USENIX'23 Artifact Appendix: Erebus: Access Control for Augmented Reality Systems

Yoonsang Kim[*] and Sanket Goutam[*], Amir Rahmati, Arie Kaufman
Stony Brook University
{yoonsakim, sgoutam, amir, ari}@cs.stonybrook.edu

A Artifact Appendix

A.1 Abstract

The core components of Erebus can be separated into 3 separate modules: (1) Policy Engine that generates Erebus policies from Natural Language inputs. (2) Language Transpiler that converts the intermediate Erebus language into the target platform code (in this implementation C#). (3) Native library implementation (in C#) that is used to build Android APKs for testing.

For reproducibility, we separate our implementation into these separate components and provide instructions on how to reproduce our results. In this Artifact, we provide instructions to reproduce the policy engine and policy generation of Erebus (components 1 and 2). We also open-source our implementation of the Erebus framework itself, developed using Unity and C#. However due to the complexity of setup required, and the number of platform-specific dependencies required to recompile the Android APK files, we skip these components for reproducibility and provide all necessary information to be used as reference.

A.2 Description & Requirements

The two components of the Policy Engine framework, included in this artifact, can be evaluated with the following system requirements.

Run-time requirements Ubuntu 20.04, Python 3.8.10, Dotnet 6.0

Hardware Minimum requirements: 2GB RAM

Expected output Policy code generated by each module into their respective text files. Module-specific instructions are provided in the README under each module.

A.2.1 Security, privacy, and ethical concerns

None

[*]These authors contributed equally to this work.

A.2.2 How to access

Stable Github Commit: https://github.com/Ethos-lab/erebus-AR_access_control/tree/artifact-final-release-v2

Github Repo: https://github.com/Ethos-lab/erebus-AR_access_control

A.2.3 Hardware dependencies

None

A.2.4 Software dependencies

All necessary software dependencies and install directions are provided in the README file.

A.2.5 Benchmarks

Any necessary data sets used for the experiements are included with the Repo. For the natural language policy generator module, we tweaked a publicly available NLP model using a custom training data set. The custom data and the final trained model are included in the repo. For evaluation, there is no need to re-train the model.

A.3 Set-up

Detailed instructions are provided in the README.

A.3.1 Installation

Detailed instructions are provided in the README.

A.3.2 Basic Test

The steps to test each module is provided in the README file under each subsection for the specific module.

A.4 Evaluation workflow

For functional and reproducible evaluation, the README file contains all the steps to evaluate the Policy Engine and Policy Transpiler components of Erebus. In our framework, these two components are sequentially chained together and compiled into an Android APK. In this artifact, we provide the steps to evaluate each of these components individually as recreating the Android APK would require substantial setup effort.

The Evaluation workflow can be summarized as below:

1. Download the Github Repo and make sure all the folders are available as per the documentation provided in README.
2. Verify each component individually based on the instructions provided in README for each component.
3. The folders for *erebus*, *prototype_apps*, and *survey* contain code and survey data used in our paper. But for the purposes of this artifact, they do not need to be evaluated for reproducibility (due to limitations of environment setup).
4. The folders for *policy_gen* and *policy_transpiler* contain the main contribution of our paper, which is the access control framework. The README file contains all the detailed instructions to verify these modules.

A.4.1 Major Claims

The main contribution of our paper is the design of an access control framework for Augmented Reality systems. This framework is designed based on a survey of existing systems, and implemented for an Android system. The main claims of our system (mainly the policy framework) include the following:

(C1): We propose a novel access control framework using a policy language design described in Section 5 and Table 3.

(C2): We also propose a mechanism to derive these policies using natural language input from developer's app descriptions, as shown in Figure 4 and Listing 3.

A.4.2 Experiments

Please refer to the README file for exact steps to test the policy framework of Erebus. Detailed steps are provided for each component, and sample policies that can be tested are also provided in the README file.

(E1): *Setup [30 human-minutes]:* Set up the software packages and install all the dependencies.

(E2): *Policy Engine [15 human-minutes]:* Follow the instructions in README for Reproducing the Policy Engine module. Test with additional sample policies provided, if needed.

(E3): *Policy Transpiler [15 human-minutes]:* Follow the instructions in README for Reproducing the Policy Transpiler module. Ensure that the generate target code matches the policy statement defined in (E2).

A.5 Notes on Reusability

Apart from the specific modules used for reproducibility, we also release the overall implementation of our framework in C# (refer *erebus* folder) that ties together each of these modules. We advise readers to use this implementation as a reference, along with the *prototype_apps* code released with this artifact.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20220926. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at `https://secartifacts.github.io/usenixsec2023/`.