Fraunhofer
SIT

Appicaptor Report

Results for Telecooperation Lab. TU Darmstadt

Fraunhofer Institute for
Secure Information Technology (SIT)

September 1, 2016

For internal use only!

„Investment in your Future“

Investments for this work were co-funded
by the European Union with European regional
development funds and by the state
government of Hessen

Fraunhofer SIT contact person
Dr. Jens Heider
Fraunhofer Institute for Secure Information Technology (SIT)
Rheinstraße 75, 64295 Darmstadt, Germany
Email: jens.heider@sit.fraunhofer.de
Phone: +49 (0) 61 51/869-233
Fax: +49 (0) 61 51/869-224

Contents

1 Terms of Use

The Results and accompanying information generated by Fraunhofer SIT and provided to the client are protected by copyright for Fraunhofer Gesellschaft e. V., all rights reserved. The Results will be provided to the client at Fraunhofer SIT' sole discretion and are be subject to strict confidentiality and use restrictions as detailed below as the Results - among others - contain benchmark test results with regard to third party software.

The client shall only be granted a non-exclusive, non-transferable, non-sublicensable right to use the Results for its own internal evaluation purposes only. The client shall not be entitled to release, transfer, assign, rent, lease, sell, disclose or otherwise publish the Results.

The client shall not be entitled to allow access to the Results - in whole or in part - or any information contained therein by any third party and shall be liable that its employees shall comply with the obligations above.

Each violation of the restrictions to use the Results as outlined above by the client shall be subject to damage claims and claims to refrain from any unauthorized use of the Results. In addition, the client shall indemnify Fraunhofer from any third party claim resulting from the client's violation of these obligations.

2 Overview

Appicaptor is a framework for semi-automated security testing of apps. Generated by the framework, this report represents an aggregated interpretation of the performed tests to answer questions about security and privacy related properties of apps.

The apps listed in Table 2.1 were selected by the customer to be tested with the Appicaptor Framework. For each app a test model was derived which describes the nature of the app best. The test model is used to configure tests and it provides information for correlating single test results to an overall result. A generic model is applied for apps that are not tagged for tests specific to a certain class of apps. The listed versions corresponds to the values specified in the app archives and may differ from those displayed in the app store if a developer had chosen to use a diffrent version string for the app store.

Table 2.1: Overview of tested apps, versions and applied test models

App Name	Version	OS	Test Model
8 Ball Pool	3.6.2	Android	Game
Candy Crush Saga	1.82.1.1	Android	Game
Clash of Clans	8.332.16	Android	Game
Cosmic Challenge	2.1	Android	Game
Criminal Case	2.12	Android	Game
Duck Hunting	1.2	Android	Game
Hill Climb Racing	1.30.0	Android	Game
Luftangriff des Helikopters	1.0.3	Android	Game
Mein Talking Tom	3.6.3.42	Android	Game
Meine Talking Angela	2.6.0.19	Android	Generic
My Dolphin Show	2.1.57	Android	Game
Racing in Car	1.1	Android	Game
Subway Surfers	1.59.1	Android	Generic
Teen Patti Gold	1.85.1	Android	Game
Temple Run	1.6.1	Android	Game
Temple Run 2	1.27	Android	Game
Township	4.0.1	Android	Generic
Traffic Rider	1.2	Android	Game
Train Simulator 2016	2.5	Android	Game
Walking Dead: Road to Survival	2.7.3.36682	Android	Game

3 Results

The presented results are based on automated test procedures. All test metrics are carefully chosen and cross-checked. For stating a single app property, multiple independent tests are conducted and correlated to prevent incorrect results. Conflicting results or results that break specified assumptions are denoted by a question mark in the results to prevent false interpretation. Those potential ambiguous results are subject to further improvements of test procedures by integrating insights of manual investigations into improved tests.

Due to the nature of automated tests, however, the correctness of the presented results can not be guaranteed. The results are based on work created to the best of our knowledge and belief.

Table 3.1: Legend

☒	tested property was found
☒ *i*	tested property was found (see detail section for limitations)
☐	tested property was not found
☐ *i*	tested property was not found (see detail section for limitations)
☑	test created proper test results
☐ –	test created no test results
☐ ?	test created conflicting results
☐ ⚡	error conditions during test

3.1 8 Ball Pool (Android)

3.1.1 Tests

The following Table 3.2 summarizes the results of the Android app `8 Ball Pool` with version *3.6.2*.

Table 3.2: Overview of summarized test results for »8 Ball Pool«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☒	*Violations of default policy? Yes.*
Communication security	

☒ *Client communication used? Yes.*
☑ *Communication endpoints: 66 entries, see details.*
☑ *Communication with country: 8 entries, see details.*
☒ *SSL/TLS used? Yes.*
☑ *Domains accessed with http AND https:* `ads.mp.mydas.mobi`, `play.google.com`
☐ *Custom SSL/TLS trust manager implemented? No.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☐ *SSL/TLS using manual domain name verification? No.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected JavaScripts? Yes.*
☒ *Unprotected communication? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.android.vending.BILLING`, `com.miniclip.eightballpool.permission.C2D-MESSAGE`, `com.google.android.c2dm.permission.RECEIVE`
☑ *Overprivileged permissions:* `GET-TASKS`, `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

− *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Installed app list accessed? Yes.*
☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `UNKNOWN`
− *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 11 entries, see details.*
☑ *Advertisment-/tracking frameworks found: 12 entries, see details.*
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*
☑ *Sensor usage:* `WIFI-Based Location`, `Acceleration/Light`

☒	*Unprotected map queries? Yes.*
Runtime Security	
☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*
☑	*Executed component after Phone Reboot:* `com.miniclip.notifications.local.LocalNotificationBootReceiver`

3.1.2 Details

The following sections describe details about the test results of `8 Ball Pool` with version *3.6.2*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: unintended use of insecure HTTP protocol for transmissions of parameters to servers capable of HTTPS.
- Reasons for category privacy risks:
  - Extensive Advertisement/Tracking: App uses more than 10 advertisement and tracking providers.
  - Unprotected Access: Disclosure of location or web query data though unprotected communication with service providers.
  - App Listing: Usage of detected functionality to access list of installed apps poses a privacy risk for detected app type.
  - Code Execution At Boot: App executes code at phone boot without user interaction, which is suspicious for detected app type.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Blacklisted by policy

- Reasons for category violations of default policy:
  - Estimated overall app risk for the enterprise exceeds the security policy threshold due to detected risks and flaws exploitable by skilled attackers without the existence of additional supporting factors.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `amzn://apps/android?p=%s`
  - `bazaar://search?q=pname:`
  - `flurry://flurrycall?event=`
  - `flurry://flurrycall?event=adWillClose`
  - `http://ads.mp.mydas.mobi/appConfigServlet?apid=`
  - `http://ads.mp.mydas.mobi/pixel?id=`
  - `http://cvt.mydas.mobi/handleConversion?firstlaunch=`
  - `http://maps.google.com/maps/api/geocode/json?latlng=`
  - `http://market.android.com/support/bin/answer.py?answer=1050566&hl=%lang%&dl=%region%`
  - `http://play.google.com/store/apps/details?id=`
  - `http://play.google.com/store/apps/details?id=com.google.android.youtube`
  - `http://www.youtube.com/playlist?list=`
  - `http://www.youtube.com/watch?v=`
  - `https://ads.mp.mydas.mobi/appConfigServlet?apid=`
  - `https://play.google.com/store/apps/details?id=`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=`

- https://www.supersonicads.com/mobile/sdk5/log?method=contextIsNotActivity
- https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppKey
- https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppUserId
- https://www.supersonicads.com/mobile/sdk5/log?method=extraParametersToJson
- https://www.supersonicads.com/mobile/sdk5/log?method=htmlControllerDoesNotExistOnFileSystem
- https://www.supersonicads.com/mobile/sdk5/log?method=injectJavaScript
- https://www.supersonicads.com/mobile/sdk5/log?method=noProductType
- https://www.supersonicads.com/mobile/sdk5/log?method=setWebViewSettings
- https://www.supersonicads.com/mobile/sdk5/log?method=unregisterConnectionReceiverIllegal
- https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadBlank
- https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadWithPath
- https://www.supersonicads.com/mobile/sdk5/log?method=webviewPause
- https://www.supersonicads.com/mobile/sdk5/log?method=webviewResume
- https://www.tumblr.com/oauth/authorize?oauth_token=%s
- market://details?id=
- market://details?id=%s
- market://details?id=%s&referrer=%s
- market://details?id=com.google.android.gms.ads
- market://details?id=com.google.android.youtube

  - market://details?id=com.miniclip.eightballpool
  - market://search?q=pname:com.google

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: .facebook.com, a.applovin.com, adlog.flurry.com, ads.flurry.com, ads.mp.mydas.mobi, analytics.query.yahoo.com, androidads23.adcolony.com, androidsdk.ads.mp.mydas.mobi, api.facebook.com, api.sponsorpay.com, api.tumblr.com, api.vungle.com, app.adjust.com, cdn.flurry.com, cdn.millennialmedia.com, connect.tapjoy.com, content-js.tapjoy.com, cvt.mydas.mobi, d.applovin.com, data.flurry.com, engine.fyber.com, engine.sponsorpay.com, facebook.com, googleads.g.doubleclick.net, graph-video.%s, graph.%s, graph.%s.facebook.com, graph.facebook.com, iframe.sponsorpay.com, images.millennialmedia.com, img.youtube.com, impact.applifier.com, impact.staging.applifier.com, ingest.vungle.com, init.supersonicads.com, live.chartboost.com, live.hyprmx.com, m.facebook.com, maps.google.com, market.android.com, media.admob.com, millennialmedia.com, mobilelogs.ec2ssa.info, outcome.supersonicads.com, placements.tapjoy.com, play.google.com, plus.google.com, proton.flurry.com, rpc.tapjoy.com, rt.applovin.com, s.ssacdn.com, service.sponsorpay.com, services.dev.miniclippt.com, services.miniclippt.com, vid.applovin.com, video.fyber.com, ws.tapjoyads.com, www.amazon.com, www.facebook.com, www.google.com, www.googleapis.com, www.supersonicads.com, www.tumblr.com, www.ultraadserver.com, www.vungle.com, www.youtube.com
- App communicates with servers in 8 countries.
- Communication with country: Netherlands, Austria, Romania, United States, Ireland, United Kingdom, Germany, unknown
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.

- Mixed usage of HTTP and HTTPS: Protected and unprotected submission of parameters to the same domain. Indicates implementation flaw or weak communication protection.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://androidsdk.ads.mp.mydas.mobi/getAd.php5?`
  - `http://rt.applovin.com/pix`
  - `http://www.ultraadserver.com/api/rest/v1.1/uniqueusers?`
  - `http://ads.mp.mydas.mobi/pixel?id=`
  - `http://s.ssacdn.com/mobileSDKController/mobileController.html`
  - `http://www.supersonicads.com/timestamp.php`
  - `http://maps.google.com/maps/api/geocode/json?latlng=`
  - `http://www.youtube.com/embed/`
  - `http://www.amazon.com/gp/mas/get-appstore/android/ref=mas_mx_mba_iap_dl`
  - `http://services.miniclippt.com/newsfeed/newsfeed.php`
  - `http://ads.mp.mydas.mobi/appConfigServlet?apid=`
  - `http://www.tumblr.com/connect/login_success.html`
  - `http://play.google.com/store/apps/details?id=`
  - `http://millennialmedia.com/android/schema`
  - `http://outcome.supersonicads.com/mediation/`
  - `http://www.youtube.com/playlist?list=`
  - `http://mobilelogs.ec2ssa.info/log`

  - http://www.youtube.com/user/
  - http://play.google.com/store/apps/details
  - http://cvt.mydas.mobi/handleConversion?firstlaunch=
  - http://services.dev.miniclippt.com/newsfeed/newsfeed.php
  - http://api.vungle.com/api/v4/

- The app loads the following JavaScript files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://media.admob.com/mraid/v1/mraid_app_interstitial.js
  - http://cdn.millennialmedia.com/mmjs/v1.7/mm.js
  - http://googleads.g.doubleclick.net/mads/static/sdk/native/sdk-core-v40.js
  - http://media.admob.com/mraid/v1/mraid_app_banner.js
  - http://media.admob.com/mraid/v1/mraid_app_expanded_banner.js

- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - http://ads.mp.mydas.mobi/appConfigServlet?apid=
  - http://ads.mp.mydas.mobi/pixel?id=
  - http://cvt.mydas.mobi/handleConversion?firstlaunch=
  - http://maps.google.com/maps/api/geocode/json?latlng=
  - http://market.android.com/support/bin/answer.py?answer=1050566&hl=%lang%&dl=%region%
  - http://play.google.com/store/apps/details?id=
  - http://play.google.com/store/apps/details?id=com.google.android.youtube

- `http://www.youtube.com/playlist?list=`

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - GET-ACCOUNTS (Allows access to the list of accounts in the Accounts Service.)
  - VIBRATE (Allows access to the vibrator.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - RECEIVE-BOOT-COMPLETED (Allows an application to receive the android.content.Intent ACTION-BOOT-COMPLETED that is broadcast after the system finishes booting. If you don't request this permission, you will not receive the broadcast at that time. Though holding this permission does not have any security implications, it can have a negative impact on the user experience by increasing the amount of time it takes the system to start and allowing applications to have themselves running without the user being aware of them. As such, you must explicitly declare your use of this facility to make that visible to the user.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - GET-TASKS (Allows an application to get information about the currently or recently running tasks.)

  - INTERNET (Allows applications to open network sockets.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - ACCESS-COARSE-LOCATION (Allows an app to access approximate location derived from network location sources such as cell towers and Wi-Fi.)

- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- Device administration features not used.

- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build hardware`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `Adcolony`, `AppLovin`, `ChartBoost`, `Doubleclick`, `Flurry`, `Fyber`, `Google AdMob`, `Google Analytics`, `MillennialMedia`, `Supersonic`, `TapJoy`, `mopub`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.miniclip.eightballpool.EightBallPoolActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different sensors. This allows the application to track the user and/or determine the environment of the user.
- App contains URL(s) that indicate an unprotected HTTP access to map providers. The transmitted location query parameters to the following map providers are in this case accesible by third parties:
  - Google Maps

Runtime Security

- The application does not contain a scheduled alarm.

- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi/libImmEndpointWarpJ.so`
  - `ARM 32 bit: lib/armeabi/libgame.so`
- The Application has the permission to start automatically after booting the device. The application can execute code without userinteraction or prevention.

Test Performance

- Execution time of all tests: 0:01:39.757

3.2 Candy Crush Saga (Android)

3.2.1 Tests

The following Table 3.3 summarizes the results of the Android app `Candy Crush Saga` with version *1.82.1.1*.

Table 3.3: Overview of summarized test results for »Candy Crush Saga«

App risks for enterprise usage
☐ *Implementation flaws? No.*
☐ *Privacy risks? No.*
☒ *Security risks? Yes.*
Blacklisted by policy
☐ *Violations of default policy? No.*
Communication security

☒	*Client communication used? Yes.*
☑	*Communication endpoints: 17 entries, see details.*
☑	*Communication with country:* `Belgium`, `United States`, `Ireland`, `United Kingdom`, `unknown`
☒	*SSL/TLS used? Yes.*
☐	*Custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☐	*SSL/TLS using manual domain name verification? No.*
☒	*Unprotected HTML? Yes.*
☒	*Unprotected communication? Yes.*

Data security

☑	*Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage:* `com.king.cross.kingapp.provider.ACCESS`, `com.android.vending.BILLING`, `com.google.android.c2dm.permission.RECEIVE`, `com.king.candycrushsaga.permission.C2D-MESSAGE`
☑	*Overprivileged permissions:* `GET-ACCOUNTS`
☒	*Is application overprivileged? Yes.*
☒	*Application defines content provider? Yes.*
⊟	*Content provider accessible without permission: None.*
☒	*JavaScript to SDK API bridge usage? Yes.*
☐	*WiFi-Direct enabled? No.*

Input interface security

⊟	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*

Privacy

☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `UNKNOWN`
⊟	*Device administration policy entries: None.*
☑	*Accessed unique identifier(s):* `build model`, `build brand`, `Wifi-MAC address`, `MMC (Mobile Country Code)`, `unique Android ID`
☑	*Advertisment-/tracking frameworks found:* `Doubleclick`
☒	*App provides public accessible activities? Yes.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
⊟	*Sensor usage: None.*

Runtime Security	
☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*Contains native libraries: Yes.*

3.2.2 Details

The following sections describe details about the test results of `Candy Crush Saga` with version *1.82.1.1*.

App risks for enterprise usage

- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`
  - `https://pubads.g.doubleclick.net/gampad/ads?sz=640x480&impl=s&gdfp_req=1&env=vp&output=xml_vast3&unviewed_position_start=1&url=&description_url=&correlator=`
  - `market://details?id=com.facebook.orca`
  - `market://details?id=com.google.ads.interactivemedia.v3`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..

- Communication endpoints: `.facebook.com`, `accounts.google.com`, `facebook.com`, `graph-video.%s`, `graph.%s`, `imasdk.googleapis.com`, `login.live.com`, `login.yahoo.com`, `play.google.com`, `plus.google.com`, `pubads.g.doubleclick.net`, `twitter.com`, `www.amazon.com`, `www.facebook.com`, `www.googleapis.com`, `www.linkedin.com`, `www.paypal.com`
- App communicates with servers in 5 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://imasdk.googleapis.com/native/sdkloader/native_sdk_v3.html`
  - `http://www.amazon.com/gp/mas/get-appstore/android/ref=mas_mx_mba_iap_dl`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - GET-ACCOUNTS (Allows access to the list of accounts in the Accounts Service.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)

- The application requires the following permissions from the protection-level: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.
- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- Device administration features not used.

- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Indicators for usage of advertisement/tracking framework were found.
- The application contains components (Activities) which are exported. This means these parts of the application are accessible or executable by other applications. An external app can write or read information/data to or from this app. Additionally components of this application can be executed. Following Activities are exported:
  - `com.king.core.VideoPlayerActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- No sensor usage Indicators found.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libcandycrushsaga.so`

Test Performance

- Execution time of all tests: 0:00:32.682

3.3 Clash of Clans (Android)

3.3.1 Tests

The following Table 3.4 summarizes the results of the Android app `Clash of Clans` with version *8.332.16*.

Table 3.4: Overview of summarized test results for »Clash of Clans«

App risks for enterprise usage	
☐	*Implementation flaws? No.*
☐	*Privacy risks? No.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 16 entries, see details.*
☑	*Communication with country:* `United States`, `Ireland`
☒	*SSL/TLS used? Yes.*
☐	*Custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☐	*SSL/TLS using manual domain name verification? No.*
☒	*Unprotected communication? Yes.*
Data security	
☑	*Cryptographic Primitives:* `"AES/CBC/NoPadding"`, `"AES/CBC/PKCS5Padding"`, `"AES/ECB/PKCS5Padding"`
☒	*Cryptographic keys found? Yes.*
☒	*Constant initialization vectors found? Yes.*
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage:* `com.supercell.clashofclans.permission.C2D-MESSAGE`, `com.android.vending.BILLING`, `com.google.android.c2dm.permission.RECEIVE`
☑	*Overprivileged permissions:* `ACCESS-WIFI-STATE`, `READ-EXTERNAL-STORAGE`
☒	*Is application overprivileged? Yes.*
☒	*JavaScript to SDK API bridge usage? Yes.*

☐	*WiFi-Direct enabled? No.*
Input interface security	
⊟	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*
Privacy	
☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `HIGH`
⊟	*Device administration policy entries: None.*
☑	*Accessed unique identifier(s): 9 entries, see details.*
☑	*Advertisment-/tracking frameworks found:* `Doubleclick`, `OpenUDID`
☐	*App provides public accessible activities? No.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Location (inactive)`
Runtime Security	
☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*Contains native libraries: Yes.*

3.3.2 Details

The following sections describe details about the test results of `Clash of Clans` with version *8.332.16*.

App risks for enterprise usage

- Reasons for category security risks:
  - Crypto: Embedded static encryption key found, which can be extracted by attackers to revert the encryption or fake the signature of the content it is used for.
  - Crypto: Constant initialization vector detected. This should be avoided, as it allows an attacker to infer relationships between seg-

ments of encrypted messages if encrypted with the same key and initialization vector.

- Crypto: Overall quality of cryptographic implementation aspects is rated poor and should be inspected in detail.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`
  - `market://details?id=`
  - `market://details?id=com.facebook.orca`
  - `market://details?id=com.google.android.gms.ads`
  - `market://play.google.com/store/apps/details?id=`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: `.facebook.com`, `accounts.google.com`, `app-measurement.com`, `csi.gstatic.com`, `facebook.com`, `googleads.g.doubleclick.net`, `graph-video.%s`, `graph.%s`, `play.google.com`, `plus.google.com`, `ssl.google-analytics.com`, `www.facebook.com`, `www.google-analytics.com`, `www.google.com`, `www.googleapis.com`, `www.googletagmanager.com`
- App communicates with servers in 2 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.

