## Enable iXGuard/DexGuard for EDGE

### Purpose

The EDGE team is required to implement code obfuscation in their project to meet security standards. This will be integrated into the Foundation App at build time, ensuring a high level of security without significantly impacting performance. We provide guidelines to achieve a balanced security level and templates for proper configuration. However, the extent of security implementation is at the discretion of the consumer, who must balance it against performance and other factors.

### Overview

This document covers:

* An introduction to the necessary tools for code obfuscation and RASP (iXGuard, DexGuard)
* Tool setup requirements (e.g., licenses)
* Infrastructure requirements and setup instructions
* Platform-specific instructions (iOS and Android)
* Modifications required in the Foundation App, including configuration templates and tool usage
* Detailed RASP setup instructions and development requirements, with configuration examples
* Testing methodologies
* Access to GuardSquare documentation and support

### Tools

GuardSquare's iXGuard (for iOS) and DexGuard (for Android) are our chosen solutions for achieving code obfuscation, jailbreak detection, and fulfilling other security requirements. These tools protect against reverse engineering, data breaches, and package tampering, which pose significant security risks.

RASP (Runtime Application Self-Protection) enables applications to detect jailbreaking, rooted devices, debuggers, and other risky third-party applications that could compromise security.

### DexGuard (Android)

DexGuard is an external tool that optimizes Android applications and libraries, making them smaller, more efficient, and resistant to reverse engineering and tampering. It functions as a post-processor that can be integrated into the existing release process.

### iXGuard (iOS)

iXGuard operates similarly to DexGuard, processing iOS applications and libraries. It can be integrated as a plugin in Xcode to create protected apps using a configuration file.

**Note:** Access to DexGuard and iXGuard licenses, documentation, and tools is available through the [GuardSquare portal](https://guardsquare.com/) (access requires credentials provided by the team leader).

### Setup Requirements

* **iXGuard Installation:** Installation files for iXGuard are provided in NAB Self Service for Macs (under Xcode).
* **DexGuard Installation:** DexGuard can be downloaded as a zip file from the GuardSquare portal, suitable for integration with Android Studio.

### Infrastructure/Pipeline Requirements

(To be provided by the DevOps team)

### Code Obfuscation Configuration (Android)

1. Place the license in the project's Android folder at **/lib/dexguard/dexguard-license.txt**.
2. Modify or create the configuration file at **android/app/dexguard-project.txt**. A template provided by NAB-X outlines the most relevant configurations, but final adjustments should be made by the consumer to balance protection and performance.
3. After building the application, DexGuard generates a protection report, which can be found at:
   * APK: **android/app/build/outputs/dexguard/reports/apk/dev/debug/protectionreport.html**
   * Bundle: **android/app/build/outputs/dexguard/reports/bundle/dev/debug/protectionreport.html**

### RASP Configuration (Android)

1. Set up DexGuard as described above.
2. Enable RASP in the **dexguard-project.txt** configuration file. Example configurations and the required callback function setup are detailed in the document.
3. The callback function (**DeviceInfoManager.OnJailBreakDetected**) updates the **jailbroken** flag in the **useragent** header of network requests upon detection.

### Code Obfuscation Configuration (iOS)

1. Place the license in the iOS folder of the application.
2. Modify or create the **ig-config.yml** file in the application's root folder. The configuration template provided by NAB-X is a starting point, but customization is required.
3. After building the application, iXGuard generates a protection report, the location of which will be specified in this document.

### RASP Configuration (iOS)

1. Set up iXGuard as detailed for code obfuscation.
2. Enable RASP in the **ig-config.yaml** file using the provided configurations. The required callback function (**XNFUtilities.sharedInstance.OnJailBreakDetected**) reacts to jailbreak detections by updating the **jailbroken** flag in network requests.

### Resources

* [NAB-X Quickstart Guides](https://.../)
* [Guard Square](https://guardsquare.com/)