus Store [21] on Oculus Go. A total of 376 items were downloaded successfully; the other items were unavailable owing to timeout or system version discrepancies. Finally, 6,422 hard-coded URLs were extracted from these APKs for static analysis. A total of 292 different sources of URLs were identified, including Java packages, native libraries, and constant string usage. A total of 242,439 packets were collected when applying the dynamic analysis to our target suspicious application.

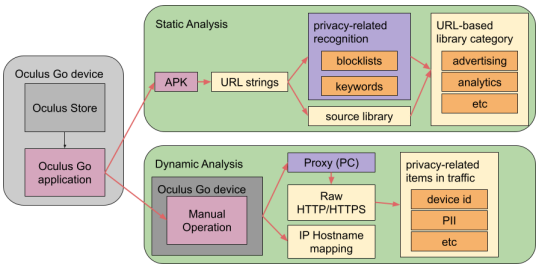

Fig. 1 Methodology overview.

3.3 Static Analysis

3.3.1 Approaching to the Source of URL Strings

Recognizing that APKs can be extracted into various formats using APK analysis tools, our study focused on the portion of code that can be executed when users install and launch applications. There are two primary locations where the source code and runtime executables are found: the Dalvik Executable (DEX) file, which contains the original source code of the application along with Java packages and shared objects (.so) located in the native library directory (/lib). For the sake of brevity, we refer to Java packages and native libraries as external libraries.

To extract the hard-coded URLs from DEX files, we used the Python Androguard [22] library for decompilation. Our focus in the static analysis is to address our first research question, which involves identifying imported packages and libraries referenced by URL strings.

For native libraries (also known as shared objects), we employed unzipped tools to obtain the desired directory structure. We searched for files using the .so extension. Native libraries may contain multiple versions of executables, depending on the Android device application binary interface (ABI).

3.3.2 Extracting Hard-coded URLs

In several cases, the URL strings present in the source code of APK files can be inferred as potential network communication destinations invoked by the relevant code, often referred to as cross-referencing URL strings. We used the Python module "re" to extract strings from both DEX files and native libraries by applying specific regular expression patterns. Our implementation focused on identifying strings that start with either the HTTP/HTTPS or FTP/FTPS protocols. We consider these protocols to be commonly employed in applications with various functionalities, such as making API calls or retrieving resources and data for display purposes.

3.3.3 Categorizing Usages of The URLs

To gain insight into libraries and external packages, it is essential to understand the purpose of including URLs in their code. Because our primary focus was to identify potential privacy concerns associated with network connections to unrelated advertising and analytics destinations, we categorized these URLs into general groups. To achieve this, we employed blocklist filtering and heuristic keyword matching to enhance our understanding of URLs.

We note that our categorization process plays a key role in exploring the privacy implications of VR applications. It is designed to uncover the complex, privacy-relevant connections be-

tween users and the servers they interact with, with a particular focus on VR-related sensor data. Using a methodology based on static analysis, we can preliminarily identify URLs that might be of interest for further investigation. However, this approach presented significant challenges, particularly in determining the specific nature of the data being transmitted. This problem is particularly pronounced in cases similar to data sent to services such as Google Analytics, where the lack of explicit naming or metadata leaves the contents of the data packets undefined.

Despite these challenges, our categorization method proved effective in several important ways. First, it allowed us to identify the types of advertising services that interact with VR applications, an essential step in understanding the privacy risks associated with VR technology. Second, it allowed us to map these services to their corresponding software packages or libraries, providing a clearer picture of how data is collected and transmitted within the VR ecosystem. This level of insight is critical to understanding the privacy dynamics at play in VR applications and forms a fundamental aspect of our analysis.

Blocklists Filtering: A modern client-side solution for preventing connections to undesirable destinations, particularly those related to advertising and aggressive analytics, is the use of Adblockers while browsing the Internet. These filter lists are frequently updated to keep up with rapidly changing links from suspicious servers. We selected the most effective blocklists, including Easylist, Easyprivacy [23], and Adguard-mobile [24], from popular Adblockers. Easylist detects advertising and analytics URLs, Easyprivacy identifies privacy-invasive URLs, and Adguard-mobile covers both on mobile devices.

Heuristic Keyword Matching: Similar to blocklists that use filter rules composed of regular expressions, patterns, and specific domains, we created our own keyword lists to facilitate a clear classification of URLs into our predefined categories. To minimize false positives in our categorization methodology, we selected only URLs that could be definitively classified into the intended groups. **Table 1** presents a detailed summary of the keywords used for the heuristic analysis, focusing on the identification of different categories within VR applications, along with a specialized list of exceptions related to advertising practices. This nuanced approach to keyword selection stems from the need to refine the categorization within the "Advertising" group. The common keywords "ad" and "ads" are prone to numerous false positives during the filtering process. To mitigate this, we have expanded our keyword list to include exceptions that have been carefully curated through manual examination of our collected URLs, thereby increasing the accuracy of our heuristic keyword matching technique.

