

nProtect AppGuard for Mobile API Reference

AppGuard for Mobile SDK API Reference

(주)잉카인터넷 개발팀

DV-210913

Ver. 2.6.0

업체 제공용

저작권

이 문서는 (주)잉카인터넷의 비밀 자산으로 AppGuard for Mobile을 Android Application에 적용 하기 위한 표준 가이드와 관련 정보를 제공하기 위하여 작성되었으며, (주)잉카인터넷의 승인 없이 제3자에게 제공하거나 제공되어나 배포할 수 없습니다.



최소 요구 사항

AppGuard Module No.	[21025] Platform11.3 Build:c4.62.1e5 (3f.a5)
AppGuard SDK	appguard-sdk_r20.jar appguard-extra_r9.jar

문서 개요

이 문서는 SDK 형태로 제공되는 nProtect AppGuard for Mobile의 API 함수 사용법을 설명한 프로그래밍 가이드입니다.

문서 목차

public class AppGuardClient	3
public interface AppGuardEventListener.....	11
public class SecurityEventParser	14
public static class SecurityEventType.Event	15
public static class SecurityEventType.Error.....	19
public static class SecurityEventType.Detect.....	22
public class SecurityEventType.S2Auth.....	28
public class MacroRule	29
public interface SecureObjectFactory.....	32
public interface SecureInteger.....	35
public interface SecureLong	44
public interface SecureBytes	54
public class AppGuardIORequestPacket	55

public class AppGuardClient

public class

AppGuardClient

com.inca.security.AppGuard.AppGuardClient

Class Overview

이 클래스는 본 AppGuard 솔루션의 코어 부분으로 Security Engine의 시작 및 종료, 중지, 재시작 등에 관한 엔진 컨트롤 기능을 제공합니다.

일반적으로 해당 클래스는 다음과 같은 코드 예시와 같이 메인 액티비티의 onCreate(Bundle savedInstanceState) 메서드에서 호출됩니다.

```
public class MainActivity extends Activity {
    private AppGuardClient mAppGuardClient = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        mAppGuardClient = new AppGuardClient(this, new MyEventListener());
        ...
    }
}
```

Summary

Public Constructors	
	AppGuardClient() AppGuardClient 객체를 생성하고 보안 서비스를 시작 합니다.
	AppGuardClient(Context context, AppGuardEventListener listener) AppGuardClient 객체를 생성하고 보안 서비스를 시작 합니다.

Public Methods	
boolean	isInitialized() 보안 서비스의 초기화가 완료되었다면 true를 돌려줍니다.
String	getVersion() 엔진 모듈의 버전을 돌려줍니다.

void	setConnectionTimeout(int timeout) 업데이트 서버 연결에 사용할 연결 timeout 값을 설정합니다.
void	setUserId(String userId) 로그 서버에 기록할 사용자의 ID를 설정합니다.
void	setUniqueClientId(String clientId, int retryTimeout) 서버 인증에 사용자 식별 키(Key)로 사용할 UniqueClientId 값을 등록합니다.
void	setCallbackHandler(Handler event, Handler error, Handler detected) 보안 로그를 수신할 이벤트 핸들러를 등록합니다.
void	setReserved1(int value) 커스텀 기능을 사용하도록 설정합니다
void	setReserved2(final String data) 인증서버의 주소를 설정합니다.
static boolean	setSecureScreen(Window w) 화면 Window 객체에 화면 캡처 방지를 설정합니다.
void	setCustomMacroRule(MacroRule macroRule) 커스텀 매크로 탐지 모드에 새로운 탐지 규칙을 설정합니다.
void	startCustomMacroMode() 커스텀 매크로 탐지 모드를 시작합니다.
void	stopCustomMacroMode() 커스텀 매크로 탐지 모드를 종료합니다.

Public Constructors

```
public AppGuardClient(Context context, AppGuardEventListener listener)
```

AppGuardClient 객체를 초기화하고 보안 서비스를 시작하는 생성자입니다.

보안 서비스는 먼저 AndroidManifest.xml에 설정된 AppGuardLicense 메타 태그에 설정되어 있는 정보에 따라 업데이트를 수행합니다. 이 때 업데이트 서버 연결에 사용될 타임아웃 값은 디폴트로 5000 Milliseconds로 설정되어 있으며, setConnectionTimeout() 메서드를 사용하여 변경할 수 있습니다.

업데이트가 완료되면 엔진이 실행되며 초기화 및 보안 검사를 수행합니다. 동작 과정에 대한 이벤트는 생성자의 두 번째 인자인 AppGuardEventListener 객체에 전달됩니다. 다음은 start() 과정에서 발생할 수 있는 주요 동작 이벤트를 보입니다.

Event 이벤트	
ENGINE_START	엔진을 시작합니다.
ENGINE_INITIALIZED	엔진 초기화를 완료하였습니다.
ENGINE_STOP	엔진이 중지되었습니다.

ENGINE_RESET	엔진이 리셋되었습니다.
UPDATE_START	업데이트를 시작합니다.
UPDATE_COMPLETE	업데이트를 완료하였습니다.
UPDATE_DOWNLOAD	업데이트 파일 다운로드가 시작됩니다.
UPDATE_DOWNLOADING	업데이트 파일을 다운로드 중입니다.
UPDATE_CHECK	업데이트가 필요한지 검사합니다.
UPDATE_FAIL	업데이트가 실패하였습니다. (UPDATE_COMPLETE 대신 전달됩니다.)
S2AUTH_CALLBACK	서버 인증을 시도하였습니다.

Error 이벤트	
ERROR_CONNECT_TO_UPDATE_SERVER	업데이트 서버에 연결할 수 없습니다.
ERROR_CONNECT_TO_GATEWAY_SERVER	게이트웨이 서버에 연결할 수 없습니다.
ERROR_DOWNLOAD_UPDATE_FILE	업데이트 정보 파일의 다운로드를 실패하였습니다.
ERROR_DOWNLOAD_MODULE_FILE	업데이트 파일의 다운로드를 실패하였습니다.
ERROR_CONNECT_TO_GATEWAY_SERVER	게이트웨이 서버에 연결할 수 없습니다.
ERROR_INVALID_UPDATE_DATA_FILE	업데이트 정보 파일이 유효하지 않습니다.
ERROR_KEY_IS_NULL	라이선스 키 정보가 설정되어 있지 않습니다.
ERROR_NETWORK_ACTIVATION	네트워크에 연결할 수 없습니다.
ERROR_OPEN_REMOTE_UPDATE_FILE	업데이트 정보 파일이 존재하지 않거나 손상되었습니다.
ERROR_ENGINE_INIT	엔진 초기화에 실패하였습니다.
ERROR_MANDATORY_FILES_DOES_NOT_EXIST	필수 파일이 존재하지 않습니다.
ERROR_LIBENGINE_FATAL	보안 엔진에서 치명적인 오류가 발생하였습니다.

이벤트에 관한 추가적인 자세한 내용은 SecurityEventType 클래스를 참조합니다.

Parameters

<i>context</i>	이 클래스가 동작하는 Context 객체
<i>listener</i>	보안 서비스 동작 이벤트를 수신할 Listener 객체

Throws

<i>AppGuardException</i>	치명적인 예외 상황이 발생하였습니다.
--------------------------	----------------------

Public Methods

```
public boolean isInitialized()
```

보안 서비스의 초기화가 유무를 질의합니다.

Returns

보안 서비스의 초기화가 완료되었다면 true를 리턴합니다.

public String getVersion()

엔진의 버전 정보를 리턴합니다.

리턴 되는 문자열의 형태는 "Platform: x.x Build: x.x.x" 형태의 문자열 값을 리턴합니다. 전체적인 빌드 버전은 보안 구성에 필요한 모듈이 모두 로드된 이후 알 수 있으므로 가급적 ENGINE_INITIALIZED 또는 ENGINE_RESET 이벤트 수신 후 호출하는 것을 권장합니다.

Returns

성공하면 Platform + Build 버전 형태의 문자열 값을 리턴합니다. 만약 실패할 경우 "UnknownYet" 문자열 값을 리턴합니다.

public void setUserId(String userId)

로그 서버에 기록할 사용자의 ID를 설정합니다.

Parameters

userId 사용자 아이디 문자열

public void setUniqueClientID(String clientID, int retryTimeout)

서버 인증에 사용될 사용자 식별 값을 설정합니다. 만약 clientID값을 null로 입력할 경우 AppGuard 시스템에 의하여 랜덤으로 UniqueClientID가 자동으로 생성됩니다. 해당 생성된 ID값은 SecurityEventType.Event.S2AUTH_CALLBACK으로 받을 수 있습니다.

(주의1: 해당 메서드는 2017/4/26일자 AppGuard 모듈부터 static에서 non-static으로 변경되었습니다.)

(주의2: 18005 모듈부터 nullKey 모드를 지원합니다.)

아래 해당 ID 값의 포맷을 보입니다.

항목	데이터 타입	설명
UniqueID	char(40)	SHA1 또는 40자로 구성된 클라이언트 유일 식별 값
TimeStamp	char(13)	14자로 구성된 해당 UniqueClientID 생성 시 TimeStamp 또는 클라이언트와 서버의 세션이 생성된 시간의 TimeStamp
LicenseKey	string	해당 앱에 발부된 AppGuard License Key값

Parameters

clientID UniqueClientID 키 포맷을 준수하는 앱 서버에서 생성한 사용자 식별 값, null 값을 입력할 경우 AppGuard에 의하여 자동으로 랜덤 ID를 생성합니다.

retryTimeout 서버인증 재시도(retry)를 종료할 최대 시간값(초단위, 기본값 SecurityEventType.S2Auth.S2AUTH_OPTION_DEFAULT_TIMEOUT(= 180))

서버 인증 재시도 Timeout 값의 설정 가능 값의 범위는 0 ~ 180(즉, 최대값 180초=3분)이며, 0으로 설정 시 재시도를 하지 않으며, 180으로 설정 시 일반적으로 3분안에 약 15번의 인증 재시도가 수행됩니다. *retryTimeout의 기본 설정값을 180로 설정하는 것을 권장합니다.*

Throws

AppGuardException clientID의 구성 포맷이 유효하지 않습니다.

```
public static void setConnectionTimeout(int timeout)
```

업데이트 서버 연결에 사용할 연결 timeout 값을 설정합니다.

