ImmuniWeb® Mobile App Scanner

Test security and privacy of your mobile application (iOS & Android), detect OWASP Mobile Top 10 and other weaknesses.

Mobile Application Audit Summary

APP NAME	APP ID	APP VERSION
Jago Sholat	org.d3ifcool.jagosholat	1.3

DEVICE TYPE	TEST STARTED	TEST FINISHED
Android	November 28th 2018, 04:04	November 28th 2018, 04:08

HIGHLIGHTS

POTENTIAL SECURITY FLAWS

Zero false-positive SLA and advanced manual testing of application is only available in ImmuniWeb® MobileSuite.

WARNING	LOW RISK	MEDIUM RISK	HIGH RISK
4	3	2	0

Mobile Application Behaviour

Mobile Application Functionality

The mobile application uses the following functionality that can endanger user's privacy under certain circumstances:

Location

The mobile application has an access to geographical location of the mobile phone.

Mobile Application Audit

The automated audit revealed the following security flaws and weaknesses that may impact the application:

PREDICTABLE RANDOM NUMBER GENERATOR [M5] [CWE-338] [SAST] — MEDIUM

Description:
The mobile application uses a predictable Random Number Generator (RNG).
Under certain conditions this weakness may jeopardize mobile application data encryption or other protection based on randomization. For example, if encryption tokens are generated inside of the application and an attacker can provide application with a predictable token to validate and then execute a sensitive activity within the application or its backend.

Example of insecure code:

```
Random random = new Random();
byte bytes[] = new byte[20];
random.nextBytes(bytes);
```

Example of secure code:

```
SecureRandom random = new SecureRandom();
byte bytes[] = new byte[20];
random.nextBytes(bytes);
```

Details:
There is 'new Random()' found in file 'org/d3ifcool/jagosholat/presenters/helpers/MethodHelper.java':

```
line 18:     private String nol_menit = "";
line 19:     private Random random = new Random();
line 20:     private String randomChar;
```

There is 'new Random()' found in file 'org/d3ifcool/jagosholat/views/fragments/CatatanFragment.java':

```
line 38:         mTextViewTanggal.setText(this.methodHelper.getDateToday());
line 39:         int getIndexArrayHadis = new Random().nextInt(((this.mHadistArab.length
- 1) - 0) + 1) + 0;
line 40:         int mResIdHadistArab = this.mHadistArab[getIndexArrayHadis];
```

CVSSv3 Base Score:
4.8 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N)

Reference:
- https://developer.android.com/reference/java/util/Random.html
- https://developer.android.com/reference/java/security/SecureRandom.html

CLEARTEXT SQLITE DATABASE [M2] [CWE-312] [DAST] — MEDIUM

Description:
The mobile application uses an unencrypted SQLite database.
This database can be accessed by an attacker with physical access to the mobile device or a malicious application with root access to the device. The application should not store sensitive information in clear text.

Details:
In file jagosholat.db:

```
TABLES:
android_metadata
data_ibadah
sqlite_autoindex_data_ibadah_1

RAW DUMP:
CREATE TABLE android_metadata (locale TEXT);CREATE TABLE data_ibadah (_id TEXT PRIMARY
KEY,tanggal TEXT NOT NULL,shalat TEXT NOT NULL,waktu TEXT NOT NULL,status TEXT NOT NULL);
```

CVSSv3 Base Score:
5.5 (AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N)

ENABLED DEBUG MODE [M2] [CWE-921] [SAST] — LOW

Description:
The mobile application has debug mode enabled. Debug mode is used by application developers during development process and

should be disabled when application is in production. This mode can expose technical information and can facilitate reverse engineering of the application.