Table 1 Comprehensive list of keywords utilized for heuristic analysis in identifying VR application categories and exceptional advertising practices.

Category	Keyword List
Analytics	track,lytics,fingerprint
Advertising	ad,ads
Fixed IP Address	ipv4, ipv6 regex
Dynamic Urls	%s, %d, %c
Documentation	doc,support,develop
Login Auth Related	login,auth
Cloud Service	cloud,cdn,aws
ChatGPT	chatgpt,chatapi,openai
Advertising (Exception rule)	head,adobe,load,ada,ready,badssl,address, salad,adult,grade,paradox,radio,read,adv, fade,shadow

URL Categories: The URL categories were designed based on two main ideas. First, we addressed our research questions by focusing on privacy-related URLs. Second, we strived to gain deeper insights into the included libraries and packages by categorizing frequently occurring and clearly identifiable services within the URL strings.

3.3.4 Categorizing External Sources

Through a static analysis, we identified Java packages and native libraries that contained hard-coded URLs. To gain insight into the purpose and usage of these libraries, we leveraged the predefined categories of URL strings. By collecting all the URLs within a given source, we could also assign the corresponding package or library to specific groups. Building on our previous classification, we established six categories of these external libraries.

3.3.5 Inconsistent External Sources

During the static analysis, we encountered inconsistent external libraries. Inconsistencies arise when a library is not only imported but also frequently extended because of the development of different VR applications. We detected these cases based on the number of URLs contained in these libraries.

3.3.6 Apps with Potential Privacy Concerns

In the final phase of the static analysis, we focused on the APKs of VR applications. We identified applications that demonstrated a large number of external libraries that fell within our defined privacy-related categories. These applications were flagged as potential candidates for further investigation. To conduct a more in-depth assessment, we conducted a dynamic analysis of a specific VR application.

3.4 Dynamic Analysis

Building upon the insights gained from the previous section on static analysis, we conducted a dynamic analysis to gain a deeper understanding of the network traffic on Oculus Go. In this section, we present our methodology for data acquisition, and measurements from the data are discussed in Section 4. It is important to note that the collected network traffic included both VR applications and platform system traffic.

3.4.1 Experimental Environment

In this study, we used Oculus Go. The Oculus Go used in the dynamic analysis was booted with root privileges, which were achieved by installing an unlocked version of the OS built from the official Oculus website.

To enable network data analysis, we used an Android debug bridge (ADB) [25] to reverse sockets to a prepared proxy server. For the proxy server, we used PolarProxy [26], a transparent TLS and SSL inspection proxy that allowed us to decrypt HTTP/HTTPS traffic. Coupled with the tool tcpdump, we were able to capture network traffic in raw HTTP/HTTP2 packets along with DNS domain mapping.

3.4.2 Extracting Privacy-related Items

We captured all the key-value pairs in HTTP headers and

HTTP bodies containing strings. We closely examined these items to determine whether user information or unique identifiers were being sent through the network to the servers, or if, in a worst-case scenario, such information was being shared across multiple domains. We applied the same blocklist filtering and keyword-matching methodology used in the static analysis to extract the items of interest and facilitate further analysis.

4. Measurement Results

4.1 Hardcoded URL Strings in the Wild

In this study, we collected 376 APKs from the Oculus Store. After extracting and filtering the strings, we obtained 6,422 hard-coded URLs from the dataset. Among these URLs, 1,946 were from Java packages, and 4,408 were from native libraries included in the code. We also found 68 URLs that were directly used in the base application code without being referred to by any class or function calls.

The distribution of URL strings among their sources is summarized in **Table 2**. We considered only one external library as a source if it contained at least one URL string. We discuss these external sources in the following sections.

Table 3 lists the top ten applications with the highest number of URL strings in their APKs. In addition, **Fig. 2** illustrates the distribution of the URL strings across the APK files. As shown in the figure, the majority of VR applications had fewer than 25 hard-coded URLs. The top ten applications contained a minimum of 74 URLs, which accounted for 19.18% of the total number of APKs collected.

Table 2 Hard-coded URLs among apps.

Type of Source	URLs	Unique URLs	Sources
Java Package	1,946	543	226
Native Library	4,408	846	62
others	68	66	4
Total	**6,422**	**1,413**	**292**

Table 3 Top-10 URLs imported in applications.