Parameters

timeout Timeout 값

```
public static void setCallbackHandler(Handler event, Handler error, Handler detected)
```

보안 로그를 수신할 이벤트 핸들러를 등록합니다.

Parameters

event EVENT 타입의 이벤트를 수신할 Handler 객체
error ERROR 타입의 이벤트를 수신할 Handler 객체
detected DETECT 타입의 이벤트를 수신할 Handler 객체

각 Handler는 보안 엔진에서 이벤트 발생 시 Message 타입의 이벤트 메시지를 수신하게 되며, 해당 객체의 데이터 필드에 포함되어 있는 값의 의미는 다음과 같습니다.

필드	설명	예시
Message.what	이벤트 코드	Ex) SecurityEventType.Detect.DETECT_RUNNING_BAD_APPLICATION
Message.obj	이벤트 데이터 해당 사항이 없을 경우 null 포함	없음

(주의: AppGuardClient 객체 생성 시, AppGuardEventListener가 등록이 되어있으면 해당 Handler를 통해 일부 메시지가 전달 되지 않을 수 있습니다.
반드시 AppGuardEventListener 또는 setCallBackHandler 둘 중 하나만 사용 하셔야 합니다.)

다음은 일반적인 사용 예를 보입니다.

```
public class MainActivity extends Activity {
    ...
    private Handler event = new Handler() {
        @Override
        public void handleMessage(Message message) {
            switch ( message.what ) {
                case SecurityEventType.Event.ENGINE_START:
                    ...
                    break;
            }
        }
    };

    private Handler error = new Handler() {
        @Override
        public void handleMessage(Message message) {
            switch ( message.what ) {
                case SecurityEventType.Error.ERROR_NETWORK_ACTIVATION:
                    ...
                    break;
            }
        }
    };

    private Handler detected = new Handler() {
        @Override
        public void handleMessage(Message message) {
            switch ( message.what ) {
                case SecurityEventType.Detect.DETECT_WEIRD_ROOT_PROCESS:
                    ...
                    break;
            }
        }
    };
    ...
    protected void onCreate(Bundle savedInstanceState) {
        ...
        mAppGuardClient.setCallbackHandler(event, error, detected);
        ...
    }
}
```



```
public void setReserved1(int value)
```

주어진 인자 value값에 따라 업체 커스터마이징된 전용 기능을 수행합니다.

기능	값	설명
IID_MACRO_MONITOR_RESTART	0x10000000	매크로 탐지를 초기화하여 재 시작합니다. 해당 함수는 "Macro Suspicious Behavior Detection"이 탐지된 이후 호출되어야만 동작합니다. 그럼으로 해당 함수는 AppGuardEventListener를 상속한 클래스 내 또는 setCallbackHandler에 등록된 핸들러에서 해당 보안 정책 위반에 대한 탐지 이벤트를 수신하였음을 확인 후 호출되어야 합니다.
IID_PROXY_SETTING_CHECK	0x20000000	HTTP 통신 시 Proxy Host를 사용하도록 설정되어 있는지 여부를 확인하여 리포트합니다. 해당 기능은 호출(다중 호출 가능) 시 1회 검사를 수행합니다. Proxy 설정이 확인될 경우 등록된 핸들러를 통하여 DETECT_PROXY 탐지 이벤트를 발생시킵니다.

```
public void setReserved2(final String data)
```

서버인증(Server-Side Authentication) 서버의 주소를 설정합니다.

setUniqueClientID 메서드를 호출하여 서버인증을 수행하기 전에 해당 메서드를 호출하여 인증을 수행할 서버의 주소를 변경할 수 있습니다.

Parameters

data 인증 서버 주소 URL (ex: https://c4-auth.appguard.co.kr)

```
public static int setSecureScreen(Window w)
```

화면 Window 객체에 화면 캡처 방지를 설정합니다.

안드로이드 3.0(Honeycomb) 이상의 버전부터 지원하는 기능으로 성공하면 true를, 실패하면 false를 리턴합니다.

다음은 일반적인 사용 예를 보입니다.

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    AppGuardClient.setSecureScreen(getWindow());
    ...
}
```

Parameters

w 화면 캡처를 적용할 Window 객체

```
public void setCustomMacroRule(MacroRule macroRule)
```

커스텀 매크로 탐지 모드에 탐지 규칙을 설정합니다.

Custom 매크로 탐지 모드는 기본적으로 Default 매크로 탐지 모드와 동일한 규칙으로 설정되어있으며, 해당 매서드를 사용하여 Default 모드와는 다른 탐지 규칙을 적용할 수 있습니다.

Default 매크로 모드와 Custom 매크로 모드는 터치 이벤트를 병렬로 처리하지 않으므로 start/stopCustomMacroMode() 메서드를 사용하여 어떤 매크로 탐지 모드를 사용할 것인지 지정해야 합니다.

Parameters

macroRule

매크로 룰셋

```
public void startCustomMacroMode()
```

커스텀 매크로 탐지 모드를 시작 합니다.

Custom 매크로 탐지 모드가 시작되면 setCustomMacroRule() 메서드로 지정된 룰셋에 따라 매크로 탐지를 수행합니다. 해당 모드로 동작 시 Default 매크로 탐지 모드는 동작하지 않습니다.

다시 Default 매크로 탐지 모드로 동작 시키기 위해서는 `stopCustomMacroMode()`를 호출하여야 합니다.

```
public void stopCustomMacroMode()
```

커스텀 매크로 탐지 모드를 종료 합니다.

수행중인 Custom 매크로 탐지 모드를 종료하고 Default 매크로 탐지 모드로 전환합니다.

public interface AppGuardEventListener

public interface

AppGuardEventListener

com.inca.security.AppGuard.AppGuardEventListener

Class Overview

보안 서비스 엔진의 동작 이벤트를 수신할 수 있는 객체에 관한 인터페이스 객체입니다. AppGuardClient 클래스는 생성자(Constructor)에 해당 인터페이스 형의 객체를 두 번째 인자로 수신하게 되어있습니다. 해당 인터페이스는 동작 이벤트(onEvent), 오류 이벤트(onError), 탐지 이벤트(onDetected)의 3개 추상 메서드로 제공함으로 이 인터페이스를 상속하여 구현된 이벤트 리스너 객체를 위 3개 메서드를 구현해야 합니다. 보안 모듈은 각각의 이벤트가 발생할 경우 사용자가 구현한 이벤트 리스너 객체에 이벤트를 전달합니다.

(주의: AppGuardClient 객체 생성 시, AppGuardEventListener를 등록 후, setCallbackHandler를 통해 Handler를 추가 등록 하면 일부 메시지가 전달 되지 않을 수 있습니다.

반드시 AppGuardEventListener 또는 setCallbackHandler 둘 중 하나만 사용 하셔야 합니다.)

해당 인터페이스를 사용하여 객체를 구현하는 예는 다음과 같습니다.

```
class MyEventListener implements AppGuardEventListener {
    @Override
    public void onEvent(int code, byte[] data) {
        switch ( code ) {
            case SecurityEventType.Event.ENGINE_START:
                ...
            }
        }
    @Override
    public void onError(int code, byte[] data) {
        switch ( code ) {
            case SecurityEventType.Error.ERROR_NETWORK_ACTIVATION:
                ...
            }
        }
    @Override
```

```
public void onDetected(int code, byte[] data) {
    switch (code) {
        case SecurityEventType.Detect.DETECT_WEIRD_ROOT_PROCESS:
            ...
    }
}
```

Summary

Public Methods	
abstract void	onEvent(int event, byte[] data) 보안 모듈의 동작 이벤트를 수신할 추상 메서드입니다.
abstract void	onError(int error, byte[] data) 보안 모듈의 오류 이벤트를 수신할 추상 메서드입니다.
abstract void	onDetected(int identity, byte[] data) 보안 모듈의 탐지 이벤트를 수신할 추상 메서드입니다.

Public Methods

```
public abstract void onEvent(int event, byte[] data)
```

보안 모듈의 동작 이벤트를 수신할 추상 메서드입니다.

동작 이벤트를 식별할 수 있는 식별 코드 값은 event 변수를 통하여 전달되며, 만약 해당 이벤트에 제공되는 추가적인 데이터가 존재한다면 data 변수를 통하여 전달됩니다.

예를 들어, SecurityEventType.Event.ENGINE_START 값은 추가적인 4 바이트(Bytes) 엔진 버전 값이 data 변수에 전달됩니다.

Parameters

event	동작 이벤트의 코드 값
data	동작 이벤트의 추가적인 데이터 값

```
public abstract void onError(int error, byte[] data)
```

보안 모듈의 오류 이벤트를 수신할 추상 메서드입니다.

오류 이벤트를 식별할 수 있는 식별 코드 값은 error 변수를 통하여 전달되며, 만약 해당

이벤트에 제공되는 추가적인 데이터가 존재한다면 data 변수를 통하여 전달됩니다.

Parameters

event	오류 이벤트의 코드 값
data	오류 이벤트의 추가적인 데이터 값

```
public abstract void onDetected(int identity, byte[] data)
```

보안 모듈의 탐지 이벤트를 수신할 추상 메서드입니다.

탐지 이벤트를 식별할 수 있는 식별 코드 값은 identity 변수를 통하여 전달되며, 만약 해당 이벤트에 제공되는 추가적인 데이터가 존재한다면 data 변수를 통하여 전달됩니다.

2016/11/25 Platform:8.2 Build:6d.33.169 (1f.28) 이상 부터 identity값은 해당 이벤트의 행위 상태인 Terminated/Reported 값을 포함하도록 변경되었습니다.

identity값 < 0, 즉 음수일 경우 reported를 의미합니다.

Identity값 > 0, 즉 양수일 경우 terminated를 의미합니다.

예를 들어 Optimized DEX Integrity의 경우 기본적으로 코드값으로 identity: 4 값을 가지며, 만약 reported로 탐지된 경우 -4값이 전달되고, teminated로 탐지된 경우 4 값이 전달됩니다.

그럼으로 SDK를 사용하여 메시지 출력을 커스터마이징한 경우 identity 값이 음수인 경우 화면 출력을 제외하고, 양수인 경우만 화면에 종료 메시지를 출력하는 것이 바람직합니다.

