Android Uygulamalarında Güvenlik Testi

written by Mert SARICA | 21 May 2012

Şubat ayında Gartner tarafından yayımlanan bir rapora göre dünya genelinde 2011 yılının 4. çeyreğinde satılan akıllı telefonların %50.9'unda Android işletim sisteminin, %23.8'inde ise IOS işletim sisteminin kurulu olduğunu görüyoruz. Web siteleri, uygulamaları üzerinden vermiş oldukları hizmetlerin tamamına yakınını mobil uygulamalar ile mobil platformlar üzerinden de vermek için uğraş veren irili ufaklı birçok firma, eskiden sadece iOS için uygulama geliştirirken pazar payını bu kadar yükselten Android karşısında kayıtsız kalamayarak Android işletim sistemi için de (misal Instagram) uygulama geliştirmek durumunda kaldılar.

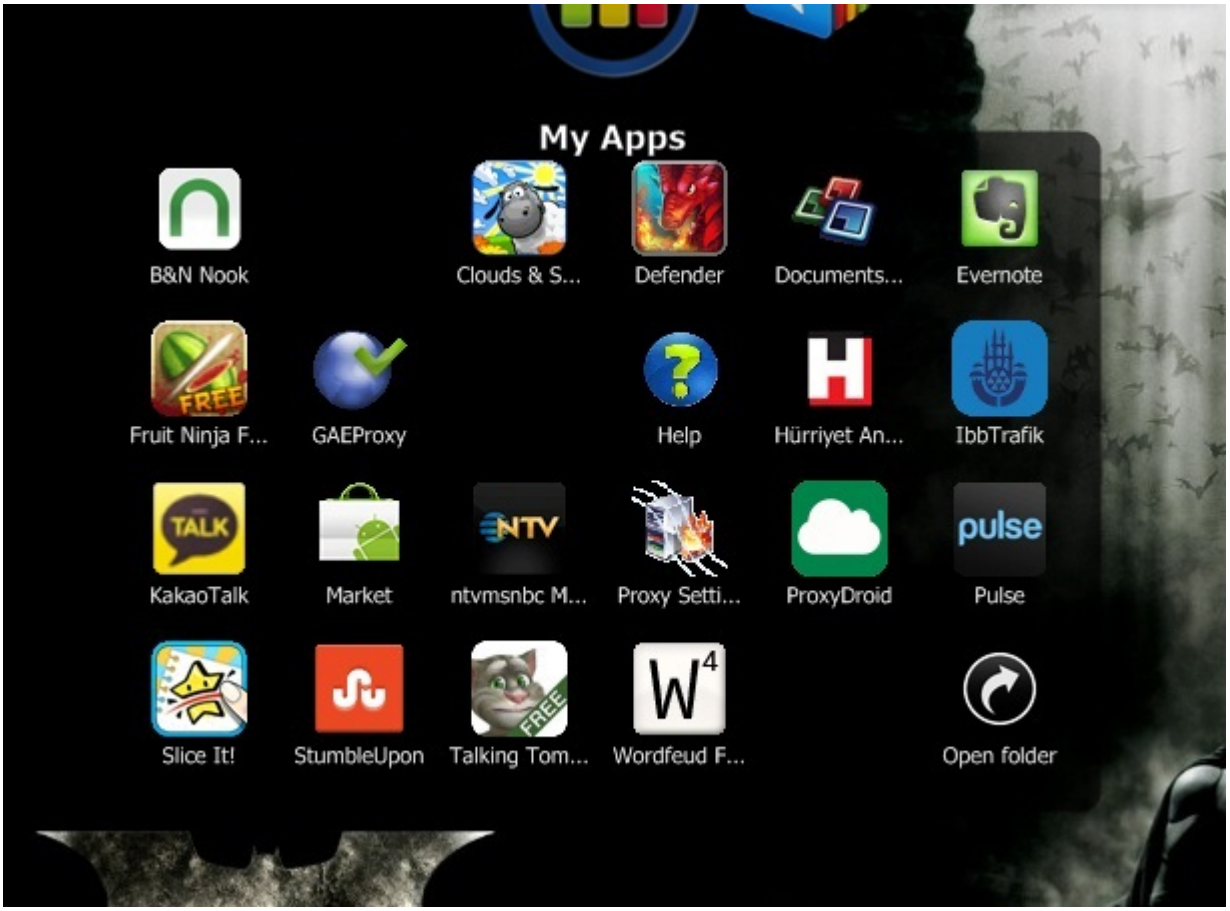
iOS işletim sistemi için geliştirilen uygulamalara kıyasla Android işletim sistemi için geliştirilen uygulamaların kaynak koduna çevrilebilir olması, ücretsiz ve platform bağımsız bir emulator sayesinde daha kolay analiz edilebilir olması beraberinde art niyetli kişilerce de güvenlik zafiyetlerinin daha kolay tespit edilebilmesine ve istismar edilebilmesine imkan tanımaktadır.