  - `http://play.google.com/store/apps/details?id=com.facebook.orca`

Data security

- ECB mode usage identified. This mode has the disadvantage, that identical plaintext blocks are encrypted into identical ciphertext blocks. Therefore it does not hide patterns well and this mode is not recommended for use in cryptographic protocols at all.
- It is considered as a bad practice to use hard-coded cryptographic keys in the application. The following hard-coded cryptographic keys were found:
  - "heF9BATUfWuISyO8"
  - "sdk"
- Use of constant initialization vectors is a bad practice. The following initialization vectors were found:
  - "fldsjfodasjifudslfjdsaofshaufihadsf"
  - "heF9BATUfWuISyO8"
- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)

  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - CHANGE-WIFI-STATE (Allows applications to change Wi-Fi connectivity state.)

- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.

- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build hardware`, `build display`, `build fingerprint`, `build brand`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.supercell.clashofclans.GameApp`
- In this application the allow backup option is enabled. This means the application and all application data will be included when performing a device backup. In case the application contains sensitive information these can be extracted from the backup archive or cloned onto other devices.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.

- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libg.so`
  - `x86 32bit: lib/x86/libg.so`

Test Performance

- Execution time of all tests: 0:00:50.918

3.4 Cosmic Challenge (Android)

3.4.1 Tests

The following Table 3.5 summarizes the results of the Android app `Cosmic Challenge` with version *2.1*.

Table 3.5: Overview of summarized test results for »Cosmic Challenge«

App risks for enterprise usage	
☐	*Implementation flaws? No.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 17 entries, see details.*
☑	*Communication with country:* `United States, Ireland`
☒	*SSL/TLS used? Yes.*
☐	*Custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☒	*SSL/TLS using manual domain name verification? Yes.*
☒	*Unprotected HTML? Yes.*
☒	*Unprotected communication? Yes.*
Data security	
☑	*Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒	*Constant initialization vectors found? Yes.*
☑	*Key derivation iteration count:* `1024`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*

☑ *Userdefined permission usage:* `com.gamedonia.sdk.permission.C2D-MESSAGE`, `com.android.vending.BILLING`, `com.android.vending.CHECK-LICENSE`, `com.google.android.c2dm.permission.RECEIVE`
☑ *Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *Application defines content provider? Yes.*
− *Content provider accessible without permission: None.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

− *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Installed app list accessed? Yes.*
☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `HIGH`
− *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 12 entries, see details.*
☑ *Advertisment-/tracking frameworks found:* `Adcolony`, `ChartBoost`, `OpenUDID`, `TapJoy`
☒ *App provides public accessible activities? Yes.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*
☑ *Sensor usage:* `Camera`, `Location (inactive)`, `Acceleration/Light`

Runtime Security

☒ *Scheduled Alarm Manager registered? Yes.*
☑ *Alarm repeating types:* `RTC-WAKEUP`
☒ *Alarm intervals dynamically? Yes.*
☐ *Alarm Manager initialized dynamically? No.*
☒ *Dynamically loaded code at runtime? Yes.*
☑ *Dynamically loaded code at runtime type(s):* `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐ *Allow app debugging Flag? No.*
☐ *Allow autoexecute after Phone Reboot? No.*
☒ *Contains native libraries: Yes.*

3.4.2 Details

The following sections describe details about the test results of `Cosmic Challenge` with version *2.1*.

App risks for enterprise usage

- Reasons for category privacy risks:
  - App Listing: Usage of detected functionality to access list of installed apps poses a privacy risk for detected app type.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - Crypto: Constant initialization vector detected. This should be avoided, as it allows an attacker to infer relationships between segments of encrypted messages if encrypted with the same key and initialization vector.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://play.google.com/store/apps/details?id=`
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`
  - `market://details?id=`
  - `market://details?id=com.facebook.orca`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: `.facebook.com`, `androidads23.adcolony.com`, `connect.tapjoy.com`, `facebook.com`, `graph-video.%s`, `graph.%s`, `impact.applifier.com`, `impact.staging.applifier.com`, `live.chartboost.com`, `market.android.com`, `placements.tapjoy.com`, `play.google.com`, `plus.google.com`, `rpc.tapjoy.com`, `ws.tapjoyads.com`, `www.amazon.com`, `www.googleapis.com`

- App communicates with servers in 2 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- Correct verification of the corresponding client hostname is important for SSL/TLS security. The app changes the secure default hostname verification by the following:
  - Interface HostnameVerifier is implemented or extended.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://play.google.com/store/apps/details?id=`
  - `http://www.amazon.com/gp/mas/get-appstore/android/ref=mas_mx_mba_iap_dl`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://play.google.com/store/apps/details?id=`
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`

Data security

- Use of constant initialization vectors is a bad practice. The following initialization vectors were found:
  - 16,74,71,-80,32,101,-47,72,117,-14,0,-29,70,65,-12,74
- Key derivation function used in the app with an amount of 1024 iterations is considered secure.
- The application requires the following permissions from the protectionlevel: NORMAL
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)

  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)

- The application requires the following permissions from the protection-level: DANGEROUS
  - CAMERA (Required to be able to access the camera device. This will automatically enforce the uses-feature manifest element for all camera features. If you do not require all camera features or can properly operate if a camera is not available, then you must modify your manifest as appropriate in order to install on devices that don't support all camera features.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - USE-CREDENTIALS (Allows an application to request authtokens from the AccountManager.)
  - INTERNET (Allows applications to open network sockets.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.

- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.

- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains components (Activities) which are exported. This means these parts of the application are accessible or executable by other applications. An external app can write or read information/data to or from this app. Additionally components of this application can be executed. Following Activities are exported:
  - `com.facebook.unity.FBUnityDeepLinkingActivity`
  - `com.facebook.unity.FBUnityAppLinkActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application contains a registered scheduled alarm. With such an alarm the application repeats the execution of the registered task for example every 10 hours. The following classes register scheduled tasks:
  - `com.Kohda.CosmicChallenge.UnityNotificationManager`
- The scheduled task gets repeated in the following intervals:
  - Dynamic interval(s)

- The alarm manager has been initialized properly.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libgpg.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`
  - `ARM 32 bit: lib/armeabi-v7a/libunity.so`
  - `x86 32bit: lib/x86/libgpg.so`
  - `x86 32bit: lib/x86/libmain.so`
  - `x86 32bit: lib/x86/libmono.so`
  - `x86 32bit: lib/x86/libunity.so`

Test Performance

- Execution time of all tests: 0:01:11.779

3.5 Criminal Case (Android)

3.5.1 Tests

The following Table 3.6 summarizes the results of the Android app `Criminal Case` with version *2.12*.

Table 3.6: Overview of summarized test results for »Criminal Case«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*

Blacklisted by policy

☒ *Violations of default policy? Yes.*

Communication security

☒ *Client communication used? Yes.*
☑ *Communication endpoints: 48 entries, see details.*
☑ *Communication with country:* `Netherlands`, `United States`, `Ireland`, `United Kingdom`, `unknown`
☒ *SSL/TLS used? Yes.*
☒ *Custom SSL/TLS trust manager implemented? Yes.*
☒ *Faulty custom SSL/TLS trust manager implemented? Yes.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☒ *SSL/TLS using manual domain name verification? Yes.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected communication? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.prettysimple.criminalcaseandroid.permission.C2D-MESSAGE`, `com.android.vending.BILLING`, `com.google.android.c2dm.permission.RECEIVE`
☑ *Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *Application defines content provider? Yes.*
− *Content provider accessible without permission: None.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

− *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Installed app list accessed? Yes.*
☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `HIGH`
− *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 12 entries, see details.*
☑ *Advertisment-/tracking frameworks found: 8 entries, see details.*
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*

☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Location (inactive), Acceleration/ Light`

Runtime Security	
☒	*Scheduled Alarm Manager registered? Yes.*
☑	*Alarm repeating types:* `RTC`
☒	*Alarm intervals dynamically? Yes.*
☐	*Alarm Manager initialized dynamically? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system. DexClassLoader(...), ClassLoader.loadClass(...), loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.5.2 Details

The following sections describe details about the test results of `Criminal Case` with version *2.12*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
- Reasons for category privacy risks:
  - Advertisement/Tracking: App uses more than 5 advertisement and tracking providers.
  - App Listing: Usage of detected functionality to access list of installed apps poses a privacy risk for detected app type.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Blacklisted by policy

- Reasons for category violations of default policy:
  - Estimated overall app risk for the enterprise exceeds the security policy threshold due to detected risks and flaws exploitable by skilled attackers without the existence of additional supporting factors.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://mobile-api.geo.kontagent.net/fb-install/%s/activities/?event=MOBILE_APP_INSTALL&attribution=%s`
  - `http://play.google.com/store/apps/details?id=`
  - `https://events.appsflyer.com/api/v2.3/androidevent?buildnumber=1.17&app_id=`
  - `https://t.appsflyer.com/api/v2.3/androidevent?buildnumber=1.17&app_id=`
  - `https://track.appsflyer.com/api/v2.3/uninsall?buildnumber=1.17`
  - `https://www.googleapis.com/games/v1management/achievements/reset?access_token=`
  - `https://www.googleapis.com/games/v1management/scores/reset?access_token=`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=contextIsNotActivity`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppKey`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppUserId`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=extraParametersToJson`

  - https://www.supersonicads.com/mobile/sdk5/log?method=htmlControllerDoesNotExistOnFileSystem
  - https://www.supersonicads.com/mobile/sdk5/log?method=injectJavaScript
  - https://www.supersonicads.com/mobile/sdk5/log?method=noProductType
  - https://www.supersonicads.com/mobile/sdk5/log?method=setWebViewSettings
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadBlank
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadWithPath
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewPause
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewResume
  - market://details?id=
  - market://details?id=%s
  - market://details?id=com.google.android.gms.ads

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: .facebook.com, a.applovin.com, accounts.google.com, androidads23.adcolony.com, api.appsflyer.com, api.facebook.com, api.geo.kontagent.net, app-measurement.com, connect.tapjoy.com, content-js.tapjoy.com, csi.gstatic.com, d.applovin.com, e.crashlytics.com, events.appsflyer.com, facebook.com, googleads.g.doubleclick.net, graph-video.%s, graph.%s, graph.%s.facebook.com, graph.facebook.com, impact.applifier.com, impact.staging.applifier.com, init.supersonicads.com, login.live.com, login.yahoo.com, m.facebook.com, mobile-api.geo.kontagent.net, mobilelogs.supersonic.com, outcome.supersonicads.com, placements.tapjoy.com, play.google.com, plus.google.com, rpc.tapjoy.com, rt.applovin.com, settings.crashlytics.com, t.appsflyer.com, track.

appsflyer.com, twitter.com, ua.supersonicads.com, vid.applovin.com, ws.tapjoyads.com, www.%s.facebook.com, www.facebook.com, www.google.com, www.googleapis.com, www.linkedin.com, www.paypal.com, www.supersonicads.com

- App communicates with servers in 5 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - Lcom/facebook/ads/internal/util/g$1.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- Correct verification of the corresponding client hostname is important for SSL/TLS security. The app changes the secure default hostname verification by the following:
  - Interface HostnameVerifier is implemented or extended.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://play.google.com/store/apps/details?id=
  - http://rt.applovin.com/pix
  - http://api.geo.kontagent.net/api/v0/ping/
  - http://mobile-api.geo.kontagent.net/fb-install/%s/activities/?event=MOBILE_APP_INSTALL&attribution=%s
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - http://mobile-api.geo.kontagent.net/fb-install/%s/activities/?event=MOBILE_APP_INSTALL&attribution=%s

  - `http://play.google.com/store/apps/details?id=`

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - GET-ACCOUNTS (Allows access to the list of accounts in the Accounts Service.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - VIBRATE (Allows access to the vibrator.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.

- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.
- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.

- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `Adcolony`, `AppLovin`, `AppsFlyer`, `Crashlytics`, `Doubleclick`, `Supersonic`, `TapJoy`, `inMobi ADs`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.prettysimple.game.CriminalCaseLauncher`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application contains a registered scheduled alarm. With such an alarm the application repeats the execution of the registered task for example every 10 hours. The following classes register scheduled tasks:
  - `com.prettysimple.notification.a`
- The scheduled task gets repeated in the following intervals:
  - Dynamic interval(s)

- The alarm manager has been initialized properly.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libcrashlytics-envelope.so`
  - `ARM 32 bit: lib/armeabi-v7a/libcrashlytics.so`
  - `ARM 32 bit: lib/armeabi-v7a/libcriminalcase.so`

Test Performance

- Execution time of all tests: 0:00:53.213

3.6 Duck Hunting (Android)

3.6.1 Tests

The following Table 3.7 summarizes the results of the Android app `Duck Hunting` with version *1.2*.

Table 3.7: Overview of summarized test results for »Duck Hunting«

App risks for enterprise usage
☒ *Implementation flaws? Yes.*
☐ *Privacy risks? No.*
☒ *Security risks? Yes.*
Blacklisted by policy
☐ *Violations of default policy? No.*

Communication security

- ☒ *Client communication used? Yes.*
- ☑ *Communication endpoints: 21 entries, see details.*
- ☑ *Communication with country:* `United States`, `Ireland`, `United Kingdom`, `unknown`
- ☒ *SSL/TLS used? Yes.*
- ☒ *Custom SSL/TLS trust manager implemented? Yes.*
- ☒ *Faulty custom SSL/TLS trust manager implemented? Yes.*
- ☒ *SSL/TLS using custom error handling? Yes.*
- ☐ *SSL/TLS using faulty custom error handling? No.*
- ☐ *SSL/TLS using manual domain name verification? No.*
- ☒ *Unprotected HTML? Yes.*

Data security

- ☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
- ☒ *Application needs normal permissions? Yes.*
- ☒ *Application needs dangerous permissions? Yes.*
- ☑ *Userdefined permission usage:* `com.android.vending.BILLING`
- ☑ *Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
- ☒ *Is application overprivileged? Yes.*
- ☒ *JavaScript to SDK API bridge usage? Yes.*
- ☐ *WiFi-Direct enabled? No.*

Input interface security

- ⊟ *App can handle documents of mimeType: None.*
- ☐ *Screenshot protection used? No.*
- ☐ *Tap Jacking Protection used? No.*

Privacy

- ☒ *Obfuscation used? Yes.*
- ☑ *Obfuscation level is:* `HIGH`
- ⊟ *Device administration policy entries: None.*
- ☑ *Accessed unique identifier(s): 9 entries, see details.*
- ☑ *Advertisment-/tracking frameworks found:* `ChartBoost`, `Doubleclick`, `Heyzap`
- ☐ *App provides public accessible activities? No.*
- ☒ *Backup of app is allowed? Yes.*
- ☒ *Log Statement Enabled? Yes.*
- ☐ *Permission to access address book? No.*
- ☑ *Sensor usage:* `Camera (inactive)`, `Location (inactive)`

Runtime Security

- ☐ *Scheduled Alarm Manager registered? No.*
- ☒ *Dynamically loaded code at runtime? Yes.*

☑ *Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐ *Allow app debugging Flag? No.*
☐ *Allow autoexecute after Phone Reboot? No.*
☒ *App uses outdated signature key? Yes.*
☒ *Contains native libraries: Yes.*

3.6.2 Details

The following sections describe details about the test results of `Duck Hunting` with version *1.2*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `market://details?id=%s&referrer=%s`
  - `market://details?id=com.google.android.gms.ads`
  - `market://details?id=com.heyzap.android`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..

- Communication endpoints: accounts.google.com, ads.heyzap.com, app-measurement.com, csi.gstatic.com, googleads.g.doubleclick.net, live.chartboost.com, login.live.com, login.yahoo.com, market.android.com, med.heyzap.com, plus.google.com, ssl.google-analytics.com, twitter.com, www.facebook.com, www.google-analytics.com, www.google.com, www.googleapis.com, www.googletagmanager.com, www.linkedin.com, www.paypal.com, x.heyzap.com
- App communicates with servers in 4 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - Lcom/heyzap/http/MySSLSocketFactory$1.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://ads.heyzap.com/in_game_api/ads

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)

  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)

- The application requires the following permissions from the protection-level: DANGEROUS
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - INTERNET (Allows applications to open network sockets.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.

- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build display`, `build fingerprint`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.unity3d.player.UnityPlayerNativeActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.

- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`
  - `ARM 32 bit: lib/armeabi-v7a/libunity.so`
  - `x86 32bit: lib/x86/libmain.so`
  - `x86 32bit: lib/x86/libmono.so`
  - `x86 32bit: lib/x86/libunity.so`

Test Performance

- Execution time of all tests: 0:01:14.525

3.7 Hill Climb Racing (Android)

3.7.1 Tests

The following Table 3.8 summarizes the results of the Android app `Hill Climb Racing` with version *1.30.0*.

Table 3.8: Overview of summarized test results for »Hill Climb Racing«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☒	*Violations of default policy? Yes.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 65 entries, see details.*
☑	*Communication with country: 6 entries, see details.*

☒ *SSL/TLS used? Yes.*
☑ *Domains accessed with http AND https:* `play.google.com`
☒ *Custom SSL/TLS trust manager implemented? Yes.*
☒ *Faulty custom SSL/TLS trust manager implemented? Yes.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☒ *SSL/TLS using manual domain name verification? Yes.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected JavaScripts? Yes.*
☒ *Unprotected communication? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/NoPadding", "AES/CBC/PKCS5Padding"`
☒ *Cryptographic keys found? Yes.*
☒ *Constant initialization vectors found? Yes.*
☒ *Cryptographic seed values found? Yes.*
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.android.vending.BILLING, com.google.android.gms.permission.ACTIVITY-RECOGNITION`
☑ *Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

⊟ *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Installed app list accessed? Yes.*
☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `UNKNOWN`
⊟ *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 11 entries, see details.*
☑ *Advertisment-/tracking frameworks found: 10 entries, see details.*
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*
☑ *Sensor usage:* `Location (inactive), Acceleration/Light`

Runtime Security

☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*Contains native libraries: Yes.*

3.7.2 Details

The following sections describe details about the test results of `Hill Climb Racing` with version *1.30.0*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
  - Possible flaw: unintended use of insecure HTTP protocol for transmissions of parameters to servers capable of HTTPS.
- Reasons for category privacy risks:
  - Advertisement/Tracking: App uses more than 5 advertisement and tracking providers.
  - App Listing: Usage of detected functionality to access list of installed apps poses a privacy risk for detected app type.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - Crypto: Embedded static encryption key found, which can be extracted by attackers to revert the encryption or fake the signature of the content it is used for.
  - Crypto: Constant initialization vector detected. This should be avoided, as it allows an attacker to infer relationships between segments of encrypted messages if encrypted with the same key and initialization vector.

  - Crypto: Constant seed detected. Using a static seed may completely replace the cryptographically strong default seed causing the random number generator to return a predictable sequence of numbers unfit for secure use.
  - Crypto: Overall quality of cryptographic implementation aspects is rated poor and should be inspected in detail.

