

MALWARE ANALYSIS

TROJAN / ADWARE FAMILIES

Marc Hofmann, Andrea Giuliani, Francesco Rubino

MALWARE SAMPLES

FILE1 - 96c2b215bc929fca8b7651e749d4a6e7

FILE2 - d65dcf5632685db88e2580ea34801d8c

FILE3 - 197548d346bd852724de6e690d502e0b

FILE4 - 0e91ebbcceb761c64d7d7b8bc5889369

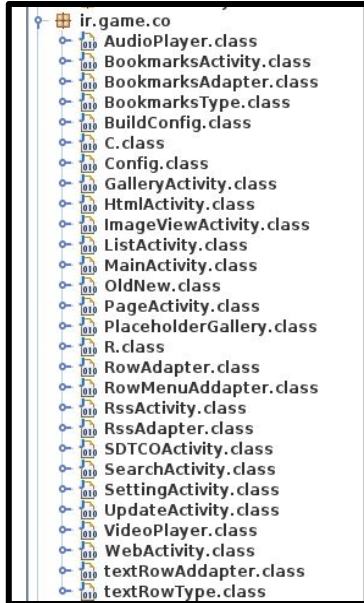
FILE5 - 0289464478c650117ca6d23780583c71

VIRUSTOTAL ANALYSIS

	THREAT CATEGORY	FAMILY CATEGORY	IPs LOCATION	IDS RULES	MITRE	SCORE
F1	TROJAN, PUA, ADWARE	DNOTUA, FRMU, AWAK	US - GB	3 LOW, 1 INFO	FOUND	32/64
F2	TROJAN, ADWARE	SMSREG, ANDR, ROOTNIK	US - CN - BE	2 HIGH, 2 LOW	NOT FOUND	35/64
F2	PUA, TROJAN, VIRUS	SMSREG, SMSPAY, GHHC	CN	NOT FOUND	NOT FOUND	36/58
F3	TROJAN, ADWARE	SMSREG, ANDR. ROOTNIK	CN - US	5 HIGH,	FOUND	35/66
F4	TROJAN, PUA, VIRUS	SMSREG, SMSPAY, ANDR	CN	NOT FOUND	NOT FOUND	33/59

FILE 1 - STATIC ANALYSIS

JD - GUI



- Main directories with critical classes for security
- Most interesting classes: **UpdateActivity.class**, **RSSAdapter.class**



FILE 1 - STATIC ANALYSIS

JD - GUI

```
public class UpdateActivity extends Activity {
    public static UpdateModel U = null;

    DownloadTask d;

    String filename = "";

    public File output_file;

    ProgressDialog p;

    public void Close(View paramView) {
        finish();
    }

    public void Update(View paramView) {
        this.d = new DownloadTask((Context)this);
        AsyncTaskCompat.executeParallel(this.d, (Object[])new String[] { U.UrlAPK });
    }

    void init() {
        TextView textView1 = (TextView)findViewById(2131361810);
        TextView textView2 = (TextView)findViewById(2131361809);
        textView1.setText(U.NewDetails);
        textView2.setText(String.valueOf(getResources().getString(2131230750)) + U.VersionCode);
        this.filename = String.valueOf(Config.getPackageName()) + ".v" + U.VersionCode + ".apk";
        this.p = new ProgressDialog((Context)this);
        this.p.setMessage(String.valueOf(getResources().getString(2131230750)) + U.VersionCode);
        this.p.setIndeterminate(true);
        this.p.setProgressStyle(1);
        this.p.setCancelable(true);
        this.p.setOnCancelListener(new DialogInterface.OnCancelListener() {
            public void onCancel(DialogInterface paramDialogInterface) {
                if (UpdateActivity.this.d != null)
                    UpdateActivity.this.d.cancel(true);
            }
        });
    }
}
```

```
package com.SDTCOSTyle.Layers;

import org.json.JSONException;
import org.json.JSONObject;

public class UpdateModel {
    public boolean HasError = false;

    public String NewDetails = "";

    public String UrlAPK = "";

    public int VersionCode = 1;

    public UpdateModel(int paramInt, String paramString1, String paramString2) {
        this.UrlAPK = paramString1;
        this.NewDetails = paramString2;
        this.VersionCode = paramInt;
    }

    public UpdateModel(String paramString) {
        try {
            JSONObject jsonObject = new JSONObject(paramString);
            this.VersionCode = jsonObject.getInt("VersionCode");
            this.UrlAPK = jsonObject.getString("UrlAPK");
            this.NewDetails = jsonObject.getString("NewDetails");
            return;
        } catch (JSONException jSONException) {
            this.HasError = true;
            jSONException.printStackTrace();
            return;
        }
    }
}
```