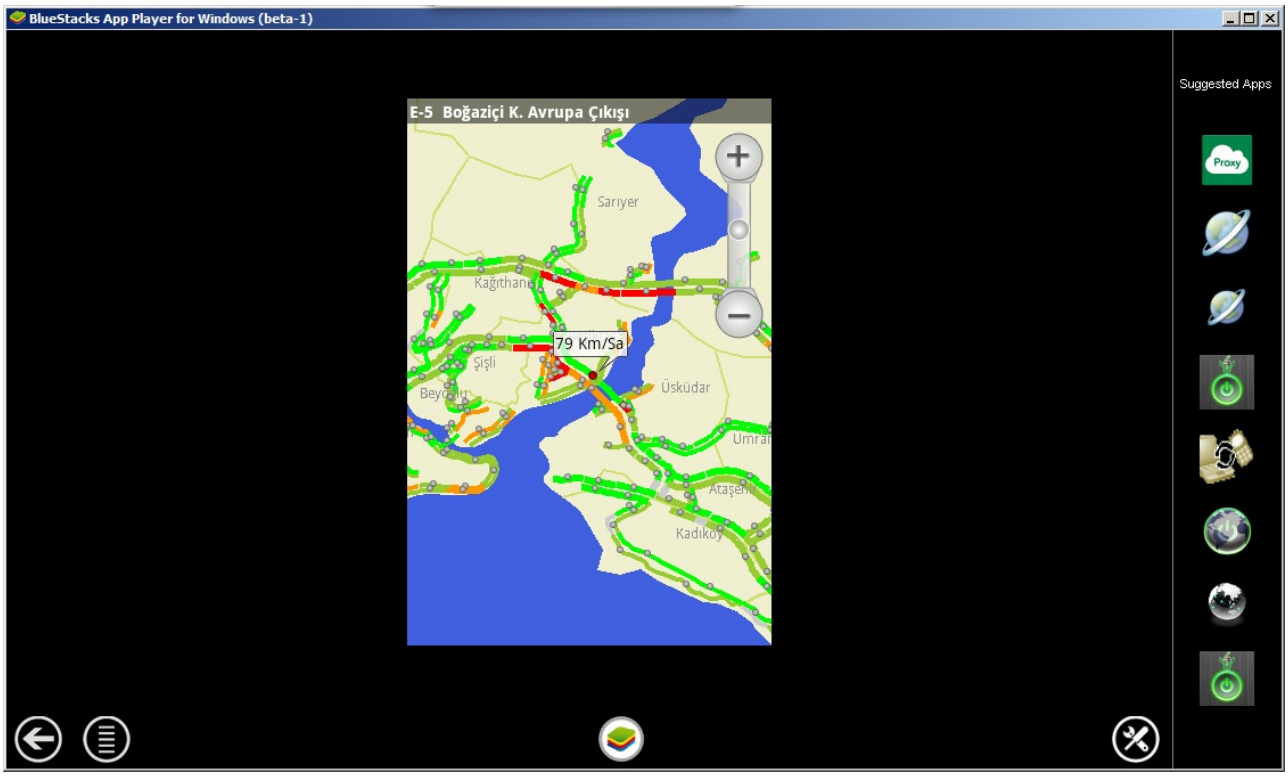
Günümüzde uyumluluktan öte müşteri güvenliği, marka değeri ve itibarı için ağ, sistem, web uygulamalarını vs. güvenlik testine tabi tutan firmalar için mobil uygulamaların da test edilmesi kaçınılmaz olduğu için bu yazımda Android uygulamalarını test ederken güvenlik adına dikkat edilmesi gereken noktalardan (ağ trafiği analizi/manipülasyonu, kaynak kodu incelemesi, tersine çevirme (disassembling), dosya sistemi incelemesi) birkaçına dikkat çekmek istedim.