Blacklisted by policy

- Reasons for category violations of default policy:
  - Estimated overall app risk for the enterprise exceeds the security policy threshold due to detected risks and flaws exploitable by skilled attackers without the existence of additional supporting factors.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `amzn://apps/android?p=`
  - `http://adelh.smaato.com/lg.php?bannerid=57708&amp.campaignid=3692&amp.zoneid=0&amp.loc=1&amp.referer=http%3A%2F%2Fadelh.smaato.com%2Faxmlrpc.php%3Fsize%3Dxlarge%26img%3Dtrue%26carrier%3DT-Mobile%2B%2528WiFi%252FWLAN%2529&amp.cb=6af462c795&amp.r_id=20b1af536e51079d611b279e5e2e5a7e&amp.r_ts=ln8ydk`
  - `http://adelh.smaato.com/lg.php?bannerid=60196&amp.campaignid=3692&amp.zoneid=0&amp.loc=1&amp.referer=http%3A%2F%2Fadelh.smaato.com%2Faxmlrpc.php%3Fsize%3Dxlarge%26img%3Dtrue%26carrier%3DT-Mobile%2B%2528WiFi%252FWLAN%2529&amp.cb=8a7475eb48&amp.r_id=c161faf29bc4cd1b964223995850ece4&amp.r_ts=ln8y6l`
  - `http://ads.fingersoft.net/mobile/apprelease?appid=`
  - `http://ads.mp.mydas.mobi/pixel?id=`

- http://api.crispwireless.com/adRequest/control/ad.gif?sitekey=DEFAULT&amp.partnerkey=afa1a1efc4977cc8bc83a8fe6a952a39&amp.zid=1418&amp.publisherid=374
- http://api.crispwireless.com/adRequest/control/noscript.gif?sitekey=DEFAULT&amp.partnerkey=afa1a1efc4977cc8bc83a8fe6a952a39&amp.zid=1418&amp.publisherid=374
- http://play.google.com/store/apps/details?id=com.google.android.youtube
- https://dl.dropboxusercontent.com/s/uushlgxxnf77zml/mraid_test_video_page.html?token_hash=AAF2-x1x1estOcg9hbncFPpJ4Q0MMkK47QbtOtFV0_5esQ&dl=1
- https://play.google.com/store/apps/details?id=
- https://www.supersonicads.com/mobile/sdk5/log?method=
- https://www.supersonicads.com/mobile/sdk5/log?method=contextIsNotActivity
- https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppKey
- https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppUserId
- https://www.supersonicads.com/mobile/sdk5/log?method=extraParametersToJson
- https://www.supersonicads.com/mobile/sdk5/log?method=htmlControllerDoesNotExistOnFileSystem
- https://www.supersonicads.com/mobile/sdk5/log?method=injectJavaScript
- https://www.supersonicads.com/mobile/sdk5/log?method=noProductType
- https://www.supersonicads.com/mobile/sdk5/log?method=setWebViewSettings
- https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadBlank

  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadWithPath
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewPause
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewResume
  - https://www.youtube.com/playlist?list=
  - https://www.youtube.com/watch?v=
  - market://details?id=
  - market://details?id=com.google.android.gms.ads
  - market://details?id=com.google.android.youtube
  - market://search?q=pub:Fingersoft

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: ad.samsungadhub.com, adelh.smaato.com, ads.fingersoft.net, ads.mp.mydas.mobi, ads3.fingersoft.net, analytics.mopub.com, androidads21.adcolony.com, api.appsflyer.com, api.crispwireless.com, api.vungle.com, app.getsentry.com, appclick.co, avr.smaato.net, cdn1.crispadvertising.com, cdn2.inner-active.mobi, csi.gstatic.com, data.flurry.com, dl.dropboxusercontent.com, events.appsflyer.com, fingersoft.net, googleads.g.doubleclick.net, i.w.inmobi.com, i.xx.openx.com, images.millennialmedia.com, img.youtube.com, impact.applifier.com, impact.staging.applifier.com, ingest.vungle.com, init.supersonicads.com, inmobisdk-a.akamaihd.net, internal.teamfreeze.com, internal2.teamfreeze.com, maps.google.com, market.android.com, marketplace-android-b56.hyprmx.com, millennialmedia.com, mobilelogs.supersonic.com, nativex-sdk-testapi.appspot.com, outcome.supersonicads.com, p25-elb-stg-mch-ad-test-681583878.us-west-1.elb.amazonaws.com, play.google.com, plus.google.com, register.appsflyer.com, relay.mobile.toboads.com, s3-eu-west-1.amazonaws.com, sdk-services.appsflyer.com, smaato-

android-sdk.s3.amazonaws.com, soma-assets.smaato.net, soma.smaato.net, staging-fsad.trafficmanager.net, stats.appsflyer.com, supersonic.ironbeast.io, t.appsflyer.com, touch.facebook.com, twitter.com, ua.supersonicads.com, wv.inner-active.mobi, www.google.com, www.googleapis.com, www.mopub.com, www.samsungapps.com, www.smaato.com, www.supersonicads.com, www.vungle.com, www.youtube.com

- App communicates with servers in 6 countries.
- Communication with country: Netherlands, Austria, United States, Ireland, Germany, unknown
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Mixed usage of HTTP and HTTPS: Protected and unprotected submission of parameters to the same domain. Indicates implementation flaw or weak communication protection.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - Lcom/nativex/volleytoolbox/IgnoreCertTrustManager.
  - Lcom/flurry/sdk/ej.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- Correct verification of the corresponding client hostname is important for SSL/TLS security. The app changes the secure default hostname verification by the following:
  - Interface HostnameVerifier is implemented or extended.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://marketplace-android-b56.hyprmx.com/viewings/
  - http://ads.mp.mydas.mobi/pixel?id=

  - http://plus.google.com/108100831193761361624/posts
  - http://maps.google.com/maps/api/geocode/json?
  - http://ads.fingersoft.net/mobile/apprelease?appid=
  - http://marketplace-android-b56.hyprmx.com/trackings/
  - http://ad.samsungadhub.com/api/android/1.0/request
  - http://internal.teamfreeze.com/internal.mobile.com/Richmedia/Preview/RichmediaTemplatePreview.aspx
  - http://marketplace-android-b56.hyprmx.com/embedded_offers/offers_available_json
  - http://millennialmedia.com/android/schema
  - http://internal2.teamfreeze.com/internal.mobile.com/Richmedia/Preview/RichmediaTemplatePreview.aspx
  - http://cdn2.inner-active.mobi/ia-android-sdk/
  - http://fingersoft.net/eula/
  - http://marketplace-android-b56.hyprmx.com/web_traffic_url_visits/create
  - http://avr.smaato.net/report
  - http://fingersoft.net/privacy/
  - http://play.google.com/store/apps/details
  - http://market.android.com/details
  - http://api.vungle.com/api/v4/
  - http://soma.smaato.net/oapi/reqAd.jsp?
  - http://twitter.com/#!/Fingersoft

- The app loads the following JavaScript files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://50.18.124.80/ormmabridge.and.js

  - http://soma-assets.smaato.net/js/ormma.js
  - http://cdn1.crispadvertising.com/afw/2.1/framework/client/adrequest.js
  - http://50.18.124.80/ormma.and.js
  - http://ad.samsungadhub.com/api/web/1.0/mraid.js
  - http://soma-assets.smaato.net/js/ormma_bridge.js

- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - http://adelh.smaato.com/lg.php?bannerid=57708&amp.campaignid=3692&amp.zoneid=0&amp.loc=1&amp.referer=http%3A%2F%2Fadelh.smaato.com%2Faxmlrpc.php%3Fsize%3Dxlarge%26img%3Dtrue%26carrier%3DT-Mobile%2B%2528WiFi%252FWLAN%2529&amp.cb=6af462c795&amp.r_id=20b1af536e51079d611b279e5e2e5a7e&amp.r_ts=ln8ydk
  - http://adelh.smaato.com/lg.php?bannerid=60196&amp.campaignid=3692&amp.zoneid=0&amp.loc=1&amp.referer=http%3A%2F%2Fadelh.smaato.com%2Faxmlrpc.php%3Fsize%3Dxlarge%26img%3Dtrue%26carrier%3DT-Mobile%2B%2528WiFi%252FWLAN%2529&amp.cb=8a7475eb48&amp.r_id=c161faf29bc4cd1b964223995850ece4&amp.r_ts=ln8y6l
  - http://ads.fingersoft.net/mobile/apprelease?appid=
  - http://ads.mp.mydas.mobi/pixel?id=
  - http://api.crispwireless.com/adRequest/control/ad.gif?sitekey=DEFAULT&amp.partnerkey=afa1a1efc4977cc8bc83a8fe6a952a39&amp.zid=1418&amp.publisherid=374
  - http://api.crispwireless.com/adRequest/control/noscript.gif?sitekey=DEFAULT&amp.partnerkey=afa1a1efc4977cc8bc83a8fe6a952a39&amp.zid=1418&amp.publisherid=374
  - http://play.google.com/store/apps/details?id=com.google.android.youtube

Data security

- It is considered as a bad practice to use hard-coded cryptographic keys in the application. The following hard-coded cryptographic keys were found:
  - "Xke-jKFBeI9gfc4V"
  - "heF9BATUfWuISyO8"
- Use of constant initialization vectors is a bad practice. The following initialization vectors were found:
  - "FJNkd+T9"
  - "heF9BATUfWuISyO8"
- Constant seeds can return constant keys, making application highly insecure. The following cryptographic seeds were found:
  - "tiJ8e+8Fb.21xd.5"
- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - INTERNET (Allows applications to open network sockets.)

- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.

- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `Adcolony`, `Doubleclick`, `Flurry`, `MillennialMedia`, `Samsung AdHub`, `Smaato`, `Supersonic`, `inMobi ADs`, `inneractive`, `mopub`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.fingersoft.game.MainActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be included when performing a device backup. In case the application contains sensitive information these can be extracted from the backup archive or cloned onto other devices.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.

- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi/libgame.so`
  - `ARM 32 bit: lib/armeabi/libImmEndpointWarpJ.so`
  - `ARM 32 bit: lib/armeabi-v7a/libgame.so`
  - `x86 32bit: lib/x86/libgame.so`

Test Performance

- Execution time of all tests: 0:01:24.179

3.8 Luftangriff des Helikopters (Android)

3.8.1 Tests

The following Table 3.9 summarizes the results of the Android app `Luftangriff des Helikopters` with version *1.0.3*.

Table 3.9: Overview of summarized test results for »Luftangriff des Helikopters«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☒	*Violations of default policy? Yes.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 21 entries, see details.*
☑	*Communication with country:* `United States`, `Ireland`
☒	*SSL/TLS used? Yes.*
☒	*Custom SSL/TLS trust manager implemented? Yes.*
☒	*Faulty custom SSL/TLS trust manager implemented? Yes.*
☐	*SSL/TLS using custom error handling? No.*

☐ *SSL/TLS using manual domain name verification? No.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected JavaScripts? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5Padding", "AES/ECB/PKCS7Padding", "DES/ECB/PKCS5Padding"`
☒ *Constant initialization vectors found? Yes.*
☑ *Key derivation iteration count:* `1024`
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.android.vending.BILLING`
☑ *Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☒ 🛈 *JavaScript to SDK API bridge vulnerability? Yes. (see details)*
☐ *WiFi-Direct enabled? No.*

Input interface security

− *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Installed app list accessed? Yes.*
☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `HIGH`
− *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 10 entries, see details.*
☑ *Advertisment-/tracking frameworks found:* `Doubleclick`, `Flurry`, `Google AdMob`, `Google Analytics`, `TapJoy`
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*
☑ *Sensor usage:* `Camera (inactive)`, `Location (inactive)`, `Acceleration/Light`

Runtime Security

☒ *Scheduled Alarm Manager registered? Yes.*
☑ *Alarm repeating types:* `RTC-WAKEUP`
☐ *Alarm intervals dynamically? No.*
☐ *Alarm Manager initialized dynamically? No.*
☒ *Dynamically loaded code at runtime? Yes.*

☑ *Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐ *Allow app debugging Flag? No.*
☐ *Allow autoexecute after Phone Reboot? No.*
☒ *App uses outdated signature key? Yes.*
☒ *Contains native libraries: Yes.*

3.8.2 Details

The following sections describe details about the test results of `Luftangriff des Helikopters` with version *1.0.3*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
  - Possible flaw: An application calling Android API methods by JavaScript and defining targetSdk version less than 17 could be vulnerable to remote code injection.
- Reasons for category privacy risks:
  - App Listing: Usage of detected functionality to access list of installed apps poses a privacy risk for detected app type.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - Crypto: Constant initialization vector detected. This should be avoided, as it allows an attacker to infer relationships between segments of encrypted messages if encrypted with the same key and initialization vector.

Blacklisted by policy

- Reasons for category violations of default policy:

  - Estimated overall app risk for the enterprise exceeds the security policy threshold due to detected risks and flaws exploitable by skilled attackers without the existence of additional supporting factors.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `bazaar://search?q=pname:`
  - `https://market.android.com/details?id=`
  - `market://details?id=`
  - `market://details?id=com.google.android.gms.ads`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: `ad.flurry.com`, `connect.tapjoy.com`, `content-js.tapjoy.com`, `data.flurry.com`, `data2.doodlemobile.com`, `ec2-184-73-77-17.compute-1.amazonaws.com`, `events.tapjoy.com`, `f2.doodlemobile.com`, `featured.perfectionholic.com`, `googleads.g.doubleclick.net`, `market.android.com`, `media.admob.com`, `newfeatureview.perfectionholic.com`, `play.google.com`, `plus.google.com`, `s3.amazonaws.com`, `tech.tapjoy.com`, `ws.tapjoyads.com`, `www.google.com`, `www.googleapis.com`, `www.googletagmanager.com`
- App communicates with servers in 2 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - `Lcom/flurry/android/n.`
- App uses the secure default error handling for SSL/TLS client communication. Error-prone modifications can be ruled out.

- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://newfeatureview.perfectionholic.com/featureview/getfeatureview/`
  - `http://f2.doodlemobile.com/feature_server/fullScreen/get.php`
  - `http://play.google.com/store/apps/`
  - `http://featured.perfectionholic.com:8080/moregames/index_app.html`
  - `http://newfeatureview.perfectionholic.com/featureview/gettime/`
  - `http://ec2-184-73-77-17.compute-1.amazonaws.com/featureview/gettime/`
  - `http://featured.perfectionholic.com:8080/feature_appserver/recommands`
  - `http://f2.doodlemobile.com/feature_server/geo-ip/test.php`
  - `http://data2.doodlemobile.com:8080/dmdata_zmm/ReceiveServlet`
  - `http://featured.perfectionholic.com:8080/moregames/index.html`
- The app loads the following JavaScript files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://media.admob.com/mraid/v1/mraid_app_interstitial.js`
  - `http://googleads.g.doubleclick.net/mads/static/sdk/native/sdk-core-v40.js`
  - `http://media.admob.com/mraid/v1/mraid_app_banner.js`
  - `http://media.admob.com/mraid/v1/mraid_app_expanded_banner.js`

Data security

- ECB mode usage identified. This mode has the disadvantage, that identical plaintext blocks are encrypted into identical ciphertext blocks. Therefore it does not hide patterns well and this mode is not recommended for use in cryptographic protocols at all.
- Use of constant initialization vectors is a bad practice. The following initialization vectors were found:
  - 16,74,71,-80,32,101,-47,72,117,-14,0,-29,70,65,-12,74
- Key derivation function used in the app with an amount of 1024 iterations is considered secure.
- The application requires the following permissions from the protectionlevel: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
- The application requires the following permissions from the protectionlevel: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.

- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- JavaScript to SDK API bridge vulnerability found: TargetSdk definition in the AndroidManifest.xml file is version: 14 . An application calling Android API methods by JavaScript and defining targetSdk version less than 17 could be vulnerable to remote code injection. For remote code injection the application has to load JavaScript or HTML code containing JavaScript code from a (generic) url.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.

- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build display`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.wordsmobile.gunship.MainActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application contains a registered scheduled alarm. With such an alarm the application repeats the execution of the registered task for example every 10 hours. The following classes register scheduled tasks:
  - `com.wordsmobile.gunship.MainActivity`
- The scheduled task gets repeated in the following intervals:
  - 24 hours
  - 72 hours
- The alarm manager has been initialized properly.

- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: assets/libs/armeabi-vfp/libmono.so`
  - `ARM 32 bit: assets/libs/armeabi-vfp/libunity.so`
  - `ARM 32 bit: lib/armeabi/libunity.so`
  - `ARM 32 bit: lib/armeabi/libmono.so`

Test Performance

- Execution time of all tests: 0:00:27.521

3.9 Mein Talking Tom (Android)

3.9.1 Tests

The following Table 3.10 summarizes the results of the Android app `Mein Talking Tom` with version *3.6.3.42*.

Table 3.10: Overview of summarized test results for »Mein Talking Tom«

App risks for enterprise usage	
☐	*Implementation flaws? No.*
☐	*Privacy risks? No.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☐	*Violations of default policy? No.*

Communication security

- ☒ *Client communication used? Yes.*
- ☑ *Communication endpoints: 43 entries, see details.*
- ☑ *Communication with country: 10 entries, see details.*
- ☒ *SSL/TLS used? Yes.*
- ☐ *Custom SSL/TLS trust manager implemented? No.*
- ☒ *SSL/TLS using custom error handling? Yes.*
- ☐ *SSL/TLS using faulty custom error handling? No.*
- ☐ *SSL/TLS using manual domain name verification? No.*
- ☒ *Unprotected HTML? Yes.*
- ☒ *Unprotected communication? Yes.*

Data security

- ☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5PADDING"`
- ☒ *Cryptographic keys found? Yes.*
- ☒ *Application needs normal permissions? Yes.*
- ☒ *Application needs dangerous permissions? Yes.*
- ☑ *Userdefined permission usage:* `com.android.vending.BILLING`, `com.google.android.c2dm.permission.RECEIVE`, `com.outfit7.mytalkingtomfree.permission.C2D-MESSAGE`
- ☑ *Overprivileged permissions:* `ACCESS-WIFI-STATE`, `READ-EXTERNAL-STORAGE`
- ☒ *Is application overprivileged? Yes.*
- ☒ *Application defines content provider? Yes.*
- ⊟ *Content provider accessible without permission: None.*
- ☒ *JavaScript to SDK API bridge usage? Yes.*
- ☐ *WiFi-Direct enabled? No.*

Input interface security

- ⊟ *App can handle documents of mimeType: None.*
- ☐ *Screenshot protection used? No.*
- ☐ *Tap Jacking Protection used? No.*

Privacy

- ☒ *Obfuscation used? Yes.*
- ☑ *Obfuscation level is:* `UNKNOWN`
- ☑ *Obfuscation framework used:* `Bangcle`
- ⊟ *Device administration policy entries: None.*
- ☑ *Accessed unique identifier(s): 9 entries, see details.*
- ☑ *Advertisment-/tracking frameworks found:* `Fyber`, `LiveRail`, `Nexage`
- ☒ *App provides public accessible activities? Yes.*
- ☒ *Backup of app is allowed? Yes.*
- ☒ *Log Statement Enabled? Yes.*

☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Camera (inactive), Location (inactive)`

Runtime Security	
☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `ClassLoader.loadClass(...), loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.9.2 Details

The following sections describe details about the test results of `Mein Talking Tom` with version *3.6.3.42*.

App risks for enterprise usage

- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - Crypto: Embedded static encryption key found, which can be extracted by attackers to revert the encryption or fake the signature of the content it is used for.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://ad5.liverail.com/?LR_IDFA_FLAG=1&iapu=0&LR_APPNAME=TestApp&wifi=true&LR_ADTYPE=3&LR_IDFA=1&os=8.3&lc=en&LR_VIDEO_POSITION=0&LR_AUTOPLAY=1&v=1.0&model=iPhone7%2C2&LR_HEIGHT=284&uid=4uT61f2yPWYD_emyE5T7yWfrBh1T_zLxO4SguN2RzD5oIK1XuEwiWfSEo8nK1gjC&LR_DURATION=3600&LR_PUBLISHER_ID=51027&lv=2.23.2&LR_FORMAT=video%2Fmp4&LR_WIDTH=320&LR_`

BUNDLE=com.outfit7.gridtestapp&LR_MUTED=0&
o7msg=1&LR_SCHEMA=vast2

  - http://api2.tnkfactory.com/tnk/ad.icon.main?
app_id=
  - http://offers.tokenads.com/show?style=xml&
client_xml&
  - http://play.google.com/store/apps/details?
id=com.facebook.orca
  - https://accounts.google.com/o/oauth2/auth?
client_id=%s&redirect_uri=%s&response_type=
code&scope=https://www.googleapis.com/auth/
youtube+https://www.googleapis.com/auth/
youtube.upload+https://www.googleapis.com/
auth/youtubepartner
  - https://m.youtube.com/create_channel?
chromeless=1&next=/channel_creation_done
  - https://oauth.vk.com/authorize?client_id=%s&
scope=%s&redirect_uri=%s&display=mobile&v=
%s&response_type=token&revoke=%d
  - https://www.googleapis.com/youtube/v3/
channels?part=status&mine=true
  - https://www.googleapis.com/youtube/v3/
subscriptions?part=snippet
  - market://details?id=
  - market://details?id=com.facebook.orca
  - outfit7p:http://apps.outfit7.com/ad/ad.jsp?
udid=

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: .facebook.com, a.archyads.
net, accounts.google.com, ad5.liverail.com,
apache.org, api.sponsorpay.com, api.w3i.com, api2.
tnkfactory.com, appdriver.jp, apps.outfit7.com,
be.outfit7.net, cdn.bee7.com, cdn.outfit7.com,
facebook.com, fb.me, graph-video.%s, graph.%s, graph.
facebook.com, java.sun.com, javax.xml.XMLConstants,
javax.xml.transform, javax.xml.transform.dom.
DOMResult, javax.xml.transform.dom.DOMSource,

javax.xml.transform.sax.SAXResult, javax.xml.transform.sax.SAXSource, javax.xml.transform.sax.SAXTransformerFactory, javax.xml.transform.stax.StAXResult, javax.xml.transform.stax.StAXSource, javax.xml.transform.stream.StreamResult, javax.xml.transform.stream.StreamSource, live.adbrix.igaworks.com, m.youtube.com, nwalsh.com, oauth.vk.com, offers.tokenads.com, play.google.com, relaxng.org, s2s.outfit7.org, sjc.ads0.nexage.com, staging.igaworks.com, storage.googleapis.com, vk.com, www.googleapis.com

- App communicates with servers in 10 countries.
- Communication with country: Netherlands, Romania, Belgium, United States, Japan, Ireland, Germany, Republic of Korea, unknown, Russia
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://apache.org/xml/features/dom/create-entity-ref-nodes
  - http://apache.org/xml/features/validation/dynamic
  - http://apache.org/xml/properties/internal/validator/dtd
  - http://apache.org/xml/properties/input-buffer-size
  - http://apache.org/xml/properties/internal/datatype-validator-factory
  - http://apache.org/xml/properties/internal/validator/schema
  - http://java.sun.com/xml/jaxp/properties/schemaSource