Application	# of URLs Found
com.GizmoVR.Virtual.Reality.Videos.GearVR	356
com.google.android.apps.youtube.vr.oculus	138
com.oculus.mediagallery	125
com.amazon.asxr	109
com.otherside.underworldoverlord_1080_build	105
com.grendelgames.relievr	93
com.oculus.cinema	81
com.visyon360.rtveplayer	77
com.ehi.viewnaija	74
jp.co.pixela.panomiru.viewer.gearvr	74
Dataset	**1,232**

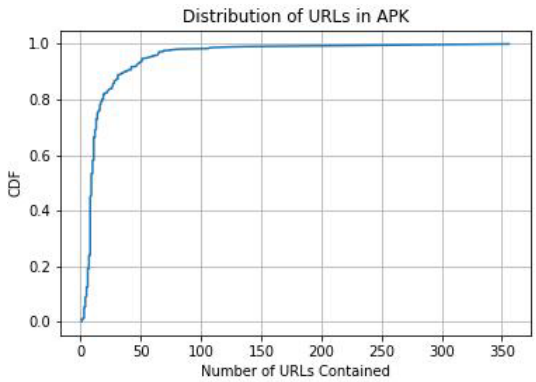

Fig. 2 The distribution of the number of URLs contained in APKs.

4.2 Statistics of the Hardcoded URL Strings

As mentioned previously, some URLs were duplicated owing to various factors. We compiled the most frequently appearing URLs along with their frequency of occurrence.

URLs from Oculus, Unity [27], Mono [28], and Google's Analytics [29] dominated the ecosystems in VR applications. URLs from Unity were often blocked by an Easyprivacy blocklist. The only URL that was not blocked by blocklists was an analytics-related URL that collects user data, similar to the previous URLs.

We observed a significant presence of Google-related domains. **Table 4** lists the top-level domains (eTLDs) with the highest frequencies in our dataset. While the URL "http://support.oculus.com" from "oculus.com" had the highest occurrence, domains related to game engine Unity and various Google services closely followed. When considering different Google domains such as "google-analytics.com", "doubleclick.net", and "gstatic.com", it is evident that Google-related domains surpass Unity in frequency. Among the Google-related domain URLs, we observed that several of them were blocked by the introduced blocklists, and it is apparent from the full URLs that these URLs are associated with advertising and analytics. Our categorization methodology handles the classification of such clear cases.

4.3 Java Packages and Native Libraries in the Dataset

We collected external URLs from Java packages and native libraries and conducted a static analysis. **Table 5** lists the most frequently imported external Java packages used for visualization. To further refine our analysis, we focused on identifying and categorizing third-level Java package names, such as com.google.android, to better understand the architecture and dependencies of modern VR applications. **Table 6** compiles these

Table 4 Top-10 eTLDs appeared in applications.

eTLD	Appearances
unity3d.com	1,209
google.com	758
oculus.com	569
googleapis.com	421
w3.org	333
microsoft.com	186
go-mono.com	128
winimage.com	101
adobe.com	87
xamarin.com	85
Total	**3,290 (51.23%)**

Table 5 Top 10 most imported external Java libraries.

Java Package	APKs
com.google.android.exoplayer2.drm	89 (23.67%)
com.google.android.exoplayer2.text.ttml	89 (23.67%)
com.google.vr.cardboard	65 (17.29%)
com.google.android.gms.common.internal	46 (12.23%)
com.google.android.gms.ads.identifier	25 (6.65%)
com.google.android.gms.drive	21 (5.59%)
com.google.android.gms.plus	20 (5.32%)
com.google.android.gms.internal	18 (4.79%)
com.google.android.gms.auth.api.credentials	18 (4.79%)
com.google.android.gms.games	16 (4.26%)
and 221 more...	

Table 6 Distribution of third-level Java package names in VR applications.

Java Package	APKs
com.google.android	131 (34.84%)
com.google.vr	65 (17.29%)
com.unity3d.services	15 (3.99%)
android.support.v4	15 (3.99%)
androidx.core.content	9 (2.39%)
io.fabric.sdk	9 (2.39%)
com.felixandpaul.FnPS	8 (2.39%)
com.facebook.internal	7 (2.13%)
com.google.tagmanager	7 (2.13%)
com.google.analytics	7 (2.13%)
and 123 more...	

Table 7 Top 10 most imported native libraries.

Native Library	APKs	#%
libvrapi	371	98.41%
libunity	264	70.03%
libvrintegrationloader	162	42.97%
libmono	128	33.95%
libil2cpp	75	19.89%
libmonobdwgc-2	70	18.57%
libMonoPosixHelper	56	14.85%
libgvr	48	12.73%
libOVRLipSync	42	11.14%
libUE4	16	4.24%
and 57 more...		

Table 8 Consistency of Java and native libraries.

Type of Source Library	Consistent	Inconsistent
Java Package	206	20
Native Library	42	20
Total	**248**	**40**

higher-level package names, highlighting their frequent reliance on common services from prominent companies such as Google, Unity, and Facebook. This approach allows us to uncover the essential support that these services provide, demonstrating their integral role in the functionality of VR applications. By focusing on third-level package names, we provide detailed insights into how external libraries are integrated and used, contributing to a broader understanding of the digital infrastructure underlying VR technology.

As discussed earlier, Google-related domains were highly prevalent in VR applications within our dataset. The top ten most imported external sources were from Google. Notably, nine out of the top ten sources belonged to the root package "com.google.android". It is already evident that there are services related to advertising and analytics, such as "com.google.android.gms.ads.identifier" and "com.google.android.gms.internal".