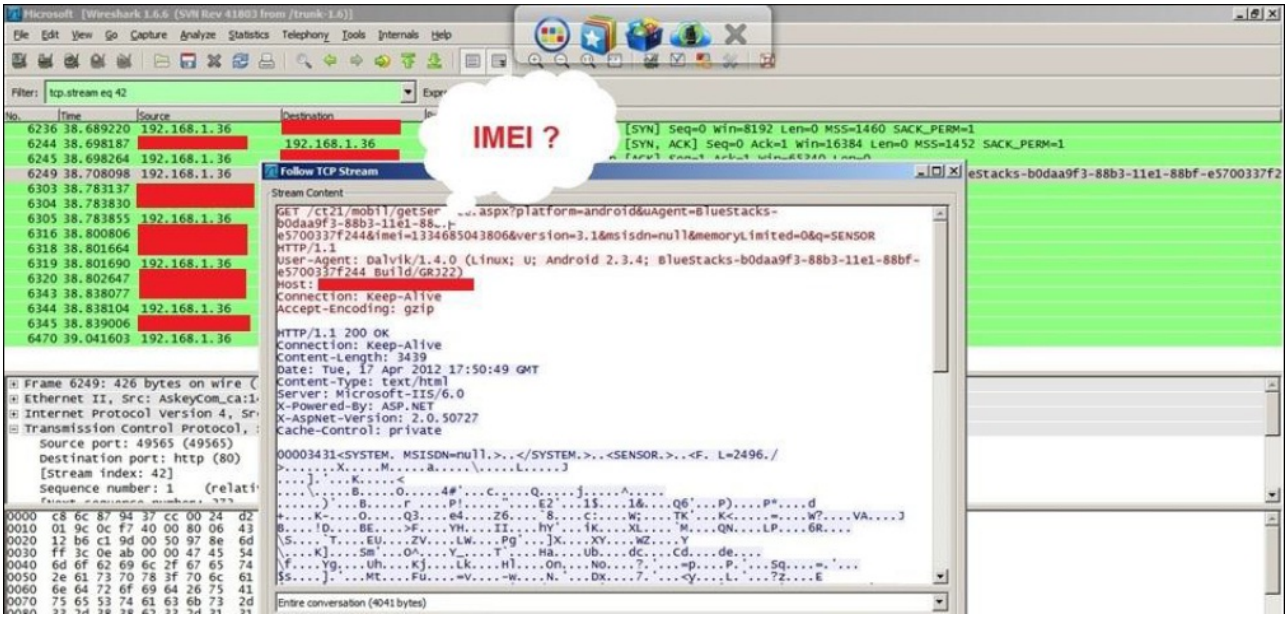
Ağ Trafiği Analizi:

- Şifreli kanal üzerinden (SSL) haberleşmeyen Native (Java ile yazılmış) ve native olmayan uygulamaların trafiğini analiz etmek için hedef uygulamayı Android SDK ile gelen emulatore yine Android SDK ile gelen adb (Android Debug Bridge) aracı ile yüklemek (adb install uygulama.apk) veya BlueStacks'in App Player aracı ile çalıştırmak ve Wireshark aracı ile trafiği analiz etmek mümkündür. (App Player, APK uzantılı tüm Android uygulamalarını Windows üzerinde çalıştırmaya yarayan gerçekten çok faydalı olduğunu düşündüğüm bir araçtır, basit