- http://apache.org/xml/properties/internal/error-handler
- http://apache.org/xml/features/validate-annotations
- http://apps.outfit7.com/rest/receipts/v1/apps
- http://apps.outfit7.com/rest/data/news-reporting
- http://be.outfit7.net/rest/talkingFriends/v3/
- http://apache.org/xml/features/xinclude
- http://apache.org/xml/serializer
- http://apps.outfit7.com/rest/talkingFriends/v1/video/report-event/
- http://apache.org/xml/features/validation/schema-full-checking
- http://apache.org/xml/features/validation/warn-on-duplicate-attdef
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Device/Offer/Click/Put
- http://apache.org/xml/properties/internal/entity-manager
- http://apache.org/xml/properties/internal/dtd-processor
- http://apps.outfit7.com/rest/video-gallery/v3/videos
- http://apache.org/xml/features/namespace-growth
- http://apache.org/xml/features/internal/parser-settings
- http://apache.org/xml/features/internal/strings-interned
- http://apps.outfit7.com/rest/talkingFriends/v1/push-notification/delete/%s/%s/
- http://apache.org/xml/features/dom/include-ignorable-whitespace

- http://apache.org/xml/features/create-cdata-nodes
- http://apache.org/xml/properties/internal/grammar-pool
- http://apache.org/xml/properties/locale
- http://apps.outfit7.com/rest/talkingFriends/v2/newsletter/is-subscribed/Android
- http://apache.org/xml/features/validation/warn-on-undeclared-elemdef
- http://javax.xml.XMLConstants/feature/secure-processing
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Device/Balance/Available/Get
- http://apache.org/xml/features/xinclude/fixup-base-uris
- http://apache.org/xml/properties/internal/error-reporter
- http://apache.org/xml/properties/internal/namespace-context
- http://apache.org/xml/features/warn-on-duplicate-entitydef
- http://apps.outfit7.com/rest/talkingFriends/v1/trackers/sources
- http://javax.xml.transform.sax.SAXTransformerFactory/feature/xmlfilter
- http://apache.org/xml/properties/internal/xpointer-handler
- http://java.sun.com/xml/jaxp/properties/schemaLanguage
- http://apache.org/xml/features/allow-java-encodings
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Offer/Qualified/Get
- http://apache.org/xml/features/internal/tolerate-duplicates
- http://s2s.outfit7.org/templates/

- http://apache.org/xml/features/include-comments
- http://apache.org/xml/features/scanner/notify-char-refs
- http://apache.org/xml/features/validation/id-idref-checking
- http://apps.outfit7.com/rest/data/1/events
- http://apache.org/xml/properties/dom/current-element-node
- http://javax.xml.transform.dom.DOMResult/feature
- http://javax.xml.transform.stax.StAXSource/feature
- http://apache.org/xml/properties/internal/document-scanner
- http://apache.org/xml/features/standard-uri-conformant
- http://apache.org/xml/features/continue-after-fatal-error
- http://apache.org/xml/features/validation/identity-constraint-checking
- http://apps.outfit7.com/rest/talkingFriends/v3/Android
- http://apache.org/xml/properties/
- http://apache.org/xml/features/honour-all-schemaLocations
- http://javax.xml.transform.stream.StreamSource/feature
- http://apps.outfit7.com/rest/data/report/client/v1/
- http://a.archyads.net/offers?
- http://apache.org/xml/features/xinclude/fixup-language
- http://apache.org/xml/features/nonvalidating/load-external-dtd

- http://apache.org/xml/properties/internal/entity-resolver
- http://javax.xml.transform.dom.DOMSource/feature
- http://apache.org/xml/features/
- http://apache.org/xml/features/generate-synthetic-annotations
- http://offers.tokenads.com/show?style=xml&client_xml&
- http://apps.outfit7.com/rest/talkingFriends/v1/ping
- http://apache.org/xml/features/dom/defer-node-expansion
- http://apache.org/xml/features/scanner/notify-builtin-refs
- http://apache.org/xml/features/disallow-doctype-decl
- http://apache.org/xml/features/validation/balance-syntax-trees
- http://apache.org/xml/properties/dom/document-class-name
- http://javax.xml.transform.stream.StreamResult/feature
- http://apps.outfit7.com/rest/talkingFriends/v1/assets-url/Android
- http://javax.xml.transform.sax.SAXResult/feature
- http://apache.org/xml/properties/internal/namespace-binder
- http://apache.org/xml/properties/internal/symbol-table
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Device/Balance/Redeem/Put
- http://java.sun.com/xml/jaxp/properties/
- http://apache.org/xml/properties/internal/validation-manager

  - http://javax.xml.transform.sax.SAXTransformerFactory/feature
  - http://apache.org/xml/properties/internal/xinclude-handler
  - http://apps.outfit7.com/rest/talkingFriends/v1/rate-app/Android
  - http://apache.org/xml/properties/security-manager
  - http://java.sun.com/jaxp/xpath/dom
  - http://apache.org/xml/features/validation/unparsed-entity-checking
  - http://javax.xml.transform.stax.StAXResult/feature
  - http://apache.org/xml/features/validation/schema
  - http://apps.outfit7.com/rest/talkingFriends/v3/Android-devel
  - http://apache.org/xml/properties/internal/dtd-scanner
  - http://javax.xml.transform.sax.SAXSource/feature

- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - http://ad5.liverail.com/?LR_IDFA_FLAG=1&iapu=0&LR_APPNAME=TestApp&wifi=true&LR_ADTYPE=3&LR_IDFA=1&os=8.3&lc=en&LR_VIDEO_POSITION=0&LR_AUTOPLAY=1&v=1.0&model=iPhone7%2C2&LR_HEIGHT=284&uid=4uT61f2yPWYD_emyE5T7yWfrBh1T_zLxO4SguN2RzD5oIK1XuEwiWfSEo8nK1gjC&LR_DURATION=3600&LR_PUBLISHER_ID=51027&lv=2.23.2&LR_FORMAT=video%2Fmp4&LR_WIDTH=320&LR_BUNDLE=com.outfit7.gridtestapp&LR_MUTED=0&o7msg=1&LR_SCHEMA=vast2
  - http://api2.tnkfactory.com/tnk/ad.icon.main?app_id=
  - http://offers.tokenads.com/show?style=xml&client_xml&

  - `http://play.google.com/store/apps/details?id=com.facebook.orca`

Data security

- It is considered as a bad practice to use hard-coded cryptographic keys in the application. The following hard-coded cryptographic keys were found:
  - "igaworks1k0i1d4a6e2i5g0ajwyobrks"
- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - RECORD-AUDIO (Allows an application to record audio.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.

- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.
- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- In general code obfuscation is done automatically by different obfuscation frameworks or obfuscation service providers. Detailed information to the detected framework Bangcle can be found under: http://www.bangcle.com/
- Device administration features not used.

- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build fingerprint`, `build brand`, `IMEI/MEID`, `country code + mobile network code for SIM provider`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains components (Activities) which are exported. This means these parts of the application are accessible or executable by other applications. An external app can write or read information/data to or from this app. Additionally components of this application can be executed. Following Activities are exported:
  - `com.outfit7.mytalkingtom.activity.Preferences`
  - `com.outfit7.mytalkingtomfree.Main`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually. Application defines a permission (android.permission.RECORD-AUDIO) accessing the microphone, but there were no specific API calls found. This could be an indication for overprivileges, developer misconfiguration or confused deputy attack.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`
  - `ARM 32 bit: lib/armeabi-v7a/libnativeutils.so`
  - `ARM 32 bit: lib/armeabi-v7a/libSoundTouchPlugin.so`
  - `ARM 32 bit: lib/armeabi-v7a/libsqlite3.so`
  - `ARM 32 bit: lib/armeabi-v7a/libunity.so`

Test Performance

- Execution time of all tests: 0:01:02.875

3.10 Meine Talking Angela (Android)

3.10.1 Tests

The following Table 3.11 summarizes the results of the Android app `Meine Talking Angela` with version *2.6.0.19*.

Table 3.11: Overview of summarized test results for »Meine Talking Angela«

App risks for enterprise usage

☐ *Implementation flaws? No.*
☐ *Privacy risks? No.*
☒ *Security risks? Yes.*

Blacklisted by policy

☐ *Violations of default policy? No.*

Communication security

☒ *Client communication used? Yes.*
☑ *Communication endpoints: 42 entries, see details.*
☑ *Communication with country: 10 entries, see details.*
☒ *SSL/TLS used? Yes.*
☐ *Custom SSL/TLS trust manager implemented? No.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☐ *SSL/TLS using manual domain name verification? No.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected communication? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5PADDING"`
☒ *Cryptographic keys found? Yes.*
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.outfit7.mytalkingangelafree.permission.C2D-MESSAGE`, `com.android.vending.BILLING`, `com.google.android.c2dm.permission.RECEIVE`
☑ *Overprivileged permissions:* `ACCESS-WIFI-STATE`, `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *Application defines content provider? Yes.*
⊟ *Content provider accessible without permission: None.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

⊟ *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `UNKNOWN`
☑ *Obfuscation framework used:* `Bangcle`
⊟ *Device administration policy entries: None.*

☑	*Accessed unique identifier(s): 8 entries, see details.*
☑	*Advertisment-/tracking frameworks found:* `Fyber`, `LiveRail`, `Nexage`
☒	*App provides public accessible activities? Yes.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Camera (inactive)`, `Location (inactive)`

Runtime Security

☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*Contains native libraries: Yes.*

3.10.2 Details

The following sections describe details about the test results of `Meine Talking Angela` with version *2.6.0.19*.

App risks for enterprise usage

- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - Crypto: Embedded static encryption key found, which can be extracted by attackers to revert the encryption or fake the signature of the content it is used for.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://ad5.liverail.com/?LR_IDFA_FLAG=1&iapu=0&LR_APPNAME=TestApp&wifi=true&LR_ADTYPE=3&LR_IDFA=1&os=8.3&lc=en&LR_VIDEO_POSITION=0&LR_AUTOPLAY=1&`

v=1.0&model=iPhone7%2C2&LR_HEIGHT=284&uid=4uT61f2yPWYD_emyE5T7yWfrBh1T_zLxO4SguN2RzD5oIK1XuEwiWfSEo8nK1gjC&LR_DURATION=3600&LR_PUBLISHER_ID=51027&lv=2.23.2&LR_FORMAT=video%2Fmp4&LR_WIDTH=320&LR_BUNDLE=com.outfit7.gridtestapp&LR_MUTED=0&o7msg=1&LR_SCHEMA=vast2

  - http://api2.tnkfactory.com/tnk/ad.icon.main?app_id=
  - http://offers.tokenads.com/show?style=xml&client_xml&
  - http://play.google.com/store/apps/details?id=com.facebook.orca
  - https://accounts.google.com/o/oauth2/auth?client_id=%s&redirect_uri=%s&response_type=code&scope=https://www.googleapis.com/auth/youtube+https://www.googleapis.com/auth/youtube.upload+https://www.googleapis.com/auth/youtubepartner
  - https://m.youtube.com/create_channel?chromeless=1&next=/channel_creation_done
  - https://oauth.vk.com/authorize?client_id=%s&scope=%s&redirect_uri=%s&display=mobile&v=%s&response_type=token&revoke=%d
  - https://www.googleapis.com/youtube/v3/channels?part=status&mine=true
  - https://www.googleapis.com/youtube/v3/subscriptions?part=snippet
  - market://details?id=
  - market://details?id=com.facebook.orca
  - outfit7p:http://apps.outfit7.com/ad/ad.jsp?udid=

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: .facebook.com, a.archyads.net, accounts.google.com, ad5.liverail.com, apache.org, api.sponsorpay.com, api2.tnkfactory.com, appdriver.jp, apps.outfit7.com, be.outfit7.net,

cdn.bee7.com, cdn.outfit7.com, facebook.com, fb.me, graph-video.%s, graph.%s, graph.facebook.com, java.sun.com, javax.xml.XMLConstants, javax.xml.transform, javax.xml.transform.dom.DOMResult, javax.xml.transform.dom.DOMSource, javax.xml.transform.sax.SAXResult, javax.xml.transform.sax.SAXSource, javax.xml.transform.sax.SAXTransformerFactory, javax.xml.transform.stax.StAXResult, javax.xml.transform.stax.StAXSource, javax.xml.transform.stream.StreamResult, javax.xml.transform.stream.StreamSource, live.adbrix.igaworks.com, m.youtube.com, nwalsh.com, oauth.vk.com, offers.tokenads.com, play.google.com, relaxng.org, s2s.outfit7.org, sjc.ads0.nexage.com, staging.igaworks.com, storage.googleapis.com, vk.com, www.googleapis.com

- App communicates with servers in 10 countries.
- Communication with country: Netherlands, Romania, Belgium, United States, Japan, Ireland, Germany, Republic of Korea, unknown, Russia
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://apache.org/xml/features/dom/create-entity-ref-nodes
  - http://apache.org/xml/features/validation/dynamic
  - http://apache.org/xml/properties/internal/validator/dtd
  - http://apache.org/xml/properties/input-buffer-size
  - http://apache.org/xml/properties/internal/datatype-validator-factory

- http://apache.org/xml/properties/internal/validator/schema
- http://java.sun.com/xml/jaxp/properties/schemaSource
- http://apache.org/xml/properties/internal/error-handler
- http://apache.org/xml/features/validate-annotations
- http://apps.outfit7.com/rest/receipts/v1/apps
- http://apps.outfit7.com/rest/data/news-reporting
- http://be.outfit7.net/rest/talkingFriends/v3/
- http://apache.org/xml/features/xinclude
- http://apache.org/xml/serializer
- http://apps.outfit7.com/rest/talkingFriends/v1/video/report-event/
- http://apache.org/xml/features/validation/schema-full-checking
- http://apache.org/xml/features/validation/warn-on-duplicate-attdef
- http://apache.org/xml/properties/internal/entity-manager
- http://apache.org/xml/properties/internal/dtd-processor
- http://apps.outfit7.com/rest/video-gallery/v3/videos
- http://apache.org/xml/features/namespace-growth
- http://apache.org/xml/features/internal/parser-settings
- http://apache.org/xml/features/internal/strings-interned
- http://apps.outfit7.com/rest/talkingFriends/v1/push-notification/delete/%s/%s/

- http://apache.org/xml/features/dom/include-ignorable-whitespace
- http://apache.org/xml/features/create-cdata-nodes
- http://apache.org/xml/properties/internal/grammar-pool
- http://apache.org/xml/properties/locale
- http://apps.outfit7.com/rest/talkingFriends/v2/newsletter/is-subscribed/Android
- http://apache.org/xml/features/validation/warn-on-undeclared-elemdef
- http://javax.xml.XMLConstants/feature/secure-processing
- http://apache.org/xml/features/xinclude/fixup-base-uris
- http://apache.org/xml/properties/internal/error-reporter
- http://apache.org/xml/properties/internal/namespace-context
- http://apache.org/xml/features/warn-on-duplicate-entitydef
- http://apps.outfit7.com/rest/talkingFriends/v1/trackers/sources
- http://javax.xml.transform.sax.SAXTransformerFactory/feature/xmlfilter
- http://apache.org/xml/properties/internal/xpointer-handler
- http://java.sun.com/xml/jaxp/properties/schemaLanguage
- http://apache.org/xml/features/allow-java-encodings
- http://apache.org/xml/features/internal/tolerate-duplicates
- http://s2s.outfit7.org/templates/
- http://apache.org/xml/features/include-comments

- http://apache.org/xml/features/scanner/notify-char-refs
- http://apache.org/xml/features/validation/id-idref-checking
- http://apps.outfit7.com/rest/data/1/events
- http://apache.org/xml/properties/dom/current-element-node
- http://javax.xml.transform.dom.DOMResult/feature
- http://javax.xml.transform.stax.StAXSource/feature
- http://apache.org/xml/properties/internal/document-scanner
- http://apache.org/xml/features/standard-uri-conformant
- http://apache.org/xml/features/continue-after-fatal-error
- http://apache.org/xml/features/validation/identity-constraint-checking
- http://apps.outfit7.com/rest/talkingFriends/v3/Android
- http://apache.org/xml/properties/
- http://apache.org/xml/features/honour-all-schemaLocations
- http://javax.xml.transform.stream.StreamSource/feature
- http://apps.outfit7.com/rest/data/report/client/v1/
- http://a.archyads.net/offers?
- http://apache.org/xml/features/xinclude/fixup-language
- http://apache.org/xml/features/nonvalidating/load-external-dtd
- http://apache.org/xml/properties/internal/entity-resolver

- http://javax.xml.transform.dom.DOMSource/feature
- http://apache.org/xml/features/
- http://apache.org/xml/features/generate-synthetic-annotations
- http://offers.tokenads.com/show?style=xml&client_xml&
- http://apps.outfit7.com/rest/talkingFriends/v1/ping
- http://apache.org/xml/features/dom/defer-node-expansion
- http://apache.org/xml/features/scanner/notify-builtin-refs
- http://apache.org/xml/features/disallow-doctype-decl
- http://apache.org/xml/features/validation/balance-syntax-trees
- http://apache.org/xml/properties/dom/document-class-name
- http://javax.xml.transform.stream.StreamResult/feature
- http://apps.outfit7.com/rest/talkingFriends/v1/assets-url/Android
- http://javax.xml.transform.sax.SAXResult/feature
- http://apache.org/xml/properties/internal/namespace-binder
- http://apache.org/xml/properties/internal/symbol-table
- http://java.sun.com/xml/jaxp/properties/
- http://apache.org/xml/properties/internal/validation-manager
- http://javax.xml.transform.sax.SAXTransformerFactory/feature
- http://apache.org/xml/properties/internal/xinclude-handler

  - http://apps.outfit7.com/rest/talkingFriends/v1/rate-app/Android
  - http://apache.org/xml/properties/security-manager
  - http://java.sun.com/jaxp/xpath/dom
  - http://apache.org/xml/features/validation/unparsed-entity-checking
  - http://javax.xml.transform.stax.StAXResult/feature
  - http://apache.org/xml/features/validation/schema
  - http://apps.outfit7.com/rest/talkingFriends/v3/Android-devel
  - http://apache.org/xml/properties/internal/dtd-scanner
  - http://javax.xml.transform.sax.SAXSource/feature

- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - http://ad5.liverail.com/?LR_IDFA_FLAG=1&iapu=0&LR_APPNAME=TestApp&wifi=true&LR_ADTYPE=3&LR_IDFA=1&os=8.3&lc=en&LR_VIDEO_POSITION=0&LR_AUTOPLAY=1&v=1.0&model=iPhone7%2C2&LR_HEIGHT=284&uid=4uT61f2yPWYD_emyE5T7yWfrBh1T_zLxO4SguN2RzD5oIK1XuEwiWfSEo8nK1gjC&LR_DURATION=3600&LR_PUBLISHER_ID=51027&lv=2.23.2&LR_FORMAT=video%2Fmp4&LR_WIDTH=320&LR_BUNDLE=com.outfit7.gridtestapp&LR_MUTED=0&o7msg=1&LR_SCHEMA=vast2
  - http://api2.tnkfactory.com/tnk/ad.icon.main?app_id=
  - http://offers.tokenads.com/show?style=xml&client_xml&
  - http://play.google.com/store/apps/details?id=com.facebook.orca

Data security

- It is considered as a bad practice to use hard-coded cryptographic keys in the application. The following hard-coded cryptographic keys were found:
  - "igaworks1k0i1d4a6e2i5g0ajwyobrks"
- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - RECORD-AUDIO (Allows an application to record audio.)
  - INTERNET (Allows applications to open network sockets.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.

- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.
- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- In general code obfuscation is done automatically by different obfuscation frameworks or obfuscation service providers. Detailed information to the detected framework Bangcle can be found under: http://www.bangcle.com/
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.

- Accessed unique identifier(s): `build model`, `build manufacturer`, `build serial`, `build fingerprint`, `build brand`, `IMEI/MEID`, `country code + mobile network code for SIM provider`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains components (Activities) which are exported. This means these parts of the application are accessible or executable by other applications. An external app can write or read information/data to or from this app. Additionally components of this application can be executed. Following Activities are exported:
  - `com.outfit7.mytalkingangelafree.Main`
  - `com.outfit7.mytalkingangela.activity.Preferences`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually. Application defines a permission (android.permission.RECORD-AUDIO) accessing the microphone, but there were no specific API calls found. This could be an indication for overprivileges, developer misscon-figuration or confused deputy attack.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.

- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`
  - `ARM 32 bit: lib/armeabi-v7a/libnativeutils.so`
  - `ARM 32 bit: lib/armeabi-v7a/libSoundTouchPlugin.so`
  - `ARM 32 bit: lib/armeabi-v7a/libsqlite3.so`
  - `ARM 32 bit: lib/armeabi-v7a/libunity.so`

Test Performance

- Execution time of all tests: 0:00:58.785

3.11 My Dolphin Show (Android)

3.11.1 Tests

The following Table 3.12 summarizes the results of the Android app `My Dolphin Show` with version *2.1.57*.

