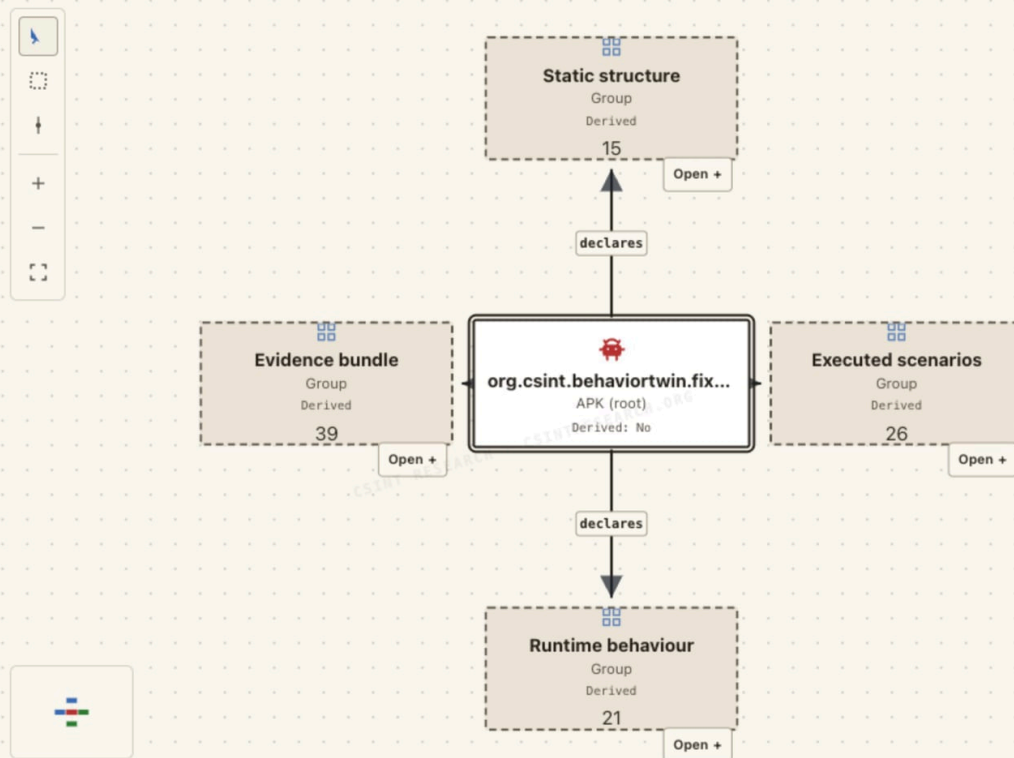




Reading a signed Android lab record with Mobile Behavior Diff

I run an Android app in my own lab, on a controlled emulator. This page opens the signed evidence bundle from that run, checks it and shows what was observed.

1 What the page is for

When a new version of an app comes out, I want to know what changed for a user who goes through the same screens. A screenshot does not tell you that. So I replay the same journey in the lab again and again and record every event.

This page does not run the app. The APK file never comes here either. The page only opens the *.evidence.zip* bundle made by the lab, in your browser, checks its signature and files, and then shows the result.