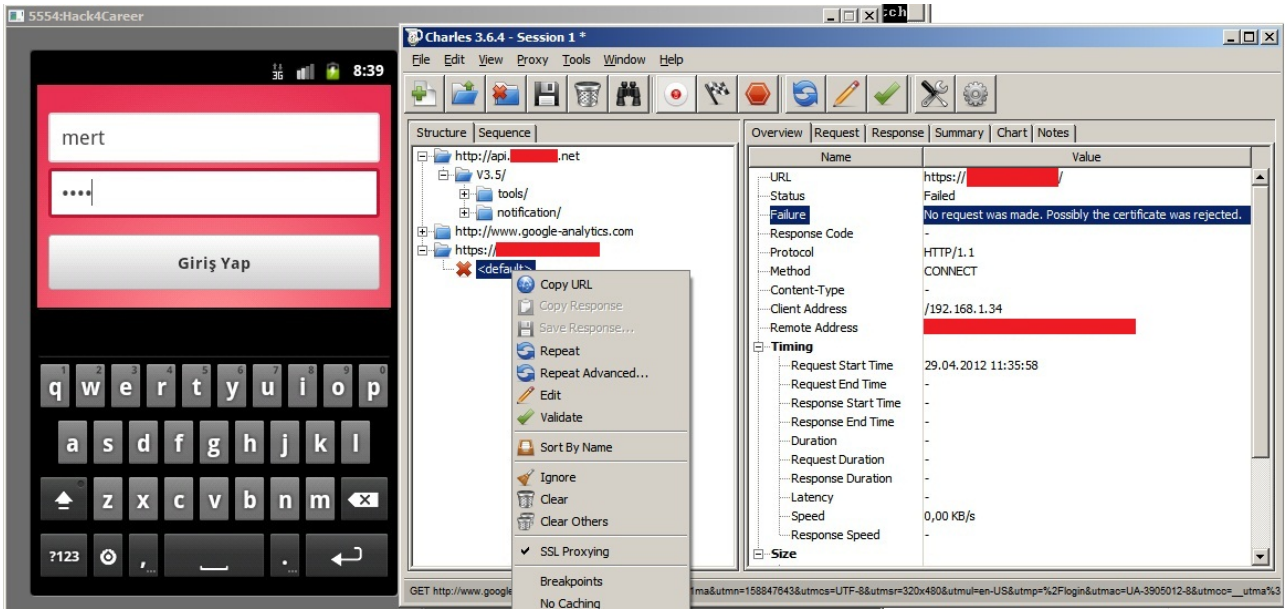
analizler için mutlaka denemenizi tavsiye ederim.)