Regarding native libraries, **Table 7** lists the ten most imported native libraries in our dataset. From the table, it can be assumed that "libvrapi" is possibly required for developing VR applications on the Oculus platform, as it appeared in 371 out of the 376 APKs. The list also reveals that Unity is present in at least 70.03% of VR applications. It is important to note that we counted only specific external library sources if they contained hardcoded URL strings. The absence of a specific native library in our study does not imply that it was excluded from the application.

Table 8 summarizes the consistency of the Java packages and native libraries. We considered libraries to be consistent if multiple libraries with the same name had the same URLs. In other words, we define "consistent" if they contain the same URLs inside since our study focused on URLs. If a source is considered "inconsistent," it means it can be editable. Conversely, libraries with different URLs were considered inconsistent. Inconsistent libraries may have introduced bias into our categorization methodology because of their variable nature. Among the external library sources analyzed, 20 Java packages and 20 native libraries were considered inconsistent, accounting for approximately 13.89% of all external library sources.

Table 9 URLs blocked by blocklists.

Blocklist	Unique	Total	#% Total
easylist	28	132	2.06%
easyprivacy	18	724	11.27%
Adguard-Mobile	28	456	7.10%
Total	57	898	13.98%

Table 10 Categorized URLs.

Category	Unique	Total	#% Total
Analytics	39	813	12.66%
Advertising	53	534	8.32%
Fixed IP Address	10	29	0.45%
Dynamic Urls	61	283	4.41%
Documentation	267	1,437	22.38%
Login Auth Related	51	297	4.62%
Cloud Service	56	1,069	16.65%
ChatGPT	0	0	0.00%
Total Categorized	482	3,346	52.10%

4.4 Categorizing Intentions by URL Strings

In this section, we provide an overview of the distribution of each category and the content identified as related to privacy issues.

First, we examined one of our key static analysis methods: blocklist filtering. As mentioned earlier, we applied various blocklists used by different adblock providers using the public Python modules adblockparser and abp_blocklist_parser. Note that an URL can be blocked using multiple blocklists. The results are listed in **Table 9**.

Table 10 shows that 52.10% of the URLs in our dataset could be categorized using the proposed method, whereas the remaining URLs were not considered because we could not determine their usage based on the URL strings alone. We also used blocklists to categorize URLs that could not be identified using keyword matching. The analytics category included URLs detected by the Easyprivacy blocklist, and the advertising category included URLs detected by the Easylist and Adguard-mobile blocklists.

Because our focus was on privacy-related URLs, we applied heuristic keyword matching to the Privacy and Advertising categories to avoid false positives. We identified hard-coded IP addresses categorized as fixed IP addresses and dynamic URL categories, including URLs with %s, %d, or %c in the hard-coded strings. It is important to note that a single URL can be assigned to multiple categories (e.g., `https://api.uca.cloud.unity3d.com/v1/events` was blocked by Easyprivacy and categorized as Analytics but was also categorized as a Cloud Service). The documentation category had the highest frequency in our methodology, as several applications provided buttons for Q&A, help, and support, resulting in a higher occurrence of URLs related to these resources (22.38% of all URLs). Cloud services were also a popular category because several applications used

cloud services. Although the trending technique ChatGPT [30] has gained popularity, we did not find any URLs related to ChatGPT in our dataset using keyword matching. This may be due to the fact that Oculus Go ceased application updates before ChatGPT was introduced. We believe that variances may be observed if we conduct this static analysis on the Oculus Quest Series.

After categorizing the URLs in our dataset, we proceeded to analyze the external library sources. We consolidated the categories into six groups, merging fixed IP addresses and dynamic URLs into the inconsistent category, and removing ChatGPT. **Table 11** summarizes the results of the proposed labeling methodology for external library sources. Among the labeled sources, we identified 46 libraries related to analytics and advertising among our labeled sources. The inconsistent category, which included sources using dynamic URLs or IP addresses, had the highest proportion (12.97%) among all sources.

Table 12 lists the top ten potential privacy-related sources determined by the total number of labels from the analytics and advertising categories. Among the top ten privacy-related sources, there were five native libraries, four Java packages, and one VR application source code. We observed that `com/google/android/gms/internal` and `libunity` appeared in both the top ten imported native libraries and the top ten imported Java packages with proportions of 4.79% and 98.41% in our APK dataset, respectively, which was significant.

Finally, we conclude our analysis with 376 VR applications in the dataset. Using the categories generated, we identified applications that imported external library sources labeled as analytics or advertising. **Figure 3** shows the distribution of the imported privacy-related sources. To further validate the findings from the static analysis, we conducted a dynamic analysis.