Update function:

- Takes update parameters from a JSONObject and has capabilities to install new APKs
- Appears to receive JSON via an RSS feed
- Risky, as app can be changed without play store or even entirely new APK can be installed

FILE 1 - STATIC ANALYSIS

JD - GUI

```
package com.SOTCOM.RSS;

import android.content.Context;
import android.content.SharedPreferences;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import org.json.JSONArray;
import org.json.JSONObject;

public class SaveRss {
    private static String CASH_NAME = "RSSCash";

    public static boolean ExistRss(Context paramContext, String paramString) {
        try {
            return paramContext.getSharedPreferences(CASH_NAME, 0).contains(paramString);
        } catch (Exception exception) {
            return false;
        }
    }

    private static String ListRssToJSON(List<RSSItem> paramList) {
        JSONArray jsonArray = new JSONArray();
        Iterator<RSSItem> iterator = paramList.iterator();
        while (true) {
            if (iterator.hasNext())
                return jsonArray.toString();
            RSSItem rssiItem = iterator.next();
            JSONObject jsonObject = new JSONObject();
            try {
                jsonObject.put("key", "JoApp");
                jsonObject.put("Content", rssiItem.getContent());
                jsonObject.put("Date", rssiItem.getDate());
                jsonObject.put("Description", rssiItem.getDescription());
                jsonObject.put("Link", rssiItem.getLink());
                jsonObject.put("Title", rssiItem.getTitle());
            } catch (Exception exception) {}
            jsonArray.put(jsonObject);
        }
    }

    public static ArrayList<RSSItem> ParseRssJSON(Context paramContext, String paramString) {
        ArrayList<RSSItem> arrayList = new ArrayList();
        if (paramString.length() == 0)
            return null;
        try {
            JSONArray jsonArray = new JSONArray(paramString);
            int i = 0;
            while (true) {
                JSONObject jsonObject;
                ArrayList<RSSItem> arrayList1 = arrayList;
                if (i == jsonArray.length()) {
                    jsonObject = jsonArray.getJSONObject(i);
                    RSSItem rssiItem = new RSSItem();
                    if (!jsonObject.isNull("Content"))
                        rssiItem.setContent(jsonObject.getString("Content").toString());
                    if (!jsonObject.isNull("Date"))
                        rssiItem.getDate(jsonObject.getString("Date").toString());
                    if (!jsonObject.isNull("Description"))
                        rssiItem.getDescription(jsonObject.getString("Description").toString());
                    if (!jsonObject.isNull("Link"))
                        rssiItem.getLink(jsonObject.getString("Link").toString());
                    if (!jsonObject.isNull("Title"))
                        rssiItem.getTitle(jsonObject.getString("Title").toString());
                    arrayList1.add(rssiItem);
                    i++;
                    continue;
                }
                return (ArrayList<RSSItem>)jsonObject;
            }
        } catch (Exception exception) {
            return null;
        }
    }
}
```