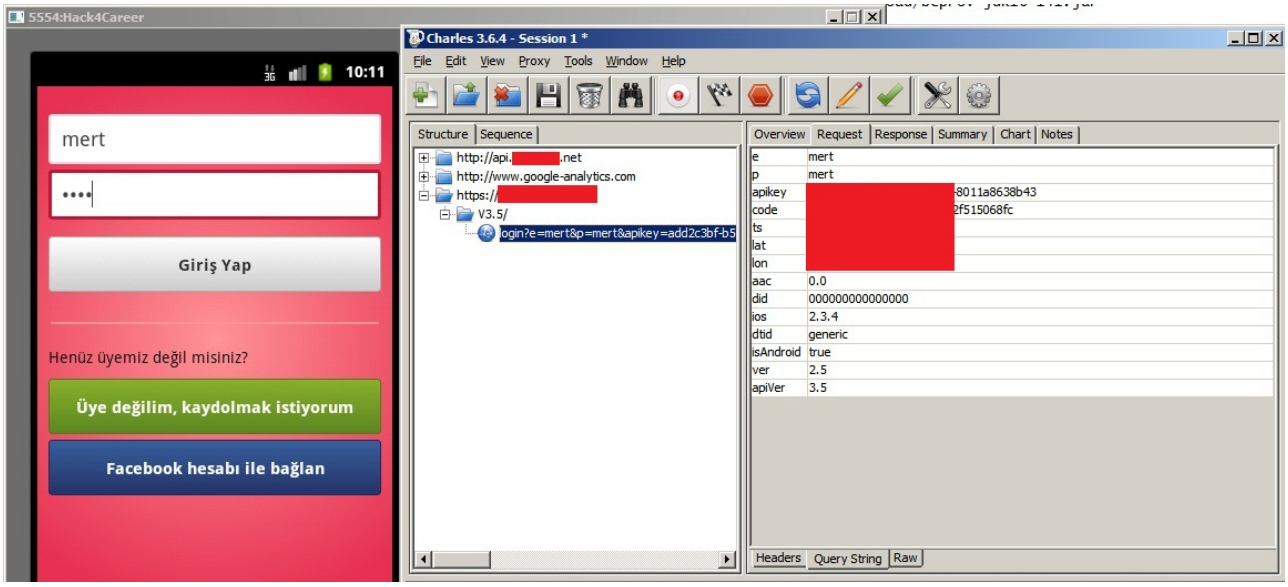
- Eğer ki uygulama native ise ve şifreli kanal üzerinden haberleşiyor ise bu durumda App Player üzerinde çalışan bir Wireshark ile trafiği analiz etmek pek mümkün değildir. Ortadaki adam saldırısını (MITM) gerçekleştirmek için yaratılmış olan Android Virtual Device (AVD)'ın herhangi bir proxy aracına (Burp Suite veya Charles Proxy) yönlendirilmesi durumunda da bu defa native uygulama sertifika hatasını gördüğü anda iletişimi kurmayacaktır. Durum böyle olunca da tek çözüm yolu Charles Proxy aracını kullanmak ve Charles Proxy aracının kök sertifikasını Android işletim sisteminin güvenilir kök sertifikalarına eklemek olacaktır. Bu sayede Charles Proxy emulatör ile web sunucusu arasına girmeye çalıştığı esnada üretmiş olduğu sahte sertifika, native uygulama tarafından reddilmeyecek ve bağlantı kesilmeyecektir. Emulatör yeniden başlatıldığında güvenilir kök sertifikalar eski, orjinal haline döndüğü için de sertifika sisteme yüklendikten sonra sistemin imajı yaffs2 aracı ile alınarak, AVD yeniden başlatıldığında bu sistem imajının kullanılması sağlanarak Charles Proxy aracının kök sertifikasının sistemde kalıcı olması sağlanacaktır. Şifreli trafik analizi için aşağıdaki ekran görüntüsünde belirtilen tüm direktiflerin izlenmesi ve komutların çalıştırılması Charles Proxy ile şifreli trafiğin analiz edilebilmesini sağlayacaktır.


5554:Hack4Career
8:39
mert
Giriş Yap
Charles 3.6.4 - Session 1 *
File Edit View Proxy Tools Window Help
Structure Sequence
http://api.[redacted].net
V3.5/
tools/
notification/
http://www.google-analytics.com
https://[redacted]
Copy URL
Copy Response
Save Response...
Repeat
Repeat Advanced...
Edit
Validate
Sort By Name
Ignore
Clear
Clear Others
SSL Proxying
Breakpoints
No Caching
Overview Request Response Summary Chart Notes
Name
Value
URL
https://[redacted]/
Status
Failed
Failure
No request was made. Possibly the certificate was rejected.
Response Code
-
Protocol
HTTP/1.1
Method
CONNECT
Content-Type
-
Client Address
/192.168.1.34
Remote Address
[redacted]
Timing
Request Start Time
29.04.2012 11:35:58
Request End Time
-
Response Start Time
-
Response End Time
-
Duration
-
Request Duration
-
Response Duration
-
Latency
-
Speed
0,00 KB/s
Response Speed
-
Size

komutlar.txt - Notepad

File Edit Format View Help

```
Yarat: AVD

İndir:   http://bouncycastle.org/download/bcprov-jdk16-141.jar

Kopyala: C:\Program Files\Java\jre6\lib\ext\bcprov-jdk16-141.jar

İndir:   http://www.charlesproxy.com/ssl.zip

Çalıştır:

emulator -avd Hack4Career

adb pull /system/etc/security/cacerts.bks cacerts.bks
keytool -keystore cacerts.bks -storetype BKS -provider
org.bouncycastle.jce.provider.BouncyCastleProvider
-storepass changeit -importcert -trustcacerts -alias somealias -file
charles-proxy-ssl-proxying-certificate.crt
-noprompt

adb remount

adb shell chmod 777 /system/etc/security/cacerts.bks

adb push cacerts.bks /system/etc/security/

İndir: http://android-group-korea.googlecode.com/files/mkfs.yaffs2.arm

Çalıştır:

adb push mkfs.yaffs2.arm /data/data/temp/mkfs.yaffs2

adb shell chmod 777 /data/data/temp/mkfs.yaffs2

adb shell

# /data/data/temp/mkfs.yaffs2 /system /sdcard/system.img
/data/data/temp/mkfs.yaffs2 /system /sdcard/system.img
mkfs.yaffs2: Android YAFFS2 Tool,Build by PowerGUI
            at http://www.openhandsetalliance.org.cn
Building...
Build Ok.
# exit
exit

adb pull /sdcard/system.img system.img

Kopyala: system.img -> C:\Users\Mert\.android\avd\Hack4Career.avd

emulator -avd Hack4Career -http-proxy http://192.168.1.34:8888
```