Table 3.12: Overview of summarized test results for »My Dolphin Show«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 34 entries, see details.*
☑	*Communication with country: 7 entries, see details.*

☒ *SSL/TLS used? Yes.*
☒ *Custom SSL/TLS trust manager implemented? Yes.*
☒ *Faulty custom SSL/TLS trust manager implemented? Yes.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☐ *SSL/TLS using manual domain name verification? No.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected communication? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5Padding", "DES/ECB/PKCS7Padding"`
☒ *Constant initialization vectors found? Yes.*
☑ *Key derivation iteration count:* `1024`
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.android.vending.BILLING, com.android.vending.CHECK-LICENSE, com.google.android.c2dm.permission.RECEIVE, com.parse.parseunitypushsample.permission.C2D-MESSAGE`
☑ *Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

– *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `HIGH`
– *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 12 entries, see details.*
☑ *Advertisment-/tracking frameworks found: 8 entries, see details.*
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*
☑ *Sensor usage:* `Camera (inactive), WIFI-Based Location, Acceleration/Light`

Runtime Security

☒ *Cordova WebApp? Yes.*

☐ *Scheduled Alarm Manager registered? No.*
☒ *Dynamically loaded code at runtime? Yes.*
☑ *Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `dalvik.system.PathClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐ *Allow app debugging Flag? No.*
☐ *Allow autoexecute after Phone Reboot? No.*
☒ *App uses outdated signature key? Yes.*
☒ *Contains native libraries: Yes.*

3.11.2 Details

The following sections describe details about the test results of `My Dolphin Show` with version *2.1.57*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
- Reasons for category privacy risks:
  - Advertisement/Tracking: App uses more than 5 advertisement and tracking providers.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - Crypto: Constant initialization vector detected. This should be avoided, as it allows an attacker to infer relationships between segments of encrypted messages if encrypted with the same key and initialization vector.
  - JavaScript Bridge attackable: App uses a bridge between web content and native code. In combination with the detected loading of unprotected web content, the functionality provided by the bridge can be exploited by man-in-the-middle attackers.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://node.veedi.com/mobile/android/server/tracker?action=`
  - `market://details?id=`
  - `market://details?id=com.google.android.gms.ads`
  - `market://search?q=pname:com.google`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: `(.*)\T1\textbackslash .amazon\T1\textbackslash .[.`, `a.applovin.com`, `ags-ext.amazon.com`, `androidads23.adcolony.com`, `androidquery.appspot.com`, `api.sponsorpay.com`, `app.adjust.io`, `applab-sdk.amazon.com`, `apptracker.spilgames.com`, `banner.fyber.com`, `cortana-gateway.amazon.com`, `d.applovin.com`, `e.crashlytics.com`, `engine.fyber.com`, `engine.sponsorpay.com`, `googleads.g.doubleclick.net`, `iframe.sponsorpay.com`, `impact.applifier.com`, `impact.staging.applifier.com`, `live.chartboost.com`, `market.android.com`, `node.veedi.com`, `plus.google.com`, `rt.applovin.com`, `service.sponsorpay.com`, `settings.crashlytics.com`, `ssl.google-analytics.com`, `vid.applovin.com`, `video.fyber.com`, `www.amazon.com`, `www.google.com`, `www.googleapis.com`, `www.googletagmanager.com`, `www.veedi.com`
- App communicates with servers in 7 countries.
- Communication with country: `Netherlands`, `Romania`, `United States`, `Ireland`, `United Kingdom`, `Germany`, `unknown`
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.

- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - `Lcom/amazon/identity/auth/device/endpoint/AbstractTokenRequest$UnsafeSslHttpClient$MySSLSocketF`
  - `Lorg/acra/util/NaiveTrustManager.`
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://rt.applovin.com/pix`
  - `http://www.amazon.com/gp/mas/get-appstore/android/ref=mas_mx_mba_iap_dl`
  - `http://node.veedi.com/mobile/android/server/tracker?action=`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://node.veedi.com/mobile/android/server/tracker?action=`

Data security

- ECB mode usage identified. This mode has the disadvantage, that identical plaintext blocks are encrypted into identical ciphertext blocks. Therefore it does not hide patterns well and this mode is not recommended for use in cryptographic protocols at all.
- Use of constant initialization vectors is a bad practice. The following initialization vectors were found:
  - 16,74,71,-80,32,101,-47,72,117,-14,0,-29,70,65,-12,74
- Key derivation function used in the app with an amount of 1024 iterations is considered secure.
- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)

  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - VIBRATE (Allows access to the vibrator.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)

- The application requires the following permissions from the protection-level: DANGEROUS
  - ACCESS-COARSE-LOCATION (Allows an app to access approximate location derived from network location sources such as cell towers and Wi-Fi.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - INTERNET (Allows applications to open network sockets.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `Adcolony`, `Adjust`, `AppLovin`, `ChartBoost`, `Crashlytics`, `Doubleclick`, `Fyber`, `Google Analytics`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.spilgames.spilsdk.SpilUnityActivityWithPrime`

- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera.Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- App contains Apache Cordova framework which enables software programmers to build applications for mobile devices using JavaScript, HTML5, and CSS3. The following Cordova plugins were detected:
- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libAmazonIapV2Bridge.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`

- ARM 32 bit: lib/armeabi-v7a/libunity.so
- x86 32bit: lib/x86/libAmazonIapV2Bridge.so
- x86 32bit: lib/x86/libmain.so
- x86 32bit: lib/x86/libmono.so
- x86 32bit: lib/x86/libunity.so

Test Performance

- Execution time of all tests: 0:01:26.880

3.12 Racing in Car (Android)

3.12.1 Tests

The following Table 3.13 summarizes the results of the Android app `Racing in Car` with version *1.1*.

Table 3.13: Overview of summarized test results for »Racing in Car«

App risks for enterprise usage	
☐	*Implementation flaws? No.*
☐	*Privacy risks? No.*
☐	*Security risks? No.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 21 entries, see details.*
☑	*Communication with country:* `United States`, `Ireland`, `United Kingdom`, `unknown`
☒	*SSL/TLS used? Yes.*
☐	*Custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☐	*SSL/TLS using manual domain name verification? No.*
Data security	
☑	*Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage:* `com.android.vending.BILLING`
☑	*Overprivileged permissions:* `READ-EXTERNAL-STORAGE`

☒	*Is application overprivileged? Yes.*
☒	*JavaScript to SDK API bridge usage? Yes.*
☐	*WiFi-Direct enabled? No.*

Input interface security

−	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*

Privacy

☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `HIGH`
−	*Device administration policy entries: None.*
☑	*Accessed unique identifier(s): 9 entries, see details.*
☑	*Advertisment-/tracking frameworks found:* `Adcolony`, `ChartBoost`, `Doubleclick`
☐	*App provides public accessible activities? No.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Camera (inactive)`, `Location (inactive)`

Runtime Security

☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.12.2 Details

The following sections describe details about the test results of `Racing in Car` with version *1.1*.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `market://details?id=`

  - market://details?id=com.google.android.gms.ads

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: accounts.google.com, androidads23.adcolony.com, app-measurement.com, csi.gstatic.com, googleads.g.doubleclick.net, impact.applifier.com, impact.staging.applifier.com, live.chartboost.com, login.live.com, login.yahoo.com, market.android.com, plus.google.com, ssl.google-analytics.com, twitter.com, www.facebook.com, www.google-analytics.com, www.google.com, www.googleapis.com, www.googletagmanager.com, www.linkedin.com, www.paypal.com
- App communicates with servers in 4 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- The application requires the following permissions from the protection-level: DANGEROUS

  - INTERNET (Allows applications to open network sockets.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)

- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.

- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build display`, `build fingerprint`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.prime31.UnityPlayerNativeActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for

dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.

- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `x86 32bit: lib/x86/libmain.so`
  - `x86 32bit: lib/x86/libmono.so`
  - `x86 32bit: lib/x86/libunity.so`

Test Performance

- Execution time of all tests: 0:01:02.519

3.13 Subway Surfers (Android)

3.13.1 Tests

The following Table 3.14 summarizes the results of the Android app `Subway Surfers` with version *1.59.1*.

Table 3.14: Overview of summarized test results for »Subway Surfers«

App risks for enterprise usage
☒ *Implementation flaws? Yes.*
☒ *Privacy risks? Yes.*
☒ *Security risks? Yes.*
Blacklisted by policy
☐ *Violations of default policy? No.*
Communication security
☒ *Client communication used? Yes.*
☑ *Communication endpoints: 55 entries, see details.*
☑ *Communication with country: 6 entries, see details.*
☒ *SSL/TLS used? Yes.*
☑ *Domains accessed with http AND https:* `play.google.com`
☐ *Custom SSL/TLS trust manager implemented? No.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☒ *SSL/TLS using manual domain name verification? Yes.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected communication? Yes.*
Data security

☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`, `"AES/CBC/PKCS7Padding"`, `"RSA/ECB/nopadding"`
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.kiloo.subwaysurf.permission.C2D-MESSAGE`, `com.android.vending.BILLING`, `com.android.vending.CHECK-LICENSE`, `com.google.android.c2dm.permission.RECEIVE`
☑ *Overprivileged permissions:* `GET-ACCOUNTS`, `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

– *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Installed app list accessed? Yes.*
☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `HIGH`
– *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 12 entries, see details.*
☑ *Advertisment-/tracking frameworks found: 9 entries, see details.*
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*
☑ *Sensor usage:* `Camera (inactive)`, `Location (inactive)`, `Acceleration/Light`

Runtime Security

☐ *Scheduled Alarm Manager registered? No.*
☒ *Dynamically loaded code at runtime? Yes.*
☑ *Dynamically loaded code at runtime type(s):* `java.net.URLClassLoader(...)`, `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐ *Allow app debugging Flag? No.*
☐ *Allow autoexecute after Phone Reboot? No.*
☒ *App uses outdated signature key? Yes.*
☒ *Contains native libraries: Yes.*

3.13.2 Details

The following sections describe details about the test results of `Subway Surfers` with version *1.59.1*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: unintended use of insecure HTTP protocol for transmissions of parameters to servers capable of HTTPS.
- Reasons for category privacy risks:
  - Advertisement/Tracking: App uses more than 5 advertisement and tracking providers.
  - App Listing: Usage of detected functionality to access list of installed apps may poses a privacy risk.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `flurry://flurrycall?event=`
  - `flurry://flurrycall?event=adWillClose`
  - `http://play.google.com/store/apps/details?id=`
  - `http://twitter.com/home?status=`
  - `https://m.google.com/app/plus/x/?v=compose&content=`
  - `https://play.google.com/store/apps/details?id=`
  - `https://www.facebook.com/dialog/feed?app_id=181821551957328&link=`

  - https://www.supersonicads.com/mobile/sdk5/log?method=
  - https://www.supersonicads.com/mobile/sdk5/log?method=contextIsNotActivity
  - https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppKey
  - https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppUserId
  - https://www.supersonicads.com/mobile/sdk5/log?method=extraParametersToJson
  - https://www.supersonicads.com/mobile/sdk5/log?method=htmlControllerDoesNotExistOnFileSystem
  - https://www.supersonicads.com/mobile/sdk5/log?method=injectJavaScript
  - https://www.supersonicads.com/mobile/sdk5/log?method=noProductType
  - https://www.supersonicads.com/mobile/sdk5/log?method=setWebViewSettings
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadBlank
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadWithPath
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewPause
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewResume
  - https://www.tumblr.com/oauth/authorize?oauth_token=%s
  - market://details?id=
  - market://details?id=com.google.android.gms.ads

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..

- Communication endpoints: .facebook.com, a.ai.inmobi.com, accounts.google.com, adlog.flurry.com, ads.flurry.com, ads.mdotm.com, analytics.query.yahoo.com, androidads23.adcolony.com, api.facebook.com, api.tumblr.com, api.vungle.com, cdn.flurry.com, connect.tapjoy.com, content-js.tapjoy.com, csi.gstatic.com, d.appsdt.com, data.flurry.com, dock.inmobi.com, e-ltvp.inmobi.com, facebook.com, googleads.g.doubleclick.net, graph-video.%s, graph.%s, graph.facebook.com, i.w.inmobi.com, ingest.vungle.com, init.supersonicads.com, inmobisdk-a.akamaihd.net, live.chartboost.com, login.live.com, login.yahoo.com, m.facebook.com, m.google.com, market.android.com, mobilelogs.supersonic.com, outcome.supersonicads.com, placements.tapjoy.com, play.google.com, plus.google.com, proton.flurry.com, rpc.tapjoy.com, rules-ltvp.inmobi.com, sdkm.w.inmobi.com, supersonic.ironbeast.io, twitter.com, ua.supersonicads.com, ws.tapjoyads.com, www.facebook.com, www.google.com, www.googleapis.com, www.linkedin.com, www.paypal.com, www.supersonicads.com, www.tumblr.com, www.vungle.com
- App communicates with servers in 6 countries.
- Communication with country: Austria, United States, Ireland, United Kingdom, Germany, unknown
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Mixed usage of HTTP and HTTPS: Protected and unprotected submission of parameters to the same domain. Indicates implementation flaw or weak communication protection.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- Correct verification of the corresponding client hostname is important for SSL/TLS security. The app changes the secure default hostname verification by the following:
  - Interface HostnameVerifier is implemented or extended.

- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://www.tumblr.com/connect/login_success.html`
  - `http://a.ai.inmobi.com/v2/ad.html`
  - `http://twitter.com/home?status=`
  - `http://play.google.com/store/apps/details?id=`
  - `http://dock.inmobi.com/carb/v1/o`
  - `http://dock.inmobi.com/carb/v1/i`
  - `http://ads.mdotm.com/ads/feed.php?`
  - `http://api.vungle.com/api/v4/`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://play.google.com/store/apps/details?id=`
  - `http://twitter.com/home?status=`

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - VIBRATE (Allows access to the vibrator.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - GET-ACCOUNTS (Allows access to the list of accounts in the Accounts Service.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)

  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)

- The application requires the following permissions from the protection-level: DANGEROUS
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - GET-TASKS (Allows an application to get information about the currently or recently running tasks.)
  - INTERNET (Allows applications to open network sockets.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.

- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `Adcolony`, `Appsdt`, `ChartBoost`, `Doubleclick`, `Flurry`, `Supersonic`, `TapJoy`, `inMobi ADs`, `mdotm`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.kiloo.unityutilities.UnityPluginActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be included when performing a device backup. In case the application contains sensitive information these can be extracted from the backup archive or cloned onto other devices.

- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`
  - `ARM 32 bit: lib/armeabi-v7a/libunity.so`
  - `x86 32bit: lib/x86/libmain.so`
  - `x86 32bit: lib/x86/libmono.so`
  - `x86 32bit: lib/x86/libunity.so`

Test Performance

- Execution time of all tests: 0:01:17.566

3.14 Teen Patti Gold (Android)

3.14.1 Tests

The following Table 3.15 summarizes the results of the Android app `Teen Patti Gold` with version *1.85.1*.

Table 3.15: Overview of summarized test results for »Teen Patti Gold«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☒	*Violations of default policy? Yes.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 21 entries, see details.*
☑	*Communication with country:* `Singapore`, `United States`, `Ireland`, `Germany`
☒	*SSL/TLS used? Yes.*
☑	*Domains accessed with http AND https:* `play.google.com`
☒	*Custom SSL/TLS trust manager implemented? Yes.*
☐	*Faulty custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☒	*SSL/TLS using manual domain name verification? Yes.*
☒	*Unprotected HTML? Yes.*
☒	*Unprotected JavaScripts? Yes.*
☒	*Unprotected communication? Yes.*
Data security	
☑	*Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage:* `com.teenpatti.hd.gold.permission.C2D-MESSAGE`, `com.android.vending.BILLING`, `com.android.vending.CHECK-LICENSE`, `com.google.android.c2dm.permission.RECEIVE`
☑	*Overprivileged permissions:* `ACCESS-FINE-LOCATION`, `READ-EXTERNAL-STORAGE`
☒	*Is application overprivileged? Yes.*
☒	*Application defines content provider? Yes.*
−	*Content provider accessible without permission: None.*
☒	*JavaScript to SDK API bridge usage? Yes.*

☐	*WiFi-Direct enabled? No.*
Input interface security	
⊟	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*
Privacy	
☒	*Installed app list accessed? Yes.*
☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `UNKNOWN`
⊟	*Device administration policy entries: None.*
☑	*Accessed unique identifier(s): 10 entries, see details.*
☑	*Advertisment-/tracking frameworks found: 6 entries, see details.*
☐	*App provides public accessible activities? No.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Acceleration/Light`
Runtime Security	
☒	*Scheduled Alarm Manager registered? Yes.*
☑	*Alarm repeating types:* `RTC-WAKEUP`
☒	*Alarm intervals dynamically? Yes.*
☐	*Alarm Manager initialized dynamically? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.14.2 Details

The following sections describe details about the test results of `Teen Patti Gold` with version *1.85.1*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: unintended use of insecure HTTP protocol for transmissions of parameters to servers capable of HTTPS.

- Reasons for category privacy risks:
  - Advertisement/Tracking: App uses more than 5 advertisement and tracking providers.
  - App tries to access the device phone number which can be use to identify the owner remotely.
  - App Listing: Usage of detected functionality to access list of installed apps poses a privacy risk for detected app type.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Blacklisted by policy

- Reasons for category violations of default policy:
  - Estimated overall app risk for the enterprise exceeds the security policy threshold due to detected risks and flaws exploitable by skilled attackers without the existence of additional supporting factors.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`
  - `http://sg-1.tp.teenpattigold.in/stats/s?p=`
  - `http://teenpattigold.com/?pid=`
  - `https://play.google.com/store/apps/details?id=com.teenpatti.hd.gold`
  - `market://details?id=com.facebook.orca`
  - `market://details?id=com.google.android.gms.ads`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..

- Communication endpoints: `.facebook.com`, `api.branch.io`, `app.adjust.io`, `bnc.lt`, `d2dejozc1aqhol.cloudfront.net`, `e.apsalar.com`, `e.crashlytics.com`, `facebook.com`, `googleads.g.doubleclick.net`, `graph-video.%s`, `graph.%s`, `graph.facebook.com`, `media.admob.com`, `play.google.com`, `plus.google.com`, `settings.crashlytics.com`, `sg-1.tp.teenpattigold.in`, `teenpattigold.com`, `www.google.com`, `www.googleapis.com`, `www.googletagmanager.com`
- App communicates with servers in 4 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Mixed usage of HTTP and HTTPS: Protected and unprotected submission of parameters to the same domain. Indicates implementation flaw or weak communication protection.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- Correct verification of the corresponding client hostname is important for SSL/TLS security. The app changes the secure default hostname verification by the following:
  - Interface HostnameVerifier is implemented or extended.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://e.apsalar.com/api/v1`
  - `http://e.apsalar.com/api/v1/canonical`
  - `http://googleads.g.doubleclick.net/mads/static/mad/sdk/native/sdk-core-v40.html`
  - `http://sg-1.tp.teenpattigold.in/stats/s?p=`
  - `http://googleads.g.doubleclick.net/mads/static/sdk/native/sdk-core-v40.html`
  - `http://e.apsalar.com/api/v1/event`
  - `http://e.apsalar.com/api/v1/resolve`
  - `http://e.apsalar.com/api/v1/start`

- The app loads the following JavaScript files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://media.admob.com/mraid/v1/mraid_app_interstitial.js`
  - `http://media.admob.com/mraid/v1/mraid_app_banner.js`
  - `http://media.admob.com/mraid/v1/mraid_app_expanded_banner.js`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`
  - `http://sg-1.tp.teenpattigold.in/stats/s?p=`
  - `http://teenpattigold.com/?pid=`

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - VIBRATE (Allows access to the vibrator.)
  - GET-ACCOUNTS (Allows access to the list of accounts in the Accounts Service.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)

- The application requires the following permissions from the protection-level: DANGEROUS
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - INTERNET (Allows applications to open network sockets.)
  - ACCESS-FINE-LOCATION (Allows an app to access precise location from location sources such as GPS, cell towers, and Wi-Fi.)
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.
- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build fingerprint`, `build brand`, `IMEI/MEID`, `phone number`, `Wifi-MAC address`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `Adjust`, `Branch Metrics`, `Crashlytics`, `Doubleclick`, `Google AdMob`, `Google Analytics`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.teenpatti.hd.gold.gold`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.

- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually. Application defines GPS Location Access Permission (android.permission.ACCESS-FINE-LOCATION) but there where no specific API calls found. This could be an indication for overprivileges, developer missconfiguration or confused deputy attack.

Runtime Security

- The application contains a registered scheduled alarm. With such an alarm the application repeats the execution of the registered task for example every 10 hours. The following classes register scheduled tasks:
  - `com.teenpatti.hd.gold.DailyBonusScheduler`
  - `com.teenpatti.hd.gold.gold`
- The scheduled task gets repeated in the following intervals:
  - Dynamic interval(s)
  - 24 hours
- The alarm manager has been initialized properly.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi/libcrashlytics.so`
  - `ARM 32 bit: lib/armeabi/libgame.so`

Test Performance

- Execution time of all tests: 0:00:39.603

3.15 Temple Run (Android)

3.15.1 Tests

The following Table 3.16 summarizes the results of the Android app `Temple Run` with version *1.6.1*.