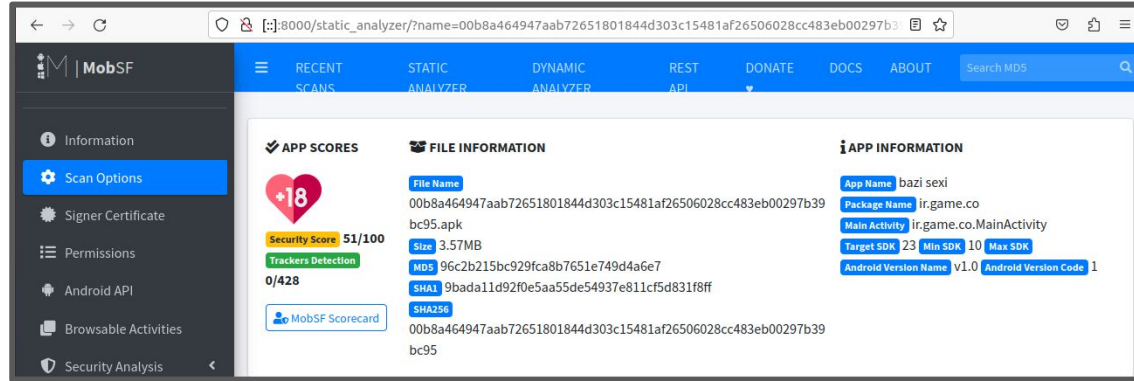
```
if (isOnline()) {
    final WebView webView = new WebView(context);
    WebSettings webSettings = webView.getSettings();
    webView.getSettings().setJavaScriptEnabled(true);
    webSettings.setDomStorageEnabled(true);
    webSettings.setDomStorageEnabled(false);
    webSettings.setAppCacheEnabled(false);
    webSettings.setCacheMode(2);
    if (isAppInstalled("com.farsitel.bazaar")) {
        webView.loadUrl("http://gamejoo.com/tabligh.html?bazaar&pn=ir.game.co&title=&#1576;&#1575;&#1586;&#1740;&#32;&#1607;&#1575;&#1740;&#32;&#1587;&#1600;&#1705;&#1587;&#1740;");
    } else {
        webView.loadUrl("http://gamejoo.com/tabligh.html?pn=ir.game.co&title=&#1576;&#1575;&#1586;&#1740;&#32;&#1607;&#1575;&#1740;&#32;&#1587;&#1600;&#1705;&#1587;&#1740;");
    }
}
webView.getSettings().setAllowFileAccess(true);
webView.setBackgroundColor(0);
webView.setLayoutParams((ViewGroup.LayoutParams)new LinearLayout.LayoutParams(-1, -1));
webView.setWebViewClient(new WebViewClient() {
    boolean offline = true;
});
```

Also appears to have functionality that provides advertising:

- While not itself malicious, happens outside of Google Adsense

FILE 1 - STATIC ANALYSIS

MOBSF



- Overview of MobSF analysis

- Certificate Analysis

CERTIFICATE ANALYSIS		
HIGH 2	WARNING 0	INFO 1
Search: <input type="text"/>		
TITLE	SEVERITY	DESCRIPTION
Application vulnerable to Janus Vulnerability	high	Application is signed with v1 signature scheme, making it vulnerable to Janus vulnerability on Android 5.0-8.0, if signed only with v1 signature scheme. Applications running on Android 5.0-7.0 signed with v1, and v2/v3 scheme is also vulnerable.
Certificate algorithm vulnerable to hash collision	high	Application is signed with SHA1withRSA. SHA1 hash algorithm is known to have collision issues.
Signed Application	Info	Application is signed with a code signing certificate
Showing 1 to 3 of 3 entries		
Previous 1 Next		

FILE 1 - STATIC ANALYSIS

MOBSF

Code Analysis with famous code vulnerabilities

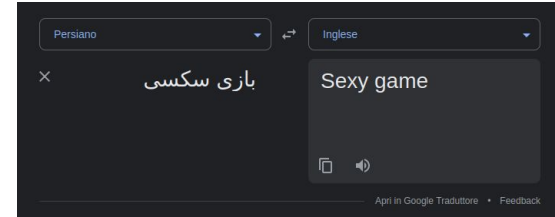
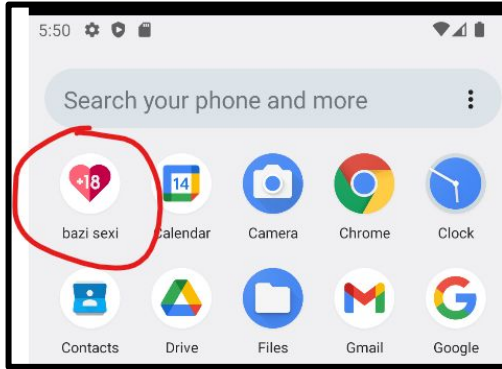
NO ↑↓	ISSUE ↑↓	SEVERITY ↑↓	STANDARDS ↑↓	FILES ↑↓	OPTIONS ↑↓
1	The App logs information. Sensitive information should never be logged.	Info	CWE: CWE-532: Insertion of Sensitive Information into Log File OWASP MASVS: MSTG-STORAGE-3	com/bumptech/glide/Glide.java com/bumptech/glide/disklruCache/DiskLruCache.java com/bumptech/glide/gifdecoder/GifDecoder.java com/bumptech/glide/gifdecoder/GifHeaderParser.java com/bumptech/glide/gifencoder/AnimatedGifEncoder.java com/bumptech/glide/load/data/AssetPathFetcher.java com/bumptech/glide/load/data/HttpUrlFetcher.java com/bumptech/glide/load/data/NetworkFetcher.java	

