Scan report

Fantasy Sports, analyzed December 17, 2018

Contents

1 Overview . . . 3
2 Assessments . . . 4
2.1 Assessment of Application Security . . . 4
2.2 Assessment of Functionality . . . 9
2.3 Assessment of Data Protection . . . 11
2.4 Assessment of Personal Information Usage . . . 12
2.5 Assessment of Communication Security . . . 14
3 Detailed Information . . . 15
3.1 Contacted IPs . . . 15
3.2 HTTP Requests . . . 15
3.3 Dynamic Class Loading . . . 15
3.4 Accessed Files . . . 15

1 Overview

This report was created by App-Ray and summarizes the scan result of the following mobile application.

Application name	com.yahoo.mobile.client.android.fantasyfootball
App version	9.5.2
Size	41246.713 KB
Hash (Sha1)	be1b119c1bd84259b7c7d99c2cbc4daa00be51ed
Minimum SDK	21

Requested Permissions

com.yahoo.mobile.client.android.fantasyfootball.permission.C2D_MESSAGE
com.yahoo.mobile.client.android.permissions.YAHOO_INTER_APP
android.permission.INTERNET
android.permission.ACCESS_NETWORK_STATE
android.permission.ACCESS_WIFI_STATE
android.permission.GET_ACCOUNTS
android.permission.USE_CREDENTIALS
android.permission.MANAGE_ACCOUNTS
android.permission.AUTHENTICATE_ACCOUNTS
android.permission.WAKE_LOCK
com.google.android.c2dm.permission.RECEIVE
android.permission.ACCESS_FINE_LOCATION
android.permission.WRITE_EXTERNAL_STORAGE
android.permission.READ_EXTERNAL_STORAGE
android.permission.VIBRATE
com.yahoo.mobile.client.android.fantasyfootball.permission.RECEIVE_ADM_MESSAGE
com.amazon.device.messaging.permission.RECEIVE

2 Assessments

2.1 Assessment of Application Security

Issues checked for

- Using cryptographic operations such as random generators in an incorrect way may lead to vulnerabilities.
- Capability leaks, allowing other apps to access protected resources through public interfaces.
- SQL injection vulnerabilities and lack of input sanitization.
- Tapjacking vulnerability.
- CloakAndDagger vulnerability.
- Basic information such as name, version, and hash.
- Code integrity check and validity of signing certificate.
- Critical binaries such as busybox and su.
- Native binaries and libraries.
- Dynamic code loading.
- Hardcoded access tokens.
- Unsafe inter-process communication which is not protected against Intent spoofing or interception.
- The nested apk may contain a malicious application, nesting an APK with other used permissions can be used to avoid detection and exploit them when running.
- Unsafe use of WebView components.
- Broken app integrity checks with SafetyNet.

Issues

8 capability leaks detected

A call path exists that makes a protected API available from a public interface

Found in the following locations

- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/app/NotificationManager;->notify(ILandroid/app/Notification;)V|android.permission.VIBRATE
- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/location/LocationManager;->getLastKnownLocation(Ljava/lang/String;)Landroid/location/Location;|android.permission.ACCESS_FINE_LOCATION

- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/location/LocationManager;->isProviderEnabled(Ljava/lang/String;)Z|android.permission.ACCESS_FINE_LOCATION
- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/location/LocationManager;->requestLocationUpdates(Ljava/lang/String;JFLandroid/location/LocationListener;Landroid/os/Looper;)V|android.permission.ACCESS_FINE_LOCATION
- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/net/ConnectivityManager;->getActiveNetworkInfo()Landroid/net/NetworkInfo;|android.permission.ACCESS_NETWORK_STATE
- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/webkit/WebView;-><init>(Landroid/content/Context;)V|android.permission.INTERNET
- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/webkit/WebView;-><init>(Landroid/content/Context;Landroid/util/AttributeSet;I)V|android.permission.INTERNET
- Lcom/flurry/android/FlurryBrowserActivity;->onCreate(Landroid/os/Bundle;)V|Ljava/net/URL;->openConnection()Ljava/net/URLConnection;|android.permission.INTERNET
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/app/NotificationManager;->notify(ILandroid/app/Notification;)V|android.permission.VIBRATE
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/location/LocationManager;->getLastKnownLocation(Ljava/lang/String;)Landroid/location/Location;|android.permission.ACCESS_FINE_LOCATION
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/location/LocationManager;->isProviderEnabled(Ljava/lang/String;)Z|android.permission.ACCESS_FINE_LOCATION
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/location/LocationManager;->requestLocationUpdates(Ljava/lang/String;JFLandroid/location/LocationListener;Landroid/os/Looper;)V|android.permission.ACCESS_FINE_LOCATION
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/net/ConnectivityManager;->getActiveNetworkInfo()Landroid/net/NetworkInfo;|android.permission.ACCESS_NETWORK_STATE
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/webkit/WebView;-><init>(Landroid/content/Context;)V|android.permission.INTERNET
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Landroid/webkit/WebView;-><init>(Landroid/content/Context;Landroid/util/AttributeSet;I)V|android.permission.INTERNET
- Lcom/flurry/android/FlurryFullscreenTakeoverActivity;->onCreate(Landroid/os/Bundle;)V|Ljava/net/URL;->openConnection()Ljava/net/URLConnection;|android.permission.INTERNET
- ... and 145 more ...

