MorningSDK
- Obfuscation Guide-

2023. 6.

Morningsoft

Document History

No.	Date	Contents
1	2023.06	Obfuscation Guide Publishing
2		
3		
4		

Contents

1. Introduction Obfuscation

Sensitive information strings included in the mobile app are separated during compilation and included in the app to be decrypted during execution. This is implemented by performing encryption in a Mac development environment separate from Android Studio and xCode. After encryption, the corresponding file is added during app build. The decryption process is provided by the library, so necessary strings can be obtained by calling the relevant API. Sensitive information requiring obfuscation includes service URLs, passwords, database schema names, etc.

2. Sensitivity Data Obfuscation

Move to the MorningSDK/Library/obfuscation folder in the terminal and execute the following command. obfuscation.txt is the table of strings that need obfuscation. It is in the format "symbol=text". After obfuscation is complete, obfuscation.dat and the generated D001.dat decryption key file will be created. The strings to be obfuscated are encrypted securely with RSA 2048 specifications, using asymmetric keys. Decryption is performed using the public key when executing from the library.

1. String Obfuscation
cd MorningSDK/Library/obfuscation
java -jar ObfuscationService.jar generate
java -jar ObfuscationService.jar obfuscate obfuscation.txt obfuscation.dat

contents of obfuscation.txt

CLIENT_PASSWORD=1234
URL=https://moningsoft.com
DATABASE_SCHEMA=secrettable

2. Initialize Obfuscation Data
A. Android using kotlin

```
        var obfuscation_data: String = "contents of obfuscation.dat"
        var obfuscation_key_data:String = "contents of D001.dat"
       getAppContext()?.let {
          MorningSDK().setStringObfuscationEnable(true)
          MorningSDK().setObfuscation(obfuscation_key_data + "_END_" +
obfuscation_data + "_END_")
       }
```

B. iOS using swift

```
   var license_data: String = “contents of obfuscation.dat"
   var license_key_data:String = “contents of D001.dat"

    MobileShield.setStringObfuscationEnable(enable: true)
    MobileShield.setObfuscation(obfuscationData: obfuscation_key_data + "_END_" +
obfuscation_data + "_END_")
```

3. Decrypt Obfuscated String

A. Android using kotlin

```
val databaseSchema = MobileShield().getObfuscation("DATABASE")
```

B. iOS using swift

```
val databaseSchema = MobileShield().getObfuscation(symbol : "DATABASE")
```