Using ThreatMetrix SDK

Using ThreatMetrix SDK

Getting the Instance

The method getInstance() is used to return a singleton instance of the TrustDefender Object. The instance is only required to be initialized once. Please note that further init() calls do not have any impact.

Initialization

ThreatMetrix SDK is initialized asynchronously at startup by calling the the init() function which is configured with the Config object. At a minimum, the Context and Org ID **must** be specified.

The init() function will perform initial setup and also start enumerating the name of all installed packages on the device. The package name enumeration is performed to support the root detection functionality and is enabled by default.

The following settings are configured with the Config object. The following optional configuration is only required to be set once during the lifetime of the app.

Mandatory Properties

setContext	Sets the Application Context.
setOrgid	Sets the Org Id

Optional Properties

setFPServer	This is the fully qualified domain name (FQDN) of the server that ThreatMetrix SDK will communicate with in order to transmit the collected device attributes. This will only need to be explicitly specified if you have Enhanced Profiling configured. Please see the Enhanced Profiling article for more information. This parameter must be specified in a FQDN format, eg: host.domain.com, NOT in URL format. See Code Samples.
setRegisterForLocationServices	Method for enabling Location Services that will gather positional information about the device being profiled. More information about Location Services may be found here: Location Services. â
setDisableWebView	ThreatMetrix SDK utilizes the WebView API to gather browser string / user agent information. Certain third-party web toolkits that offer JavaScript and HTML frameworks (such as Apache's Cordova) have been identified to cause conflicts with our use of WebView. If you are using such toolkits and identify issues relating to WebView please use the disableWebview option. ThreatMetrix SDK will then use a different method of constructing the user agent and as such, the accuracy of the device profiling is not affected.
setDisableInitPackageScan	Method used for enumerating the package names present on the device.
setScreenOffTimeout	This option is used to configure the time (in seconds) from the time the screen is off to when profiling should be disabled. The default value is 180 seconds.

setDisableLocSerOnBatteryLow	This option is used to disable Location services when the battery is low.
setEnableCallbackOnFailure	This option is used to enable the callback method for all instances regardless of the status of the initial result of profiling.
setCertificateHash	This option is used to add the SHA1 of the certificate of the FP server to the SDK. See Certificate Pinning.

Profiling

Profiling is the term given to collecting a large collection of device attributes and transmitting them to the ThreatMetrix Platform for evaluation. Profiling begins when the doProfileRequest()is called after the init() call has completed. The setEndNotifier() method must be called to get the result of the profiling. Additionally, during this request, a session id, up to 5 custom attributes and a custom location may be optionally passed.

As with the init() function, the doProfileRequest() function will also perform initial setup and also start enumerating the name of all installed packages on the device. The package name enumeration is performed to support the root detection functionality and is enabled by default. The doProfileRequest() function always interrupts any other function and its associated scanning, including the package name enumeration during init() and doPackageScan().

The customer's web server then performs a Session Query API to get detailed information about the session.

These options may be configured with the ProfilingOptions object:

Mandatory Property

setEndNotifier	Method used for getting the result once profiling is complete. Please note that the callback pertains to the current profiling session.

Please note that if the **setEnableCallbackOnFailure** is set to true, then the callback occurs regardless of whether the result is a success or failure, or if the profiling request was cancelled.

Optional Properties

setLocation	Sets a location for use during the next profiling request. If the parent application already has loc before profiling occurs, instead of registering for location updates.
setCustomAttributes	Sets up to 5 custom attributes to be passed during the next profiling request. The data passed Platform in the following attributes: custom_mobile_1 -5. These should be passed in the form o used. They will be stored sequentially into the custom_mobile_n attributes when transmitted to
setSessionID	The customer may optionally pass a Session ID that has been generated by the parent applica customer is already generating a unique session identifier.

Please review the Session ID article for more information on the Session ID as well at the formatting requirements before continuing.

Package Scanning for Mobile Application Reputation

After the doProfileRequest() has completed, the doPackageScan() function should be called to begin full package scanning. This differs from the Init and Profile package scans in that instead of just collecting package names, it will generate an MD5 for every package present on the device and also evaluate the associated metadata for each application. This metadata is used for Mobile Application Reputation and Malware Detection.

This functionality has deliberate been split from the doProfileRequest() function to give customers maximum flexibility when implementing ThreatMetrix SDK. Depending on the number of packages installed and the hardware capabilities of the device, the doPackageScan() function can can take several seconds to complete. This results of the full package scan are then stored in a local database to improve the performance of subsequent package scans.

The doPackageScan() function has one mandatory argument: a timeout, specified in milliseconds. Passing a value of 0 to the timeout argument will disable the timeout. Please see the Code Samples article for an example.

Please read the Mobile Application Reputation article for more information.

Obtaining Profilingâ Results

The result of profiling can be obtained from the argument of the following callback method:

```
void complete (ProfilingResult result);
```

Please note that the getResult() method is deprecated in SDK 4.1 and will be removed in the next major release.

Additional Operations

Cancelling Profiling Requests

Cancelling requests can be achieved using the cancel() method.

Code Samples

The Code Samples page contains example usage code for all of the options described above. Please use this article as a guideline during your own implementation of ThreatMetrix SDK.