2	This App uses SSL certificate pinning to detect or prevent MITM attacks in secure communication channel.	secure	OWASP MASVS: MSTG-NETWORK-4	com/SDTCOM/RSS/SimpleFeedParser.java	
3	App can read/write to External Storage. Any App can read data written to External Storage.	warning	CWE: CWE-276: Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	rapid/decoder/cache/DiskLruCache.java	

4	MD5 is a weak hash known to have hash collisions.	warning	CWE: CWE-327: Use of a Broken or Risky Cryptographic Algorithm OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-4	rapid/decoder/cache/DiskLruCache.java	
5	Files may contain hardcoded sensitive information like usernames, passwords, keys etc.	warning	CWE: CWE-312: Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14	com/bumptech/glide/load/engine/EngineKey.java	

FILE 1 - DYNAMIC ANALYSIS

ANDROID STUDIO



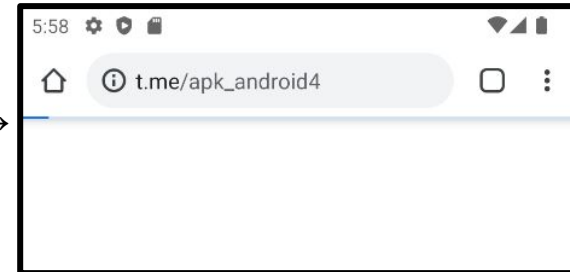
- Installed app in emulator, we find that is called "bazi sexy" (translated from Persian means "sexy game")



Explains the software

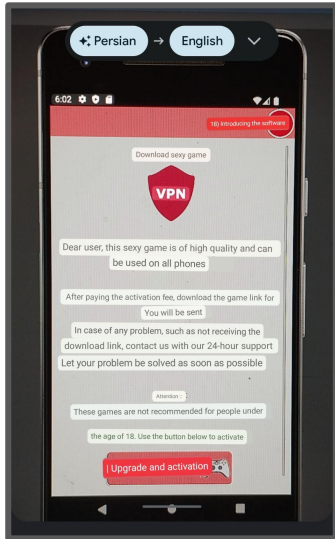
Opens browser to t.me
Telegram application

Purchase and activate



FILE 1 - DYNAMIC ANALYSIS

ANDROID STUDIO



Described how the game can be purchased

- Payment appears to happen outside play store
- Download link for new APK will be send
- Bit sketchy