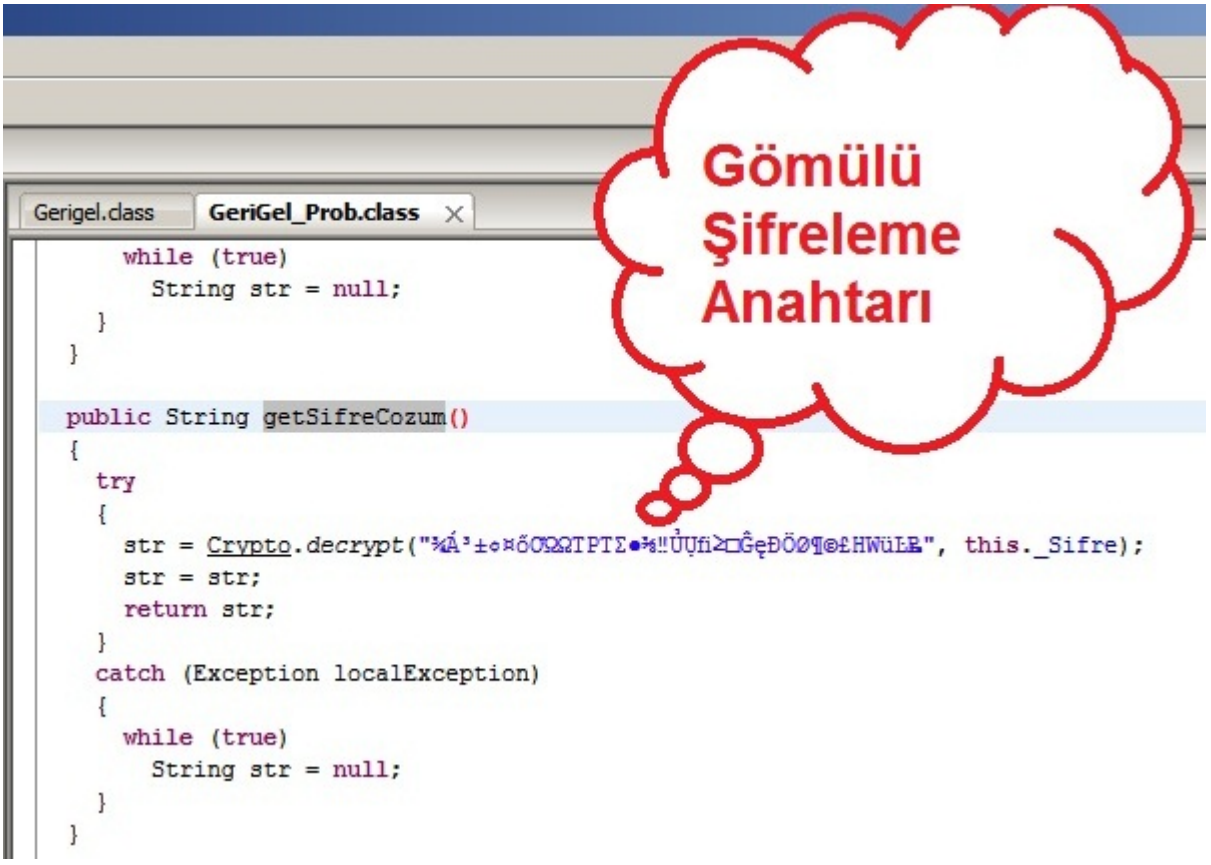


Kaynak Kodu İncelemesi:

