

Android용 MorningSDK v1.1.0
-개발자 매뉴얼-

2024. 3.

Morningsoft

제•개정 이력

순번	제•개정 일	제•개정 내역
1	2022.05	“Android용 개발자 매뉴얼” 발간
2	2023.05	개발자 라이선스 기능 추가
3	2024.01	라이선스 기능 추가
4	2024.03	전자결제, HTTP 2.0 지원

Contents

1. MorningSDK v1.1.0 소개
2. 라이브러리 설치 및 빌드
3. MorningSDK 사용 샘플
4. Android용 API 목록

1. MorningSDK 소개

가. MorningSDK

모바일 앱의 해킹 기법들이 발달함에 따라 앱 마켓에서 서비스하는 앱들이 최신 기술을 활용하여 해킹되는 사례가 발생하고 있습니다. 고객을 위해 서비스를 제공하는 기관이나 기업들은 이런 최신 기술이 포함된 해킹 도구들로 부터 개인정보를 보호할 수 있는 방안이 필요합니다. MorningSDK 모바일 라이브러리는 전세계적으로 알려진 해킹 도구들인 Frida, Magisk, Magisk Hide 등의 설치를 확인하여 민감정보의 보호가 필요한 앱의 실행을 방지하는 기능을 포함하고 있습니다. 또한 일부 해킹 소프트웨어들을 이용해서 디버깅 혹은 메모리 침범등을 시도할 경우 이를 사전에 방지하는 기능을 가지고 있습니다. 따라서 이미 알려진 해킹 도구만을 차단하는것이 아닌 새로운 해킹 도구를 차단할 수 있는 도구를 포함합니다.

또한 민감정보(사용자 아이디, 주민번호) 문자열이 소스코드에 포함되었을때 이를 난독화 하여 이미 배포한 앱으로 부터 해당 정보의 추출을 어렵게 만드는 기능을 포함하고 있습니다.

나. 세부 기능

포함 기능	분류
앱 무결성 및 위변조 방지	Integrity
문자열 난독화	Obfuscation
민감정보 저장	Secure Storage
암호화	Cryptography
고유 식별자 발급	AUUUID
멀웨어 탐지	Vaccine
화면 캡처 방지	DLP
mVoip 보안	VOIP
전자 결제 및 문서 위변조 방지	Sign

- AUUUID : Application Universally Unique Identifier
- DLP : Data Loss Prevention

다. 컴파일 및 실행 환경

항목	내용
Android Studio	Android Studio Flamingo 2023.2.1 이상
Kotlin	v1.9.20 이상
실행 기기	Galaxy S6 이상

실행 운영체제	Android 6.0 이상 (api 23)
---------	-------------------------

2. 라이브러리 설치 및 빌드

MorningSDK 압축 파일의 압축을 풀고 Library/Android/kotlin에 포함된 MorningSDK.aar 파일을 libs에 추가한후 MorningSDK 의 초기화 함수를 호출하여 MorningSDK 기능을 활성화 시킵니다.

가. 라이브러리 설치

1. MorningSDK.aar 복사

2. 앱 gradle 파일 수정

```
implementation fileTree(include: ['*.jar', '*.aar'], dir: 'libs')
implementation files('libs/MorningSDK.aar')
```

```
repositories {
    ...
    flatDir { dirs 'libs' }
}
```

3. MainApplication에서 초기화를 수행합니다.

```
< kotlin >
```

```
getApplicationContext()?.let{
    MorningSDK().setLicense(mContext, " Your-License-Data", "voip.yourdomain.com\n
yourdomain.com\ncom.yourdomain.voip\nYour-Developer-License")
}
```

```
< java >
```

```
// MorningSDK Java Example
```

```
public static com.morningsoft.security.MorningSDK mMorningSDK;
```

```
...
```

```
if(mContext != null){
    mMorningSDK.setLicense(mContext, "Your-License-Data", "voip.morningsoft.com\n
nmorningsoft.com\ncom.morningsoft.voip\nYour-Developer-License");
}
```

4. 민감정보 저장 API를 사용하기 위해서는 AndroidManifest.xml에 다음을 설정해야합니다. 이는 안드로이드 EncryptedSharedPreferences 기능을 사용하기 위한 설정입니다.

```
<application
    ...
    android:allowBackup="false"
    android:fullBackupContent="false"
    ...
```

3. MorningSDK 사용 샘플

가. 라이브러리 초기화 샘플

라이브러리 초기화 샘플

```
MorningSDK().setLicense(mContext, "XXXX", "A\nB\nC\nD")
MorningSDK().setTamperProtectEnable(true)
MorningSDK().setMagiskHideProtectEnable(true)
MorningSDK().setDebuggerProtectEnable(true)
MorningSDK().setSignatureVerifyEnable(true)
MorningSDK().setFridaProtectEnable(true)
MorningSDK().setRootingCheckEnable(true)
MorningSDK().setScreenProtectEnable(true)
MorningSDK().setMarketVerifyEnable(true)
MorningSDK().setDebuggableProtectEnable(true)
```