```
public class SecurityEventParser
```

```
public class
```

SecurityEventParser

```
com.inca.security.Common.SecurityEventParser
```

Class Overview

이 클래스는 보안 서비스 엔진에서 수신 받은 Callback에 포함된 데이터값을 파싱(Parsing)하기 위하여 사용되는 클래스입니다.

Summary

Public Methods	
static SecurityEventParser	getInstance() SecurityEventParser 객체를 리턴합니다.
Object[]	parse(int type, int code, byte[] data) 데이터를 파싱하여 Object[] 형태의 결과 값을 리턴합니다.

Public Methods

```
public static void getInstance()
```

데이터 파싱에 사용할 SecurityEventParser 객체를 리턴합니다.

Returns

성공하면 SecurityEventParser 객체를 리턴합니다.

```
public Object[] parse(int type, int code, byte[] data)
```

바이트 형태의 데이터를 파싱하여 Object[] 형태의 결과 값을 리턴합니다.

Parameters

<i>type</i>	Callback에서 수신된 이벤트 타입 값
<i>code</i>	Callback에서 수신된 이벤트 코드 값
<i>data</i>	Callback에서 수신된 이벤트 데이터 값

Returns

성공하면 파싱된 Object[] 타입의 객체를 리턴하고, 실패하면 null을 리턴한다.

public class SecurityEventType.Event

public class

SecurityEventType.Event

com.inca.security.Common.SecurityEventType.Event

Class Overview

동작 이벤트의 식별 값의 열거자 클래스입니다.

Summary

Fields		
int	ENGINE_START	엔진을 시작합니다.
int	ENGINE_INITIALIZED	엔진 초기화를 완료하였습니다.
int	ENGINE_STOP	엔진이 중지되었습니다.
int	ENGINE_RESET	엔진이 리셋되었습니다.
int	UPDATE_START	업데이트를 시작합니다.
int	UPDATE_COMPLETE	업데이트를 완료하였습니다.
int	UPDATE_DOWNLOAD	업데이트 파일 다운로드가 시작됩니다.
int	UPDATE_DOWNLOADING	업데이트 파일을 다운로드 중입니다.
int	UPDATE_CHECK	업데이트가 존재하는지 검사합니다.
int	UPDATE_FAIL	업데이트가 실패하였습니다.
Int	S2AUTH_CALLBACK	서버 인증을 시도하였습니다.

Fields

public static final int **ENGINE_START**

보안 엔진 초기화를 시작한다는 이벤트입니다.

AppGuardEventListener::onEvent()를 통하여 해당 이벤트 수신 시 추가 데이터로 버전 정보 (int 형 데이터)를 전달 받습니다.

다음은 추가 데이터를 파싱하는 예시를 보입니다.

```
Object[] ver = SecurityEventParser.getInstance().parse(
    SecurityEventType.EVENT,
    SecurityEventType.Event.ENGINE_START,
    data);
```

Constant Value : 0 (0x00000000)

public static final int ENGINE_INITIALIZED

보안 엔진 초기화를 완료하였다는 이벤트입니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 1 (0x00000001)

public static final int ENGINE_STOP

보안 엔진이 정지 되었다는 이벤트입니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 2 (0x00000002)

public static final int ENGINE_RESET

보안 엔진이 리셋 되었다는 이벤트입니다. (해당 이벤트는 Platform:8.2 Build:71.37.16e (20.2d) 버전 이후부터 추가된 이벤트로 일반적으로 재실행 시 발생합니다.)

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 3 (0x00000003)

public static final int UPDATE_START

업데이트 엔진을 시작하였음을 의미합니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 5 (0x00000005)

public static final int UPDATE_COMPLETE

업데이트가 완료되었음을 의미하는 이벤트입니다.

실제로 업데이트가 진행되어 완료되었거나, 또는 현재 로컬의 파일이 최신 상태로 유지되어 있을 경우도 해당 이벤트를 수신합니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 6 (0x00000006)

public static final int UPDATE_DOWNLOAD

업데이트 파일이 존재하여 업데이트 다운로드의 시작을 의미합니다.

AppGuardEventListener::onEvent()를 통하여 해당 이벤트 수신 시 추가 데이터로 업데이트 받을 파일의 수(int 형 데이터)를 전달 받습니다.

Constant Value : 7 (0x00000007)

public static final int UPDATE_DOWNLOADING

업데이트 파일을 다운로드 받고 있음을 의미하는 이벤트입니다.

AppGuardEventListener::onEvent()를 통하여 해당 이벤트 수신 시 추가 데이터로 다운로드 % (Percentage) 정보를 받습니다. 해당 값은 1%부터 100%까지 증가합니다.

다음은 추가 데이터를 파싱하는 예시를 보입니다.

```
Object[] percent= SecurityEventParser.getInstance().parse(  
                                SecurityEventType.EVENT,  
                                SecurityEventType.Event.UPDATE_DOWNLOADING,  
                                data);
```

Constant Value : 8 (0x00000008)

public static final int UPDATE_CHECK

업데이트 파일이 존재하여 업데이트 다운로드의 시작을 의미합니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 9 (0x00000009)

public static final int UPDATE_FAIL

업데이트가 실패하였음을 의미합니다. 해당 이벤트 발생 시 오류(Error) 이벤트가 발행하여 실패 이유를 리포트 합니다.

Constant Value : 10 (0x0000000A)

public static final int **S2AUTH_CALLBACK**

앱가드 모듈이 서버 인증을 시도하였다는 것을 의미합니다. 해당 이벤트로 인증서버에 대한 서버인증 요청의 상태를 성공/재시도/실패로 확인할 수 있습니다.

일반적으로 해당 이벤트 수신 시 앱 서버를 통하여 해당 클라이언트에 대한 인증 확인을 인증 서버로 요청하는 시점으로 사용할 수 있습니다.

다음은 추가 데이터를 파싱하는 예시를 보입니다.

```
Object[] result = SecurityEventParser.getInstance().parse(
    SecurityEventType.EVENT,
    SecurityEventType.Event.S2AUTH_CALLBACK,
    data);