References

- CWE-749 - Exposed Dangerous Method or Function

Potential SQL injections found in 3 places

Injection of SQL statements is possible where raw SQL queries are constructed at runtime. Applications should rather use prepared statements or define raw queries only as string constants.

Found in the following locations

- Lcom/c/a/a;->onUpgrade(Landroid/database/sqlite/SQLiteDatabase;II)V
- Lcom/facebook/stetho/inspector/database/SqliteDatabaseDriver;->executeRawQuery(Landroid/database/sqlite/SQLiteDatabase;Ljava/lang/String;Lcom/facebook/stetho/inspector/protocol/module/BaseDatabaseDriver$ExecuteResultHandler;)Ljava/lang/Object;
- Lcom/facebook/stetho/inspector/database/SqliteDatabaseDriver;->executeSelect(Landroid/database/sqlite/SQLiteDatabase;Ljava/lang/String;Lcom/facebook/stetho/inspector/protocol/module/BaseDatabaseDriver$ExecuteResultHandler;)Ljava/lang/Object;

References

- CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Legacy cryptography code

A method defined in java.security.Permission is used. The class or one of its superclasses is marked as legacy code by Google and should therefore not be used anymore.

Found in the following locations

- Ljava/io/SerializablePermission;-><init>(Ljava/lang/String;)V
- Ljava/io/SerializablePermission;-><init>(Ljava/lang/String;Ljava/lang/String;)V

References

- CWE-327 - Use of a Broken or Risky Cryptographic Algorithm
- OWASP M5 - Insufficient Cryptography

This app accesses files of another app

The app retrieves the PackageContext of other apps and might be able to retrieve files or execute code from these apps. The PackageContext of the following known packages was accessed: com.google.android.gms, unknown

Found in the following locations

- Lcom/google/android/gms/common/s;->getRemoteContext(Landroid/content/Context;)Landroid/content/Context;
- Lcom/google/android/gms/dynamite/DynamiteModule;->a(Landroid/content/Context;)Lcom/google/android/gms/dynamite/k;
- Lcom/google/android/gms/flags/impl/FlagProviderImpl;->init(Lcom/google/android/gms/a/a;)V
- Lcom/google/android/gms/internal/fb;->a(Landroid/content/Context;Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;
- Lcom/google/android/youtube/player/a/j;->b(Landroid/content/Context;)Landroid/content/Context;

The app contains 2 access keys to web services or APIs

Sometimes, developers forget to delete access tokens such as OAuth, AWS, SAML, or web service credentials like for facebook or twitter in their app's source code. These should be removed to prevent abuse.

Found in the following locations

- Lcom/flurry/android/impl/ads/tumblr/oauth/tumblroauth/OAuthBase;
- Lcom/google/android/gms/internal/akx;

References

- OWASP M10 - Extraneous Functionality

Unsafe WebView: Untrusted web pages may execute code in the context of this app

Web pages can make calls into this application. This can lead to execution of malicious code in the context of this app.