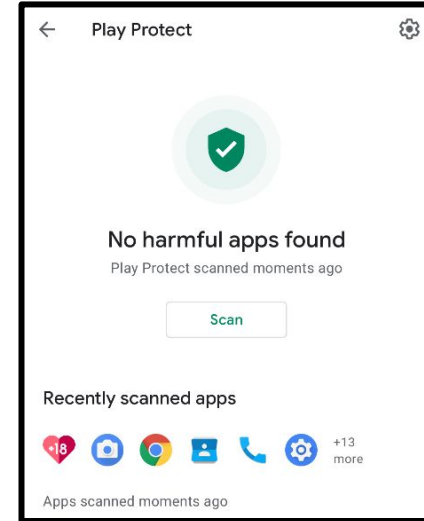
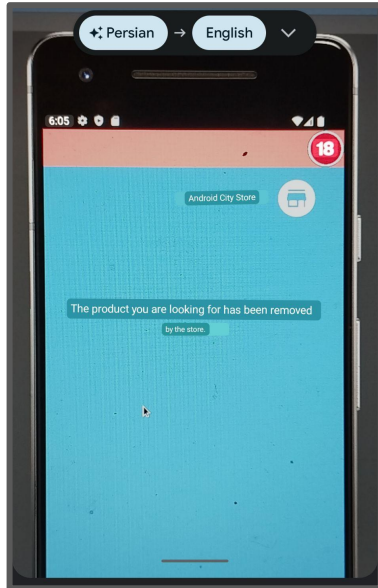
21686	64.336094284	10.0.2.15	142.251.209.10	TCP	76 57662 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2767755850 TSecr=0 WS=128
21687	64.352305526	142.251.209.10	10.0.2.15	TCP	62 443 → 57662 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
21688	64.352341722	10.0.2.15	142.251.209.10	TCP	56 57662 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
21713	64.441253056	10.0.2.15	142.251.209.10	TLSv1.2	230 Client Hello
21728	64.466387509	142.251.209.10	10.0.2.15	TCP	62 443 → 57662 [ACK] Seq=1 Ack=175 Win=32594 Len=0
21729	64.473272127	142.251.209.10	10.0.2.15	TLSv1.2	2976 Server Hello
21730	64.473291112	10.0.2.15	142.251.209.10	TCP	56 57662 → 443 [ACK] Seq=175 Ack=2921 Win=62780 Len=0
21731	64.473455051	142.251.209.10	10.0.2.15	TCP	1516 443 → 57662 [ACK] Seq=2921 Ack=175 Win=32594 Len=1460 [TCP segment of a reassembled PDU]
21732	64.473463791	10.0.2.15	142.251.209.10	TCP	56 57662 → 443 [ACK] Seq=175 Ack=4381 Win=62780 Len=0
21733	64.473586763	142.251.209.10	10.0.2.15	TLSv1.2	277 Certificate, Server Key Exchange, Server Hello Done
21734	64.473606094	10.0.2.15	142.251.209.10	TCP	56 57662 → 443 [ACK] Seq=175 Ack=4602 Win=62780 Len=0
21735	64.593388355	10.0.2.15	142.251.31.188	TCP	76 45260 → 5228 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1564234551 TSecr=0 WS=128
21736	64.627398748	142.251.31.188	10.0.2.15	TCP	62 5228 → 45260 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
21737	64.627346285	10.0.2.15	142.251.31.188	TCP	56 45260 → 5228 [ACK] Seq=1 Ack=1 Win=64240 Len=0
21738	64.642852806	10.0.2.15	142.251.31.188	TLSv1.3	573 Client Hello
21739	64.647573484	10.0.2.15	142.251.209.10	TLSv1.2	149 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
21740	64.663565758	142.251.209.10	10.0.2.15	TLSv1.2	420 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message, Application Data
21741	64.663601954	10.0.2.15	142.251.209.10	TCP	56 57662 → 443 [ACK] Seq=268 Ack=4966 Win=62780 Len=0
21742	64.676785855	142.251.31.188	10.0.2.15	TLSv1.3	2976 Server Hello, Change Cipher Spec
21743	64.676818773	10.0.2.15	142.251.31.188	TCP	56 45260 → 5228 [ACK] Seq=518 Ack=2921 Win=62780 Len=0
21744	64.676960625	142.251.31.188	10.0.2.15	TCP	1372 5228 → 45260 [PSH, ACK] Seq=2921 Ack=518 Win=32251 Len=1316 [TCP segment of a reassembled PDU]
21745	64.676973074	10.0.2.15	142.251.31.188	TCP	56 45260 → 5228 [ACK] Seq=518 Ack=4237 Win=62780 Len=0
21746	64.677951550	142.251.31.188	10.0.2.15	TCP	1516 5228 → 45260 [ACK] Seq=4237 Ack=518 Win=32251 Len=1460 [TCP segment of a reassembled PDU]

FILE 1 - DYNAMIC ANALYSIS

ANDROID STUDIO

If we click on pay and activate:

- Product has been removed by the store
- Store appears to be called Android City Store, but did not find it online



Google Play Protect does not recognize the app as malicious

FILE 1 - DYNAMIC ANALYSIS

MOBSF

To understand how the malware interact with the outside we dynamically analyzed it using the HTTP Tool of MobSF to capture the traffic generated:

- interaction with a server in San Francisco;
- interaction with a server of Telegram in England;

DOMAIN MALWARE CHECK

DOMAIN	STATUS	GEOLOCATION
forush.co	good	IP: 104.21.51.24 Country: United States of America Region: California City: San Francisco Latitude: 37.775700 Longitude: -122.395203 View: Google Map
t.me	good	IP: 149.154.167.99 Country: United Kingdom of Great Britain and Northern Ireland Region: England City: Lowestoft Latitude: 52.475201 Longitude: 1.751590 View: Google Map