Example of insecure code:

```
android:debuggable="true"
```

Example of secure code:

```
android:debuggable="false"
```

Details:
There is 'android:debuggable="true"' found in file 'android/AndroidManifest.xml':

```
line 6:     <application android:allowBackup="true"
android:appComponentFactory="android.support.v4.app.CoreComponentFactory"
android:debuggable="true" android:icon="@mipmap/ic_logo_jago_sholat"
android:label="@string/app_name" android:roundIcon="@mipmap/ic_logo_jago_sholat"
android:supportsRtl="true" android:theme="@style/AppTheme">
```

CVSSv3 Base Score:
3.3 (AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

Reference:
- https://developer.android.com/guide/topics/manifest/application-element.html

ENABLED APPLICATION BACKUP [M2] [CWE-921] [SAST] LOW

Description:
The mobile application uses external backup functionality (default Android backup mechanism) that may store inside sensitive data from the application. In certain conditions, this may lead to information disclosure (e.g. when a backup server or your Gmail account is compromised).

Example of insecure code:

```
android:allowBackup="true"
```

Example of secure code:

```
android:allowBackup="false"
```

Details:
There is 'android:allowBackup="true"' found in file 'android/AndroidManifest.xml':

```
line 6:     <application android:allowBackup="true"
android:appComponentFactory="android.support.v4.app.CoreComponentFactory"
android:debuggable="true" android:icon="@mipmap/ic_logo_jago_sholat"
android:label="@string/app_name" android:roundIcon="@mipmap/ic_logo_jago_sholat"
android:supportsRtl="true" android:theme="@style/AppTheme">
```

CVSSv3 Base Score:
3.3 (AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N)

Reference:
- https://developer.android.com/guide/topics/manifest/application-element.html
- http://resources.infosecinstitute.com/android-hacking-security-part-15-hacking-android-apps-using-backup-techniques/

MISSING TAPJACKING PROTECTION [M1] [CWE-451] [SAST] LOW

Description:
The mobile application does not have a tapjacking protection required to mitigate tapjacking attacks.
By default, Android OS permits a mobile application to display its user interface over the user interface of another application installed and run on the device. When user touches the screen, application may pass the touch event to another application below its user interface layer that the user does not see, serving like a proxy to pass unintended touch activities. This attack is quite similar to clickjacking but for mobile devices. In order to be successfully exploited, a malicious application shall be already installed on the mobile phone of the victim. An example of exploitation would be a malware app that tricks user to unwittingly tap on a payment button (or any other functionality) of a sensitive application when playing a game or doing other innocent activity in the malicious application screen.

Example of secure code:

```
public class MyActivity extends Activity {
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);

        final Button myButton = (Button)findViewById(R.id.button_id);
        myButton.setFilterTouchesWhenObscured(true);

        myButton.setOnClickListener(new View.OnClickListener() {
            // Perform action on click
        }
    }
}

<Button
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="@string/self_destruct"
    android:onClick="selfDestruct"
    android:filterTouchesWhenObscured="true" />
```

Details:
There is android:filterTouchesWhenObscured="true" missing in files:

- android/res/layout-v22/abc_alert_dialog_button_bar_material.xml
- android/res/layout-v22/fragment_statistik_harian.xml
- android/res/layout-v22/fragment_jadwal.xml
- android/res/layout-v22/fragment_statistik.xml
- android/res/layout-v22/content_statistik_harian.xml
- android/res/layout-watch-v20/abc_alert_dialog_button_bar_material.xml
- android/res/layout-watch-v20/abc_alert_dialog_title_material.xml
- android/res/layout/content_tatacara_image.xml
- android/res/layout/abc_list_menu_item_checkbox.xml
- android/res/layout/fragment_main.xml
- android/res/layout/content_catatan_button.xml
- android/res/layout/design_layout_snackbar_include.xml
- android/res/layout/content_statistik_empty.xml
- android/res/layout/abc_alert_dialog_button_bar_material.xml
- android/res/layout/abc_activity_chooser_view_list_item.xml
- android/res/layout/notification_template_big_media.xml
- android/res/layout/design_layout_tab_icon.xml
- android/res/layout/activity_tentang_kami.xml
- android/res/layout/abc_screen_content_include.xml
- android/res/layout/abc_activity_chooser_view.xml
- android/res/layout/abc_select_dialog_material.xml
- android/res/layout/abc_action_bar_title_item.xml
- android/res/layout/abc_screen_toolbar.xml
- android/res/layout/activity_detail.xml
- android/res/layout/abc_dialog_title_material.xml
- android/res/layout/abc_search_view.xml
- android/res/layout/abc_search_dropdown_item_icons_2line.xml
- android/res/layout/abc_action_menu_item_layout.xml
- android/res/layout/design_bottom_sheet_dialog.xml
- android/res/layout/activity_splash_screen.xml
- android/res/layout/fragment_statistik_harian.xml

