Examples

Shielder Plug-in to update the ShieldSDK-callbacks inside an apk

If you have an apk (for example a customer application) which uses the ShieldSDK-callbacks and you want to Shield that apk with a newer Shield version, then Shielder complains about a version mismatch.

```
Shield SDK version mismatch! ...
```

In some cases it is enough to Shield the application with the *hidden* Shielder command line option `--ignore-version-mismatch`.

However, if in some Shield release the new ShieldSDK contains new Java classes, then Shield will cause a crash at run time, when the native code tries to access these new classes. In such a case the app needs to be recompiled with the matching ShieldSDK.

If you can't recompile the application, then you can use Shielder with this Shielder-plugin to update the ShieldSDK-callback classes inside the application while Shielding the app.

To update the ShieldSDK-callback classes you need

- Shielder
- `ShielderPlugin-update-ShieldSDK-callbacks.dat`
- The app (for example `app.apk`)

```
$ java -jar path/to/Shielder.jar \
    --plugin ShielderPlugin-update-ShieldSDK-callbacks.dat \
        [... any arguments you want to pass to Shielder...] \
        app.apk
```

This updates the ShieldSDK-callback classes inside the app while shielding the app.

See ShieldSDK/Android/java/Tests/ShielderPlugins/ShielderPlugin-update-ShieldSDK-callbacks

Shielder Plug-in inject a simple ShieldSDK-callback handler

The plug-in provides a simple implementation of ShieldSDK-callback's `ExtendedObserver` interface.

This implementation of the `ExtendedObserver` prints the callback information as Android log messages to the Android log console with the tag `"Shield/Observer"` and priority "Info".

To filter the callback information from this observer you can use

```
$ adb logcat Shield/Observer:I *:S
```

To fuse this plug-in with an application you need

- Shielder
- `ShielderPlugin-ShieldObserver.dat`
- The app (for example `app.apk`)

```
$ java -jar path/to/Shielder.jar \
    --plugin ShielderPlugin-ShieldObserver.dat \
        [... any arguments you want to pass to Shielder...] \
        app.apk
```

This fuses the plug-in's ShieldObserver implementation with the application (while shielding the application).

Note: an application can register a single `ExtendedObserver` instance. If the application has its own `ExtendedObserver` instance, then the instance that is registered last in time will replace the instance that was registered first.

See ShieldSDK/Android/java/Tests/ShielderPlugins/ShielderPlugin-ShieldObserver

Shielder Plug-in to print Java class-name reflection

A common problem with Shielder’s class-name obfuscation is, that applications often use Java’s class name reflection that is:

- Class::forName(name, initialize, loader) | Android Developers
- Class::forName(name) | Android Developers

So **SHAND-3131** - Getting issue details... STATUS shall provide a Shielder plug-in that

- logs the requested class name (to the android log)
- calls the original `Class::forName()` method (and returns that result)

Such a Shielder plug-in can be used to

- identify exceptions for Shielder’s class name obfuscation rules.

signhero

DEV-Examples-131222-1100.pdf

Sent on 09.06.2023 by yik
408299651@qq.com

yik

408299651@qq.com

Signed on 09.06.2023