Freemium SDK

> **NOTE:** This overview refers to the App Protector free SDK! To view the full overview of the App Protector solution go to the "App Protector SDK" page.

Introduction

App Protector Freemium SDK enables limited threat detection on mobile devices.

Supported detections are:

- Jailbreaking detection for iOS devices;
- Rooting detection for Android devices;

Supported reactions to detected threats are:

- Notify the user;

Supported detections and reactions are described below.

Security detections

Jaibreak detection

Jailbreaking is the process of removing the limitations put in place by a device's manufacturer. Jailbreaking is generally performed on Apple iOS devices, such as the iPhone or iPad. Jailbreaking removes the restrictions Apple puts in place, allowing you to install third-party software from outside the app store. Some people may have the perception that jailbreaking is only used for piracy, but this is not the case – jailbreaking allows you to do things like change your iPhone's default browser and mail client. Essentially, jailbreaking allows you to use software that Apple doesn't approve of. Also, it removes restrictions in inter-application communications, accessing files of other applications, and gives the user root access.

App Protector detects if the device is jailbroken and also has a large "blacklist" of unwanted libraries and checks if some library from the "blacklist" exists on the client's device. Furthermore, App Protector checks if the client's device has access permission to suspicious directories (directories to which the client does not have access permission by default).

Root detection

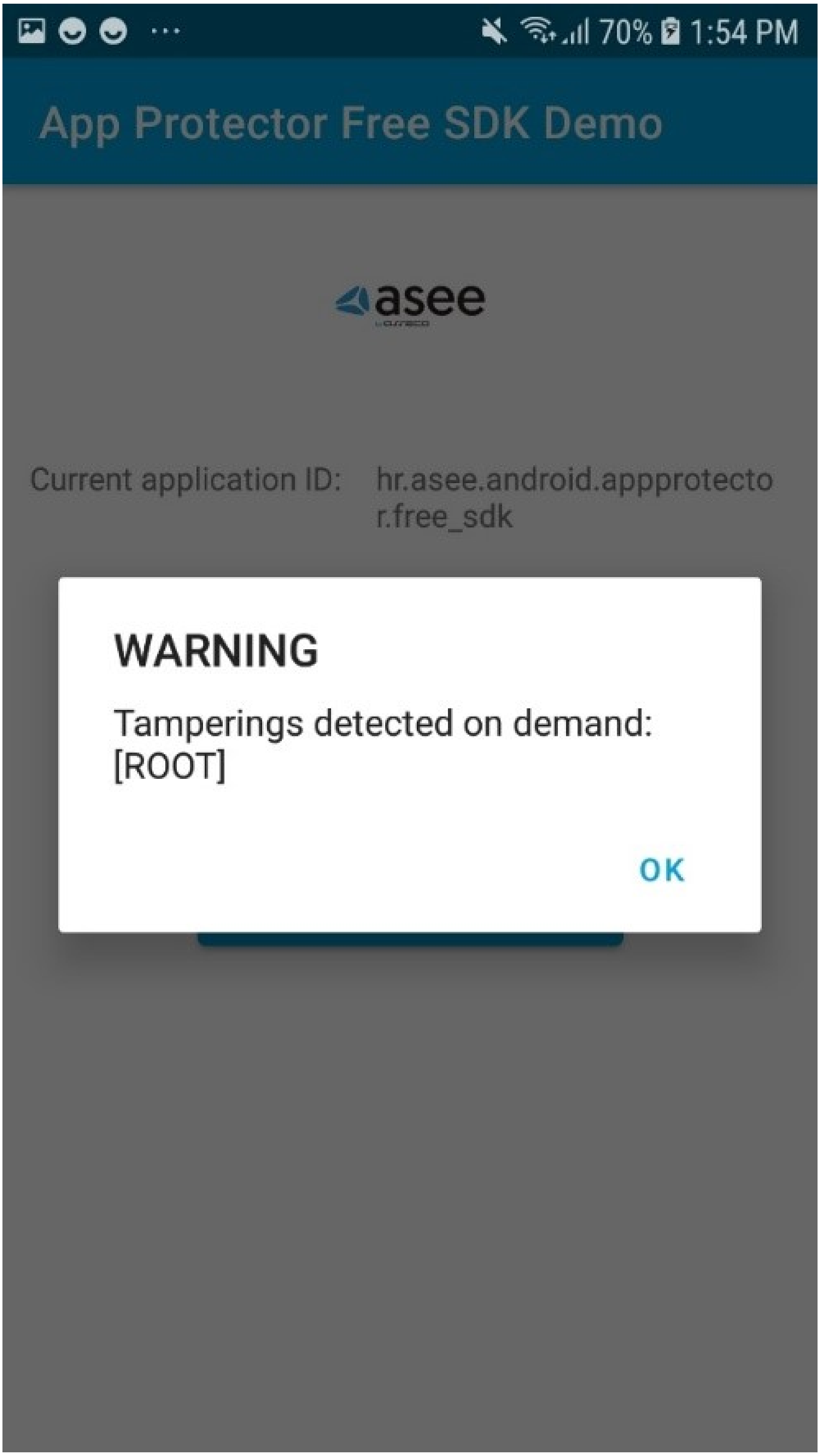
Rooting is the process of removing the limitations put in place by a device's manufacturer and gaining superuser access to device resources as on Linux. Rooting is generally performed on Android devices. The process is usually done in order to overcome the limitations that carriers put on mobile devices. Thus, rooting gives permission to alter or replace system applications and settings. It also gives the user permission to execute privileged commands which are not available to the device in stock configuration. It is also required in order for more advanced and potentially dangerous operations including modifying or deleting system files, removing preinstalled applications, and low-level access to the hardware itself.

App Protector will detect whether any of the known root packages exist on the client device and whether the kernel that is installed was signed with non-release keys while it was compiled. Besides that, App Protector will detect whether root privileges, which are not available on a non-rooted device, are available on the client device.

Security detections reactions

Notify user

When a security event is detected, App Protector returns information about the detected problem on the device – to the application that implements it. This information can be shown to the user. This reaction is on the information level, only reporting the detected event. These events can be handled on the application level, i.e., for root detection, the user can be just informed about the possible security issue, or the application can be terminated. Example of notifying user:



Current application ID: hr.asee.android.appprotector.free_sdk

WARNING

Tamperings detected on demand: [ROOT]

Notify the user