int status = result [0];
String uniqueClientID = result [1];
```

리턴된 Object[] 타입 값은 index 0의 경우(예- ver[0]) Integer 형으로 아래와 같은 상태값을 포함하고 있습니다.

SecurityEventType.S2Auth 클래스의 Field 중 하나를 전달 받습니다.

Constant Value		
int	S2AUTH_RESULT_SUCCESS	인증서버에 서버인증 요청을 성공했습니다. 해당 상태일 경우 추가적으로 인증에 사용된 Unique Client ID값이 제공됩니다.
int	S2AUTH_RESULT_RETRY	인증서버에 접속이 원활하지 않아 재시도합니다.
int	S2AUTH_RESULT_FAIL	인증서버에 서버인증 요청을 실패하였습니다.

리턴된 Object[] 타입 값은 index 1의 경우(예- ver[1]) String 형으로 서버 인증에 사용된 Unique Client ID 값을 리턴합니다. 만약 AppGuardClient.setUniqueClientID()의 첫번째 인자로 null을 지정하였다면 해당 리턴값을 사용하여 인증에 사용된 ID값을 확보할 수 있습니다.

위 index 1의 값은 S2AUTH_RESULT_SUCCESS 상태일 때만 리턴됩니다.
나머지 상태의 경우 empty값을 리턴합니다.

Constant Value에 관한 추가적인 자세한 내용은 [SecurityEventType.S2Auth](#) 클래스를 참조합니다.

Constant Value : 20 (0x00000014)

Public class SecurityEventType.Error

Public class

SecurityEventType.Error

com.inca.security.Common.SecurityEventType.Error

Class Overview

오류 이벤트의 식별 값의 열거자 클래스입니다.

Summary

Fields		
int	ERROR_NOTHING	오류가 발생하지 않았습니다.
int	ERROR_CONNET_TO_UPDATE_SERVER	업데이트 서버에 연결할 수 없습니다.
int	ERROR_DOWNLOAD_UPDATE_FILE	업데이트 정보 파일의 다운로드를 실패하였습니다.
int	ERROR_DOWNLOAD_MODULE_FILE	업데이트 파일의 다운로드를 실패하였습니다.
int	ERROR_CONNECT_TO_GATEWAY_SERVER	게이트웨이 서버에 연결할 수 없습니다.
int	ERROR_KEY_IS_NULL	라이선스 키 정보가 설정되어 있지 않습니다.
int	ERROR_NETWORK_ACTIVATION	네트워크에 연결할 수 없습니다.
int	ERROR_OPEN_REMOTE_UPDATE_FILE	업데이트 정보 파일이 존재하지 않거나 손상되었습니다.
int	ERROR_ENGINE_INIT	엔진 초기화에 실패하였습니다.
int	ERROR_MANDATORY_FILE_DOES_NOT_EXIST	필수 파일이 존재하지 않습니다.
int	ERROR_LIBENGINE_FATAL	보안 엔진에서 치명적인 오류가 발생하였습니다.

Fields

public static final int **ERROR_NOTHING**

오류가 발생하지 않았다는 의미의 이벤트입니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 0 (0x00000000)

public static final int ERROR_CONNET_TO_UPDATE_SERVER

업데이트 서버에 연결할 수 없음을 의미하는 이벤트입니다.

일반적으로 네트워크 연결이 불가능 하거나, 업데이트 서버가 동작하지 않을 경우 발생합니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 1 (0x00000001)

public static final int ERROR_DOWNLOAD_UPDATE_FILE

업데이트 정보 파일의 다운로드를 실패하였음을 의미하는 이벤트입니다.

일반적으로 네트워크 연결 상태가 불안정할 때 발생합니다.

해당 오류 이벤트를 수신하였어도 어플리케이션은 정상 실행 될 수 있으며, 로컬에 저장되어 있는 보안 모듈이 정상적으로 보안 서비스를 실행합니다. 단, 만약 해당 어플리케이션에 서버 인증 기능이 적용되어 있다면 서버에서 해당 클라이언트의 버전이 유효하지 않다는 이유로 연결 세션을 끊어 버릴 것입니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 2 (0x00000002)

public static final int ERROR_DOWNLOAD_MODULE_FILE

업데이트 파일의 다운로드를 실패하였음을 의미하는 이벤트입니다.

일반적으로 네트워크 연결 상태가 불안정할 때 발생합니다.

해당 오류 이벤트를 수신하였어도 어플리케이션은 정상 실행 될 수 있으며, 로컬에 저장되어 있는 보안 모듈이 정상적으로 보안 서비스를 실행합니다. 단, 만약 해당 어플리케이션에 서버 인증 기능이 적용되어 있다면 서버에서 해당 클라이언트의 버전이 유효하지 않다는 이유로 연결 세션을 끊어 버릴 것입니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 3 (0x00000003)

public static final int ERROR_CONNECT_TO_GATEWAY_SERVER

게이트웨이 서버에 연결을 실패하였음을 의미합니다.

일반적으로 네트워크 연결이 불가능 하거나, 게이트웨이 서버가 동작하지 않을 경우 발생합니다.

Constant Value : 4 (0x00000004)

public static final int ERROR_KEY_IS_NULL

라이선스 키 정보가 설정되어 있지 않음을 의미하는 이벤트입니다.

AppGuard를 적용하는 어플리케이션은 AndroidManifest.xml의 <meta-data> 태그를 사용하여 제품 구매 시 발부된 라이선스 키를 등록해야 합니다. 만약 해당 키 정보가 존재하지 않으면 해당 오류 이벤트가 발생합니다. 또한 라이선스 키가 올바르게 설정되어 있지 않으면, 정상적으로 보안정책 적용 및 업데이트가 진행되지 않습니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 5 (0x00000005)

public static final int ERROR_NETWORK_ACTIVATION

네트워크에 연결할 수 없음을 의미합니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 6 (0x00000006)

public static final int ERROR_OPEN_REMOTE_UPDATE_FILE

업데이트 정보 파일이 존재하지 않거나 손상되었음을 의미합니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 7 (0x00000007)

public static final int ERROR_ENGINE_INIT

보안 서비스 엔진 초기화에 실패하였습니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 8 (0x00000008)

public static final int ERROR_MANDATORY_FILE_DOES_NOT_EXIST

실행에 필요한 필수 파일이 존재하지 않습니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 9 (0x00000009)

public static final int ERROR_LIBENGINE_FATAL

보안 엔진 운영 중에 복구할 수 없는 치명적인 오류가 발생하여 더 이상 보안 서비스를 제공할 수 없음을 의미합니다.

해당 이벤트를 수신할 경우 어플리케이션은 반드시 종료되어야 합니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 9999 (0x0000270F)

public class SecurityEventType.Detect

public class

SecurityEventType.Detect

com.inca.security.Common.SecurityEventType.Detect

Class Overview

탐지 이벤트의 식별 값의 열거자 클래스입니다.

탐지 이벤트는 크게 두 개의 파트로 나누어져 있습니다. 이벤트 코드 1 ~ 31까지의 값을 가지는 이벤트는 Critical Level Detection 이벤트로 반드시 해당 이벤트를 수신하는 어플리케이션이 실행을 정지하고 종료 상태로 변화되는 것을 권장하는 이벤트입니다. 모듈 업데이트에 따라 현재 해당 열거자에 존재하지 않는 보안 기능이 추가될 수 있으며 이 경우 SecurityEventType::IsCriticalLevelDetection(int code) 메서드를 사용하여 구별할 수 있습니다.

이벤트 코드 32 ~ 62까지의 값을 가지는 이벤트는 Warning Level Detection 이벤트로 종료될 수 있으나 필요에 따라 정상 진행을 선택할 수 있는 탐지 이벤트를 의미합니다.

Summary

Fields (Normal)		
int	DETECT_NOTHING	보안 위반 탐지가 발생하지 않았습니다.

Fields (Critical Level Detection)		
int	DETECT_WEIRD_ROOT_PROCESS	루트 권한을 가지고 있는 비허용 프로세스가 탐지 되었습니다.
int	DETECT_INVALID_APK_SIGN	APK 파일 유효하지 않음이 탐지 되었습니다.
int	DETECT_INVALID_ODEX_FILE	실행 파일인 ODEX 파일의 변조가 탐지 되었습니다.
int	DETECT_INVALID_CORE_SYSTEM_LIBRARY	기본 시스템 라이브러리 파일의 변조가 탐지 되었습니다.
int	DETECT_INVALID_APPLIB_LIBRARY	어플리케이션에 포함된 네이티브 라이브러리 파일이 유효하지 않습니다.
int	DETECT_INVALID_SECURITY_ENGINE	보안 엔진 파일이 유효하지 않습니다.
int	DETECT_DEBUGGING	디버깅 시도가 탐지 되었습니다.
int	DETECT_MEMORY_TAMPERING	보호 변수의 데이터 영역에 대한 비정상적인

		접근이 탐지 되었습니다.
int	DETECT_INVALID_FRAMEWORK_FILES	시스템의 프레임워크 파일의 변조가 탐지 되었습니다.
int	DETECT_INVALID_LICENSE_USES	현재 앱을 라이선스 정책을 위반하여 사용됨이 탐지되었습니다.
int	DETECT_INVALID_UNITY_ASSEMBLY	메모리에 로드된 UNITY 어셈블리 파일이 변조가 탐지 되었습니다.
int	DETECT_MEMORY_SCANNING	내, 외부 모듈이 앱의 메모리 영역을 스캔하는 행위가 탐지 되었습니다.
int	DETECT_INVALID_SYSTEMAPP_FILES	필수적인 시스템 앱(App) 파일의 손상이 탐지 되었습니다. 일반적으로 구글 결제 우회 공격 기술 중 시스템 앱(App)의 실행 코드를 변조하는 방식의 공격을 탐지합니다.
int	DETECT_RUNNING_BAD_APPLICATION	해킹툴(Hacktool) 어플리케이션이 동작중에 있음이 탐지 되었습니다.
int	DETECT_NATIVE_DEBUGGING	AppGuard 서버 접속 시 MITM 공격이 탐지 되었습니다.
int	DETECT_MEMORY_DUMPING	메모리 덤프 행위가 탐지 되었습니다.
int	DETECT_RUNNING_BAD_PROCESS	해킹툴(Hacktool) 어플리케이션에 소속된 프로세스가 현재 메모리에 로드되어 있음이 탐지 되었습니다.
Int	DETECT_SPEEDHACK	앱 시간(Time)/속도(Speed) 변조 행위가 탐지 되었습니다.
Int	DETECT_MACRO_SUSPICIOUS	매크로 동작 의심 행위가 탐지 되었습니다.

Fields (Warning Level Detection)		
int	DETECT_EMULATOR	에뮬레이터 환경 등 비 표준적인 환경에서 실행됨이 탐지 되었습니다.
int	DETECT_INSTALLED_BAD_APPLICATION	해킹툴(Hacktool) 어플리케이션이 설치되어 있음이 탐지 되었습니다.
int	DETECT_ROOTING_ENVIRONMENT	루팅된 장치에서 실행 됨이 탐지 되었습니다.
int	DETECT_CUSTOMROM	커스텀롬 장치에서 실행 됨이 탐지 되었습니다.
int	DETECT_USBDEBUGGING_FLAG	USB 디버깅이 가능하도록 설정되어 있음이 탐지 되었습니다.
int	DETECT_INVALID_LOCAL_HASH	업데이트 파일이 손상됨이 탐지 되었습니다.
int	DETECT_PROXY	프록시 서버를 사용하도록 설정되어 있음이 탐지 되었습니다.

Fields

public static final int **DETECT_NOTHING**

보안 위반이 탐지 되지 않았음을 의미하는 이벤트입니다.

Constant Value : 0 (0x00000000)

public static final int **DETECT_WEIRD_ROOT_PROCESS**

루트 권한을 가지고 있는 정상적이지 않다고 판단되는 프로세스가 탐지되었음을 의미합니다.

Constant Value : 1 (0x00000001)

public static final int **DETECT_INVALID_APK_SIGN**

APK 파일 유효하지 않음이 탐지 되었습니다.

일반적으로 APK 파일이 리패키징 되었거나 AppGuard Manager Web서버를 통한 보안 서명이 되어 있지 않은 경우 또한 발생할 수 있다.

Constant Value : 3 (0x00000003)

public static final int **DETECT_INVALID_ODEX_FILE**

실행 파일인 ODEX 파일의 변조가 탐지 되었습니다.

Constant Value : 4 (0x00000004)

public static final int **DETECT_INVALID_CORE_SYSTEM_LIBRARY**

기본 시스템 라이브러리 파일의 변조가 탐지 되었습니다.

Constant Value : 6 (0x00000006)

public static final int **DETECT_INVALID_APPLIB_LIBRARY**

어플리케이션에 포함된 네이티브 라이브러리 파일이 유효하지 않습니다.

Constant Value : 9 (0x00000009)

public static final int **DETECT_INVALID_SECURITY_ENGINE**

보안 엔진 파일이 유효하지 않습니다.

Constant Value : 10 (0x0000000a)

public static final int DETECT_DEBUGGING

디버깅 시도가 탐지 되었음을 의미합니다.

디버깅 시도의 탐지는 최상의 보안 위반 사항 중 하나로 취급되어야 하며 어플리케이션은 반드시 어플리케이션을 강제 종료 상태로 유도해야 합니다. 만약 보안 검사 주기에서 2회를 초과하는 디버깅 시도가 탐지되면 강제로 어플리케이션을 종료 시킵니다.

해당 이벤트는 추가 데이터를 제공하지 않습니다.

Constant Value : 11 (0x0000000b)

public static final int DETECT_MEMORY_TAMPERING

보호 변수(SecureInteger 클래스의 인스턴스)의 데이터 영역에 대한 손상되었거나 데이터 영역에 대한 메모리 검사의 수행이 탐지되었음을 의미합니다.

Constant Value : 12 (0x0000000c)

public static final int DETECT_INVALID_FRAMEWORK_FILES

시스템의 프레임워크 파일의 변조가 탐지 되었습니다.

Constant Value : 13 (0x0000000d)

public static final int DETECT_INVALID_LICENSE_USES

현재 앱을 라이선스 정책을 위반하여 사용됨이 탐지되었습니다.

Constant Value : 14 (0x0000000e)

public static final int DETECT_INVALID_UNITY_ASSEMBLY

메모리에 로드된 UNITY 어셈블리 파일이 변조가 탐지 되었습니다.

Constant Value : 17 (0x00000011)

public static final int DETECT_MEMORY_SCANNING

내. 외부 모듈이 앱의 메모리 영역을 스캔하는 행위가 탐지 되었습니다.

Constant Value : 18 (0x00000012)

public static final int DETECT_INVALID_SYSTEMAPP_FILES

필수적인 시스템 앱(App) 파일의 손상이 탐지 되었습니다. 일반적으로 구글 결제 우회 공격

기술 중 시스템 앱(App)의 실행 코드를 변조하는 방식의 공격을 탐지합니다.

Constant Value : 20 (0x00000014)

public static final int **DETECT_RUNNING_BAD_APPLICATION**

해킹(Hacktool) 어플리케이션이 동작중에 있음이 탐지 되었습니다.

다음은 추가 데이터를 파싱하는 예시를 보입니다.