4.5 Network Traffic Generated by VR Applications in Practice

To gain deeper insight, we selected a VR application that imported the most privacy-related external library sources into its APK for our dynamic analysis. This VR application allows users to watch videos on its platform. To launch this application, the Oculus system requires users to download an official application known as Oculus TV, on which the VR application can be executed. We captured all traffic during our experiment, including system traffic and Oculus TV traffic. We filtered out most packets that did not contain the information of interest, focusing on HTTP/HTTP2 data after decryption. This resulted in 1,275 packets with 398 unique URLs for our analysis, with three unique URLs being blocked by our blocklists.

However, it is important to note that the URL `611d5307(snip...).cws.conviva.com` was identified as an extension of `cws.conviva.com`, and `t.appsflyer.com` was categorized as a dynamic URL in the APK, formatted as `t.%s`. Additionally, since our PCAP data encompassed system traffic, some domains may relate to system packages rather than the targeted VR application.

Among all the packets, keyword matching extracted 18 different items from the HTTP header and 46 different items from the HTTP body, which were identified as potentially privacy-related based on the proposed methodology. Based on the domain names, `graph.oculus.com` includes the ID and device information in the HTTP headers. In the HTTP body, four of the nine domains (44.44%) matched the items with our keywords in their communication. Among these, three were advertising and analytics services. `t.appsflyer.com` collected 12 matched items, including GAID, android_id, and device information. `611d5307(snip...).cws.conviva.com` and `sessions.bugsnag.com` collected 11 and 6 matched items, respectively, including IDs and device information. `graph.oculus.com` collected three items, including app_id, token, and device id.

We also observed that the proposed keyword-matching

Table 11 Categorized source libraries.

Category	Libraries	#%
Analytics	28	9.56%
Advertising	25	8.53%
Inconsistent	38	12.97%
Documentation	25	8.53%
Login Authentication Related	23	7.85%
Cloud Service	31	10.58%
Total Categorized	127	43.34%

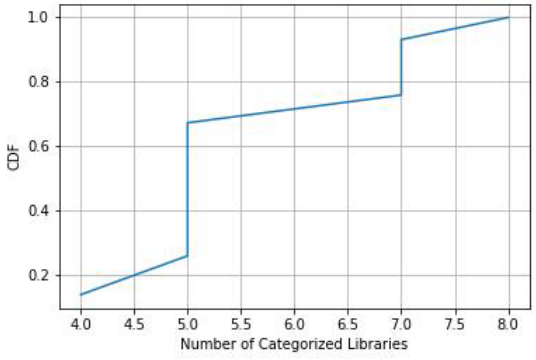

Fig. 3 The distribution of imported privacy-related sources among VR applications.

Table 12 Top 10 most likely privacy-related libraries. The columns display the distribution of URL labels across each external library source.

Source Library	Privacy	Ads	IP	Dynamic	Docs	Auth	Cloud
com.google.android.gms.internal	4	7	0	0	0	0	0
libalohabromium	3	7	0	3	7	2	1
Youtube VR (Oculus)	1	9	1	0	1	8	0
com.google.android.gms.internal.ads	1	9	0	0	1	0	0
com.google.firebase.crashlytics.internal.settings	5	0	0	4	0	0	0
libcronet	0	5	1	0	2	0	2
libil2cpp	3	2	0	0	230	0	1
com.xiaomi.mistatistic.sdk.a	4	1	1	0	0	0	0
com.flurry.sdk	0	4	0	0	0	0	0
libunity	2	1	0	1	0	0	3

methodology could not be applied to Google-related advertising traffic packets because of the use of hidden key names. Google employed indices with keys such as `cd12=163a8635f18bfdd7&cd121=&cd44=unspecified`. In a single packet, up to 427 items were collected.

5. Threat Case Studies

After confirming the presence of privacy concerns within the VR application dataset, we explored threat case studies related to VR applications and the Oculus platform.

5.1 Lack of Awareness of Privacy Issues and Data Leakage

As mentioned previously, the adopted methodology revealed that VR applications send privacy-related information to platforms, applications, and third-party servers. In well-established computer and mobile ecosystems, user agreement and privacy policy usage are highly adopted by most application developers and are well managed by operating systems and application stores. However, during our VR application experiments on Oculus Go and Oculus Quest2, we observed that user agreement pop-ups and privacy policies were not frequently presented. Specifically, in the case of Red Bull TV, no information regarding the inclusion of "google-analytics" [29], "bugsnag" [31], "appsflyer" [32], and "conviva" [33] was disclosed during regular Oculus Go usage, nor the connection to "graph.oculus.com" by the platform or application. In comparison with computers and mobile devices, Oculus devices lack well-established transparency concerning user privacy, presenting a concerning situation. We believe that VR applications could potentially be misused by data-hungry companies to collect user information before Oculus platform establishes stringent rules and measures.

5.2 Absence of Optional Privacy Protections

Following the discovery of privacy-related connections, the lack of optional measures to protect users from VR applications connected to remote advertising and analytics servers without their consent has become a critical privacy concern. Oculus VR devices have design choices that are not friendly to privacy-conscious users. Currently, the only one solution to address this privacy issue is to avoid purchasing or using Oculus devices.