- android/res/layout/design_bottom_navigation_item.xml
- android/res/layout/abc_action_bar_up_container.xml
- android/res/layout/notification_template_icon_group.xml
- android/res/layout/abc_list_menu_item_layout.xml
- android/res/layout/fragment_catatan.xml
- android/res/layout/design_navigation_item_separator.xml
- android/res/layout/notification_media_cancel_action.xml
- android/res/layout/notification_action_tombstone.xml
- android/res/layout/fragment_jadwal.xml
- android/res/layout/abc_action_menu_layout.xml
- android/res/layout/notification_template_part_time.xml
- android/res/layout/notification_action.xml
- android/res/layout/abc_list_menu_item_radio.xml
- android/res/layout/mtrl_layout_snackbar_include.xml
- android/res/layout/fragment_statistik.xml
- android/res/layout/fragment_tatacara.xml
- android/res/layout/abc_screen_simple_overlay_action_mode.xml
- android/res/layout/abc_popup_menu_item_layout.xml
- android/res/layout/fragment_main_tab.xml
- android/res/layout/design_menu_item_action_area.xml
- android/res/layout/select_dialog_singlechoice_material.xml
- android/res/layout/select_dialog_item_material.xml
- android/res/layout/design_navigation_menu.xml
- android/res/layout/support_simple_spinner_dropdown_item.xml
- android/res/layout/notification_media_action.xml
- android/res/layout/notification_template_media.xml
- android/res/layout/abc_alert_dialog_material.xml
- android/res/layout/content_tatacara_text_niat.xml
- android/res/layout/design_navigation_item.xml
- android/res/layout/abc_tooltip.xml
- android/res/layout/activity_main_backup.xml
- android/res/layout/abc_action_mode_bar.xml
- android/res/layout/content_tatacara_text_doa.xml
- android/res/layout/design_navigation_item_header.xml
- android/res/layout/notification_template_lines_media.xml
- android/res/layout/content_statistik_harian.xml
- android/res/layout/design_layout_snackbar.xml
- android/res/layout/notification_template_big_media_narrow_custom.xml
- android/res/layout/abc_action_mode_close_item_material.xml
- android/res/layout/design_navigation_item_subheader.xml
- android/res/layout/abc_list_menu_item_icon.xml
- android/res/layout/fragment_tatacara_text.xml
- android/res/layout/content_statistik_update.xml
- android/res/layout/fragment_tatacara_image.xml
- android/res/layout/abc_expanded_menu_layout.xml
- android/res/layout/notification_template_media_custom.xml
- android/res/layout/abc_cascading_menu_item_layout.xml
- android/res/layout/activity_main.xml
- android/res/layout/fragment_statistik_grafik.xml
- android/res/layout/design_text_input_password_icon.xml
- android/res/layout/mtrl_layout_snackbar.xml
- android/res/layout/design_navigation_menu_item.xml
- android/res/layout/notification_template_custom_big.xml
- android/res/layout/abc_popup_menu_header_item_layout.xml
- android/res/layout/abc_alert_dialog_title_material.xml
- android/res/layout/notification_template_big_media_custom.xml
- android/res/layout/notification_template_big_media_narrow.xml
- android/res/layout/notification_template_part_chronometer.xml
- android/res/layout/abc_screen_simple.xml
- android/res/layout/select_dialog_multichoice_material.xml
- android/res/layout/fragment_kiblat.xml
- android/res/layout/design_layout_tab_text.xml
- android/res/layout-sw600dp/design_layout_snackbar.xml
- android/res/layout-sw600dp/activity_main.xml
- android/res/layout-sw600dp/mtrl_layout_snackbar.xml
- android/res/layout-v26/abc_screen_toolbar.xml