Table 3.16: Overview of summarized test results for »Temple Run«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☐	*Privacy risks? No.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 41 entries, see details.*
☑	*Communication with country: 9 entries, see details.*
☒	*SSL/TLS used? Yes.*
☑	*Domains accessed with http AND https:* `foursquare.com`
☒	*Custom SSL/TLS trust manager implemented? Yes.*
☒	*Faulty custom SSL/TLS trust manager implemented? Yes.*
☐	*SSL/TLS using custom error handling? No.*
☐	*SSL/TLS using manual domain name verification? No.*
☒	*Unprotected HTML? Yes.*
☒	*Unprotected communication? Yes.*
Data security	
☑	*Cryptographic Primitives:* `"AES/ECB/PKCS7Padding", "PBEWithMD5AndDES"`
☑	*Key derivation iteration count:* `20`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage:* `com.android.vending.BILLING`
☑	*Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒	*Is application overprivileged? Yes.*
☒	*JavaScript to SDK API bridge usage? Yes.*
☐	*WiFi-Direct enabled? No.*

Input interface security	
⊟	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*

Privacy	
☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `UNKNOWN`
⊟	*Device administration policy entries: None.*
☑	*Accessed unique identifier(s): 8 entries, see details.*
☑	*Advertisment-/tracking frameworks found:* `Flurry`
☒	*App provides public accessible activities? Yes.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Camera (inactive)`, `Location (inactive)`, `Acceleration/Light`

Runtime Security	
☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.PathClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses insecure signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.15.2 Details

The following sections describe details about the test results of `Temple Run` with version *1.6.1*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
  - Possible flaw: unintended use of insecure HTTP protocol for transmissions of parameters to servers capable of HTTPS.
- Reasons for category security risks:

  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://api.kaixin001.com/oauth/authorize?oauth_token=%s`
  - `http://api.t.163.com/oauth/authenticate?oauth_token=%s`
  - `http://api.t.163.com/oauth/authorize?oauth_token=%s`
  - `http://api.t.sina.com.cn/oauth/authorize?oauth_token=%s`
  - `http://api.t.sohu.com/oauth/authorize?oauth_token=%s`
  - `http://api.w3i.com/AfppApi/PrivacyPolicy.aspx?PlatformType=2`
  - `http://foursquare.com/oauth/authorize?oauth_token=%s`
  - `http://vimeo.com/oauth/authorize?oauth_token=%s`
  - `http://www.plurk.com/OAuth/authorize?oauth_token=%s`
  - `http://www.plurk.com/m/authorize?oauth_token=%s`
  - `https://api.linkedin.com/uas/oauth/authorize?oauth_token=%s`
  - `https://api.login.yahoo.com/oauth/v2/request_auth?oauth_token=%s`
  - `https://api.twitter.com/oauth/authenticate?oauth_token=%s`
  - `https://api.twitter.com/oauth/authorize?oauth_token=%s`

- https://api.vkontakte.ru/oauth/authorize?client_id=%s&redirect_uri=%s&response_type=code
- https://foursquare.com/oauth2/access_token?grant_type=authorization_code
- https://foursquare.com/oauth2/authenticate?client_id=%s&response_type=code&redirect_uri=%s
- https://iap.samsungapps.com/iap/appsItemVerifyIAPReceipt.as?protocolVersion=2.0
- https://id.sapo.pt/oauth/authorize?oauth_token=%s
- https://market.android.com/details?id=
- https://oauth.constantcontact.com/ws/oauth/confirm_access?oauth_token=%s
- https://oauth.live.com/authorize?client_id=%s&redirect_uri=%s&response_type=code
- https://oauth.live.com/authorize?client_id=%s&redirect_uri=%s&response_type=code&scope=%s
- https://oauth.live.com/token?grant_type=authorization_code
- https://open.t.qq.com/cgi-bin/authorize?oauth_token=%s
- https://sandbox.evernote.com/oauth?oauth_token=%s
- https://www.dropbox.com/0/oauth/authorize?oauth_token=
- https://www.evernote.com/OAuth.action?oauth_token=%s
- https://www.facebook.com/dialog/oauth?client_id=%s&redirect_uri=%s
- https://www.facebook.com/dialog/oauth?client_id=%s&redirect_uri=%s&scope=%s
- https://www.google.com/accounts/OAuthAuthorizeToken?oauth_token=%s

  - https://www.lovefilm.com/activate?oauth_token=%s
  - https://www.yammer.com/oauth/authorize?oauth_token=%s
  - market://details?id=

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: ad.flurry.com, androidscreens.hit.bg, api.dropbox.com, api.facebook.com, api.kaixin001.com, api.linkedin.com, api.login.yahoo.com, api.t.163.com, api.t.sina.com.cn, api.t.sohu.com, api.twitter.com, api.vkontakte.ru, api.w3i.com, data.flurry.com, dl5.neospotlight.com, fc05.deviantart.net, fc08.deviantart.net, files.softicons.com, foursquare.com, graph.facebook.com, iap.samsungapps.com, id.sapo.pt, m.facebook.com, market.android.com, oauth.constantcontact.com, oauth.live.com, open.t.qq.com, openapi.lovefilm.com, sandbox.evernote.com, upload.twitter.com, vimeo.com, www.amazon.com, www.dropbox.com, www.evernote.com, www.facebook.com, www.file-extensions.org, www.google.com, www.lovefilm.com, www.plurk.com, www.yammer.com, www.youtube.com
- App communicates with servers in 9 countries.
- Communication with country: Czech Republic, United States, China, Ireland, Bulgaria, Portugal, Germany, Russia, unknown
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Mixed usage of HTTP and HTTPS: Protected and unprotected submission of parameters to the same domain. Indicates implementation flaw or weak communication protection.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - Lcom/w3i/advertiser/EasySSLSocketFactory$1.

  - Lcom/flurry/android/n.

- App uses the secure default error handling for SSL/TLS client communication. Error-prone modifications can be ruled out.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:

  - http://www.plurk.com/OAuth/request_token
  - http://api.w3i.com/PublicServices/MobileTrackingApiRestV1.svc/ActionTaken/Put
  - http://api.t.163.com/oauth/request_token
  - http://api.kaixin001.com/oauth/authorize?oauth_token=%s
  - http://api.t.sina.com.cn/oauth/request_token
  - http://www.plurk.com/m/authorize?oauth_token=%s
  - http://api.t.sohu.com/oauth/access_token
  - http://api.w3i.com/PublicServices/CtaApiRestV1.svc/Offer/Qualified/Get
  - http://www.plurk.com/OAuth/access_token
  - http://openapi.lovefilm.com/oauth/access_token
  - http://api.t.sohu.com/oauth/request_token
  - http://api.twitter.com/oauth/access_token
  - http://api.t.sina.com.cn/oauth/access_token
  - http://api.w3i.com/PublicServices/MobileTrackingApiRestV1.svc/AppWasRunV2/Put
  - http://api.w3i.com/PublicServices/MobileTrackingApiRestV1.svc/Session/End/Put
  - http://api.w3i.com/AfppApi/PrivacyPolicy.aspx?PlatformType=2
  - http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Offer/Featured/Get
  - http://api.kaixin001.com/oauth/request_token
  - http://api.w3i.com/PublicServices/MobileTrackingApiRestV1.svc/Session/Get

- http://vimeo.com/oauth/access_token
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Device/Offer/Click/Put
- http://api.t.163.com/oauth/authenticate?oauth_token=%s
- http://vimeo.com/oauth/authorize?oauth_token=%s
- http://openapi.lovefilm.com/oauth/request_token
- http://api.t.sina.com.cn/oauth/authorize?oauth_token=%s
- http://api.w3i.com/PublicServices/MobileTrackingApiRestV1.svc/AppWasRun/Put
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Device/Offer/History/Get
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Device/Balance/Available/Get
- http://www.plurk.com/OAuth/authorize?oauth_token=%s
- http://api.t.sohu.com/oauth/authorize?oauth_token=%s
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Device/Balance/Redeem/Put
- http://www.amazon.com/gp/mas/get-appstore/android/ref=mas_mx_mba_iap_dl
- http://foursquare.com/oauth/access_token
- http://api.w3i.com/PublicServices/AfppApiRestV1.svc/Offer/Qualified/Get
- http://api.kaixin001.com/oauth/access_token
- http://vimeo.com/oauth/request_token
- http://api.t.163.com/oauth/access_token
- http://api.w3i.com/PublicServices/CtaApiRestV1.svc/Device/Offer/Click/Put
- http://foursquare.com/oauth/request_token

  - `http://api.t.163.com/oauth/authorize?oauth_token=%s`
  - `http://api.twitter.com/oauth/request_token`
  - `http://foursquare.com/oauth/authorize?oauth_token=%s`

- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://api.kaixin001.com/oauth/authorize?oauth_token=%s`
  - `http://api.t.163.com/oauth/authenticate?oauth_token=%s`
  - `http://api.t.163.com/oauth/authorize?oauth_token=%s`
  - `http://api.t.sina.com.cn/oauth/authorize?oauth_token=%s`
  - `http://api.t.sohu.com/oauth/authorize?oauth_token=%s`
  - `http://api.w3i.com/AfppApi/PrivacyPolicy.aspx?PlatformType=2`
  - `http://foursquare.com/oauth/authorize?oauth_token=%s`
  - `http://vimeo.com/oauth/authorize?oauth_token=%s`
  - `http://www.plurk.com/OAuth/authorize?oauth_token=%s`
  - `http://www.plurk.com/m/authorize?oauth_token=%s`

Data security

- ECB mode usage identified. This mode has the disadvantage, that identical plaintext blocks are encrypted into identical ciphertext blocks. Therefore it does not hide patterns well and this mode is not recommended for use in cryptographic protocols at all.
- Key derivation functions with less than 1000 interations are considered vulnerable to bruteforce attacks. Therefore, this app with 20 iterations is considered vulnerable.

- The application requires the following permissions from the protection-level: NORMAL
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
- The application requires the following permissions from the protection-level: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.

- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains components (Activities) which are exported. This means these parts of the application are accessible or executable by other applications. An external app can write or read information/data to or from this app. Additionally components of this application can be executed. Following Activities are exported:
  - `com.flurry.android.CatalogActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls

accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a bit-length of less than 1024 bits (1021 bits).
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`
  - `ARM 32 bit: lib/armeabi-v7a/libunity.so`

Test Performance

- Execution time of all tests: 0:00:17.068

3.16 Temple Run 2 (Android)

3.16.1 Tests

The following Table 3.17 summarizes the results of the Android app `Temple Run 2` with version *1.27*.

Table 3.17: Overview of summarized test results for »Temple Run 2«

App risks for enterprise usage
☒ *Implementation flaws? Yes.*
☐ *Privacy risks? No.*

☒ *Security risks? Yes.*

Blacklisted by policy

☐ *Violations of default policy? No.*

Communication security

☒ *Client communication used? Yes.*
☑ *Communication endpoints: 28 entries, see details.*
☑ *Communication with country:* `United States`, `Ireland`, `United Kingdom`
☒ *SSL/TLS used? Yes.*
☒ *Custom SSL/TLS trust manager implemented? Yes.*
☒ *Faulty custom SSL/TLS trust manager implemented? Yes.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☐ *SSL/TLS using manual domain name verification? No.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected communication? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`, `"DES/ECB/PKCS7Padding"`
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.android.vending.BILLING`
☑ *Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

− *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `HIGH`
− *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 13 entries, see details.*
☑ *Advertisment-/tracking frameworks found:* `Adcolony`, `ChartBoost`, `Doubleclick`, `Flurry`
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*

☑	*Sensor usage:* `Camera (inactive), Location (inactive)`

Runtime Security	
☒	*Cordova WebApp? Yes.*
☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...), dalvik.system.PathClassLoader(...), ClassLoader.loadClass(...), loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.16.2 Details

The following sections describe details about the test results of `Temple Run 2` with version *1.27*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - JavaScript Bridge attackable: App uses a bridge between web content and native code. In combination with the detected loading of unprotected web content, the functionality provided by the bridge can be exploited by man-in-the-middle attackers.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `amzn://apps/android?p=%s`

  - flurry://flurrycall?event=
  - flurry://flurrycall?event=adWillClose
  - http://bidder.kochava.com/adserver/request/?w=
  - http://www.amazon.com/gp/mas/dl/android?p=%s
  - https://market.android.com/details?id=
  - https://play.google.com/store/apps/details?id=
  - https://play.google.com/store/apps/details?id=%s
  - https://twitter.com/intent/tweet?source=webclient
  - https://twitter.com/intent/tweet?source=webclient&text=
  - https://www.tumblr.com/oauth/authorize?oauth_token=%s
  - market://details?id=
  - market://details?id=%s
  - market://details?id=com.google.android.gms.ads
  - mraid://useCustomClose/?useCustomClose=true&callId=

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: adlog.flurry.com, ads.flurry.com, ags-ext.amazon.com, androidads23.adcolony.com, api.tumblr.com, api.vungle.com, applab-sdk.amazon.com, bidder.kochava.com, cdn.flurry.com, content.bitsontherun.com, control.kochava.com, cortana-gateway.amazon.com, csi.gstatic.com, data.flurry.com, googleads.g.doubleclick.net, impact.applifier.com, impact.staging.applifier.com, live.chartboost.com, market.android.com, play.google.com, plus.google.com, proton.flurry.com, twitter.com, www.amazon.com, www.google-analytics.com, www.google.com, www.googleapis.com, www.tumblr.com

- App communicates with servers in 3 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - `Lcom/amazon/identity/auth/device/endpoint/AbstractTokenRequest$MyHttpClient$MySSLSocketFactory$`
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://www.tumblr.com/connect/login_success.html`
  - `http://api.vungle.com/api/v1/`
  - `http://www.amazon.com/gp/mas/dl/android?p=%s`
  - `http://bidder.kochava.com/adserver/request/`
  - `http://www.amazon.com/gp/mas/get-appstore/android/ref=mas_mx_mba_iap_dl`
  - `http://bidder.kochava.com/adserver/request/?w=`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://bidder.kochava.com/adserver/request/?w=`
  - `http://www.amazon.com/gp/mas/dl/android?p=%s`

Data security

- ECB mode usage identified. This mode has the disadvantage, that identical plaintext blocks are encrypted into identical ciphertext blocks. Therefore it does not hide patterns well and this mode is not recommended for use in cryptographic protocols at all.

- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - INTERNET (Allows applications to open network sockets.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.

- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build hardware`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.imangi.unityactivity.ImangiUnityProxyActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.

- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- App contains Apache Cordova framework which enables software programmers to build applications for mobile devices using JavaScript, HTML5, and CSS3. The following Cordova plugins were detected:
- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `x86 32bit: lib/x86/libAmazonIapV2Bridge.so`
  - `x86 32bit: lib/x86/libmain.so`
  - `x86 32bit: lib/x86/libmono.so`
  - `x86 32bit: lib/x86/libunity.so`

Test Performance

- Execution time of all tests: 0:01:12.587

3.17 Township (Android)

3.17.1 Tests

The following Table 3.18 summarizes the results of the Android app `Township` with version *4.0.1*.

Table 3.18: Overview of summarized test results for »Township«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☒	*Privacy risks? Yes.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 49 entries, see details.*
☑	*Communication with country: 9 entries, see details.*
☒	*SSL/TLS used? Yes.*
☑	*Domains accessed with http AND https:* `play.google.com`
☐	*Custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☐	*SSL/TLS using manual domain name verification? No.*
☒	*Unprotected HTML? Yes.*
☒	*Unprotected communication? Yes.*
Data security	
☑	*Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☑	*Key derivation iteration count:* `65536`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage:* `com.android.vending.BILLING`, `com.android.vending.CHECK-LICENSE`, `com.google.android.c2dm.permission.RECEIVE`, `com.playrix.township.permission.C2D-MESSAGE`
☑	*Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒	*Is application overprivileged? Yes.*
☒	*Application defines content provider? Yes.*
−	*Content provider accessible without permission: None.*
☒	*JavaScript to SDK API bridge usage? Yes.*
☐	*WiFi-Direct enabled? No.*
Input interface security	

⊟	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*

Privacy	
☒	*Installed app list accessed? Yes.*
☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `UNKNOWN`
⊟	*Device administration policy entries: None.*
☑	*Accessed unique identifier(s): 10 entries, see details.*
☑	*Advertisment-/tracking frameworks found: 8 entries, see details.*
☐	*App provides public accessible activities? No.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Location (inactive)`

Runtime Security	
☒	*Scheduled Alarm Manager registered? Yes.*
☑	*Alarm repeating types:* `RTC`
☒	*Alarm intervals dynamically? Yes.*
☐	*Alarm Manager initialized dynamically? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*Contains native libraries: Yes.*

3.17.2 Details

The following sections describe details about the test results of `Township` with version *4.0.1*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: unintended use of insecure HTTP protocol for transmissions of parameters to servers capable of HTTPS.
- Reasons for category privacy risks:
  - Advertisement/Tracking: App uses more than 5 advertisement and tracking providers.

  - App Listing: Usage of detected functionality to access list of installed apps may poses a privacy risk.

- Reasons for category security risks:

  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:

  - `(https?.ftp.file)://[-a-zA-Z0-9+&@#/%?=~_.!:,..\(\)]*[.\.]+\.(mp4.mkv.flv.webm.avi.wmv)`
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`
  - `http://web.playrix.com/township_ios/og2.php?city=`
  - `http://xml.playrix.com/township_ios/og1_v2.php?object=`
  - `http://xml.playrix.com/township_ios/og2.php?city=`
  - `https://play.google.com/store/apps/details?id=com.playrix.township`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=contextIsNotActivity`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppKey`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=encodeAppUserId`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=extraParametersToJson`
  - `https://www.supersonicads.com/mobile/sdk5/log?method=htmlControllerDoesNotExistOnFileSystem`

  - https://www.supersonicads.com/mobile/sdk5/log?method=noProductType
  - https://www.supersonicads.com/mobile/sdk5/log?method=setWebViewSettings
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadBlank
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewLoadWithPath
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewPause
  - https://www.supersonicads.com/mobile/sdk5/log?method=webviewResume
  - market://details?id=
  - market://details?id=%s
  - market://details?id=com.facebook.orca
  - market://details?id=com.google.ads.interactivemedia.v3
  - market://details?id=com.playrix.township
  - market://search?q=pname:com.google

- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: .facebook.com, a.applovin.com, api.sponsorpay.com, api.vungle.com, banner.fyber.com, cdn.playrix.com, connect.tapjoy.com, csi.gstatic.com, d.applovin.com, engine.fyber.com, engine.sponsorpay.com, facebook.com, forum.playrix.com, googleads.g.doubleclick.net, graph-video.%s, graph.%s, graph.%s.facebook.com, graph.facebook.com, iframe.sponsorpay.com, imasdk.googleapis.com, impact.applifier.com, impact.staging.applifier.com, ingest.vungle.com, instagram.com, live.chartboost.com, market.android.com, mobile.twitter.com, play.google.com, plrx.gs, rink.hockeyapp.net, rpc.tapjoy.com, rt.applovin.com, sdk.hockeyapp.net, service.sponsorpay.com, township-ios.playrix.com, vdo.pokkt.com, vid.applovin.com, video.fyber.com, web.playrix.com, ws.tapjoyads.com, www.%s.facebook.com,

www.appitrk.com, www.apple.com, www.facebook.com, www.googleapis.com, www.pokkt.com, www.supersonicads.com, www.vungle.com, xml.playrix.com

- App communicates with servers in 9 countries.
- Communication with country: Netherlands, Romania, Singapore, Belgium, United States, Ireland, Germany, unknown, Russia
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Mixed usage of HTTP and HTTPS: Protected and unprotected submission of parameters to the same domain. Indicates implementation flaw or weak communication protection.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - http://xml.playrix.com/township_ios/og2.php?city=
  - http://rt.applovin.com/pix
  - http://xml.playrix.com/township_ios/og1_v2.php?object=
  - http://township-ios.playrix.com/404
  - http://xml.playrix.com/township_android/
  - http://facebook.com/TownshipMobile
  - http://cdn.playrix.com/%1$s/help/help-%2$s.html
  - http://www.apple.com/404
  - http://web.playrix.com/township_ios/og2.php?city=
  - http://plrx.gs/township_ios
  - http://instagram.com/township_mobile

  - http://forum.playrix.com/forumdisplay.php?1-Township
  - http://web.playrix.com/%1$s/help/help-%2$s.html
  - http://api.vungle.com/api/v4/
  - http://xml.playrix.com/township-tracking/api/TrackPurchase

- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - http://play.google.com/store/apps/details?id=com.facebook.orca
  - http://web.playrix.com/township_ios/og2.php?city=
  - http://xml.playrix.com/township_ios/og1_v2.php?object=
  - http://xml.playrix.com/township_ios/og2.php?city=

Data security

- Key derivation function used in the app with an amount of 65536 iterations is considered secure.
- The application requires the following permissions from the protection-level: NORMAL
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - GET-ACCOUNTS (Allows access to the list of accounts in the Accounts Service.)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)

  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - VIBRATE (Allows access to the vibrator.)