Two initialization steps are required for Oculus devices. First, users are required to bind Oculus devices to their mobile phones through an Oculus mobile application. Second, a meta-analysis is mandatory. Modern mobile phones require users to log in with their Google or Apple accounts to use the respective application stores. Therefore, binding a mobile phone is not justifiable and can result in privacy concerns related to sharing unrelated information with Meta/Oculus.

Furthermore, once inside the Oculus device menu, no option is available to ensure privacy during VR application usage or internet surfing through the default browser. The operating system of Oculus devices is fixed and restricted, prohibiting users from protecting themselves during VR application use. Attempts to download other browsers or tools have proven futile, as no such options are available in the Oculus Store. The only recourse is to manually install non-root-privileged applications using ADB, which is a non-straightforward solution for regular users.

5.3 Undeclared Usage of External Libraries

The use of external libraries or sources is a common practice in the efficient development of computer science. Developers should not only include user agreements and privacy policies for their application functionalities but also disclose the usage of external library sources. Trimananda et al. [2] reported that VR applications collected on Oculus Quest2 often lacked privacy policies regarding the use of external library sources.

6. Discussion

In this section, we discuss possible mitigation for this problem from the perspectives of developers, privacy researchers, and regular VR users. We conclude with key points of attention for future VR devices and potential directions for future research.

6.1 Enhancements to User Privacy Protection

As mentioned previously, the root cause of privacy concerns was the lack of clear declarations regarding the use of advertising and analytics in VR applications during our experiment. Depending on the operating system adopted by the device manufacturer, the options available to users to protect their privacy may vary. Our focus was on the operating system adopted by Oculus, which is based on Android.

We propose the following enhancements to ensure user privacy:

Providing Privacy Options on the Oculus Platform: Currently, the lack of options for users to protect themselves is a significant issue. Oculus requires users to log in to their Meta accounts to use their VR devices. We urge the Oculus platform to provide privacy options similar to those of modern web browsers, including features that prevent user tracking, such as disabling cookies, fingerprinting, tracking identifiers, and rejecting analytics. Additionally, incorporating a "Do Not Track" signal to avoid tracking through advertising and analytics services would be beneficial. These functionalities protect the rights of users.

Application Development Regulations for VR: To address the lack of awareness of privacy issues, we propose that VR applications should contain visible user consent prompts upon their initial launch. In addition, the use of the included external library sources should be explicitly stated in the user agreement of the VR application. We recommend that VR platforms enforce application developers to adhere to strict rules during application development and ensure that user consent is obtained at the initial launch of each application. In our measurement study, we observed that privacy-related connections could be identified through an inspection of external library sources. Therefore, VR applications must disclose the external library sources and related information. Introducing transparency in data collection and the services included can significantly enhance trust in applications.

6.2 VR Applications on Other Devices

The market offers various VR devices manufactured by different companies and tailored for different purposes. VR devices possess a wide array of features and capabilities that meet the di-

verse needs and preferences of users, including their operating systems. In this study, Oculus OS was based on Android. We developed our methodology by considering the Oculus Go as a mobile wearable device because of its similar operating system and permission management. The measurement study only represented Oculus Go, and possibly Oculus Quest series. We speculate that VR devices from other manufacturers may also face similar privacy issues and that their operating systems may not offer the same degree of permission control as Android. To analyze VR applications on other operating systems, such as Linux, Windows, Playstation 4, we need to understand the application file format on these platforms. Conducting a comprehensive VR privacy analysis on other platforms is essential.

In this context, our study utilized the Oculus Go as our experimental device, acknowledging its status as an increasingly obsolete model within the rapidly advancing virtual reality technology landscape. Our findings revealed significant use of the now-depreciated libvrapi as a native library across a wide range of VR applications. This underscores the critical importance of continued monitoring and evaluation of the evolving vrapi library, especially as we navigate the transition. As part of our future efforts, we intend to expand our research to further understand the use and impact of these libraries within the VR development ecosystem.

6.3 Meta Oculus Browser

During our experiments with the Oculus Go, we initially attempted to analyze the pre-installed applications within the Oculus system. However, we ultimately decided to focus our dataset on applications available in the Oculus Store. We also discovered that the default internet browser on the Oculus Go, known as the Meta Oculus browser, established several connections to advertising and analytics destinations during browsing sessions. Our inspection revealed that Meta Oculus browser is built based on chromium [34]; however, it lacks several customizable options. Although it offers an option to disable analytics from Meta Facebook, it does not sufficiently prevent user-tracking services by other parties. Furthermore, users have no choice but to select an alternative browser because there are no other browsers available in the Oculus Store. Investigating privacy issues in Meta Oculus browser could be a potential area for future research.