- android/res/layout-v21/abc_screen_toolbar.xml
- android/res/layout-v21/notification_template_icon_group.xml
- android/res/layout-v21/notification_action_tombstone.xml
- android/res/layout-v21/notification_action.xml
- android/res/layout-v21/notification_template_custom_big.xml
- android/res/layout-v17/design_layout_snackbar_include.xml
- android/res/layout-v17/abc_alert_dialog_button_bar_material.xml
- android/res/layout-v17/notification_template_big_media.xml
- android/res/layout-v17/abc_select_dialog_material.xml
- android/res/layout-v17/abc_dialog_title_material.xml
- android/res/layout-v17/abc_search_view.xml
- android/res/layout-v17/notification_action_tombstone.xml
- android/res/layout-v17/notification_action.xml
- android/res/layout-v17/mtrl_layout_snackbar_include.xml
- android/res/layout-v17/select_dialog_singlechoice_material.xml
- android/res/layout-v17/notification_template_media.xml
- android/res/layout-v17/content_tatacara_text_niat.xml
- android/res/layout-v17/abc_tooltip.xml
- android/res/layout-v17/notification_template_lines_media.xml
- android/res/layout-v17/notification_template_big_media_narrow_custom.xml
- android/res/layout-v17/abc_action_mode_close_item_material.xml
- android/res/layout-v17/notification_template_media_custom.xml
- android/res/layout-v17/notification_template_custom_big.xml
- android/res/layout-v17/abc_popup_menu_header_item_layout.xml
- android/res/layout-v17/abc_alert_dialog_title_material.xml
- android/res/layout-v17/notification_template_big_media_custom.xml
- android/res/layout-v17/notification_template_big_media_narrow.xml
- android/res/layout-v17/select_dialog_multichoice_material.xml
- android/res/layout-v17/fragment_kiblat.xml
- android/res/layout-v16/design_bottom_sheet_dialog.xml
- android/res/layout-v16/notification_template_custom_big.xml

There is 'extends ViewGroup' found in file 'com/github/mikephil/charting/charts/Chart.java':

```
line 58: @SuppressLint({"NewApi"})
line 59: public abstract class Chart<T extends ChartData<? extends IDataSet<? extends
Entry>>> extends ViewGroup implements ChartInterface {
line 60:     public static final String LOG_TAG = "MPAndroidChart";
```

There is 'extends RelativeLayout' found in file 'com/github/mikephil/charting/components/MarkerView.java':

```
line 15:
line 16: public class MarkerView extends RelativeLayout implements IMarker {
line 17:     private MPPointF mOffset = new MPPointF();
```

CVSSv3 Base Score:
3.9 (AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:N)

Reference:
- https://developer.android.com/guide/topics/ui/declaring-layout.html
- https://developer.android.com/guide/topics/resources/layout-resource.html
- https://blog.lookout.com/blog/2010/12/09/android-touch-event-hijacking/

USAGE OF INTENT FILTER [M1] [CWE-927] [SAST]

WARNING

Description:
The mobile application uses an intent filter that may be a serious security risk if not properly implemented and filtered. Developers should not solely rely on intent filters for security purposes because they place no restrictions on explicit intents. Intent filters are defined in the Android Manifest file, they let developers choose which type of intents their application components are supposed to receive and handle.