나. Android Keystore를 이용한 민감 정보 저장 샘플

Android Keystore를 이용한 민감 정보 저장 샘플

```
// 앱에서 데이터를 저장하기 위해서는 반드시 암호화 해야 함
// MorningSDK는 암호화 키를 안드로이드의 키스토어를 사용하므로 안전함
// 주의 : 앱을 재설치 할 경우나 앱데이터를 삭제할 경우 기존 데이터는 복원되지 않음
var mUserPassword : String = "password##!!"
Log.d(TAG, "User Password : " + mUserPassword)
MorningSDK().writeString("UserPassword", mUserPassword)
Log.d(TAG, "User Password : " + MorningSDK().readString("UserPassword"))
var mUserNumber : Int = "12341234"
Log.d(TAG, "User Number : " + mUserNumber)
MorningSDK().writeInteger("UserNumber", mUserNumber)
Log.d(TAG, "User Number : " + MorningSDK().readInteger("UserNumber"))
var mUserLogin : Boolean = true
MorningSDK().writeBool("UserLogin", mUserLogin)
Log.d(TAG, "User Login : " + MorningSDK().readBool("UserLogin"))
```

다. 문자열 난독화 샘플

문자열 난독화 샘플

```
// 기존 소스코드
var mAESPassword : String = "password##!!"
Log.d(TAG, "AES Password : " + mAESPassword)

// IDS_AES_PASSWORD는 문자열의 Symbol로 빌드전 난독화 ObfuscationService.jar 프로그램으로 암호화로 추가함
// 중요 정보가 앱의 소스코드에 문자열로 포함되지 않아 유출 가능성이 없음
// 수정된 소스코드
Log.d(TAG, "AES Password : " + MorningSDK().getObfuscation(IDS_AES_PASSWORD))
```

4. Android용 API 목록

가. 앱 무결성 및 위변조 방지

함수 원형	fun securityCheck()
설명	MorningSDKActivity의 멤버 함수로 주요 Activity에서 MorningSDKActivity를 상속받아 사용할 경우 자동으로 호출됨
매개변수	없음
결과값	무결성 검증 실패 및 멀웨어가 탐지 되었을 경우 앱을 종료함
비고	

함수 원형	fun isMarketVerifyEnabled() : Boolean
설명	배포 마켓 검증 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun isRootingCheckEnabled() : Boolean
설명	루팅 체크 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun isSignatureVerifyEnabled() : Boolean
설명	앱 시그너처 검증 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun isDebuggableProtectEnabled() : Boolean
설명	디버깅 가능 설정을 확인하는 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun isFridaProtectEnabled() : Boolean
설명	프리다 방지 기능 사용 여부

매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun isDebuggerProtectEnabled() : Boolean
설명	디버거 방지 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun isMagiskHideProtectEnabled() : Boolean
설명	Magisk Hide 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun isTamperProtectEnabled() : Boolean
설명	템퍼 방지 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun setMarketVerifyEnable(enable : Boolean?)
설명	배포 마켓 검증 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

함수 원형	fun setRootingCheckEnable(enable : Boolean?)
설명	루팅 체크 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

함수 원형	fun setSignatureVerifyEnable(enable : Boolean?)
설명	앱 시그너처 검증 기능 설정

매개변수	enable : 설정값
결과값	없음
비고	

함수 원형	fun setDebuggableProtectEnable(enable : Boolean?)
설명	디버깅 가능 방지 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

함수 원형	fun setFridaProtectEnable(enable : Boolean?)
설명	Frida 방지 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

함수 원형	fun setDebuggerProtectEnable(enable : Boolean?)
설명	디버거 실행 방지 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

함수 원형	fun setMagiskHideProtectEnable(enable : Boolean?)
설명	Magisk Hide 방지 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

함수 원형	fun setTamperProtectEnable(enable : Boolean?)
설명	Tamper 방지 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

나. 문자열 난독화

함수 원형	fun getObfuscation(symbol: String) : String?
설명	난독화 문자열을 복원함
매개변수	symbol : 문자열 key

결과값	복원된 평문
비고	

함수 원형	fun setObfuscation(obfuscationData : String) -> Void
설명	난독화 데이터 설정
파라미터	obfuscationData : 별도의 프로그램으로 생성한 난독화 데이터
결과값	없음
비고	

함수 원형	fun isStringObfuscationEnabled() : Boolean
설명	문자열 난독화 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun setStringObfuscationEnable(enable : Boolean?)
설명	문자열 난독화 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

다. 민감 정보 저장

함수 원형	fun writeString(key: String?, value: String?)
설명	(키, 텍스트) 형식의 데이터를 keystore를 이용해서 안전하게 저장함
매개변수	key: 데이터와 연관된 키값 value : 텍스트 데이터
결과값	없음
비고	

함수 원형	fun readString(key: String) : String?
설명	Key에 해당하는 텍스트 형식의 데이터를 얻어옴
매개변수	key: 데이터와 연관된 키값
결과값	텍스트 데이터
비고	