Found in the following locations

- Lcom/facebook/react/views/webview/ReactWebViewManager$ReactWebView;->setMessagingEnabled(Z)V
- Lcom/facebook/react/views/webview/ReactWebViewManager;->setJavaScriptEnabled(Landroid/webkit/WebView;Z)V
- Lcom/flurry/android/impl/ads/views/AdUnityView;->initLayout()V
- Lcom/flurry/android/impl/ads/views/FlurryWebView;->initialize(Landroid/content/Context;)V
- Lcom/flurry/android/impl/ads/views/TileAdWebView;->loadContent(Ljava/lang/String;Ljava/lang/String;)V
- Lcom/flurry/android/impl/ads/views/TileAdWebView;->sendMessage(Ljava/lang/String;Ljava/lang/Object;)V
- Lcom/google/android/gms/ads/internal/ar;-><init>(Landroid/content/Context;Lcom/google/android/gms/internal/ahi;Ljava/lang/String;Lcom/google/android/gms/internal/iv;)V
- Lcom/google/android/gms/internal/aff;->run()V

References

- CERT DRD13
- Android Developers Blog: Description and How To Fix

This app loads code dynamically

DexClassLoader allows applications to load and execute additional DEX code (the code runs within Android's ART virtual machine) from device storage, another app or from a remote source after being retrieved at runtime. Dynamic code loading may rise several security concerns. The most obvious vulnerability would be if a developer uses a world-writable directory (such as the SD card) for the dexPath. This may lead to code injection: a malicious application could simply replace the intended DEX code with malicious code. Dynamically loaded code may cause other flaws, too. Since dynamically loaded code is likely out of scope for static analysis, it is an attractive target for malware writers: if the malicious code is not stored inside the APK itself or not even on the device it is easier to evade malware detection. However, detecting such problems is rather possible using Dynamic Analysis.

Found in the following locations

- Lcom/google/android/gms/dynamite/h;-><init>(Ljava/lang/String;Ljava/lang/ClassLoader;)V
- Lcom/google/android/gms/dynamite/h;->loadClass(Ljava/lang/String;Z)Ljava/lang/Class;
- Lcom/google/android/gms/internal/ul;->a(Landroid/content/Context;Ljava/lang/String;Ljava/lang/String;Z)Lcom/google/android/gms/internal/ul;
- Lcom/google/android/gms/internal/yu;->b()V

References

- CWE-95 - Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')
- OWASP M10 - Extraneous Functionality

2.2 Assessment of Functionality

Issues checked for

- Automated start of the app at boot.
- Ability to unlock the screen.
- Calls to in-app billing services.
- Ability to send text messages or make phone calls.
- Device Admin rights can be aquired by a broadcast receiver and are tools for companies and admins to control their devices, but can also be used to create ransomware or other maliciouse applications. This module detects broadcast receivers that can acquire device admin rights.
- Known malware signatures.
- Unneeded permissions.
- Analysis evasion techniques and attempts to detect emulator.

Issues

The app can fingerprint the runtime environment

The application uses several techniques to gather information about its running environment - such as attempts to retrieve timestamps, collecting details about the architecture, MAC-addresses, build version, system properties, network types, saved Wi-Fi networks and other network-related information. This can be an indicator that the app, depending on its environment, may behave differently - e.g. by providing standard functionality during analysis but using spyware/malware components later on your device. In this case we look for indirect evidence such as Fingerprinting in order to flag potentially harmful applications.

Found in the following locations

- La/a;->a(Ljava/util/concurrent/ThreadPoolExecutor;Z)V
- Landroid/a/g$6;->run()V
- Landroid/a/g;-><clinit>()V
- Landroid/support/design/widget/CoordinatorLayout;->onTouchEvent(Landroid/view/MotionEvent;)Z
- Landroid/support/design/widget/CoordinatorLayout;->performIntercept(Landroid/view/MotionEvent;I)Z
- Landroid/support/design/widget/CoordinatorLayout;->resetTouchBehaviors()V
- Landroid/support/v4/a/ah$d;-><init>(Landroid/content/Context;Ljava/lang/String;)V
- Landroid/support/v4/content/a;->a(Landroid/support/v4/content/a$a;Ljava/lang/Object;)V
- Landroid/support/v4/content/a;->a(Ljava/lang/String;Ljava/io/FileDescriptor;Ljava/io/PrintWriter;[Ljava/lang/String;)V

- Landroid/support/v4/content/a;->b(Landroid/support/v4/content/a$a;Ljava/lang/Object;)V
- Landroid/support/v4/content/a;->c()V
- Landroid/support/v4/media/session/MediaSessionCompat;->b(Landroid/support/v4/media/session/PlaybackStateCompat;Landroid/support/v4/media/MediaMetadataCompat;)Landroid/support/v4/media/session/PlaybackStateCompat;
- Landroid/support/v4/widget/a;->b()V
- Landroid/support/v4/widget/DrawerLayout;->cancelChildViewTouch()V
- Landroid/support/v7/app/aa;->a()Z
- Landroid/support/v7/app/aa;->a(Landroid/location/Location;)V
- ... and 385 more ...