6.4 Differences Between Android APKs and Oculus-specific APKs

We adopted several Android APK analyzing techniques from past research, and we hope to understand the data used in communications currently. In conclusion, our experiments involved the analysis of Oculus-specific APKs as normal Android APKs, such that no significant differences between Android apps and Oculus-specific apps were observed. However, we believe that the data being transmitted might vary. Consequently, we aim to capture VR-specific user information in tracking services such as advertising and analytics. We believe that our future study will involve a large-scale, high-level perspective data investigation, to fully understand possible privacy concerns.

6.5 Dynamic Analysis: Potential and Limitations

In this study, we primarily used static analysis to evaluate the privacy implications of hardcoded advertising URLs within VR applications. This approach allowed us to examine application binaries to identify privacy-relevant URLs without running the applications. While static analysis is effective at uncovering potential privacy risks, it does not provide insight into whether these URLs are actively used during the application's runtime. Given the limitations of static analysis alone, we see dynamic analysis as a promising complement. Dynamic analysis involves running applications in a controlled environment to observe their actual behavior, including network requests, API calls, and other runtime operations that could affect user privacy. This method can provide a more accurate picture of how and when privacy-relevant URLs are used.

However, dynamic analysis of VR applications comes with its own set of challenges. The primary limitation is the difficulty in ensuring comprehensive coverage of all application functionality due to the complex and interactive nature of VR environments. Achieving this requires sophisticated automation capable of simulating a wide range of user interactions, which remains a significant hurdle. In addition, the resource-intensive nature of dynamic analysis limits its scalability across multiple applications and platforms.

To summarize, while our study focused primarily on static analysis due to its feasibility and the scope of our research, we recognize the value of integrating dynamic analysis for a more thorough privacy assessment. Future work will explore the development of automated tools for dynamic analysis in VR environments to overcome current limitations and provide a comprehensive view of privacy risks.

7. Conclusion

We summarize the potential privacy concerns regarding VR applications and their platforms following the disclosure of advertising and analytics services. We anticipate that privacy leakage may not be confined to Oculus Go, as different VR device manufacturers may exhibit varying privacy-invasive behaviors depending on the adopted operating system and the optional measurements provided by the system. Our future goal is to conduct large-scale VR privacy analyses on other platforms. Provided that advertising and analytics services exist, user rights must always be considered from a critical perspective.

References

[1] Nair, V., Garrido, G.M., Song, D. and O'Brien, J.: Exploring the Privacy Risks of Adversarial VR Game Design, *23rd Privacy Enhancing Technologies Symposium* (*PETS 23*) (online), DOI: 10.56553/popets-2023-0108 (2023).

[2] Trimananda, R., Le, H., Cui, H., Ho, J.T., Shuba, A. and Markopoulou, A.: OVRseen: Auditing Network Traffic and Privacy Policies in Oculus VR, *31st USENIX Security Symposium* (*USENIX Security 22*), pp.3789–3806, USENIX Association (2022).

[3] VR Headset Sales and Market Share in 2023 (How Many Sold?), available from ⟨https://thesmallbusinessblog.net/vr-headset-sales-and-market-share/⟩.

[4] Nair, V., Guo, W., Mattern, J., Wang, R., O'Brien, J.F., Rosenberg, L. and Song, D.: Unique identification of 50,000+ virtual reality users from head & hand motion data, *32nd USENIX Security Symposium* (*USENIX Security 23*), pp.895–910 (2023).

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

[6] Luo, S., Nguyen, A., Farooq, H., Sun, K. and Yan, Z.: Eavesdropping on Controller Acoustic Emanation for Keystroke Inference Attack in Virtual Reality, *The Network and Distributed System Security Symposium* (*NDSS*) (2024).

[7] Farrukh, H., Mohamed, R., Nare, A., Bianchi, A. and Celik, Z.B.: LocIn: Inferring Semantic Location from Spatial Maps in Mixed Reality, *32nd USENIX Security Symposium* (*USENIX Security 23*), pp.877–894 (2023).

[8] Nair, V., Garrido, G.M. and Song, D.: Exploring the unprecedented privacy risks of the metaverse, arXiv preprint arXiv:2207.13176 (2022).

[9] Bierma, M., Gustafson, E., Erickson, J., Fritz, D. and Choe, Y.R.: Andlantis: Large-scale Android dynamic analysis, arXiv preprint arXiv:1410.7751 (2014).

[10] Li, Y., Yang, Z., Guo, Y. and Chen, X.: Droidbot: A lightweight ui-guided test input generator for android, *2017 IEEE/ACM 39th International Conference on Software Engineering Companion* (*ICSE-C*), pp.23–26, IEEE (2017).

[11] Tramontana, P., Amalfitano, D., Amatucci, N. and Fasolino, A.R.: Automated functional testing of mobile applications: A systematic mapping study, *Software Quality Journal*, Vol.27, pp.149–201 (2019).

[12] Pariwono, E., Chiba, D., Akiyama, M. and Mori, T.: Don't throw me away: Threats caused by the abandoned internet resources used by android apps, *Proc. 2018 on Asia Conference on Computer and Communications Security*, pp.147–158 (2018).