SERVER LOCATIONS



CAPTURED TRAFFIC

🔗 <https://forush.co>

- 📄 GET /31875/991366/
- 📄 GET /static/js/popup.js?v=dddbbc344a5a895944e2da7891da9a2a
- 📄 GET /static/js/product.js?v=5915901a8d6f16ba23a67e0b82feb8ad
- 📄 GET /static/img/defaults/store-blue.svg?v=fa82294f9705c1e38fb1504f7c30d42d
- 📄 GET /static/img/forush-logo/favicon-16x16.png?v=bb040faf184350abaf69154c35038320
- 📄 GET /static/img/forush-logo/favicon-96x96.png?v=9f9f9481d0fafd8c02f3d81579a8a9ba
- 📄 GET /static/img/forush-logo/android-icon-192x192.png?v=c149c72785c4c4e9f49485c0ae211eed
- 📄 GET /static/img/forush-logo/favicon-32x32.png?v=480bec9b58cbaa9537af30c8e3255e7e
- 📄 GET /31875/991366/

🔗 <https://t.me>

- 📄 GET /apk_android4

FILE 1 - DYNAMIC ANALYSIS

MOBSF

Other information found during the analysis are:

- Interaction with internal databases used to gather information about users;
- Tests about TLS/SSL connections resulted into TLS Pinning and Certification Transparency vulnerabilities.

TESTS	RESULT
TLS Misconfiguration Test	✓
TLS Pinning/Certificate Transparency Test	✗
TLS Pinning/Certificate Transparency Bypass Test	✗
Cleartext Traffic Test	✓

 **SQLITE DATABASE**

FILES

[data/data/ir.game.co/app_webview/Cookies](#)

[data/data/ir.game.co/app_webview/Web Data](#)

autofill

name	value	value_lower	date_created	date_last_used	count
------	-------	-------------	--------------	----------------	-------

credit_cards

guid	name_on_card	expiration_month	expiration_year	card_number_encrypted	date_modified	origin	use_count	use_date	billing_address_id
------	--------------	------------------	-----------------	-----------------------	---------------	--------	-----------	----------	--------------------

autofill_profiles

guid	company_name	street_address	dependent_locality	city	state	zipcode	sorting_code	country_code	date_modified	origin	language_code	use_count	use_date	validity_bitfield	is_client_validity_states_update
------	--------------	----------------	--------------------	------	-------	---------	--------------	--------------	---------------	--------	---------------	-----------	----------	-------------------	----------------------------------

autofill_profile_names



guid	first_name	middle_name	last_name	full_name
------	------------	-------------	-----------	-----------

autofill_profile_emails

guid	email
------	-------

OTHER FILES

MOBSF ANALYSIS

Recent Scans					
APP	FILE	TYPE	HASH	SCAN DATE	ACTIONS
 调皮女仆 - 2.9.9 com.ktdvau.myidglux  Static Report Dynamic Report	0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6.apk		47c4957a533f66a0020381a0431da57c	June 16, 2023, 3:17 p.m.	  Diff or Compare Delete Scan
 调皮女仆 - 2.9.9 com.aejpln.duhixqsh  Static Report Dynamic Report	0c40fb505fb96ca9aed220f48a3c6c22318d889efa62bc7aaeee98f3a740afab.apk		f57ccdedeef0f933b60f32a6aec963d4	June 16, 2023, 3:16 p.m.	  Diff or Compare Delete Scan
 调皮女仆 - 2.9.9 com.jfvocq.trjuscni  Static Report Dynamic Report	0c05e5035951e260725d15392c8792a4941f92f868558e8b90b52977d832a70d.apk		0e42593023e52e207886e96f736e41d4	June 16, 2023, 3:15 p.m.	  Diff or Compare Delete Scan
 调皮女仆 - 2.9.9 com.yxfhjo.muagktts  Static Report Dynamic Report	0b41181a6b9c85b8fa5c8e8c836ac24dd6e738a0d843f0b81b46ffe41b925818.apk		b940e276fcd8ccdbe917832219eee9f	June 16, 2023, 3:14 p.m.	  Diff or Compare Delete Scan



