WOOT'26 Artifact Appendix: Exploiting Android Apps with Counterfeit Art

Rokhaya-Diamil Fall
EPFL, Lausanne, Switzerland

Philipp Mao
EPFL, Lausanne, Switzerland

Martin Wagner
Asymmetric Research

Mathias Payer
EPFL, Lausanne, Switzerland

Abstract

The artifact contains an emulation and debugging environment in addition to exploitation scripts for reproducing both the local and remote attack scenarios.

1 Artifact Appendix

1.1 Accessing the artifact

The artifact is publicly available at https://doi.org/10.5281/zenodo.20076101 and https://github.com/HexHive/artow-artifact.

1.2 Running the artifact

1.2.1 Dependencies

We provide a Docker environment (using an Ubuntu 20.04 container) that sets up an Android 16 emulator and all required software dependencies. The host system must have Docker 28.2+ installed on an x86_64 Linux machine with at least 25 GB of free disk space and 16 GB of RAM.

1.2.2 Setup

The provided docker container can be built with the `build-docker.sh` script and executed via `run-docker.sh` script.

The emulator is initialized from within the container by running `setup.sh`. Detailed instructions are provided in the `README.md` file.

1.3 Evaluation

We provide exploitation scripts for the local and attack scenarios described in the paper. Both exploits target the Google Chrome app pre-installed with the emulator.

Local Exploit: Executed with `local.py`. It triggers an arbitrary write bug during app image decompression, resulting in a target application crash. The corresponding crash report is stored under `crash_artifacts/hijack_local` in the docker container.

Remote Exploit: Executed with `local.py`. It leverages app image relocation to inject bytecode executing a sleep command by default. The script verifies execution by querying the process list and saves the output to `crash_artifacts/hijack_remote` in the docker container.

1.4 Badges Claimed

- Available
- Functional
- Reproduced

1.5 Security Note

This artifact includes functional exploit scripts for research purposes. To ensure proper isolation and prevent unintended system interference, please only execute these scripts within the provided Docker environment.