[13] Luoshi, Z., Yan, N., Xiao, W., Zhaoguo, W. and Yibo, X.: A3: Automatic analysis of android malware, *1st International Workshop on Cloud Computing and Information Security*, pp.89–93, Atlantis Press (2013).

[14] Calciati, P., Kuznetsov, K., Bai, X. and Gorla, A.: What did really change with the new release of the app?, *Proc. 15th International Conference on Mining Software Repositories*, pp.142–152 (2018).

[15] Pourali, S., Samarasinghe, N. and Mannan, M.: Hidden in Plain Sight: Exploring Encrypted Channels in Android Apps, *Proc. 2022 ACM SIGSAC Conference on Computer and Communications Security*, *CCS '22*, pp.2445–2458, Association for Computing Machinery (online), DOI: 10.1145/3548606.3560665 (2022).

[16] Kuzuno, H. and Tonami, S.: Signature generation for sensitive information leakage in android applications, *2013 IEEE 29th International Conference on Data Engineering Workshops* (*ICDEW*), pp.112–119 (online), DOI: 10.1109/ICDEW.2013.6547438 (2013).

[17] Oculus Go - Specification, available from ⟨https://www.niora.net/en/p/oculus_go#⟩.

[18] Unlocking Oculus Go, available from ⟨https://developer.oculus.com/blog/unlocking-oculus-go/⟩.

[19] APKPure, available from ⟨https://apkpure.com⟩.

[20] APKMirror, available from ⟨https://www.apkmirror.com/⟩.

[21] Go Store: VR Games, Apps, & More, available from ⟨https://www.oculus.com/experiences/go/⟩.

[22] Androguard - GitHub, available from ⟨https://github.com/androguard/androguard⟩.

[23] EasyList - Overview, available from ⟨https://easylist.to/⟩.

[24] Adguard, available from ⟨https://adguard.com/⟩.

[25] Android Debug Bridge (adb) — Android Studio, available from ⟨https://developer.android.com/tools/adb⟩.

[26] PolarProxy TLS proxy - Netresec, available from ⟨https://www.netresec.com/?page=PolarProxy⟩.

[27] Unity Real-Time Development Platform, available from ⟨https://unity.com/⟩.

[28] Home — Mono, available from ⟨https://www.mono-project.com/⟩.

[29] Google Analytics, available from ⟨https://analytics.google.com/analytics/web/provision/#/provision⟩.

[30] Introducing ChatGPT - OpenAI, available from ⟨https://openai.com/blog/chatgpt⟩.

[31] BugSnag: Error Monitoring & App Stability Management, available from ⟨https://www.bugsnag.com/⟩.

[32] AppsFlyer — Make good data-driven choices, available from ⟨https://www.appsflyer.com/⟩.

[33] Conviva — #1 in Streaming Analytics, available from ⟨https://www.conviva.com/⟩.

[34] The Chromium Projects, available from ⟨https://www.chromium.org/chromium-projects/⟩.

Shu-Pei Huang graduated from Waseda University, specializing in the study of user privacy within mobile devices, including smartphones, wearables, and VR. His research focuses on developing solutions that enhance user understanding while safeguarding their privacy. He aspires to bridge the gap between technology and user comprehension, envisioning a future where users feel secure and informed in their interactions with evolving technologies.

Takuya Watanabe received B.E. and M.E. degrees in computer science and engineering, and a Ph.D. in engineering from Waseda University in 2014, 2016, and 2020, respectively. Since joining the Nippon Telegraph and Telephone Corporation (NTT) in 2016, he has been engaged in research on system security and privacy from the perspective of an attacker, particularly in web and mobile applications. He is currently with the Cyber Security Project of NTT Social Informatics Laboratories.

Mitsuaki Akiyama received his M.E. and Ph.D. in engineering from Nara Institute of Science and Technology in 2007 and 2013. Since joining the Nippon Telegraph and Telephone Corporation (NTT) in 2007, he has been engaged in research and development on cybersecurity. He is currently a Senior Distinguished Researcher at NTT Social Informatics Laboratories. He has received the Cybersecurity Encouragement Award from the Minister for Internal Affairs and Communications in 2020 and IPSJ/IEEE Computer Society Young Computer Researcher Award in 2022. His research interests include cybersecurity measurements, offensive security, and usable security and privacy. He is a senior member of IPSJ and a member of IEEE, SIGCHI, and IEICE.

Tatsuya Mori is currently a professor at Waseda University, Tokyo, Japan. He received B.E. and M.E. degrees in Applied Physics and Ph.D. degree in Information Science from Waseda University in 1997, 1999, and 2005, respectively. He joined NTT laboratory in 1999 and moved to Waseda University in 2013. From March 2007 to March 2008, he was a visiting researcher at the University of Wisconsin-Madison. He has been engaged in research on network measurement, security, and privacy. He has received several best paper awards, including NDSS 2020 and EuroUSEC 2021. Dr. Mori is a member of the ACM, IEEE, IEICE, and IPSJ.