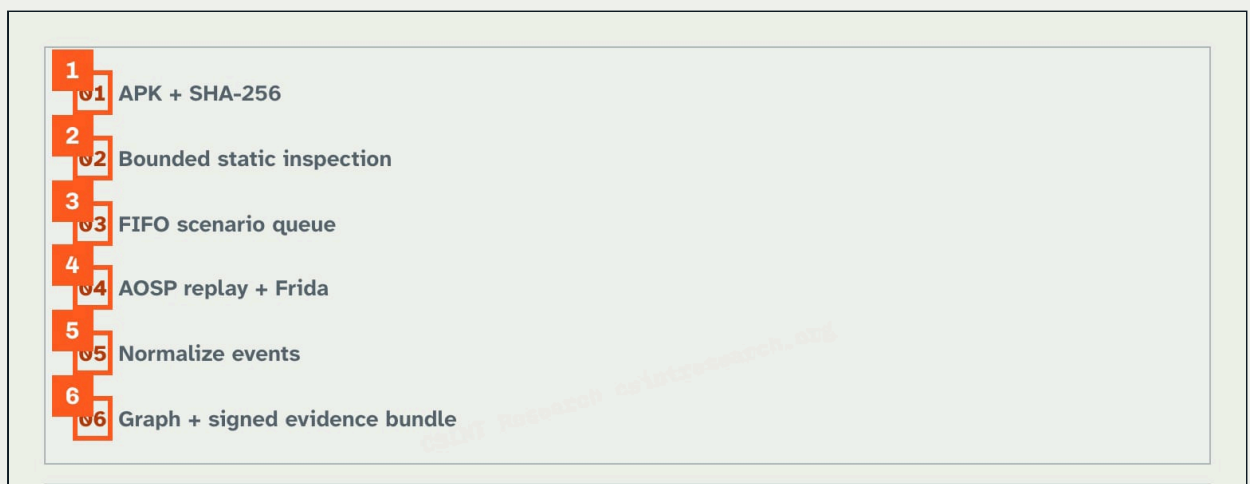
There is no built-in sample. The screen stays empty on purpose until a lab bundle is opened. It never shows made-up or sample results.

A few words

APK	The install file of an Android app.
Emulator	A virtual Android phone that runs on a computer. The lab never touches a real phone.
SHA - 256	A fingerprint of a file. If one byte of the file changes, the fingerprint changes too.
Ed25519 signature	A digital signature that shows the bundle content has not changed since it was signed.
Frida	A tool that records which functions a running app calls. Here it only watches and changes no value.
Control and treatment	Two runs of the same journey. Control is the base of the comparison, treatment is the changed condition (for example a permission turned off, or a new version).

2 How a bundle is made

The six-step row at the top of the page is not a result. It is the order the lab works in. Knowing it helps you see where each part of a bundle comes from.



The six lab stages. The numbers match the notes below.

1 APK and fingerprint

The SHA-256 of the app file is taken. Everything in the bundle is tied to it.

2 Bounded static reading

Before the app runs, its permissions, package name and version are read.

3 Scenario queue

Scenarios run one by one, in the order they arrive. Each starts from the same clean state.

4 Replay on the emulator

The same screen journey is replayed on the emulator. Frida only watches while it runs.

