A Artifact Appendix

A.1 Abstract

We developed OVRSEEN, a methodology and system for collecting, and analyzing network traffic and privacy policies on OVR. OVRSEEN consists of two main parts: network traffic and privacy policy. OVRSEEN's network traffic part consists of traffic collection and post-processing. First, traffic collection consists of repackaging the app's APK, and running the traffic decryption scripts simultaneously with AntMonitor, the traffic collection app. Second, post-processing consists of scripts that perform analysis on the collected network traffic.

OVRSEEN's privacy policy part consists of network-to-policy consistency analysis, and purpose extraction. First, the network-to-policy consistency analysis comes in the form of PoliCheck that has been adapted with VR data and entity ontologies to perform its analysis on VR apps. Second, the purpose extraction comes in the form of scripts that send privacy policies to Polisis website (https://www.pribot.org/) using the provided REST APIs, and scripts that provide a translation from PoliCheck data flows into text segments that have been annotated by Polisis with purposes.

A.2 Artifact check-list (meta-information)

- **Datasets:** lists of apps, network traffic dataset, privacy policy files, manual validation of PoliCheck and Polisis spreadsheets, and intermediate output files.
- **Run-time environment:** Python scripts tested on Python 3.8 and 3.9; we also provide a VM that runs Ubuntu 20.04.3 LTS with all the dependencies installed.
- **Hardware:** an Oculus Quest 2 device and a standard machine with Linux/MacOS (or using the provided VM).
- **Execution:** We have provided a set of steps to demo OVRseen (*e.g.,* for artifact evaluation), which takes a few hours approximately.
- **Metrics:** number of packets, TCP flows, numbers of apps, domain names/entities, and data flows.
- **Output:** console output (*e.g.,* debug/error messages), intermediate output files, final analysis results.
- **Experiments:** Please see OVRSEEN's Github page (*i.e.,* "Try OVRSEEN Yourself" Wiki page in particular).
- **How much disk space required (approximately)?:** Our VM provides 30GB of disk space (more than 25GB will be occupied when running OVRSEEN).
- **How much time is needed to prepare workflow (approximately)?:** Downloading and booting up the provided VM should take less than 1 hour (plus additional time to install VirtualBox/VMWare).
- **How much time is needed to complete experiments (approximately)?:** We have a quick demo for OVRSEEN that would take a few hours (at least 2 hours).
- **Publicly available?:** We have released OVRSEEN, along with the datasets, publicly.
- **Code licenses (if publicly available)?:** Code licenses information is available on OVRSEEN's Github page.
- **Data licenses (if publicly available)?:** Datasets licenses information is available on the datasets release page.
- **Archived (provide DOI or stable reference)?:** https://doi.org/10.5281/zenodo.5565170

A.3 Description

A.3.1 How to access

We have made OVRSEEN available at https://athinagroup.eng.uci.edu/projects/ovrseen/ and our datasets at https://athinagroup.eng.uci.edu/projects/ovrseen-datasets/. These two links have the complete information about the paper, OVRSEEN, and datasets. For convenience, the link to OVRSEEN's Github page is https://github.com/UCI-Networking-Group/OVRseen.

A.3.2 Hardware dependencies

OVRSEEN's network traffic and privacy policy analyses can be run on our datasets on a standard machine that runs Linux/MacOS, or using the provided VM. The information to download the VM can be found at https://github.com/UCI-Networking-Group/OVRseen#getting-started. Please see the "Virtual Machine" section: https://github.com/UCI-Networking-Group/OVRseen#virtual-machine.

OVRSEEN's network traffic collection needs a real Oculus Quest 2 device for the most part. Thus, our quick demo mainly assumes that one just runs OVRSEEN on our datasets (without collecting network traffic on the device).

A.3.3 Software dependencies

The dependencies for OVRSEEN are explained in detail on OVRSEEN's Github Wiki page at https://github.com/UCI-Networking-Group/OVRseen/wiki#dependencies. These dependencies have been properly installed and set up in the provided VM.

A.3.4 Datasets

OVRSEEN has a number of datasets: (1) list of apps in our corpus (*i.e.,* two files that contain apps information obtained by crawling the Oculus and SideQuest app stores, and four files that contain the information of the top 150 apps); (2) network traffic dataset in the form of PCAP files from 140 VR apps; (3) 102 privacy policy files; (4) manual validation results for PoliCheck and Polisis (*i.e.,* two spreadsheets); and (5) intermediate outputs (*i.e.,* a CSV file containing TCP flows, pre-processed privacy policy files, a CSV file containing data flows, PoliCheck output files, JSON files containing Polisis output for text segment annotation, and a CSV file that contains the translation/mapping from PoliCheck data flows into the annotated text segments from Polisis). For artifact evaluation purposes, we provide the download link for our datasets through hotcrp.com. In the future, these datasets will be shared

to OVRSEEN users after they submit the consent form at https://athinagroup.eng.uci.edu/projects/ovrseen-datasets/.

