Reverse Engineering React Native Apps

Decompiling for Effective Penetration Testing

By: John Ascher

As mobile applications continue to play a crucial role in modern business operations, the need for comprehensive security assessments has never been more critical. React Native-based apps have become increasingly popular due to their ability to accelerate development and support multiple platforms.

React Native is a framework developed by Meta that allows developers to build mobile applications using the React JavaScript library. React Native enables developers to create platform-agnostic apps that can run on both iOS and Android devices, accelerating development timelines and time-to-market. However, the complexity of these apps presents a unique challenge for security teams that can expose organizations to significant risks, such as data breaches, unauthorized access, and reputational damage, if not addressed.

By equipping your security team with the tools and techniques outlined in this white paper, you can ensure that your organization's React Native-based apps undergo thorough and effective penetration testing, reducing the risk of successful attacks and safeguarding your organization's assets and customer data.

About React Native Bundles

The most important thing to know about React Native apps is that while they contain some native code in Kotlin or Swift, the bulk of the application logic is stored in a file that can be easily overlooked if you are not familiar with the React Native framework.

On iOS, this file is often called `main.jsbundle` and is typically found in the base directory of the IPA file. On Android, the file is often called `index.android.bundle` and is usually located in the `resources/assets/` directory of the APK file.

These files are the source React Native JavaScript behind the application. However, there is another “gotcha” to know about. In 2019, an open-source project called Hermes debuted and later became the default engine for React Native in 2022. Hermes converts the React Native code to bytecode for improved performance and to minimize resource consumption. This bytecode has made it more challenging to review the source code for security vulnerabilities. Most modern React Native apps will now be using Hermes by default, so gone are the days of easy source code access to React Native code. So, what can we do?

Extracting the APK and Decompiling the Bytecode

There are a few tools out there that disassemble and decompile the Hermes bytecode to give a decompiled version of the source JavaScript. Let’s look at how we can pull an APK from a device and decompile the Hermes bytecode into JavaScript.

Step 1: Use Android Debug Bridge (ADB) to get the APK from the device.

```
# Find the target package
adb list packages

# Get the path of the target package
adb shell pm path com.example.app

# Pull the APK using the path obtained from the previous command
adb pull path/to/base.apk
```

Step 2: Use APKTool to decompile the APK:

```
apktool d your-app.apk
```

Step 3: Install hermes-dec by using pip:

```
sudo pip3 install --upgradegit+https://github.com/P1sec/hermes-dec
```

Step 4: Using hermes-dec, we can disassemble and decompile the bundle:

ANDROID

```
# Disassemble into HASM
hbc-disassembler assets/index.android.bundle disassembled.hasm

# Decompile into JavaScript
hbc-decompiler assets/index.android.bundle decompiled.js
```

IOS

```
# Disassemble into HASM
hbc-disassembler main.jsbundle disassembled.hasm

# Decompile into JavaScript
hbc-decompiler main.jsbundle decompiled.js
```

You should now have a decompiled.js file containing the decompiled JavaScript. This is not the exact JavaScript code that was written by the developers. However, this code can still be analyzed for hard-coded secrets such as API keys, client secrets, or passwords. Additionally, the full deeplink routes and the expected parameters can be found here, which can be helpful during blackbox assessments.

```
1   _fun0: for(var _fun0_ip = 0; ; ) switch(_fun0_ip) {
2   case 0:
3       __BUNDLE_START_TIME__ = undefined;
4       __DEV__ = undefined;
5       process = undefined;
6       __METRO_GLOBAL_PREFIX__ = undefined;
7       r4 = this;
8       r1 = r4.nativePerformanceNow;
9       r0 = global;
10      if(r1) { _fun0_ip = 52; continue _fun0 }
11  case 35:
12      r2 = r0.Date;
13      r1 = r2.now;
14      r1 = r1.bind(r2)();
15      _fun0_ip = 64; continue _fun0;
16  case 52:
17      r5 = r0.nativePerformanceNow;
18      r2 = undefined;
19      r1 = r5.bind(r2)();
20  case 64:
21      r0['__BUNDLE_START_TIME__'] = r1;
22      r1 = false;
23      r0['__DEV__'] = r1;
24      r1 = r4.process;
25      if(r1) { _fun0_ip = 88; continue _fun0 }
26  case 86:
27      r1 = {};
28  case 88:
```

Tools and Resources

hermes-dec

A tool that can disassemble the code into a HASM file, and then decompile the HASM code in JavaScript.

Github: https://github.com/P1sec/hermes-dec

Website: https://www.p1sec.com/blog/releasing-hermes-dec-an-open-source-disassembler-and-decompiler-for-the-react-native-hermes-bytecode

hasmer

Hasmer is still a work in progress and will hopefully be the next best tool to work with these bytecode bundles, so keep your eye on it.

Github: https://github.com/lucasbaizer2/hasmer

Website: https://lucasbaizer2.github.io/hasmer/

Apktool

Tool to decompile, compile, and inspect APK files.

Github: https://github.com/iBotPeaches/Apktool

Website: https://apktool.org/

ADB

This tool is part of the Android SDK and is used to interact with Android devices.

Website: https://developer.android.com/tools/adb

SECURITY INNOVATION
A BUREAU VERITAS COMPANY

About Security Innovation/ Bureau Veritas

Security Innovation is a leader in software security, providing comprehensive assessment solutions to secure software from design to deployment, across all environments, including web, cloud, IoT, and mobile. Leveraging decades of expertise and as part of Bureau Veritas, a global leader in Testing, Inspection, and Certification, we seamlessly integrate world-class security into development processes, safeguarding the way companies build and deliver products.

Security Innovation is a Bureau Veritas company. Bureau Veritas (BV) is a publicly listed company specialized in testing, inspection and certification. BV was founded in 1828, has over 80,000 employees and is active in 140 countries.

Other Resources

- https://reactnative.dev/docs/getting-started
- https://pilfer.github.io/mobile-reverse-engineering/react-native/reverse-engineering-and-instrumenting-react-native-apps/

BUREAU VERITAS 1828

Interested?

Contact us today to start raising your cyber resilience.

sisales@securityinnovation.com
+1 877 839 7598
securityinnovation.com