```
Object[] result = SecurityEventParser.getInstance().parse(  
    SecurityEventType.DETECT  
    SecurityEventType.Event.DETECT_RUNNING_BAD_PROCESS,  
    data);  
for ( int i = 0; i < result.length; i++ ) {  
    Log.e("debug", (String)result[i]);  
}
```

Constant Value : 21 (0x00000015)

public static final int **DETECT_NATIVE_DEBUGGING**

AppGuard 실행 중 AppGuard 서버 접속 시 MITM(중간자공격)이 탐지되었습니다.

Constant Value : 22 (0x00000016)

public static final int **DETECT_MEMORY_DUMPING**

해킹(Hacktool) 어플리케이션이 메모리에 접근 하는 행위가 탐지 되었습니다.

Constant Value : 23 (0x00000017)

public static final int **DETECT_RUNNING_BAD_PROCESS**

해킹(Hacktool) 어플리케이션에 소속된 프로세스가 현재 메모리에 로드되어 있음이 탐지 되었습니다.

Constant Value : 25 (0x00000019)

public static final int **DETECT_MACRO_SUSPICIOUS**

화면 터치 이벤트 동작이 매크로 동작으로 의심되는 유형으로 판단되어 있음이 탐지 되었습니다.

Constant Value : 28 (0x0000001C)

public static final int **DETECT_EMULATOR**

에뮬레이터 환경 등 비 표준적인 환경에서 실행 됨이 탐지 되었습니다.

Constant Value : 32 (0x00000020)

public static final int **DETECT_INSTALLED_BAD_APPLICATION**

해킹(Hacktool) 어플리케이션이 설치 또는 동작중에 있음이 탐지 되었습니다.

Constant Value : 33 (0x00000021)

public static final int **DETECT_ROOTING_ENVIRONMENT**

루팅된 장치에서 실행 됨이 탐지 되었습니다.

Constant Value : 34 (0x00000022)

public static final int **DETECT_SPEEDHACK**

앱 시간 속도(Speed) 변조 행위가 탐지 되었습니다.

Constant Value : 35 (0x00000023)

public static final int **DETECT_CUSTOMROM**

커스텀롬 장치에서 실행 됨이 탐지 되었습니다.

Constant Value : 37 (0x00000025)

public static final int **DETECT_USBDEBUGGING_FLAG**

커스텀롬 장치에서 실행 됨이 탐지 되었습니다.

Constant Value : 38 (0x00000026)

public static final int **DETECT_INVALID_LOCAL_HASH**

업데이트 파일이 손상 됨이 탐지 되었습니다.

Constant Value : 39 (0x00000027)

public static final int **DETECT_PROXY**

프록시 서버를 사용하도록 설정되어 있음이 탐지 되었습니다.

Constant Value : 41 (0x00000029)

public class SecurityEventType.S2Auth

public class

SecurityEventType.S2Auth

com.inca.security.Common.SecurityEventType.S2Auth

Class Overview

서버인증 이벤트 식별값의 열거자 클래스입니다.

Fields

public static final int **S2AUTH_RESULT_SUCCESS**

보안모듈이 서버인증을 위해 인증서버에 시도한 인증 요청이 성공했음을 의미하는 이벤트입니다.

Constant Value : 1 (0x00000001)

public static final int **S2AUTH_RESULT_RETRY**

보안모듈이 서버인증을 위해 인증서버에 시도한 인증 요청이 진행중임을 의미하는 이벤트입니다. 해당 이벤트는 네트워크가 불안정하거나 인증서버의 장애 발생시 발생 할 수 있습니다.

Constant Value : 2 (0x00000002)

public static final int **S2AUTH_RESULT_FAIL**

보안모듈이 서버인증을 위해 인증서버에 시도한 인증 요청이 실패했음을 의미하는 이벤트입니다.

Constant Value : 3 (0x00000003)

public class MacroRule

public class

MacroRule

com.inca.security.Macro.MacroRule

Class Overview

커스텀 매크로 탐지 정책을 설정하는 클래스입니다.

매크로 탐지 모드는 Default와 Custom 모드가 존재하며 앱 시작 시 기본적으로 Custom 모드는 Default 모드와 동일한 룰셋으로 설정됩니다.

그럼으로 제공되는 Default 정책에 EventRepeat, EventTime, EventDensity가 활성화되었고 만약 커스텀 매크로에 EventTime 모드만 사용하고자 한다면 EventRepeat과 EventDensity의 Enable을 명시적으로 OFF 하여야합니다.

Summary

Public Methods	
void	setRepeatEnable(boolean enable) EventRepeat 탐지 기능을 활성화 여부를 설정합니다.
void	setRepeatCount(int count) EventRepeat 탐지 기능의 규칙인 반복 패턴의 기준 카운트값을 설정합니다.
void	setEventTimeEnable(boolean enable) EventTime 탐지 기능을 활성화 여부를 설정합니다.
void	setEventTimeSamplingCount(int count) EventTime 탐지 기능의 규칙인 시간값 계산 대상이될 이벤트 샘플링 카운트값을 설정합니다.
void	setEventTimeThresholdRange(int minTime, int maxTime) EventTime 탐지 기능의 규칙인 샘플링된 이벤트의 평균 이벤트 시간의 임계치값을 설정합니다.
void	setDensityEnable(boolean enable) EventDensity 탐지 기능을 활성화 여부를 설정합니다.
void	setDensityRatio(int value) EventDensity 탐지 기능의 규칙인 이벤트 밀집도 기준값을 설정합니다.

Public Methods

```
public void setRepeatEnable(boolean enable)
```

EventRepeat 탐지 기능을 활성화 여부를 설정합니다.

Parameters

<i>enable</i>	활성화 유무값(true/false) / default : true
---------------	--------------------------------------

```
public void setRepeatCount(int count)
```

EventRepeat 탐지 기능의 규칙인 반복 패턴의 기준 카운트값을 설정합니다.

N(count) 개의 패턴이 연속적일 때 탐지로 판단합니다. 만약 해당 값이 너무 낮게 설정되어 있을 경우 오진 발생 가능성이 높아집니다.

Parameters

count 반복 패턴 기준 카운트값 / default : 32

```
public void setEventTimeEnable(boolean enable)
```

EventTime 탐지 기능을 활성화 여부를 설정합니다.

Parameters

<i>enable</i>	활성화 유무값(true/false) / default : true
---------------	--------------------------------------

```
public void setEventTimeSamplingCount (int count)
```

EventTime 탐지 기능의 규칙인 평균 이벤트 시간값 계산에 사용할 최소 이벤트 샘플링 카운트 값을 설정합니다.

만약 해당값을 12으로 지정할 경우 이벤트가 최소 12이상 수집되기 전에는 탐지 유무를 판단하지 않습니다. 만약 게임의 특성에 따라 터치 발생이 적다면 해당 값을 조정하여야하며 너무 낮을 경우 오진이 발생할 가능성이 높아집니다.

Parameters

<i>count</i>	이벤트 샘플링 카운트 값 / default : 12
--------------	------------------------------

```
public void setEventTimeThresholdRange(int minTime, int maxTime)
```

EventTime 탐지 기능의 규칙인 샘플링된 이벤트의 평균 이벤트 시간의 임계치값을 설정합니다.

Parameters

<i>minTime</i>	비정상 범위로 판단한 최소 시간 값(밀리세컨드) / default 0
<i>maxTime</i>	비정상 범위로 판단한 최대시간 값(밀리세컨드) / default 8

```
public void setDensityEnable(boolean enable)
```

EventDensity 탐지 기능을 활성화 여부를 설정합니다.

Parameters

<i>enable</i>	활성화 유무값(true/false) / default : true
---------------	--------------------------------------