A.3.5 Security, privacy, and ethical concerns

Please keep in mind that VR apps collect PII and other sensitive information: OVRSEEN collects such sensitive information as well when used to collect and analyze network traffic. Our network traffic dataset, however, contains PII that is associated only with a test account/persona (*i.e.,* no human subjects were involved).

A.4 Installation

We have provided complete instructions on how to download, install, and use OVRSEEN on its Github page: https://github.com/UCI-Networking-Group/OVRseen (please see the README and Wiki pages). The instructions also include how to download and use our VM that has all the dependencies installed.

A.5 Experiment workflow

We have created a Wiki page (called "Try OVRSEEN Yourself") on OVRSEEN's Github page (*i.e.,* https://github.com/UCI-Networking-Group/OVRseen/wiki/Try-OVRseen-Yourself). This Wiki page contains a set of steps that one can follow to quickly demo OVRSEEN's workflow using our datasets.

A.6 Evaluation and expected results

Main claims. Our paper presents OVRSEEN, a methodology and system for collecting, and analyzing network traffic and privacy policies on OVR. In our paper, we first claimed that, using OVRSEEN, we decrypted, captured, and analyzed network traffic of VR apps. Then, we made the following claims based on our findings:

- *More centralized, more tracking, but less advertising*: the OVR ecosystem is more centralized, and driven by tracking and analytics, instead of by third-party advertising.
- *Data types exposure*: data types exposed by VR apps include the traditional PII and, most notably, VR specific data types.
- *Inconsistent disclosures*: the majority of data type exposures of an app are inconsistent with the disclosures in the app's privacy policy.
- *Non-core purposes*: many data exposures occurred for purposes unrelated to an app's core functionality.

Key results. Next, we outline the key results that support our main claims:

- *More centralized, more tracking, but less advertising*: We found that OVR exposes data primarily to tracking and analytics services, and has a less diverse tracking ecosystem. We found no evidence of data exposure to advertising services as ads on OVR is still in its infancy (see Section 3.3).
- *Data types exposure*: We discovered that there were 21 data types exposed, namely *PII*, *Fingerprint*, and *VR Sensory Data* data types (see Section 3.4).
- *Inconsistent disclosures*: First, we found that approximately 70% of data flows from VR apps were inconsistent with their privacy policies: only 30% were consistent. Second, apps' privacy policies often neglected declaring privacy policies from the libraries they used. We discovered that by including these other parties' privacy policies in OVRSEEN's network-to-policy consistency analysis, 74% of data flows became consistent (see Section 4.1).
- *Non-core purposes*: We discovered that there were 69% of data flows that have purposes unrelated to the core functionality, *e.g.,* advertising, marketing campaigns, and analytics (see Section 4.2).

Reproducing key results. To reproduce the key results, we recommend our artifact reviewers to follow the instructions at https://github.com/UCI-Networking-Group/OVRseen/wiki/Try-OVRseen-Yourself that we also describe in detail in the following.

To prepare OVRSEEN, please follow the instructions in the "Virtual Machine" section in the README (*i.e.,* https://github.com/UCI-Networking-Group/OVRseen#virtual-machine) to first download and run our pre-configured VM (with all the dependencies installed). Then, our reviewers can download, install (*e.g.,* in the home directory), and run OVRSEEN on the running VM.

First, collecting network traffic using OVRSEEN is not possible without installing AntMonitor and running the certificate validation bypass scripts on a real Oculus Quest 2 device. Further, it is impractical to repeat our network traffic collection steps on 140 VR apps for the purpose of artifact evaluation. Thus, we release our network traffic dataset in the form of PCAP files that we captured using AntMonitor and the certificate validation bypass scripts. We welcome our reviewers to download and use our datasets and run OVRSEEN on them: this will be sufficient to reproduce all results we reported in our paper.

Since OVRSEEN's *traffic collection* is impractical to perform for our reviewers, we invite our reviewers to look at the complete source code for AntMonitor and the certificate validation bypass scripts. We also invite our reviewers to look at https://github.com/UCI-Networking-Group/OVRseen/wiki/Traffic-Collection to review the instructions: these have been tested using our VM and a real Quest 2 device. One part of the OVRSEEN's *traffic collection* that our reviewers can still run in the quick demo is the app repackaging step—we provide a sample app to test with (please see https://github.com/UCI-Networking-Group/OVRseen/wiki/Try-OVRseen-Yourself#traffic-collection).