- The application requires the following permissions from the protection-level: DANGEROUS
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - INTERNET (Allows applications to open network sockets.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.
- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.
- The obfuscation level UNKNOWN means that the application has the capability to dynamically load code from outside, which currently is not part of the analysis. Therefore, the obfuscation strength is not evaluated.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build display`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `AppLovin`, `ChartBoost`, `Doubleclick`, `Fyber`, `HockeyApp`, `Supersonic`, `TapJoy`, `inneractive`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.playrix.township.Launcher`
- In this application the allow backup option is enabled. This means the application and all application data will be included when performing a device backup. In case the application contains sensitive information these can be extracted from the backup archive or cloned onto other devices.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.

- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application contains a registered scheduled alarm. With such an alarm the application repeats the execution of the registered task for example every 10 hours. The following classes register scheduled tasks:
  - `com.app.pokktsdk.notification.NotificationScheduler`
- The scheduled task gets repeated in the following intervals:
  - Dynamic interval(s)
- The alarm manager has been initialized properly.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- Loadable libraries found:
  - `x86 32bit: lib/x86/libgame.so`
  - `ARM 32 bit: lib/armeabi-v7a/libgame.so`

Test Performance

- Execution time of all tests: 0:00:47.668

3.18 Traffic Rider (Android)

3.18.1 Tests

The following Table 3.19 summarizes the results of the Android app `Traffic Rider` with version *1.2*.

Table 3.19: Overview of summarized test results for »Traffic Rider«

App risks for enterprise usage	
☐	*Implementation flaws? No.*
☐	*Privacy risks? No.*
☐	*Security risks? No.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 21 entries, see details.*
☑	*Communication with country:* `United States`, `Ireland`, `United Kingdom`, `unknown`
☒	*SSL/TLS used? Yes.*
☐	*Custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☐	*SSL/TLS using manual domain name verification? No.*
Data security	
☑	*Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage:* `com.android.vending.BILLING`
☑	*Overprivileged permissions:* `READ-EXTERNAL-STORAGE`
☒	*Is application overprivileged? Yes.*
☒	*JavaScript to SDK API bridge usage? Yes.*
☐	*WiFi-Direct enabled? No.*
Input interface security	
⊟	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*
Privacy	
☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `HIGH`
⊟	*Device administration policy entries: None.*

☑	*Accessed unique identifier(s): 9 entries, see details.*
☑	*Advertisment-/tracking frameworks found:* `Adcolony`, `ChartBoost`, `Doubleclick`
☐	*App provides public accessible activities? No.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Camera (inactive)`, `Location (inactive)`, `Acceleration/Light`

Runtime Security

☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.18.2 Details

The following sections describe details about the test results of `Traffic Rider` with version *1.2*.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `amzn://apps/android?p=`
  - `market://details?id=`
  - `market://details?id=com.google.android.gms.ads`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: `accounts.google.com`, `androidads23.adcolony.com`, `app-measurement.com`, `csi.gstatic.com`, `googleads.g.doubleclick.net`, `impact.applifier.com`, `impact.staging.applifier.`

com, live.chartboost.com, login.live.com, login.yahoo.com, market.android.com, plus.google.com, ssl.google-analytics.com, twitter.com, www.facebook.com, www.google-analytics.com, www.google.com, www.googleapis.com, www.googletagmanager.com, www.linkedin.com, www.paypal.com

- App communicates with servers in 4 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)

- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build display`, `build fingerprint`, `Wifi-MAC address`, `country code + mobile network code for SIM provider`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.

- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.prime31.UnityPlayerNativeActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:

- ARM 32 bit: lib/armeabi-v7a/libmain.so
- ARM 32 bit: lib/armeabi-v7a/libmono.so
- ARM 32 bit: lib/armeabi-v7a/libunity.so

Test Performance

- Execution time of all tests: 0:01:05.707

3.19 Train Simulator 2016 (Android)

3.19.1 Tests

The following Table 3.20 summarizes the results of the Android app `Train Simulator 2016` with version *2.5*.

Table 3.20: Overview of summarized test results for »Train Simulator 2016«

App risks for enterprise usage	
☒	*Implementation flaws? Yes.*
☐	*Privacy risks? No.*
☒	*Security risks? Yes.*
Blacklisted by policy	
☐	*Violations of default policy? No.*
Communication security	
☒	*Client communication used? Yes.*
☑	*Communication endpoints: 24 entries, see details.*
☑	*Communication with country:* `United States`, `Ireland`, `United Kingdom`, `unknown`
☒	*SSL/TLS used? Yes.*
☑	*Domains accessed with http AND https:* `play.google.com`
☐	*Custom SSL/TLS trust manager implemented? No.*
☒	*SSL/TLS using custom error handling? Yes.*
☐	*SSL/TLS using faulty custom error handling? No.*
☐	*SSL/TLS using manual domain name verification? No.*
☒	*Unprotected HTML? Yes.*
☒	*Unprotected communication? Yes.*
Data security	
☑	*Cryptographic Primitives:* `"AES/CBC/PKCS5Padding"`
☒	*Application needs normal permissions? Yes.*
☒	*Application needs dangerous permissions? Yes.*
☑	*Userdefined permission usage: 10 entries, see details.*
☐	*Is application overprivileged? No.*
☒	*Application defines content provider? Yes.*

☐	*Content provider accessible without permission: None.*
☒	*JavaScript to SDK API bridge usage? Yes.*
☐	*WiFi-Direct enabled? No.*

Input interface security

☐	*App can handle documents of mimeType: None.*
☐	*Screenshot protection used? No.*
☐	*Tap Jacking Protection used? No.*

Privacy

☒	*Obfuscation used? Yes.*
☑	*Obfuscation level is:* `HIGH`
☐	*Device administration policy entries: None.*
☑	*Accessed unique identifier(s): 8 entries, see details.*
☑	*Advertisment-/tracking frameworks found:* `Doubleclick`
☒	*App provides public accessible activities? Yes.*
☒	*Backup of app is allowed? Yes.*
☒	*Log Statement Enabled? Yes.*
☐	*Permission to access address book? No.*
☑	*Sensor usage:* `Camera (inactive)`, `Location (inactive)`

Runtime Security

☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.19.2 Details

The following sections describe details about the test results of `Train Simulator 2016` with version *2.5*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: unintended use of insecure HTTP protocol for transmissions of parameters to servers capable of HTTPS.
- Reasons for category security risks:

  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`
  - `https://market.android.com/details?id=`
  - `https://play.google.com/store/apps/details?id=`
  - `https://www.googleapis.com/games/v1management/achievements/reset?access_token=`
  - `market://details?id=`
  - `market://details?id=com.facebook.orca`
  - `market://details?id=com.google.android.gms.ads`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: `.facebook.com`, `accounts.google.com`, `app-measurement.com`, `csi.gstatic.com`, `facebook.com`, `googleads.g.doubleclick.net`, `graph-video.%s`, `graph.%s`, `login.live.com`, `login.yahoo.com`, `market.android.com`, `onesignal.com`, `play.google.com`, `plus.google.com`, `ssl.google-analytics.com`, `timuz.com`, `twitter.com`, `www.facebook.com`, `www.google-analytics.com`, `www.google.com`, `www.googleapis.com`, `www.googletagmanager.com`, `www.linkedin.com`, `www.paypal.com`
- App communicates with servers in 4 countries.
- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.

- Mixed usage of HTTP and HTTPS: Protected and unprotected submission of parameters to the same domain. Indicates implementation flaw or weak communication protection.
- App uses the secure default SSL/TLS implementation for client communication. Error-prone modifications were not detected.
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://timuz.com/mobilegames/privacypolicy.html`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://play.google.com/store/apps/details?id=com.facebook.orca`

Data security

- The application requires the following permissions from the protection-level: NORMAL
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - VIBRATE (Allows access to the vibrator.)
- The application requires the following permissions from the protection-level: DANGEROUS
  - INTERNET (Allows applications to open network sockets.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Userdefined permission usage: `com.sonyericsson.home.permission.BROADCAST_BADGE`, `com.htc.launcher.permission.READ_SETTINGS`, `com.android.vending.BILLING`, `com.majeur.launcher.permission.UPDATE_BADGE`, `com.timuzsolutions.trainsimulator2016.permission.C2D_MESSAGE`, `com.htc.launcher.permission.UPDATE_SHORTCUT`, `com.sec.android.`

`provider.badge.permission.WRITE`, `com.sec.android.provider.badge.permission.READ`, `com.anddoes.launcher.permission.UPDATE-COUNT`, `com.google.android.c2dm.permission.RECEIVE`

- No indicators for overprivilege/redundant permissions found! The defined permission can not abused by foreign apps.
- The application uses a content provider for interacting with data set structures. Content providers are the standard interface that connects data in one process with code running in another process.
- Every ContentProvider defined in the application is protected by a permission. To access the interface from an external application it must request access to it. The interface is only available if an application defines these permissions.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- Code obfuscation techniques were detected for the app.
- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.

- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- The application contains components (Activities) which are exported. This means these parts of the application are accessible or executable by other applications. An external app can write or read information/data to or from this app. Additionally components of this application can be executed. Following Activities are exported:
  - `com.facebook.unity.FBUnityDeepLinkingActivity`
  - `com.facebook.unity.FBUnityAppLinkActivity`
- In this application the allow backup option is enabled. This means the application and all application data will be considered by doing a device backup. If an application contains sensitive information these can be cloned by backing up the data and extracted from the backup archive off device.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no Permission defined for camera usage, but the application contains specific API calls accessing the camera. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.

- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `ARM 32 bit: lib/armeabi-v7a/libmain.so`
  - `ARM 32 bit: lib/armeabi-v7a/libmono.so`
  - `ARM 32 bit: lib/armeabi-v7a/libunity.so`

Test Performance

- Execution time of all tests: 0:01:06.057

3.20 Walking Dead: Road to Survival (Android)

3.20.1 Tests

The following Table 3.21 summarizes the results of the Android app `Walking Dead: Road to Survival` with version *2.7.3.36682*.

Table 3.21: Overview of summarized test results for »Walking Dead: Road to Survival«

App risks for enterprise usage
☒ *Implementation flaws? Yes.*
☒ *Privacy risks? Yes.*
☒ *Security risks? Yes.*
Blacklisted by policy
☒ *Violations of default policy? Yes.*
Communication security
☒ *Client communication used? Yes.*
☑ *Communication endpoints: 32 entries, see details.*
☑ *Communication with country:* `Romania`, `United States`, `Ireland`, `Germany`
☒ *SSL/TLS used? Yes.*

☒ *Custom SSL/TLS trust manager implemented? Yes.*
☒ *Faulty custom SSL/TLS trust manager implemented? Yes.*
☒ *SSL/TLS using custom error handling? Yes.*
☐ *SSL/TLS using faulty custom error handling? No.*
☐ *SSL/TLS using manual domain name verification? No.*
☒ *Unprotected HTML? Yes.*
☒ *Unprotected JavaScripts? Yes.*
☒ *Unprotected communication? Yes.*

Data security

☑ *Cryptographic Primitives:* `"AES/CBC/NoPadding", "AES/CBC/PKCS5Padding"`
☒ *Cryptographic keys found? Yes.*
☒ *Constant initialization vectors found? Yes.*
☒ *Cryptographic salt values found? Yes.*
☑ *Key derivation iteration count:* `1000, 1024`
☒ *Application needs normal permissions? Yes.*
☒ *Application needs dangerous permissions? Yes.*
☑ *Userdefined permission usage:* `com.android.vending.BILLING, com.android.vending.CHECK_LICENSE, com.scopely.headshot.permission.C2D_MESSAGE, com.google.android.c2dm.permission.RECEIVE`
☑ *Overprivileged permissions:* `READ_EXTERNAL_STORAGE`
☒ *Is application overprivileged? Yes.*
☒ *JavaScript to SDK API bridge usage? Yes.*
☐ *WiFi-Direct enabled? No.*

Input interface security

– *App can handle documents of mimeType: None.*
☐ *Screenshot protection used? No.*
☐ *Tap Jacking Protection used? No.*

Privacy

☒ *Installed app list accessed? Yes.*
☒ *Obfuscation used? Yes.*
☑ *Obfuscation level is:* `HIGH`
– *Device administration policy entries: None.*
☑ *Accessed unique identifier(s): 12 entries, see details.*
☑ *Advertisment-/tracking frameworks found: 7 entries, see details.*
☐ *App provides public accessible activities? No.*
☒ *Backup of app is allowed? Yes.*
☒ *Log Statement Enabled? Yes.*
☐ *Permission to access address book? No.*
☑ *Sensor usage:* `Location (inactive), Acceleration/Light`

Runtime Security

☐	*Scheduled Alarm Manager registered? No.*
☒	*Dynamically loaded code at runtime? Yes.*
☑	*Dynamically loaded code at runtime type(s):* `dalvik.system.DexClassLoader(...)`, `ClassLoader.loadClass(...)`, `loadLibrary(...)`
☐	*Allow app debugging Flag? No.*
☐	*Allow autoexecute after Phone Reboot? No.*
☒	*App uses outdated signature key? Yes.*
☒	*Contains native libraries: Yes.*

3.20.2 Details

The following sections describe details about the test results of `Walking Dead: Road to Survival` with version *2.7.3.36682*.

App risks for enterprise usage

- Reasons for category implementation flaws:
  - Possible flaw: App contains insecure code for communication protection with SSL/TLS. Common source for flawed communication protection against man-in-the-middle attacks.
- Reasons for category privacy risks:
  - Advertisement/Tracking: App uses more than 5 advertisement and tracking providers.
  - App Listing: Usage of detected functionality to access list of installed apps poses a privacy risk for detected app type.
- Reasons for category security risks:
  - Unprotected Web Content: App loads active web content (e.g. JavaScript or HTML files) without integrity protection. This poses a risk as man-in-the-middle attackers can modify the loaded web content and change the functionality of the app.
  - Crypto: Embedded static encryption key found, which can be extracted by attackers to revert the encryption or fake the signature of the content it is used for.
  - Crypto: Constant initialization vector detected. This should be avoided, as it allows an attacker to infer relationships between segments of encrypted messages if encrypted with the same key and initialization vector.

  - Crypto: Constant salt detected. This should be avoided, as it can make app vulnerable to bruteforce attacks.
  - Crypto: Overall quality of cryptographic implementation aspects is rated poor and should be inspected in detail.

Blacklisted by policy

- Reasons for category violations of default policy:
  - Estimated overall app risk for the enterprise exceeds the security policy threshold due to detected risks and flaws exploitable by skilled attackers without the existence of additional supporting factors.

Communication security

- Client communication detected. The application can establish a network connection to one or more specific host systems. URLs with parameters found:
  - `http://play.google.com/store/apps/details?id=`
  - `market://details?id=com.google.android.gms.ads`
  - `market://search?q=pname:com.google`
- Communication endpoints is a list of all potential communication endpoints Appicaptor was able to detect. This allows quick enumeration of suspicious domains, raw IP Addresses, etc..
- Communication endpoints: `.facebook.com`, `androidads21.adcolony.com`, `api.facebook.com`, `api.nanigans.com`, `api.sponsorpay.com`, `app.adjust.com`, `be.sponsorpay.com`, `collector.scopely.io`, `connect.tapjoy.com`, `content-js.tapjoy.com`, `engine.sponsorpay.com`, `facebook.com`, `googleads.g.doubleclick.net`, `graph-video.%s`, `graph.%s`, `graph.facebook.com`, `iframe.sponsorpay.com`, `m.facebook.com`, `media.admob.com`, `placements.tapjoy.com`, `play.google.com`, `plus.google.com`, `puck.scopely.io`, `rink.hockeyapp.net`, `rpc.tapjoy.com`, `sdk.hockeyapp.net`, `service.sponsorpay.com`, `ws.tapjoyads.com`, `www.google.com`, `www.googleapis.com`, `www.googletagmanager.com`, `www.youtube.com`
- App communicates with servers in 4 countries.

- Usage of SSL/TLS can protect the App's communication from adversaries. Tests indicate that communication is at least partly protected with SSL/TLS.
- Modifications of trust management found. Interface X509TrustManager is implemented or extended.
- The SSL trust management for socket communication is modified in an insecure way. The following implementations of the X509TrustManager interface should be checked:
  - `Lcom/iugome/igl/IugoX509TrustManager.`
- Modifications of the SSL error handling detected: Class WebViewClient is extended and onReceivedSslError(...) is overwritten.
- The app loads the following HTML files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://play.google.com/store/apps/details?id=`
  - `http://api.nanigans.com/disallowed.php?`
  - `http://googleads.g.doubleclick.net/mads/static/mad/sdk/native/sdk-core-v40.html`
  - `http://www.youtube.com/embed/`
  - `http://googleads.g.doubleclick.net/mads/static/sdk/native/sdk-core-v40.html`
- The app loads the following JavaScript files via unprotected communication (http), which can be exploited by attackers to remotely change the displayed content and functionality of the app:
  - `http://media.admob.com/mraid/v1/mraid_app_interstitial.js`
  - `http://media.admob.com/mraid/v1/mraid_app_banner.js`
  - `http://media.admob.com/mraid/v1/mraid_app_expanded_banner.js`
- The unprotected communication of the App via http connections can be eavesdroped or maliciously modified.
  - `http://play.google.com/store/apps/details?id=`

Data security

- It is considered as a bad practice to use hard-coded cryptographic keys in the application. The following hard-coded cryptographic keys were found:
  - "heF9BATUfWuISyO8"
- Use of constant initialization vectors is a bad practice. The following initialization vectors were found:
  - "–l3anplum–iv–"
  - "heF9BATUfWuISyO8"
  - 16,74,71,-80,32,101,-47,72,117,-14,0,-29,70,65,-12,74
- Use of constant salts can make application vulnerable to bruteforce attacks. The following constant salts were found:
  - "L3@nP1Vm"
- Key derivation function used in the app with an amount of 1000,1024 iterations is considered secure.
- The application requires the following permissions from the protection-level: NORMAL
  - ACCESS-WIFI-STATE (Allows applications to access information about Wi-Fi networks)
  - WAKE-LOCK (Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.)
  - READ-EXTERNAL-STORAGE (Allows an application to read from external storage. Any app that declares the WRITE-EXTERNAL-STORAGE permission is implicitly granted this permission. Currently, this permission is not enforced and all apps still have access to read from external storage without this permission. That will change in a future release and apps will require this permission to read from external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - GET-ACCOUNTS (Allows access to the list of accounts in the Accounts Service.)
  - ACCESS-NETWORK-STATE (Allows applications to access information about networks.)
  - VIBRATE (Allows access to the vibrator.)

- The application requires the following permissions from the protection-level: DANGEROUS
  - WRITE-EXTERNAL-STORAGE (Allows an application to write to external storage. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - READ-PHONE-STATE (Allows read only access to phone state. Note: If both minSdkVersion and targetSdkVersion values are set to 3 or lower, the system implicitly grants this permission to the app.)
  - INTERNET (Allows applications to open network sockets.)
  - CHANGE-WIFI-STATE (Allows applications to change Wi-Fi connectivity state.)
- Application uses userdefined permissions. Application can access data of a foreign application which requires this permission to access data.
- Application is propably overprivileged. Application has too much permissions. Foreign applications may be able to abuse this permission.
- Indicator for JavaScript bridge to Android API usage found. JavaScript used in the application (localy stored or loaded dynamicaly) may access and execute Android SDK API calls.
- Wifi-Direct is not enabled. There is no risk for exploiting a vulnerability in the wpa-supplicant module responsible for the wlan management. (http://www.coresecurity.com/advisories/android-wifi-direct-denial-service)

Input interface security

- No indicators for file handling found. The app does not define a filter scheme to process specific files.
- The app does not use protection measures for preventing screenshots. For apps displaying sensitive data it is recommended to disable screenshots.
- The application is vulnerable to tapjacking. When the protection is not used inside an exported activity another application is able to redirect touch events to the exported activity without the users consent.

Privacy

- The Application gathers a list of installed applications. Even though some legitimate applications may use this functionality, it can be misused to send this information to third parties.
- Code obfuscation techniques were detected for the app.

- Obfuscation levels are rated as LOW, MEDIUM, ABOVE MEDIUM, HIGH or UNKNOWN. The detected obfuscation level of HIGH provides sophisticated protection against manual analysis which requires a high effort and deep knowledge to reverse the functionality of the app.
- Device administration features not used.
- Application reads out different unique device Ids. These unique identifiers allows to identify the device and to distinguish it from other devices. Another option for reading out these IDs allow to determine the environment. The application can determine if it is running on a real device or on a virtual/emulated device.
- Accessed unique identifier(s): `build model`, `build manufacturer`, `build product`, `build serial`, `build hardware`, `build display`, `build fingerprint`, `build brand`, `IMEI/MEID`, `Wifi-MAC address`, `MMC (Mobile Country Code)`, `unique Android ID`
- Indicators for usage of advertisement/tracking framework were found.
- Advertisment-/tracking frameworks found: `Adcolony`, `Doubleclick`, `Fyber`, `Google AdMob`, `Google Analytics`, `HockeyApp`, `TapJoy`
- The application contains no specific exported activity. The application has only launchable activities which are implicit exported. This means there are no activities which can be accessed by an external application. The start activity is:
  - `com.iugome.igl.Activity`
- In this application the allow backup option is enabled. This means the application and all application data will be included when performing a device backup. In case the application contains sensitive information these can be extracted from the backup archive or cloned onto other devices.
- Logging statements found in app. This might leak security or privacy relevant information.
- Permission READ-CONTACTS not used.
- Application reads information from different Sensors. This allows the application to track the user and/or determine the environment of the user. There was no permission defined for location sensors, but the application contains API calls accessing location information. Missing permissions despite of API calls could be an indication for missconfiguration or plugin/library code which is not used. For more detailed information application has to be reviewed manually.