2 permissions are requested but might not be required

No call to an API which requires the following permissions was found:
android.permission.USE_CREDENTIALS
com.amazon.device.messaging.permission.RECEIVE

2.3 Assessment of Data Protection

Issues checked for

- Static data flow analysis revealing potential leaks of private information.
- Registration of secret dialer codes.
- Reading or writing to unprotected external storage.
- Google Cloud Backup.

Issues

The app can read and write the SD card

The app requires permission WRITE_EXTERNAL_STORAGE and can therefore read and write external storage External storage is world read- and writable and must not be used to store private information.

Found in the following locations

- AndroidManifest.xml

The app can read the contents of the SD card

The app requires permission READ_EXTERNAL_STORAGE and can therefore read from the SD card. The SD card is world read- and writable. Storing sensitive information on it is a commonly made mistake.

Found in the following locations

- AndroidManifest.xml

2.4 Assessment of Personal Information Usage

Issues checked for

- Accesses to contact data.
- Tracking of user locations.
- Retrieval of unique phone identifiers such as IMEI or IMSI
- Retrieval of phone number.
- Attempts to identify the current execution environment, accesses to system logfiles, and communication with tracking servers.
- Audio recording capability.
- Ability to take screenshots.
- Ability to use Bluetooth.
- Ability to take pictures using the built-in camera.
- Communication with advertisement servers or usage of ad libraries.

Issues

1 ad library is used

References to the following libraries are found in the app:
inmobi

Found in the following locations

- Landroid/content/pm/ActivityInfo;
- Landroid/graphics/Point;
- Landroid/os/Bundle;
- Landroid/util/SparseArray;
- Landroid/widget/LinearLayout$LayoutParams;
- Lcom/flurry/android/AdCreative;
- Lcom/flurry/android/AdNetworkView;
- Lcom/flurry/android/impl/ads/mediation/AdNetworkTakeover;
- Lcom/flurry/android/impl/ads/mediation/inmobi/InMobiBannerView$AdBannerListener;
- Lcom/flurry/android/impl/ads/mediation/inmobi/InMobiBannerView;
- Lcom/flurry/android/impl/ads/mediation/inmobi/InMobiSizeMapper;
- Lcom/flurry/android/impl/ads/mediation/inmobi/InMobiTakeover$AdInterstitialListener;
- Lcom/flurry/android/impl/ads/mediation/inmobi/InMobiTakeover;
- Lcom/flurry/android/impl/ads/mediation/ThirdPartyAdApi;
- Lcom/flurry/android/impl/ads/mediation/ThirdPartyAdCreator;
- Lcom/flurry/android/impl/core/log/Flog;
- ... and 16 more ...

References

- EU-GDPR
- CWE-359 - Exposure of Private Information ('Privacy Violation')

The app accesses your GPS location

The app can determine your accurate location using GPS and Wifi localization.

Found in the following locations

- AndroidManifest.xml

References

- EU-GDPR
- CWE-359 - Exposure of Private Information ('Privacy Violation')

The app accesses your wifi state which can potentially leak your location

The app requires permission ACCESS_WIFI_STATE and can therefore be able to determine your location based on the visible WIFI networks.

Found in the following locations

- AndroidManifest.xml

References

- EU-GDPR
- CWE-359 - Exposure of Private Information ('Privacy Violation')

2.5 Assessment of Communication Security

Issues checked for

- Communication over unencrypted HTTP.
- Access to email attachments.
- Deactivated TLS server certificate validation.

Issues

The app communicates over unencrypted HTTP

The app contains the string 'http://', indicating that it communicates over unencrypted HTTP.
http://%s/%s.bundle?platform=android&dev=%s&hot=%s&minify=%s
http://%s/%s
http://
http://%s/open-stack-frame
http://%s/symbolicate
http://%s/jscheapcaptureupload
http://%s/status
http://%s/launch-js-devtools
http://%s/inspector/device?name=%s
...
http://xml.org/sax/properties/lexical-handler
http://%s/onchange

Found in the following locations

- Lcom/caverock/androidsvg/g;
- Lcom/facebook/react/devsupport/DevServerHelper;
- Lcom/facebook/react/views/toolbar/ReactToolbar;

References

- CWE-311 - Missing Encryption of Sensitive Data

3 Detailed Information

3.1 Contacted IPs

No communication recorded.

3.2 HTTP Requests

No plaintext HTTP communication observed.

3.3 Dynamic Class Loading

No dynamic class loading observed.

3.4 Accessed Files

No file accesses recorded.