```
public void setDensityRatio(int value)
```

EventDensity 탐지 기능의 규칙인 이벤트 밀집도 기준값을 설정합니다.

해당 value가 1에 가까울 경우 발생 이벤트가 특정 좌표로 수렴하고 있다는 의미이며 반대로 100에 가까울 경우에는 이벤트 좌표가 난수성을 가진다는 의미입니다.

Parameters

<i>count</i>	이벤트 밀집도 기준값 / default : 5
--------------	---------------------------

public interface SecureObjectFactory

public interface

SecureObjectFactory

com.inca.security.AppGuard.SecureObjectFactory

Class Overview

보안 기능 객체를 생성하는 ObjectFactory 클래스입니다.

Summary

Public Methods	
abstract SecureInteger	newInstanceOfSecureInteger() SecureInteger 보안 객체를 생성하여 리턴합니다.
abstract SecureInteger	newInstanceOfSecureInteger(int val) SecureInteger 보안 객체를 생성하여 설정된 값으로 초기화하고 리턴합니다.
abstract SecureLong	newInstanceOfSecureLong() SecureLong 보안 객체를 생성하여 리턴합니다.
abstract SecureLong	newInstanceOfSecureLong(long val) SecureLong 보안 객체를 생성하여 설정된 값으로 초기화하고 리턴합니다.
abstract SecureBytes	newInstanceOfSecureBytes() SecureBytes 보안 객체를 생성하여 리턴합니다.
abstract SecureBytes	newInstanceOfSecureBytes(byte[] data) SecureBytes 보안 객체를 생성하여 설정된 값으로 초기화하고 리턴합니다.

Public Methods

```
public abstract SecureInteger newInstanceOfSecureInteger()
```

SecureInteger 보안 객체를 생성하여 리턴합니다.

Returns

성공하면 SecureInteger 타입의 객체를 리턴하고, 실패하면 null을 리턴합니다.

Throws

AppGuardException 치명적인 예외 상황이 발생하였습니다.


```
public abstract SecureInteger newInstanceOfSecureInteger(int val)
```

SecureInteger 보안 객체를 생성하여 설정된 값으로 초기화하고 리턴합니다.

Parameters

val 초기화 값

Returns

성공하면 SecureInteger 타입의 객체를 리턴하고, 실패하면 null을 리턴합니다.

Throws

AppGuardException 치명적인 예외 상황이 발생하였습니다.

```
public abstract SecureLong newInstanceOfSecureLong()
```

SecureLong 보안 객체를 생성하여 리턴합니다.

Returns

성공하면 SecureLong 타입의 객체를 리턴하고, 실패하면 null을 리턴합니다.

Throws

AppGuardException 치명적인 예외 상황이 발생하였습니다.

```
public abstract SecureLong newInstanceOfSecureLong(int val)
```

SecureLong 보안 객체를 생성하여 설정된 값으로 초기화하고 리턴합니다.

Parameters

val 초기화 값

Returns

성공하면 SecureLong 타입의 객체를 리턴하고, 실패하면 null을 리턴합니다.

Throws

AppGuardException 치명적인 예외 상황이 발생하였습니다.

```
public abstract SecureInteger newInstanceOfSecureBytes()
```

SecureBytes 보안 객체를 생성하여 리턴합니다.

Returns

성공하면 SecureBytes 타입의 객체를 리턴하고, 실패하면 null을 리턴합니다.

Throws

AppGuardException 치명적인 예외 상황이 발생하였습니다.

```
public abstract SecureBytes newInstanceOfSecureBytes(byte[] data)
```

SecureBytes 보안 객체를 생성하여 설정된 값으로 초기화하고 리턴합니다.

Parameters

data 초기화 값

Returns

성공하면 SecureBytes 타입의 객체를 리턴하고, 실패하면 null을 리턴합니다.

Throws

AppGuardException 치명적인 예외 상황이 발생하였습니다.

public interface SecureInteger

public interface

SecureInteger

com.inca.security.Interface.SecureInteger

Class Overview

암호화된 int형 데이터 연산 기능을 제공하는 클래스입니다.

다음 해당 클래스의 사용 예를 보입니다.

```
SecureInteger c, a, b, d;

// a = 1;
// b = 2;
a = SecureObjectFactory.newInstanceOfSecureInteger(1);
b = SecureObjectFactory.newInstanceOfSecureInteger(2);

// c = -a + (b * b) + (b / a);
//   = -1 + (2 * 2) + (2 / 1);
//   = -1 + 4 + 2;
//   = 5;
c = a.nagate().add(b.multiply(b)).add(b.divide(a));

// c = (a^2 + b) / 2 * 4 + (a * 8) * -b + 12;
//   = ((1 + 2) / 2 * 4) + (8 * -2) + 12;
//   = 3 / 2 * 4 + -16 + 12;
//   = 4 + -16 + 12;
//   = -12 + 12;
//   = 0
c = a.pow(2).add(b).divide(2).multiply(4).add(a.multiply(8).multiply(b.nagate())).add(12);

// d = c;
d.set(c);
```

Summary

Public Methods	
abstract	add(int val)
SecureInteger	현재 값과 int 값을 더하기(+)하고 결과 값을 리턴한다.

abstract SecureInteger	add(SecureInteger obj) 현재 값과 SecureInteger 값을 더하기(+)하고 결과 값을 리턴한다.
abstract SecureInteger	subtract(int val) 현재 값과 int 값을 빼고(-)하고 결과 값을 리턴한다.
abstract SecureInteger	subtract(SecureInteger obj) 현재 값과 SecureInteger 값을 빼고(-)하고 결과 값을 리턴한다.
abstract SecureInteger	multiply(int val) 현재 값과 int 값을 곱하기(*)하고 결과 값을 리턴한다.
abstract SecureInteger	multiply(SecureInteger obj) 현재 값과 SecureInteger 값을 곱하기(*)하고 결과 값을 리턴한다.
abstract SecureInteger	divide(int val) 현재 값과 int 값을 나누기(/)하고 결과 값을 리턴한다.
abstract SecureInteger	divide(SecureInteger obj) 현재 값과 SecureInteger 값을 나누기(/)하고 결과 값을 리턴한다.
abstract SecureInteger	mod(int val) 현재 값과 int 값을 나눈 몫을 결과 값으로 리턴한다.
abstract SecureInteger	mod(SecureInteger obj) 현재 값과 SecureInteger 값을 나눈 몫을 결과 값으로 리턴한다.
abstract SecureInteger	pow(int val) 현재 값과 int 값을 거듭제곱하고 결과 값을 리턴한다.
abstract SecureInteger	pow(SecureInteger obj) 현재 값과 SecureInteger 값을 거듭제곱하고 결과 값을 리턴한다.
abstract SecureInteger	min(int val) 현재 값과 int 값 중 작은 값을 리턴한다.
abstract SecureInteger	min(SecureInteger obj) 현재 값과 SecureInteger 값 중 작은 값을 리턴한다.
abstract SecureInteger	max(int val) 현재 값과 int 값 중 큰 값을 리턴한다.
abstract SecureInteger	max(SecureInteger obj) 현재 값과 SecureInteger 값 중 큰 값을 리턴한다.
abstract SecureInteger	decrease() 현재 값을 1 감소 시키고 결과 값을 리턴한다.
abstract SecureInteger	increase() 현재 값을 1 증가 시키고 결과 값을 리턴한다.
abstract SecureInteger	abs() 현재 값의 부호를 제거한 절대 값을 리턴한다.
abstract	nagate()

SecureInteger	현재 값에 마이너스(-) 부호값을 추가한 결과 값을 리턴한다.
abstract SecureInteger	and(int val) 현재 값에 bitwise AND 연산을 수행한 결과 값을 리턴한다.
abstract SecureInteger	and(SecureInteger obj) 현재 값에 bitwise AND 연산을 수행한 결과 값을 리턴한다.
abstract SecureInteger	or(int val) 현재 값에 bitwise OR 연산을 수행한 결과 값을 리턴한다.
abstract SecureInteger	or(SecureInteger obj) 현재 값에 bitwise OR 연산을 수행한 결과 값을 리턴한다.
abstract SecureInteger	xor(int val) 현재 값에 bitwise XOR 연산을 수행한 결과 값을 리턴한다.
abstract SecureInteger	xor(SecureInteger obj) 현재 값에 bitwise XOR 연산을 수행한 결과 값을 리턴한다.
abstract boolean	equal(int val) 현재 값과 int 값을 비교하여 동일하다면 true를 리턴한다.
abstract boolean	equal(SecureInteger obj) 현재 값과 SecureInteger 값을 비교하여 동일하다면 true를 리턴한다.
abstract boolean	isZero() 현재 값이 0이라면 true를 리턴한다.
abstract String	toString() 현재 값을 String 형으로 변환하여 리턴한다.
abstract int	intValue() 현재 값을 int 형으로 변환하여 리턴한다.
abstract short	shortValue() 현재 값을 short 형으로 변환하여 리턴한다.
abstract float	floatValue() 현재 값을 float 형으로 변환하여 리턴한다.
abstract double	doubleValue() 현재 값을 double 형으로 변환하여 리턴한다.
abstract void	set(int val) int 값을 현재 객체의 값으로 설정한다.
abstract void	set(SecureInteger obj) SecureInteger 값을 현재 객체의 값으로 설정한다.

Public Methods

```
public abstract SecureInteger add(int val)
```

현재 값과 int 값을 더하기(+)하고 결과 값을 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger add(SecureInteger obj)
```

현재 값과 SecureInteger 값을 더하기(+)하고 결과 값을 리턴한다.

Parameters

obj 설명

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger subtract(int val)
```

현재 값과 int 값을 빼고(-)하고 결과 값을 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger subtract(SecureInteger obj)
```

현재 값과 SecureInteger 값을 빼고(-)하고 결과 값을 리턴한다.

Parameters

obj 설명

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger multiply(int val)
```

현재 값과 int 값을 곱하기(*)하고 결과 값을 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger multiply(SecureInteger obj)
```

현재 값과 SecureInteger 값을 곱하기(*)하고 결과 값을 리턴한다.