함수 원형	fun writeInteger(key: String?, value: Int?)
설명	(키, 정수) 형식의 데이터를 keystore를 이용해서 안전하게 저장함
매개변수	key: 데이터와 연관된 키값

	value : 정수 데이터
결과값	없음
비고	

함수 원형	fun readInteger(key: String) : Int
설명	Key에 해당하는 정수 형식의 데이터를 얻어옴
매개변수	key: 데이터와 연관된 키값
결과값	정수 데이터
비고	

함수 원형	fun writeBool(key: String?, value: Bool?)
설명	(키, 이진) 형식의 데이터를 keystore를 이용해서 안전하게 저장함
매개변수	key: 데이터와 연관된 키값 value : 이진 데이터
결과값	없음
비고	

함수 원형	fun readBool(key: String) : Bool
설명	Key에 해당하는 이진 형식의 데이터를 얻어옴
매개변수	key: 데이터와 연관된 키값
결과값	이진 데이터
비고	

라. 암호화

함수 원형	fun initRSA2048String()
설명	텍스트(String)의 RSA 암호화를 위한 public/private 키 생성(2048 bits)
매개변수	없음
결과값	없음
비고	

함수 원형	fun encryptRSA2048String(plainText: String) : String
설명	텍스트(String)의 RSA 암호화를 수행하여 암호문을 생성함
매개변수	plainText : 평문 텍스트
결과값	Base64형식의 암호문 텍스트
비고	암호화를 위한 공용키는 keystore 에 저장된 최근 키를 사용함 만약 키가 생성되지 않았을 경우 암호화 전에 자동 생성됨

함수 원형	fun decryptRSA2048String(cipherText: String) : String
설명	텍스트(String)의 RSA 복호화를 수행하여 평문을 생성함

매개변수	cipherText : Base64형식의 암호문 텍스트
결과값	평문 텍스트
비고	복호화를 위한 공용키는 keystore에 저장된 최근 키를 사

함수 원형	fun initAES256String()
설명	AES(256bits)로 평문 텍스트(String)을 암호화 하기 위한 키 생성 함수
매개변수	없음
결과값	없음
비고	이전 키는 초기화되어 더이상 사용하지 못함

함수 원형	fun encryptAES256String(plainText: String) : String
설명	AES(256bits)로 평문 텍스트(String)를 암호화 하여 암호문을 생성함
매개변수	plainText : 평문 텍스트(String)
결과값	없음
비고	암호화를 위한 키는 자동 생성되어 keystore에 안전하게 보관됨

함수 원형	fun decryptAES256String(cipherText: String) : String
설명	AES(256bits)로 암호문 텍스트(String)를 복호화 하여 평문을 얻음
매개변수	cipherText : Base64로 인코딩된 암호문
결과값	평문 텍스트(String)
비고	복호화를 위한 키는 자동 생성되어 keystore에 안전하게 보관됨

함수 원형	fun computeSHA256String(plainText: String) : String
설명	SHA256으로 해시 문자열 생성
매개변수	plainText : 평문 텍스트
결과값	해시 텍스트(String)
비고	

마. 고유 식별자 발급

함수 원형	fun getAUUID() : String
설명	AUUID를 생성함, 모바일 단말의 식별자로 사용할 수 있음
매개변수	없음
결과값	AUUID
비고	앱의 재설치 및 앱 데이터 삭제시 AUUID는 새로 생성됨

바. 멀웨어 탐지

함수 원형	fun runScan()
설명	멀웨어 스캔 시작

매개변수	없음
결과값	없음
비고	

사. 화면 캡처 방지

함수 원형	fun isScreenProtectEnabled() : Boolean
설명	스크린 캡처 방지 기능 사용 여부
매개변수	없음
결과값	true : 설정 false : 설정 안됨
비고	

함수 원형	fun setScreenProtectEnable(enable : Boolean?)
설명	스크린 캡처 방지 기능 설정
매개변수	enable : 설정값
결과값	없음
비고	

아. 자가 테스트

함수 원형	fun testMorningSDK()
설명	라이브러리 자가 테스트
매개변수	없음
결과값	없음
비고	Logcat 으로 테스트 결과를 확인할 수 있음

자. 라이선스 설정

함수 원형	fun setLicense(mContext : ApplicationContext, licenseData: String, configData : String) -> Void
설명	라이선스 설정
파라미터	mContext : 프로그램 실행시 획득한 ApplicationContext licenseData : 발급받은 라이선스 데이터 configData : 사용을 위한 설정
결과값	없음
비고	

차. 로깅 설정

함수 원형	fun setLogLevel(logLevel : Int) -> Void
-------	---

설명	로깅 레벨 설정
파라미터	logLevel : 로깅 레벨을 설정함
결과값	없음
비고	LogUtil.LEVEL_NONE LogUtil.LEVEL_ERROR LogUtil.LEVEL_WARN LogUtil.LEVEL_INFO LogUtil.LEVEL_DEBUG