- Native uygulamalar yukarıda da belirttiğim gibi çoğunlukla Java programlama dili ile geliştirilmektedir. APK uzantılı Android uygulamasının uzantısı ZIP olarak değiştirilerek Winzip veya Winrar gibi araçlar ile açılabilir, ardından içerisinde yer alan Dex uzantılı yani Dalvik sanal makinesinde çalıştırılabilen Android uygulaması d2j-dex2jar.bat aracı ve d2j-dex2jar.bat classes.dex komutu ile jar dosyasına çevrilebilir. Ardından JD-GUI aracı ile bu jar dosyası kaynak koduna çevirebilir ve incelenebilir.
- Kaynak kodu incelemesi ile kaynak kodu içinde yer alan gömülü şifreleme anahtarları, web servis bilgileri gibi birçok bilgi elde edilebilir.
- Kimi zaman bazı durumlarda kaynak koduna çevirmek istenilen sonuçları üretmez bu gibi durumlarda dex dosyası tersine çevrilerek (disassembling) analiz edilebilir.

UtilityClass.class

```
public static ArrayList<ImkbObject> getHisseList()
{
  dling = true;
  Object localObject1 = new DefaultHttpClient();
  Object localObject2 = new HttpGet("http://iphone.[redacted].com.tr/[redacted]push/Service.asmx/getHisseList");
  ArrayList localArrayList = new ArrayList();
  try
  {
    localObject1 = ((HttpClient)localObject1).execute((HttpUriRequest)localObject2).getEntity();
    int i;
    if (localObject1 != null)
    {
      localObject1 = convertStreamToString(((HttpEntity)localObject1).getContent());
      localObject1 = ((String)localObject1).substring(((String)localObject1).indexOf('A'));
      localObject2 = ((String)localObject1).substring(0, ((String)localObject1).indexOf('<')).split("¦¦");
      i = 0;
      int j = localObject2.length;
      if (i < j);
    }
    else
    {
      label97: dling = false;
      return localArrayList;
    }
    ImkbObject localImkbObject = new ImkbObject();
    String str = localObject2[i].replace('|', ';');
    String[] arrayOfString = str.split(";");
    for (int k = 0; ; k++)
    {
      if (k >= arrayOfString.length)
      {
```