Runtime Security

- The application does not contain a scheduled alarm.
- Indicators found for dynamic code loading. The application loads executable code during runtime from a local or external source.
- Android dalvik code is loaded dynamically by the listed methods. Native code by Java Native Interface (for dynamic loading) is used.
- In the AndroidManifest.xml file the debuggable option is disabled. This prevents some attempts for debugging the application over the adb debug bridge with jdb. Depending of the used Android operating system this flag is not mandatory, in custom ROMs or rooted devices the OS may ignore this flag. On a non stock Android ROM this can still be misused for dynamic analyzes of the application or for doing runtime manipulation. This option should be disabled in released applications.
- The app is signed with a key that has a strength of 1024 bits. Google recommends to use a key with a strength of 2048 bit or more.
- Loadable libraries found:
  - `x86 32bit: lib/x86/libapp.so`
  - `x86 32bit: lib/x86/libclient.so`
  - `x86 32bit: lib/x86/libopenal.so`

Test Performance

- Execution time of all tests: 0:01:20.018

4 Glossary

3DES

Triple DES or 3DES is the common name for the Triple Data Encryption Algorithm (TDEA or Triple DEA) symmetric-key block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. The original DES cipher's key size of 56 bits was generally sufficient when that algorithm was designed, but the availability of increasing computational power made brute-force attacks feasible.
URL: `http://en.wikipedia.org/wiki/Triple_DES`

Address book

All sorts of information about a person can be stored within the global address book including email addresses, phone numbers, addresses, websites, chat names, and more. Apps can access the address book based on different requirements or methods (Android: permission based, iOS: access with user interaction or direct access without user interaction (deprecated)). Appicaptor evaluates the methods and API function calls of address book access as well as their context (e.g. user interaction, permission analysis)
URL: `http://developer.android.com/reference/android/Manifest.permission.html#READ_CONTACTS`, `https://developer.apple.com/library/ios/documentation/ContactData/Conceptual/AddressBookProgrammingGuideforiPhone/Introduction.html`

Advertisement frameworks

Appicaptor evaluates different advertisement and tracking frameworks e.g., Apple ID Support for Ads, Google AdMob, Apple iAd, OpenUDID, Google Analytics, possibly other AD/-Tracking, MillennialMedia, mopub, MobClix, TapJoy, Flurry, inMobi AD Tracker, MobFox, mdotm, AdWhirl, Crashlytics, inneractive, AdFonic, Mocean Mobile, GreyStripe, inMobi ADs, RevMob Ads, AdMarvel, Madvertise, Crittercism, Adobe Omniture Tracker, Burstly, Jumptap, Urban Airship, Unity3D. Advertisement frameworks grant apps access to identifiers that can be used for serving advertisements or ad tracking.

Content provider (Android)
: Content providers manage access to a structured set of data. They encapsulate the data, and provide mechanisms for defining data security. Content providers are the standard interface that connects data in one process with code running in another process. As content providers are one potential way to leak data to other apps Appicaptor searches for content provider creation in apps.
URL: `http://developer.android.com/guide/topics/providers/content-providers.html`

AES
: Advanced Encryption Standard (AES) is the standard symmetric-key block encryption algorithm with a block size of 128 bits and encryption key length of 128, 192 or 256 bits.
URL: `http://en.wikipedia.org/wiki/Advanced_Encryption_Standard`

ARC (iOS)
: see Automatic reference counting (ARC)

ASLR-PIE (iOS)
: Address space layout randomization (ASLR) protects apps from buffer overflow attacks. In order to prevent an attacker from reliably jumping to a particular exploited function in memory, ASLR involves randomly arranging the positions of key data areas of a program, including the base of the executable and the positions of the stack, heap, and libraries, in a process's address space. For full ASLR protection, the app has to be compiled with support for PIE (position-independent executable). Appicaptor evaluates whether or not the ASLR-PIE compile option was set during app creation.
URL: `http://en.wikipedia.org/wiki/Address_space_layout_randomization`, `https://developer.apple.com/library/ios/qa/qa1788/_index.html`

Automatic reference counting (ARC)
(iOS)

In Objective-C programming, Automatic Reference Counting (ARC) is a memory management enhancement where the burden of keeping track of an object's reference count is lifted from the programmer to the compiler. In traditional Objective-C, the programmer would send retain and release messages to objects in order to mark objects for deallocation or to prevent deallocation. Under ARC, the compiler does this automatically by examining the source code and then adding the retain and release messages in the compiled code.Appicaptor evaluates whether or not the ARC compile option was set during app deployment.
URL: `http://en.wikipedia.org/wiki/Automatic_Reference_Counting`, `https://developer.apple.com/library/ios/releasenotes/ObjectiveC/RN-TransitioningToARC/Introduction/Introduction.html`

Background activities

If the user performs an action that starts another app or switches to another app, the operating system moves the previously running app into the background (where the activity is no longer visible, but the instance and its state remains intact). Appicaptor evaluates the methods and API function calls of iOS background modes for audio (play and record audible content in background), location (provide location-based information to the user), voip (provide Voice-over-IP services and automatically launch after system boot so that the app can reestablish VoIP services (and is allowed to play and record background audio)), newsstand-content (process content that was recently downloaded in the background using the Newsstand Kit framework), external-accessory (communicate with an accessory that delivers data at regular intervals), bluetooth-central (use the Core-Bluetooth framework to communicate with a Bluetooth accessory while in the background), bluetooth-peripheral (use the CoreBluetooth framework to communicate in peripheral mode with a Bluetooth accessory), remote-notification (use remote notifications to resume or launch the app in the background for downloading new content), fetch (request a launch or resume by the system to fetch new content from the network on a regular basis).
URL: `https://developer.apple.com/library/ios/#documentation/general/Reference/InfoPlistKeyReference/Articles/iPhoneOSKeys.html#//apple_ref/doc/uid/TP40009252-SW22`

Blacklist Application blacklisting is a common administration practice to prevent the execution of undesirable programs. Such programs may include apps known to contain security threats or vulnerabilities but also those that are deemed inappropriate within an organization. Appicaptor will mark an app as blacklisted when Appicaptor findings are not compliant to your policy rule set.

CAST CAST is a symmetric-key block cipher with a block size of 64 bits and encryption key length of 40 to 128 bits. It is used in a number of products, notably as the default cipher in some versions of GPG and PGP.
URL: `http://en.wikipedia.org/wiki/CAST-128`

CBC In Cipher-block chaining (CBC) mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block depends on all plaintext blocks processed up to that point. To make each message unique, an initialization vector must be used in the first block.
URL: `http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation`

Client communication The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware. A server host runs one or more server programs which share their resources with clients. A client requests a server's content or service function and therefore initiate communication sessions with servers which await incoming requests. Appicaptor evaluates the methods and API function calls that initiate, perform and end communication processes with external entities.
URL: `http://en.wikipedia.org/wiki/Client%E2%80%93server_model`

Communication security Secure communication is achieved when two entities are communicating in a way not susceptible to eavesdropping, interception and manipulation. Appicaptor validates the communication security characteristics in terms of correct communication counterpart authenticity check implementations, and communication protection characteristics (integrity and encryption).
URL: `http://en.wikipedia.org/wiki/Secure_communication`

Compiler Flags The compiler transforms source code written in a programming language into another computer language (the target language, often resulting in a binary form known as object code). Several compile-time options can be used to help hardening a resulting binary e.g., against memory corruption attacks. Appicaptor evaluates the compile-time options applied during app deployment.

Custom SSL/TLS trust manager See SSL Trust Management Modification

Data Protection Data at rest on the mobile device is subject to multiple threats. To prevent this data from being unauthorizedly accessed, modified or stolen, mobile operating systems employ security protection measures such as password protection, data encryption, or a combination of both.

Data Protection (iOS) Data protection is available for iOS devices that offer hardware encryption, including iPhone 3GS and later, all iPad models, and iPod touch (3rd generation and later). Data protection enhances the built-in hardware encryption by protecting the hardware encryption keys with the device passcode. This provides an additional layer of protection for specific data on rest. Especially if a device is lost.
URL: `http://support.apple.com/kb/ht4175`

Data protection classes (iOS) When a new file is created on an iOS device, it is assigned to a specific class by the app that creates it or the default class is utilized when no specific class is assigned. The default class is NSFileProtectionComplete when an app was installed on iOS 7 whereas it is NSFileProtectionNone when an app was installed on iOS6 or prior. Each class uses different policies to determine when the data is accessible. The basic classes and policies are as follows: complete protection (NSFileProtectionComplete), protected unless open (NSFileProtectionCompleteUnlessOpen), protected until first user authentication (NSFileProtectionCompleteUntilFirstUserAuthentication) and no protection (NSFileProtectionNone). Appicaptor evaluates all file generation and modification processes within the evaluated app and monitors the (default) assignment of data protection classes to these files.
URL: `https://www.apple.com/privacy/docs/iOS_Security_Guide_Oct_2014.pdf`

Data security Appicaptor evaluates different aspects of data security: data protection (data on rest protection, see data protection), permission analysis, etc.

Default trust anchor

DES	The Data Encryption Standard (DES) is an outdated symmetric-key encryption algorithm which is now considered to be insecure for many applications. URL: `http://en.wikipedia.org/wiki/Data_Encryption_Standard`
Document types	If an app is capable of opening specific types of files, the app may indicate that support to the operating system. This allows other apps to offer the user the option to hand off those files to that mentioned app. Appicaptor extracts all document types an app can handle. URL: `https://developer.apple.com/library/ios/Documentation/FileManagement/Conceptual/DocumentInteraction_TopicsForIOS/Articles/RegisteringtheFileTypesYourAppSupports.html` , `http://developer.android.com/reference/android/content/Intent.html`
Domains accessed with HTTP and HTTPS	See Mixed usage of HTTP and HTTPS
Dynamically loaded code (Android)	Loading (external) executable code while an app is running.
ECB	The simplest of the encryption modes of a block cipher algorithm is the electronic codebook (ECB) mode. The message is divided into blocks, and each block is encrypted separately. URL: `http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation`
Flaw	A software flaw is an error, failure, or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.
fstack-protector-all (iOS)	iOS applications can apply stack smashing protection at compile time. This can be achieved by specifying the compiler option named fstack-protector-all

iCloud Usage iCloud is a cloud storage and cloud computing service provided by Apple. It allows data syncing for email, contacts, calendars, bookmarks, notes, reminders (to-do lists), iWork documents, photos and other data. The service also allows users to wirelessly back up their iOS devices to iCloud. Appicaptor examines iCloud usage as an option to store private or sensitive data with potentially different protection measures than the app's selected protection measures on the mobile device.
URL: https://www.icloud.com/

Implementation flaw See flaw

InApp purchase In-App purchase in apps enables the app developer to sell content or features directly within a free or paid app, e.g., premium content, virtual goods, or subscriptions.

JavaScript to SDK API bridge (Android) WebViews JavaScript API Calls to all Android Java methods are possible in case the app is executed on Android before 4.2 (remote code injection)
URL: http://developer.android.com/reference/android/webkit/WebView.html#addJavascriptInterface%28java.lang.Object,%20java.lang.String%29, http://sseblog.ec-spride.de/2013/09/java-script-attack-vector/

Keychain (iOS) Apps need to handle passwords and other sensitive data, such as keys or tokens. The iOS keychain provides a way to store these items. Rather than limiting access to a single process or app, access groups allow keychain items to be shared between apps. Keychain items can only be shared between apps from the same developer.
URL: https://www.apple.com/privacy/docs/iOS_Security_Guide_Oct_2014.pdf

Keychain classes (iOS)
The basic classes are as follows: Access to keychain entries when device is unlocked (kSecAttrAccessibleWhenUnlocked), after first unlock (kSecAttrAccessibleAfterFirstUnlock) or always (kSecAttrAccessibleAlways). Apps with background refresh services in iOS 7 require the keychain class kSecAttrAccessibleAfterFirstUnlock for keychain items when that information is accessed during background updates. Each keychain class has a "This device only" counterpart, which is always protected with device specific Key (the UID-key) when being copied from the device during a backup, rendering it useless if restored to a different device. Appicaptor evaluates all keychain generation and modification processes within the evaluated app and monitors the assignment of keychain entry classes.
URL: `https://www.apple.com/privacy/docs/iOS_Security_Guide_Oct_2014.pdf`

Log Statement
For e.g., application debugging there is the opportunity to utilize log statements to write data to the global device log. As the usage of log statements is one potential way to leak data Appicaptor searches for the usage of log statements in apps.

Malicious behaviour
Malicious app behavior affects the app user directly e.g. through some action within a malicious app that harms the user's data, information or processes. Malicious actions could be e.g. unauthorized data leakage, data modification or social engineering.

MD5
The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value. The security of the MD5 hash function is severely compromised, as a collision attack exists that can find collisions within seconds.
URL: `http://en.wikipedia.org/wiki/MD5`

Message UI (iOS)
The Message UI framework provides view controllers for presenting composition interfaces for email and SMS messages within a 3rd party app without requiring the user to leave the app.
URL: `https://developer.apple.com/library/ios/Documentation/MessageUI/Reference/MessageUI_Framework_Reference/_index.html`

Mixed usage of HTTP and HTTPS
When an app transmits data to a server via http that is capable of https the app does not utilize the maximum amount of protection that is offered by its communication counterpart. To detect potential but avoidable information leakage based on unprotected communication Appicaptor searches and documents for http usage when the target server is capable of https communication, as this characteristic is crucial to data in transit protection.

OpenSSL Usage
The OpenSSL Project develops a Open Source toolkit implementing the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. The project is managed by a worldwide community of volunteers. Appicaptor checks whether or not OpenSSL used within an app.
URL: `https://www.openssl.org/`

Overprivileged
Serveral apps ask for more permissions than necessary (according to their app functionality and utilized API methods within the app). This is because they are integrated with the operating system at a low level by device manufacturers or app developer requests more permissions than required (e.g., within Android app manifest file).

Padding
A block cipher works on units of a fixed size (known as a block size), but messages come in a variety of lengths. So some modes (namely ECB and CBC) require that the final block be padded before encryption. Several padding schemes exist. The simplest is to add null bytes to the plaintext to bring its length up to a multiple of the block size, but care must be taken so that the original length of the plaintext can be recovered. As an example the value of each added byte by PKCS7 padding is the number of bytes that are added.
URL: `http://en.wikipedia.org/wiki/Padding_(cryptography)`

Passbook (iOS)
With Passbook apps can store boarding passes, event tickets, retail coupons, store cards and generic passes. These elements include barcodes that can be scanned in order to convey information stored in the pass to perform actions in the physical world. As the usage of passbook is one potential way to leak data Appicaptor searches for the usage of passbook in apps.
URL: `https://developer.apple.com/passbook/`

Pasteboard Types (iOS)
When the user requests a copy or cut operation on a selection in the user interface an object in the app writes data to a pasteboard. Another object in the same or a different app then reads that data from the pasteboard and presents it to the user at a new location; this usually happens when the user requests a paste operation. The copy and paste actions can be processed with two different apps. To share data with any other app, the app can either use the system-wide pasteboard; or to share data with another app that has the same team ID as the initial app, the app-specific pasteboards can be utilized. As the usage of pasteboards is one potential way to leak data Appicaptor searches for the utilized pasteboard type and the usage of the system-wide pasteboard if available.
URL: `https://developer.apple.com/library/ios/documentation/uikit/reference/UIPasteboard_Class/Reference.html`

Permission (Android)
Android is a privilege-separated operating system, in which each application runs with a distinct system identity (Linux user ID and group ID). Additional finer-grained security features are provided through a "permission" mechanism that enforces restrictions on the specific operations that a particular process can perform, and per-URI permissions for granting ad hoc access to specific pieces of data.
URL: `http://developer.android.com/guide/topics/security/permissions.html`

PIE (iOS)
see ASLR-PIE

Privacy
Data privacy deals with the ability of an organization or individual to restrict the sharing of data with third parties.

Privacy violations
Privacy violations refers to a process in which personal, sensitive information are exposed to unauthorized third parties. Appicaptor detects privacy violations based on e.g., unauthorized screenshot captures, access to device identifiers, address book usage without notification, advertisement-/tracking frameworks usage, sensor usage (location, microphone, camera, etc.), log statements utilized, message UI usage, iCloud usage, Pasteboard or passbook usage, etc.

RC2
RC2 a symmetric-key block cipher with a block size of 64 bits and encryption key length of 8–1024 bits, in steps of 8 bits.
URL: `http://en.wikipedia.org/wiki/RC2`

RC4 Stream cipher used in popular protocols such as Transport Layer Security (TLS) (to protect Internet traffic) and WEP (to secure wireless networks). While remarkable for its simplicity and speed in software, RC4 has weaknesses that argue against its use in new systems.
URL: `http://en.wikipedia.org/wiki/RC4`

Runtime Security Runtime security summarizes Appicaptor test cases that refer to methods to harden the application binary based on compile-time options as well as the ability to execute dynamically loaded code.

Security violations Security violations refers to a circumstance that a process or data handling is not protected in an appropriate manner.

Sensor usage App's access to smartphone sensors, with or without user interaction. Appicaptor detects access to sensor data such as location data and location updates, microphone, and camera data.

SHA1 The SHA1 message-digest algorithm is a widely used cryptographic hash function producing a 160-bit (20-byte) hash value. Attacks were found on SHA-1 therefore it is recommended to move to SHA-2.
URL: `http://en.wikipedia.org/wiki/SHA-1`

Social Network usage App's interaction with social networks, based on social network framework or library usage. Appicaptor detects social network interaction with Twitter, Facebook and Weibo.

SSL Secure Sockets Layer (SSL), and its successor Transport Layer Security (TLS), are cryptographic protocols which were designed to provide communication security (integrity, authenticity and confidentiality) over untrusted communication channels.
URL: `http://tools.ietf.org/html/rfc6101`

SSL Error Handling Modification If using WebViews in coordination with SSL/TLS the app developer can modify the SSLErrorHandler. One intention to do so is to accept self-signed or even all certificates, even incorrect ones. Appicaptor detects and notifies SSL error handling modifications as these open the opportunity to improper SSL error handling and therefore facilitate Man-in-the-Middle attacks.
URL: `http://developer.android.com/reference/android/webkit/SslErrorHandler.html`

SSL/TLS usage See SSL or TLS

SSL/TLS using custom error handling	See SSL Error Handling Modification
SSL/TLS using faulty custom error handling	This refers also to SSL Error Handling Modification, but in this circumstance there is at least one point of execution where the communication proceeds even if an error is indicated. Appicaptor detects and notifies faulty custom SSL error handling modifications as these open the opportunity to improper SSL error handling and therefore facilitate Man-in-the-Middle attacks.
SSL/TLS using improper certificate validation	The communications security of SSL/TLS bases on the authenticity and integrity of the utilized server certificates. If an app implements a SSL/TLS certificate check itself and does not use the operating system's functions to validate certificates. Faulty checks can render the SSL/TLS usage for communication security useless. Appicaptor detects improper certificate validation as this opens the opportunity for Man-in-the-Middle attacks.
SSL/TLS using manual domain name verification	The ALLOW_ALL HostnameVerifier essentially turns hostname verification off. URL: `http://developer.android.com/reference/org/apache/http/conn/ssl/AllowAllHostnameVerifier.html`
SSL/TLS with changed cipher list	Appicaptor detects wether or not the app implementation changes the default SSL/TLS cipher sets.
stack smashing protection (iOS)	Stack buffer overflows occur when a program writes to a memory address on the program's call stack outside of the intended data structure. The stack smashing protection is a compile-time option to mitigate the effects of stack buffer overflows.
Static passwords in URLs	Some apps transmit certain static credentials in URL parameters. As URL parameters are not protected as they are part of the HTTP header, this is a potential way to unintentionally leak sensitive data.
TLS	Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), is a cryptographic protocol which is designed to provide communication security (integrity and confidentiality) over untrusted communication channels URL: `http://tools.ietf.org/html/rfc2246`, `http://tools.ietf.org/html/rfc4346` , `http://tools.ietf.org/html/rfc5246`

Tracking framework	See Advertisement frameworks
URL schemata	Apps that support custom URL schemes can use those schemes to receive messages. Appicaptor searches if an app registers for these URL schemes to receive external data. URL: `https://developer.apple.com/library/ios/featuredarticles/iPhoneURLScheme_Reference/Introduction/Introduction.html`
Web view	A Web View is an element that displays web pages within apps without starting a dedicated stand alone browser. Appicaptor checks if Web Views are used within apps. URL: `http://developer.android.com/reference/android/webkit/WebView.html` , `https://developer.apple.com/library/ios/documentation/uikit/reference/UIWebView_Class/Reference/Reference.html`