Parameters

obj 설명

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger divide(int val)
```

현재 값과 int 값을 나누기(/)하고 결과 값을 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger divide(SecureInteger obj)
```

현재 값과 SecureInteger 값을 나누기(/)하고 결과 값을 리턴한다.

Parameters

obj SecureInteger형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger mod(int val)
```

현재 값과 int 값을 나눈 몫을 결과 값으로 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger mod(SecureInteger obj)
```

현재 값과 SecureInteger 값을 나눈 몫을 결과 값으로 리턴한다.

Parameters

obj SecureInteger형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger pow(int val)
```

현재 값과 int 값을 거듭제곱하고 결과 값을 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger pow(SecureInteger obj)
```

현재 값과 SecureInteger 값을 거듭제곱하고 결과 값을 리턴한다.

Parameters

obj SecureInteger형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger min(int val)
```

현재 값과 int 값 중 작은 값을 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.


```
public abstract SecureInteger min(SecureInteger obj)
```

현재 값과 SecureInteger 값 중 작은 값을 리턴한다.

Parameters

obj SecureInteger형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger max(int val)
```

현재 값과 int 값 중 큰 값을 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger max(SecureInteger obj)
```

현재 값과 SecureInteger 값 중 큰 값을 리턴한다.

Parameters

obj SecureInteger형 변수 값

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger decrease()
```

현재 값을 1 감소 시키고 결과 값을 리턴한다.

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger increase()
```

현재 값을 1 증가 시키고 결과 값을 리턴한다.

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger abs()
```

현재 값의 부호를 제거한 절대 값을 리턴한다.

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureInteger negate()
```

현재 값에 마이너스(-) 부호값을 추가한 결과 값을 리턴한다.

Returns

성공하면 새로운 SecureInteger 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract boolean equal(int val)
```

현재 값과 int 값을 비교하여 동일하다면 true를 리턴한다.

Parameters

val int형 변수 값

Returns

성공하면 true를 리턴하고, 실패하면 false를 리턴한다.

```
public abstract boolean equal(SecureInteger obj)
```

현재 값과 SecureInteger 값을 비교하여 동일하다면 true를 리턴한다.

Parameters

<i>obj</i>	SecureInteger형 변수 값
------------	---------------------

Returns

성공하면 true를 리턴하고, 실패하면 false를 리턴한다.

```
public abstract boolean isZero()
```

현재 값이 0이라면 true를 리턴한다.

Returns

성공하면 true를 리턴하고, 실패하면 false를 리턴한다.

```
public abstract String toString()
```

현재 값을 String 형으로 변환하여 리턴한다.

Returns

성공하면 String 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract int intValue()
```

현재 값을 int 형으로 변환하여 리턴한다.

Returns

int 형 데이터를 리턴한다.

```
public abstract short shortValue()
```

현재 값을 short 형으로 변환하여 리턴한다.

Returns

short 형 데이터를 리턴한다.

```
public abstract float floatValue()
```

현재 값을 float 형으로 변환하여 리턴한다.

Returns

float 형 데이터를 리턴한다.

```
public abstract double doubleValue()
```

현재 값을 double 형으로 변환하여 리턴한다.

Returns

double 형 데이터를 리턴한다.

```
public abstract void set(int val)
```

int 값을 현재 객체의 값으로 설정한다.

Parameters

val int형 변수 값

```
public abstract void set(SecureInteger obj)
```

SecureInteger 값을 현재 객체의 값으로 설정한다.

Parameters

obj SecureInteger형 변수 값

public interface SecureLong

public interface

SecureLong

com.inca.security.Interface.SecureLong

Class Overview

암호화된 long형 데이터 연산 기능을 제공하는 클래스입니다.

다음 해당 클래스의 사용 예를 보입니다.

```
SecureLong c, a, b, d;

// a = 1;
// b = 2;
a = SecureObjectFactory.newInstanceOfSecureLong(1);
b = SecureObjectFactory.newInstanceOfSecureLong(2);

// c = -a + (b * b) + (b / a);
//   = -1 + (2 * 2) + (2 / 1);
//   = -1 + 4 + 2;
//   = 5;
c = a.nagate().add(b.multiply(b)).add(b.divide(a));

// c = (a^2 + b) / 2 * 4 + (a * 8) * -b + 12;
//   = ((1 + 2) / 2 * 4) + (8 * -2) + 12;
//   = 3 / 2 * 4 + -16 + 12;
//   = 4 + -16 + 12;
//   = -12 + 12;
//   = 0
c = a.pow(2).add(b).divide(2).multiply(4).add(a.multiply(8).multiply(b.nagate())).add(12);

// d = c;
d.set(c);
```

Summary

Public Methods	
abstract	add(long val)

SecureLong	현재 값과 long 값을 더하기(+)하고 결과 값을 리턴한다.
abstract SecureLong	add(SecureLong obj) 현재 값과 SecureLong 값을 더하기(+)하고 결과 값을 리턴한다.
abstract SecureLong	subtract(long val) 현재 값과 long 값을 빼고(-)하고 결과 값을 리턴한다.
abstract SecureLong	subtract(SecureLong obj) 현재 값과 SecureLong 값을 빼고(-)하고 결과 값을 리턴한다.
abstract SecureLong	multiply(long val) 현재 값과 long 값을 곱하기(*)하고 결과 값을 리턴한다.
abstract SecureLong	multiply(SecureLong obj) 현재 값과 SecureLong 값을 곱하기(*)하고 결과 값을 리턴한다.
abstract SecureLong	divide(long val) 현재 값과 long 값을 나누기(/)하고 결과 값을 리턴한다.
abstract SecureLong	divide(SecureLong obj) 현재 값과 SecureLong 값을 나누기(/)하고 결과 값을 리턴한다.
abstract SecureLong	mod(long val) 현재 값과 long 값을 나눈 몫을 결과 값으로 리턴한다.
abstract SecureLong	mod(SecureLong obj) 현재 값과 SecureLong 값을 나눈 몫을 결과 값으로 리턴한다.
abstract SecureLong	pow(long val) 현재 값과 long 값을 거듭제곱하고 결과 값을 리턴한다.
abstract SecureLong	pow(SecureLong obj) 현재 값과 SecureLong 값을 거듭제곱하고 결과 값을 리턴한다.
abstract SecureLong	min(long val) 현재 값과 long 값 중 작은 값을 리턴한다.
abstract SecureLong	min(SecureLong obj) 현재 값과 SecureLong 값 중 작은 값을 리턴한다.
abstract SecureLong	max(long val) 현재 값과 long 값 중 큰 값을 리턴한다.
abstract SecureLong	max(SecureLong obj) 현재 값과 SecureLong 값 중 큰 값을 리턴한다.
abstract SecureLong	decrease() 현재 값을 1 감소 시키고 결과 값을 리턴한다.
abstract SecureLong	increase() 현재 값을 1 증가 시키고 결과 값을 리턴한다.
abstract SecureLong	abs() 현재 값의 부호를 제거한 절대 값을 리턴한다.

abstract SecureLong	nagate() 현재 값에 마이너스(-) 부호값을 추가한 결과 값을 리턴한다.
abstract SecureLong	and(long val) 현재 값에 bitwise AND 연산을 수행한 결과 값을 리턴한다.
abstract SecureLong	and(SecureLong obj) 현재 값에 bitwise AND 연산을 수행한 결과 값을 리턴한다.
abstract SecureLong	or(long val) 현재 값에 bitwise OR 연산을 수행한 결과 값을 리턴한다.
abstract SecureLong	or(SecureLong obj) 현재 값에 bitwise OR 연산을 수행한 결과 값을 리턴한다.
abstract SecureLong	xor(long val) 현재 값에 bitwise XOR 연산을 수행한 결과 값을 리턴한다.
abstract SecureLong	xor(SecureLong obj) 현재 값에 bitwise XOR 연산을 수행한 결과 값을 리턴한다.
abstract boolean	equal(long val) 현재 값과 long 값을 비교하여 동일하다면 true를 리턴한다.
abstract boolean	equal(SecureLong obj) 현재 값과 SecureLong 값을 비교하여 동일하다면 true를 리턴한다.
abstract boolean	isZero() 현재 값이 0이라면 true를 리턴한다.
abstract String	toString() 현재 값을 String 형으로 변환하여 리턴한다.
abstract int	intValue() 현재 값을 long 형으로 변환하여 리턴한다.
abstract long	longValue() 현재 값을 long 형으로 변환하여 리턴한다.
abstract short	shortValue() 현재 값을 short 형으로 변환하여 리턴한다.
abstract float	floatValue() 현재 값을 float 형으로 변환하여 리턴한다.
abstract double	doubleValue() 현재 값을 double 형으로 변환하여 리턴한다.
abstract void	set(long val) long 값을 현재 객체의 값으로 설정한다.
abstract void	set(SecureLong obj) SecureLong 값을 현재 객체의 값으로 설정한다.

Public Methods

```
public abstract SecureLong add(long val)
```

현재 값과 long 값을 더하기(+)하고 결과 값을 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong add(SecureLong obj)
```

현재 값과 SecureLong 값을 더하기(+)하고 결과 값을 리턴한다.

Parameters

obj 설명

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong subtract(long val)
```

현재 값과 long 값을 빼고(-)하고 결과 값을 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong subtract(SecureLong obj)
```

현재 값과 SecureLong 값을 빼고(-)하고 결과 값을 리턴한다.

Parameters

obj 설명

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong multiply(long val)
```

현재 값과 long 값을 곱하기(*)하고 결과 값을 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong multiply(SecureLong obj)
```

현재 값과 SecureLong 값을 곱하기(*)하고 결과 값을 리턴한다.

Parameters

obj 설명

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong divide(long val)
```

현재 값과 long 값을 나누기(/)하고 결과 값을 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong divide(SecureLong obj)
```

현재 값과 SecureLong 값을 나누기(/)하고 결과 값을 리턴한다.

Parameters

obj SecureLong형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong mod(long val)
```

현재 값과 long 값을 나눈 몫을 결과 값으로 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong mod(SecureLong obj)
```

현재 값과 SecureLong 값을 나눈 몫을 결과 값으로 리턴한다.