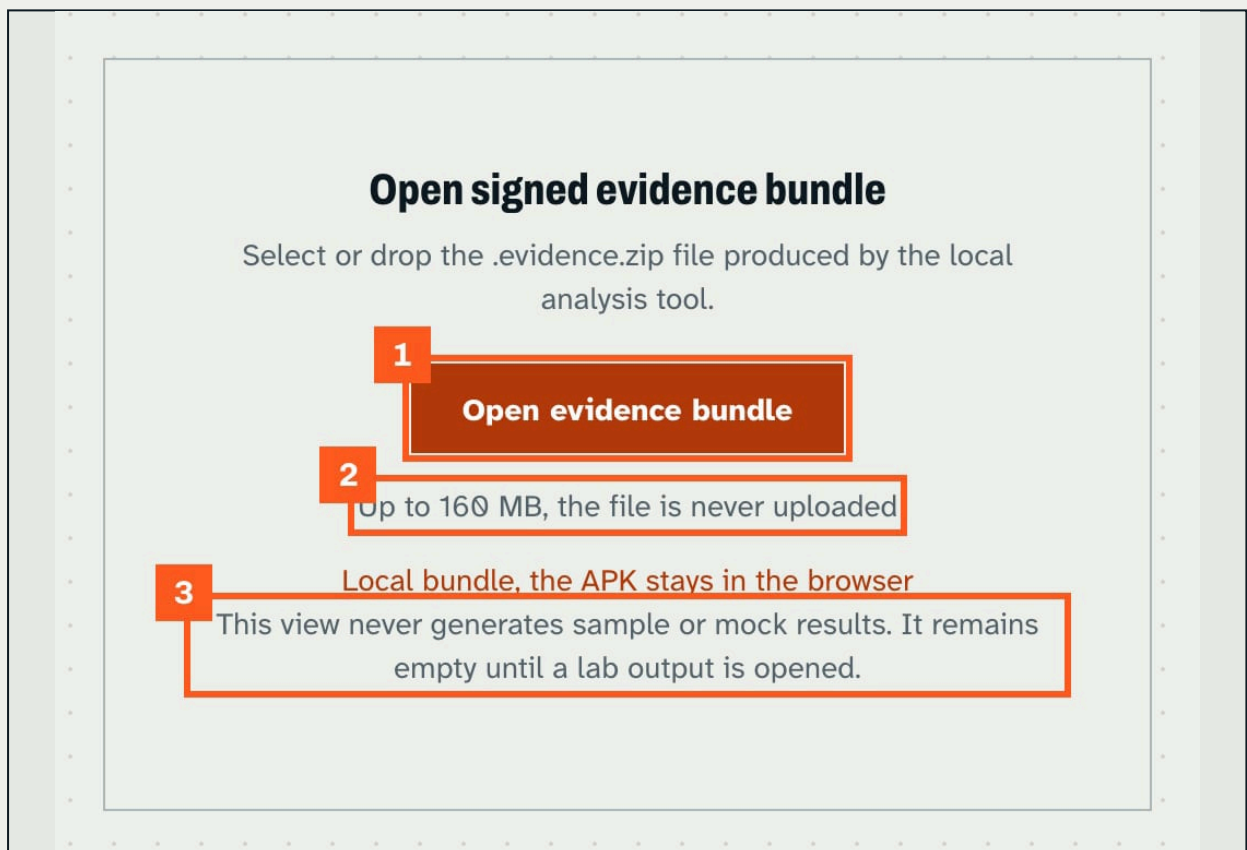
5 Tidying the events

Logs, network traffic and file changes are turned into one event format.

6 Graph and signed bundle

Events are linked into a graph. Every file is listed with its SHA-256 and the list is signed with Ed25519.

3 Opening a bundle



The bundle picker. The page stays like this until a bundle is opened.

1 Pick the bundle

Press the button and choose the file, or drop the file on the box.

2 Mind the limit

A file over 160 MB, or one that is not .evidence.zip, is refused straight away.

3 An empty screen is normal

This line is a reminder that the page makes no result on its own.