MALWARES COMPARISON

MOBSF

APP INFORMATION

	com.ktdvau.myidglux - 2.9.9	com.jfvocq.trjuscncq - 2.9.9
File name	0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6.apk	0c05e5035951e260725d15392c8792a4941f92f868558e8b90b52977d832a70d.apk
MD5	47c4957a533f66a0020381a0431da57c	0e42593023e52e207886e96f736e41d4
Size	6.27MB	6.26MB
Certificate	Subject: C=marc, ST=amrc, L=mamr, O=amrc, OU=marcm, CN=marc	No subject

ICON

com.ktdvau.myidglux - 2.9.9	com.jfvocq.trjuscncq - 2.9.9
	

COMPONENTS

	ACTIVITIES	EXPORTED ACTIVITIES	SERVICES	EXPORTED SERVICES	RECEIVERS	EXPORTED RECEIVERS	PROVIDERS	EXPORTED PROVIDERS
com.ktdvau.myidglux - 2.9.9	7	2	13	3	3	3	0	0
com.jfvocq.trjuscncq - 2.9.9	7	2	14	3	4	4	0	0

MobSF Comparison Tool:

- Malware almost the same
- Just different Names
- Other than that they appear to be identical

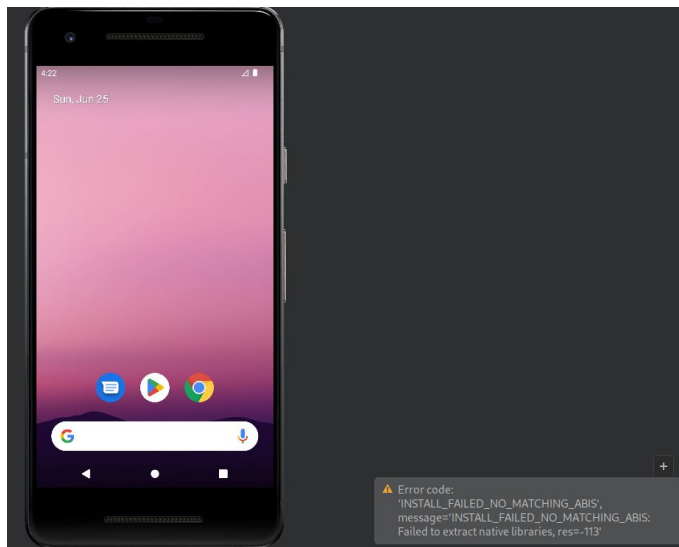
FILE 1 AND FILE 2 COMPARISON

MOBSF

Common	Only in com.yxfhjo.muaqkts - 2.9.9	Only in ir.game.co - v1.0
Java Reflection Get System Service Inter Process Communication HTTP Connection Message Digest Local File I/O Operations Starting Activity Loading Native Code (Shared Library) Query Database of SMS, Contacts etc HTTPS Connection	Base64 Encode Base64 Decode Crypto Send SMS HTTP Requests, Connections and Sessions Execute OS Command Get Subscriber ID Get SIM Serial Number Get WiFi Details Get Network Interface information Get SIM Provider Details Get Phone Number Kill Process Starting Service Sending Broadcast Get Cell Location Dynamic Class and Dexloading TCP Socket Certificate Handling WebView JavaScript Interface Load and Manipulate Dex Files	WebView GET Request Android Notifications

FILE 2 - Overview Static & Dynamic Analysis

- Static Analysis performed on all four files
- Dynamic analysis not possible as, dependencies are missing when building
 - Tries to build and create certificates
 - Errors during the start in emulator



Error

Description

This APK cannot be installed. Is this APK compatible the Android VM/Emulator? adb install failed