Parameters

obj SecureLong형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong pow(long val)
```

현재 값과 long 값을 거듭제곱하고 결과 값을 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong pow(SecureLong obj)
```

현재 값과 SecureLong 값을 거듭제곱하고 결과 값을 리턴한다.

Parameters

obj SecureLong형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong min(long val)
```

현재 값과 long 값 중 작은 값을 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong min(SecureLong obj)
```

현재 값과 SecureLong 값 중 작은 값을 리턴한다.

Parameters

<i>obj</i>	SecureLong형 변수 값
------------	------------------

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong max(long val)
```

현재 값과 long 값 중 큰 값을 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong max(SecureLong obj)
```

현재 값과 SecureLong 값 중 큰 값을 리턴한다.

Parameters

<i>obj</i>	SecureLong형 변수 값
------------	------------------

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong decrease()
```

현재 값을 1 감소 시키고 결과 값을 리턴한다.

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong increase()
```

현재 값을 1 증가 시키고 결과 값을 리턴한다.

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong abs()
```

현재 값의 부호를 제거한 절대 값을 리턴한다.

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract SecureLong negate()
```

현재 값에 마이너스(-) 부호값을 추가한 결과 값을 리턴한다.

Returns

성공하면 새로운 SecureLong 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract boolean equal(long val)
```

현재 값과 long 값을 비교하여 동일하다면 true를 리턴한다.

Parameters

val long형 변수 값

Returns

성공하면 true를 리턴하고, 실패하면 false를 리턴한다.

```
public abstract boolean equal(SecureLong obj)
```

현재 값과 SecureLong 값을 비교하여 동일하다면 true를 리턴한다.

Parameters

<i>obj</i>	SecureLong형 변수 값
------------	------------------

Returns

성공하면 true를 리턴하고, 실패하면 false를 리턴한다.

```
public abstract boolean isZero()
```

현재 값이 0이라면 true를 리턴한다.

Returns

성공하면 true를 리턴하고, 실패하면 false를 리턴한다.

```
public abstract String toString()
```

현재 값을 String 형으로 변환하여 리턴한다.

Returns

성공하면 String 객체를 리턴하고, 실패하면 null을 리턴한다.

```
public abstract int intValue()
```

현재 값을 long 형으로 변환하여 리턴한다.

Returns

int 형 데이터를 리턴한다.

```
public abstract long longValue()
```

현재 값을 long 형으로 변환하여 리턴한다.

Returns

long 형 데이터를 리턴한다.

```
public abstract short shortValue()
```

현재 값을 short 형으로 변환하여 리턴한다.

Returns

short 형 데이터를 리턴한다.

```
public abstract float floatValue()
```

현재 값을 float 형으로 변환하여 리턴한다.

Returns

float 형 데이터를 리턴한다.

```
public abstract double doubleValue()
```

현재 값을 double 형으로 변환하여 리턴한다.

Returns

double 형 데이터를 리턴한다.

```
public abstract void set(long val)
```

long 값을 현재 객체의 값으로 설정한다.

Parameters

val long형 변수 값

```
public abstract void set(SecureLong obj)
```

SecureLong 값을 현재 객체의 값으로 설정한다.

Parameters

obj SecureLong형 변수 값

public interface SecureBytes

public interface

SecureBytes

com.inca.security.Interface.SecureBytes

Class Overview

Bytes 데이터 또는 JSON 형태와 같은 String 데이터를 암호/복호화 기능을 제공하는 클래스입니다.

다음 해당 클래스의 사용 예를 보입니다.

```
SecureBytes json;

json = SecureObjectFactory.newInstanceOfSecureBytes();

String jsonStr = "JSON => (name:value)";
json.set(jsonStr.getBytes());

Log.i(TAG, new String(json.get()));
```

Summary

Public Methods	
abstract void	set(byte[] data) 주어진 데이터를 암호화 하여 저장합니다.
abstract byte[]	get() 해당 객체에 저장된 데이터를 복호화 하여 리턴합니다.

Public Methods

```
public abstract void set(byte[] data)
```

주어진 데이터를 암호화 하여 저장합니다.

Parameters

data byte[]형 변수 값

```
public abstract byte[] get()
```

해당 객체에 저장된 데이터를 복호화 하여 리턴합니다.

Returns

성공하면 새로운 byte[] 객체를 리턴하고, 실패하면 null을 리턴한다.

public class AppGuardIORequestPacket

public class

AppGuardIORequestPacket

com.inca.security.AppGuard.AppGuardIORequestPacket

Class Overview

이 클래스는 AppGuard End-to-End Encryption을 적용 시 클라이언트에서 패킷 암호/복호화를 위하여 제공되는 클래스입니다.

하나의 앱은 하나의 종단간 암호화 터널(tunnel) 만을 만들 수 있으며, 만약 여러 개의 해당 클래스 객체를 생성하였다고 하여도 모든 객체는 동일한 터널 정보를 공유합니다. 즉, 클라이언트는 오직 하나의 암호화 서버와 통신을 수행할 수 있습니다.

해당 클래스를 사용하기 위해서는 반드시 서버 측에서도 'appguard-server-sdk' 라이브러리를 사용한 적용 작업이 선행되어야 합니다.

Summary

Public Methods	
static AppGuardIORequestPacket	getInstance() AppGuardIORequestPacket 객체를 반환합니다.
byte[]	buildPacket(byte[] data) 서버로 전달할 암호화된 패킷을 생성하여 반환합니다.
String	buildJsonPacket(String json) 서버로 전달할 Json등의 String데이터를 암호화된 패킷을 생성하여 String 형태의 암호화된 값으로 반환합니다.
byte[]	parsePacket(byte[] data) 서버에서 수신한 암호화된 패킷을 복호화하여 반환합니다.
String	parseJsonPacket(String json) 서버에서 수신한 String형태의 암호화된 패킷을 생성하여 String 형태의 복호화된 값으로 반환합니다.
void	setUuidAttachment() 패킷에 클라이언트 UniqueClientID를 첨부하여 전송하도록 설정합니다.

void	prepare(byte[] data) 테스트 서버와 통신을 위하여 Null 키(Key) 모드로 전환합니다.
void	reset() 해당 객체를 초기화 합니다.

Public Methods

```
public static AppGuardIORequestPacket getInstance()
```

Singleton 형태의 객체 반환 메서드로 해당 클래스의 공용 객체를 생성하여 반환합니다.

```
public byte[] buildPacket(byte[] data)
```

인자로 전달받은 byte[] 형태의 데이터를 서버로 전달할 암호화된 패킷으로 변환하여 해당 데이터를 반환합니다.

Parameters

data 서버로 전달할 Plain 데이터

public String **buildJsonPacket**(String json) throws Exception

인자로 전달받은 String 형태의 데이터를 서버로 전달할 String 형태의 암호화된 패킷으로 변환하여 해당 데이터를 반환합니다.

일반적으로 Json 형태의 데이터를 서버로 전송 시 해당 메서드를 사용하는 것을 권장하여 서버에서는 해당 메서드로 생성된 패킷을 수신 시 `parseJsonPacket`으로 복호화하여야 합니다.

Parameters

json 클라이언트로 전달할 Json등의 String 형태의 Plain 데이터

Returns

암호화된 패킷을 리턴합니다.

```
public byte[] parsePacket(byte[] data)
```

서버에서 전달 받은 암호화된 byte[] 형태의 데이터 패킷을 복호화 하여 반환합니다.

Parameters

data 서버에서 전달 받은 Encrypted 데이터

public String **parseJsonPacket**(String json) throws Exception

서버에서 전달 받은 String 형태(서버에서 buildJsonPacket으로 생성한)의 암호화된 String 형태의 데이터 패킷을 복호화 하여 반환합니다.

서버에서 buildJsonPakcet을 사용하여 암호화된 Json 형태의 패킷을 클라이언트로 전송 시 해당 메서드를 사용하는 복호화하여야 합니다.

Parameters

json	클라이언트에서 전달 받은 String 형식의 Encrypted 데이터
------	--

Returns

복호화된 패킷을 리턴합니다.

```
public void setUuidAttachment()
```

패킷을 생성 시 클라이언트의 UniqueClientID를 첨부합니다.

서버에서 UniqueClientID가 포함된 패킷을 파싱(Parsing) 시 어떤 클라이언트가 생성한 패킷인지 확인할 수 있습니다. 해당 API를 호출한 시점 이후에 패킷에만 해당 정보가 포함되어며 데이터 패킷의 용량이 증가할 수 있습니다.

REST 방식을 서버를 설계하여 클라이언트 식별이 불가능한 경우 해당 API를 사용할 수 있습니다.

```
public void prepare(byte[] data)
```

테스트 서버와 통신을 위하여 Null 키(Key) 모드로 전환합니다. 해당 메서드를 사용하기 위해서는 인자로 반드시 null 값을 전달하여야 합니다.

AppGuard 빌드 서버(Manager 서버)를 이용한 Appguarding 수행 전에 종단간 암호화 테스트를 위해서는 해당 메시지를 사용하여 암호화 운영 모드를 Null Key 모드로 임시 전환할 수 있습니다.

Parameters

data 반드시 null 값을 사용

```
public void reset()
```

암호화 객체를 초기화(reset) 합니다.

AppGuardIORequestPacket 클래스는 객체 첫 호출 시점에 반드시 `reset()` 메서드를 호출할 것을 요구합니다. 일반적으로 `MainActivity` 또는 `LauncherActivity`의 생명 주기 메서드 중 하나인 `onCreate()` 에서 해당 메서드를 호출하는 것을 권장합니다.



■ (주)잉카인터넷

메일: appguard@inca.co.kr

서울시 구로구 구로3동 235-2 에이스 하이엔드타워 12층 1201호 (우 152-848)

Security, For a More Joyful play.

Copyright ©INCA Internet Corp. All rights reserved.