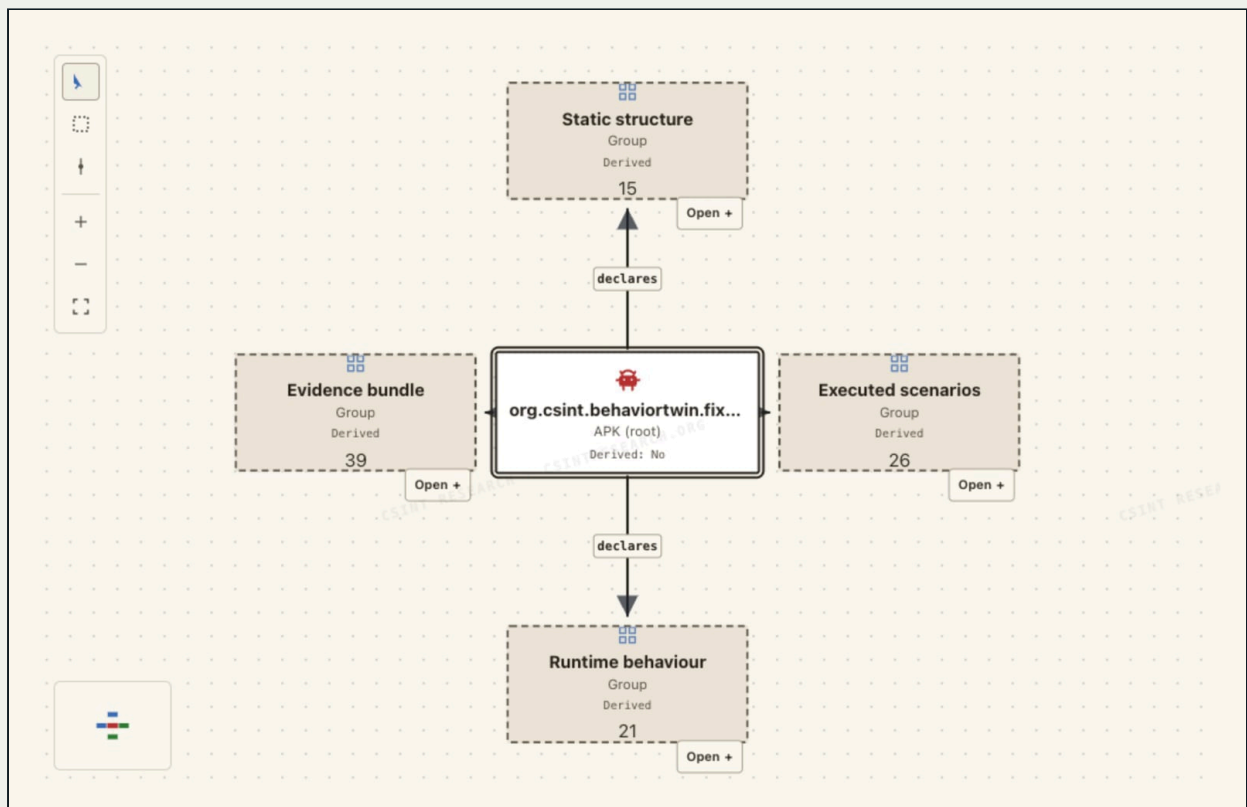
4 Wait for the check

The browser checks the file list, the fingerprint of every file, the signature and the record format. If one of them fails, the bundle does not open and an error code is shown.

You need	A .evidence.zip file made by the lab tool. No other extension is accepted.
Size	Up to 160 MB.
Account	Not needed.
Where data goes	The bundle is opened and checked in your browser. It is not sent to a server.

The bundle carries its own signing key. The signature proves the bundle has not changed. On its own it does not prove who owns the key.

4 What you see after opening



The first view after opening the bundle of the harmless test app. The APK is in the middle, with four groups around it and the record count of each group.

The app sits in the middle of the graph. Around it sit four groups for static structure, executed scenarios, runtime behaviour and the evidence bundle. The *Open* button on a group shows the real records inside it. These are permissions, network addresses, files, database tables, loaded libraries and the functions Frida saw.

Below the graph there are thirteen views. These are the ones I use most.

Overview	Number of scenarios, events, differences found and signed files.
Timeline	The events of the selected run in order. Press an event to see its detail and its evidence file.
Network	Addresses the app tried to reach. A connection is not the same as leaking data.
YIOC	Behaviour differences between the control and treatment runs.
Notes	The bundle's own limits and the safety tier it ran in.

Every finding has an *Open raw evidence* button. It opens the file the finding is based on, straight from the bundle. I suggest you look at that file before you trust a result.

5 Reading a difference

In the YIOC view every difference has a label. The label is *newly-observed*, *disappeared*, or says its frequency went up or down. The two numbers next to it show how many times it was seen in the control run and in the treatment run. A new behaviour starts as *correlated*. If it is seen again in a separate confirmation run, it becomes *reproduced*.

- ✗ Make sure the two runs you compare are the same journey, the same starting state and the same environment. A difference between unmatched runs is not a version difference.
- ✗ A permission on the list does not mean it was used. Static records and runtime records are kept apart.
- ✗ If encrypted traffic cannot be read, the page does not guess. It leaves it as not observable.

6 What it does not do

- ✗ It does not run or upload an APK. In the browser it only reads the signed bundle.
- ✗ The lab is set up only for the harmless test app for now. Random or harmful APKs are refused.
- ✗ A result holds only for the experiment in that bundle, not for every device, network or user.
- ✗ It does not replay the experiment in the browser and does not decide whether an app is harmful or safe. A new run is made only with the local lab tool.