Example of insecure code:

```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.PICK" />
    <category android:name="android.intent.category.DEFAULT" />
    <data mimeType:name="vnd.android.cursor.dir/vnd.google.note" />
</intent-filter>
```

Example of secure code:

```
// When you use intent-filter, you have to perform input validation in your code.
```

Details:
There is '<intent-filter>' found in file 'android/AndroidManifest.xml':

```
line 7:         <activity android:label="@string/app_name"
android:name="org.d3ifcool.jagosholat.views.activities.SplashScreenActivity"
android:noHistory="true" android:theme="@style/AppTheme.NoActionBar">
line 8:             <intent-filter>
line 9:                 <action android:name="android.intent.action.MAIN"/>
```

Reference:
- https://developer.android.com/guide/components/intents-filters.html
- https://developer.android.com/training/articles/security-tips.html

DYNAMIC LOAD OF CODE [M7] [CWE-94] [SAST] WARNING

Description:
The mobile application uses dynamic load of executable code. Under certain circumstances, dynamic load of code can be dangerous. For example, if the code is located on an external storage (e.g. SD card), this can lead to code injection vulnerability if the external storage is world readable and/or writable and an attacker can access it.

Example of insecure code:

```
Object test = loader.loadClass("Test", true).newInstance();
```

Example of secure code:

```
// If you are using code from unsafe place (like external storage),
// you should sign and cryptographically verify your code.
```

Details:
There is 'ClassLoader' found in file 'androidx/versionedparcelable/VersionedParcelParcel.java':

```
line 182:     public <T extends Parcelable> T readParcelable() {
line 183:         return this.mParcel.readParcelable(getClass().getClassLoader());
line 184:     }
```

```
line 186:     public Bundle readBundle() {
line 187:         return this.mParcel.readBundle(getClass().getClassLoader());
line 188:     }
```

There is 'ClassLoader' found in file 'androidx/versionedparcelable/VersionedParcel.java':

```
line 693:         int code = 0;
line 694:         if ((e instanceof Parcelable) && e.getClass().getClassLoader() ==
Parcelable.class.getClassLoader()) {
line 695:             code = EX_PARCELABLE;
```

```
line 1008:                 protected Class<?> resolveClass(ObjectStreamClass osClass)
throws IOException, ClassNotFoundException {
line 1009:                     Class<?> c = Class.forName(osClass.getName(), false,
getClass().getClassLoader());
line 1010:                     if (c != null) {
```

```
line 1032:         try {
line 1033:             return (VersionedParcelable) Class.forName(parcelCls, true,
VersionedParcel.class.getClassLoader()).getDeclaredMethod("read", new Class[]
{VersionedParcel.class}).invoke(null, new Object[]{versionedParcel});
line 1034:         } catch (IllegalAccessException e) {
```

```
line 1069:     private static Class findParcelClass(Class<? extends VersionedParcelable>
cls) throws ClassNotFoundException {
line 1070:         return Class.forName(String.format("%s.%sParcelizer", new Object[]
{cls.getPackage().getName(), cls.getSimpleName()}), false, cls.getClassLoader());
line 1071:     }
```

There is 'ClassLoader' found in file 'com/github/mikephil/charting/data/Entry.java':

```
line 90:         if (in.readInt() == 1) {
line 91:             setData(in.readParcelable(Object.class.getClassLoader()));
line 92:         }
```

Reference:

- https://developer.android.com/reference/java/lang/ClassLoader.html
- https://developer.android.com/reference/dalvik/system/DexClassLoader.html
- https://developer.android.com/reference/java/security/SecureClassLoader.html
- https://developer.android.com/reference/java/net/URLClassLoader.html

MISSING ANTI-EMULATION [SAST]

WARNING

Description:

The mobile application does not use any anti-emulation or anti-debugger techniques (e.g. detecting rooted devices or checking if contacts are authentic).
This can significantly facilitate application debugging and reverse-engineering processes.

Reference:

- https://github.com/strazzere/anti-emulator

NETWORK SECURITY CONFIGURATION IS NOT PRESENT [SAST]

WARNING

Description:

The mobile application does not use Network Security Configuration to define which certificates and Certificate Authorities (CA) can be used for different environments (e.g. Development, Test and Production). The Network Security Configuration on Android feature lets application developers customize their network security settings in a safe, declarative configuration file without modifying the application code.

Reference:

- https://developer.android.com/training/articles/security-config.html