Next, using the provided network traffic dataset (and our other datasets), our reviewers can perform the following steps when running OVRSEEN.

- *More centralized, more tracking, but less advertising*: OVRSEEN's *post-processing* scripts can be run to analyze the network traffic dataset we provide; the final product of OVRSEEN's *post-processing* is a combined CSV file that contains information on TCP flows: each TCP flow, among other information, records app ID (*i.e.,* app name), PII types, and endpoints; for now, we recommend that OVRSEEN is run partially on our network traffic dataset (due to the limitations of RAM and disk space in the VM), but we provide the intermediate outputs generated when we ran OVRSEEN on our complete

network traffic dataset and scripts that use these outputs to reproduce Table 1 (discussed in Section 3.2.1), and Table 2 and Figure 2 (discussed in Section 3.3) in our paper.

- *Data types exposure*: OVRSEEN's *post-processing* also includes scripts that use the intermediate outputs to reproduce Table 3 that summarizes data types exposures, destinations, and blocklists' effectiveness for 21 data types; we discuss these in Section 3.4 in our paper.
- *Inconsistent disclosures*: OVRSEEN's *network-to-policy consistency* analysis consists of PoliCheck that has been adapted and improved for VR apps, and VR (data and entity) ontologies; our reviewers can run OVRSEEN's *network-to-policy consistency* analysis using the provided intermediate outputs to reproduce our results reported in Section 4 in our paper: more specifically, we provide scripts to reproduce Figures 4, 5, and 6; further, we release the HTML files of the 102 privacy policies we collected, the scripts that pre-process these into text files suitable as an input to PoliCheck, and the text files themselves; in addition to privacy policies, PoliCheck also takes a CSV file that contains data flows information extracted from the CSV file that contains TCP flows information (*i.e.,* output of OVRSEEN's *post-processing*): we release the scripts to produce this data flows CSV file along with the CSV file itself; finally, we also provide the output CSV files from PoliCheck's disclosure classification and the spreadsheet that contains the results of our manual validation for PoliCheck.
- *Non-core purposes*: OVRSEEN's *purpose extraction* consists of scripts that perform the extraction of purposes for text segments in privacy policies using Polisis, and scripts that perform translation/mapping from PoliCheck data flows to the text segments annotated by Polisis; Polisis website requires a special token for the APIs to work with our scripts; unfortunately, while the token can be acquired by contacting Polisis authors, they had to discontinue their online service as of September 2021 due to some technical issue; thus, we provide the JSON files that contain the Polisis analysis output we obtained for our 102 privacy policies; using these files, one can run our scripts to reproduce the statistics/results we reported in Section 4.2 and Figure 7 in our paper; further, we also release the spreadsheet that contains the results of our manual validation for Polisis.

Thus, we believe that the instructions we provide at https://github.com/UCI-Networking-Group/OVRseen/wiki/Try-OVRseen-Yourself are sufficient to quickly demo OVRSEEN. While, parts of OVRSEEN's workflow will not be possible to perform (*e.g.,* the network traffic collection that requires a Quest 2 device, the Polisis online service that has been discontinued, *etc.*), these instructions, coupled with our datasets, will allow our artifact reviewers to reproduce our (key) results to support the main claims in the paper.

A.7 Experiment customization

If one has a local machine that allows the provided VM to be provisioned with more RAM and disk space, they can try to increase the RAM and disk space for the VM, and run OVRSEEN on our entire datasets. Please see https://github.com/UCI-Networking-Group/OVRseen/wiki/Try-OVRseen-Yourself for more information.

Further, OVRSEEN can be used to collect network traffic from other apps on Quest 2. The PCAP files can then be post-processed and analyzed (together with the apps' privacy policies) using OVRSEEN. For other devices, other than Quest 2 (or even non-VR devices), one has to adapt the network traffic collection part to decrypt network traffic on the device. Other parts of OVRSEEN also may need adjustments if the collected network traffic contains new data types. For instance, new network traffic dataset and/or privacy policies may change PoliCheck's data and entity ontologies.

Additionally, we also release our app crawler scripts that we used to collect app information we present in our lists of apps, and the curated lists of top apps. Please see https://github.com/UCI-Networking-Group/OVRseen/wiki/App-Corpus for more information on how to use them. Nevertheless, we do not consider these crawler scripts to be part of the main OVRSEEN's workflow.