FILE 2 - ENCRYPTION FUNCTIONALITY

```
package com.mobile.bumptechnology.ordinary.miniSDK.SDK;

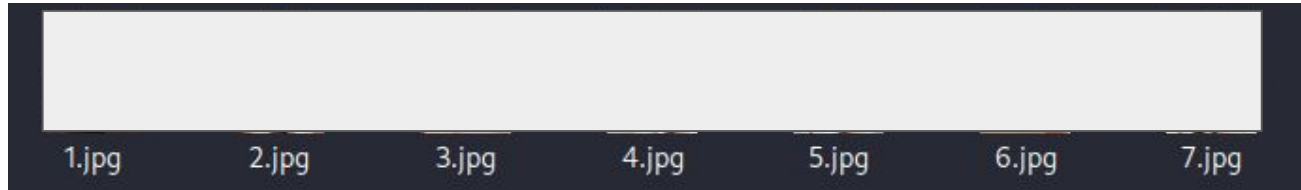
import android.util.Base64;

public final class a {
    public static String a(String paramString) {
        byte[] arrayOfByte;
        if (paramString != null && paramString.length() != 0) {
            byte[] arrayOfByte1 = Base64.decode(paramString, 2);
            if (arrayOfByte1 != null && arrayOfByte1.length != 0) {
                arrayOfByte = new byte[arrayOfByte1.length];
                for (int i = 0; i < arrayOfByte1.length; i++)
                    arrayOfByte[i] = (byte)(arrayOfByte1[i] ^ 0x42);
                return new String(arrayOfByte);
            }
        }
        return (String)arrayOfByte;
    }
}
```

This is an example of crypto functionality:

- It exploits a simple encryption of main strings trying to elude antimalware signature based
- It uses a BASE64 code and a XOR operator with each character and 0x42 hex number.

INSIDE FOLDERS



Appears to be some pornographic video game, with lots of naked girls and mp3 file often called "sexy".

Lots of chinese references

- China Unicom Logo - Could be for payment processing
- Phishing content for phone bills payment



FILE 2 - PERMISSION

MOBSF

android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents
android.permission.READ_PHONE_STATE	dangerous	read phone state and identity
android.permission.READ_EXTERNAL_STORAGE	dangerous	read external storage contents
android.permission.READ_SMS	dangerous	read SMS or MMS
android.permission.RECEIVE_SMS	dangerous	receive SMS
android.permission.SEND_SMS	dangerous	send SMS messages
android.permission.WRITE_SMS	dangerous	edit SMS or MMS
android.permission.MOUNT_UNMOUNT_FILESYSTEMS	dangerous	mount and unmount file systems
android.permission.RECEIVE_USER_PRESENT		Unknown permission
android.permission.GET_TASKS	dangerous	retrieve running applications

android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location
android.permission.CALL_PHONE	dangerous	directly call phone numbers
android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location

android.permission.WRITE_SETTINGS	dangerous	modify global system settings
android.permission.SYSTEM_OVERLAY_WINDOW		Unknown permission
android.permission.MOUNT_FORMAT_FILESYSTEMS	dangerous	format external storage

Manifests are equals with lots of dangerous and unnecessary permission - for a porn app

- Writing settings
- Format Filesystem
- Call and SMS
- Access to location
- Access to external memory

FILE 2 - MANIFEST

```
<receiver android:name="com.mn.kt.rs.RsRe">
  <intent-filter android:priority="2147483647">
    <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    <action android:name="android.intent.action.BATTERY_CHANGED" />
    <action android:name="android.intent.action.SIM_STATE_CHANGED" />
    <action android:name="android.intent.action.NOTIFICATION_ADD" />
    <action android:name="android.intent.action.SERVICE_STATE" />
    <action android:name="android.intent.action.NOTIFICATION_REMOVE" />
    <action android:name="android.intent.action.NOTIFICATION_UPDATE" />
    <action android:name="android.bluetooth.adapter.action.STATE_CHANGED" />
    <action android:name="android.intent.action.ANY_DATA_STATE" />
    <action android:name="android.net.wifi.STATE_CHANGE" />
    <action android:name="android.intent.action.BOOT_COMPLETED" />
    <action android:name="android.intent.action.SCREEN_ON" />
    <action android:name="android.intent.action.USER_PRESENT" />
  </intent-filter>
</receiver>
```

```
<supports-screens android:anyDensity="true" android:largeScreens="true" android:normalScreens="true" android:smallScreens="true" android:xlargeScreens="true" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.WRITE_SMS" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.RECEIVE_USER_PRESENT" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.SYSTEM_OVERLAY_WINDOW" />
<uses-permission android:name="android.permission.MOUNT_FORMAT_FILESYSTEMS" />
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION" />
<uses-permission android:name="android.permission.RUN_INSTRUMENTATION" />
<uses-permission android:name="android.permission.READ_SETTINGS" />
<uses-permission android:name="android.permission.RECEIVE_MMS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.RESTART_PACKAGES" />
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.RECEIVE_WAP_PUSH" />
</manifest>
```