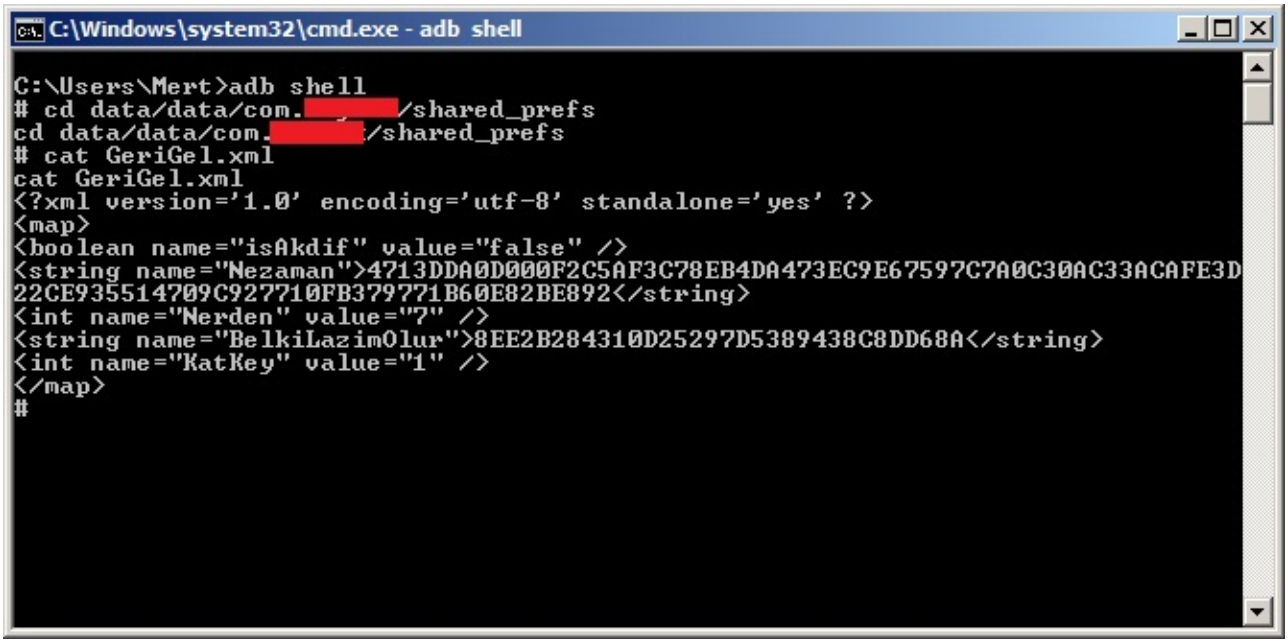


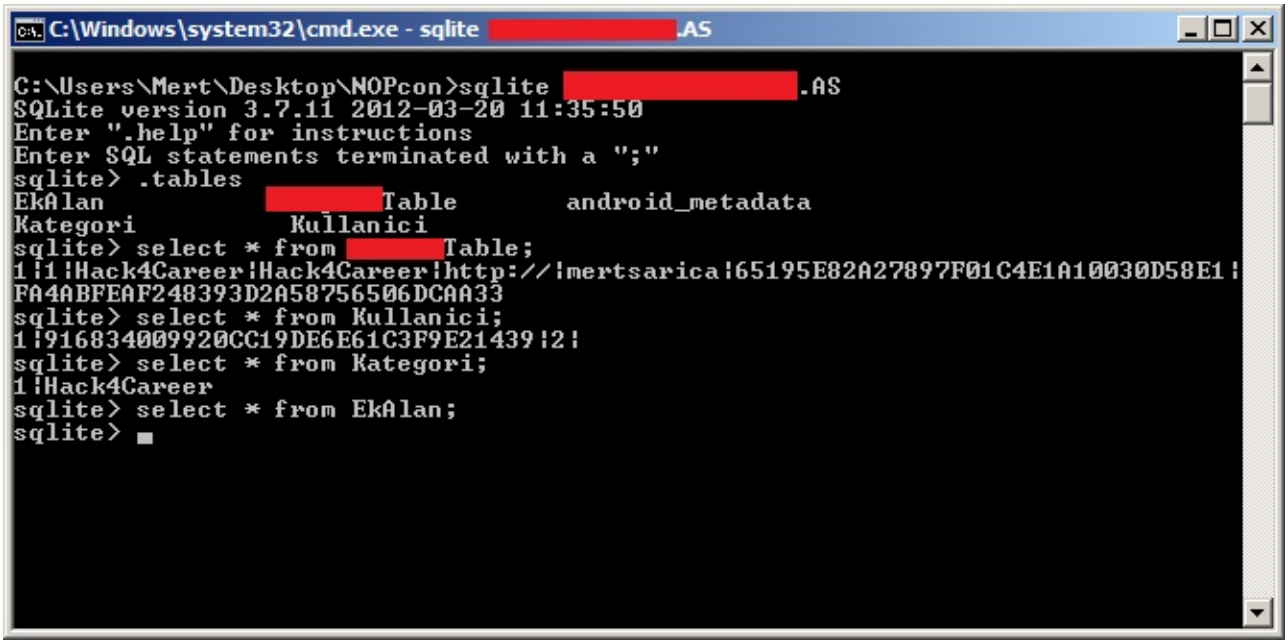
Tersine Çevirme:

- Kaynak koduna çevirmenin yeterli olmadığı durumlarda veya program akışının değiştirilme ihtiyacı olduğu durumlarda dex dosyası android-apktool aracı ve java -jar apktool.jar d Uygulama.apk komutu ile tersine çevrilebilir (disassembling) ardından java -jar apktool.jar b Uygulama Uygulama.apk komutu ile tekrar birleştirilebilir. (assembling)
- Assembly programlama diline hakimseniz bu sayfada yer alan bilgiler ışığında Dalvik op kodlarını analiz edebilir ve değiştirebilirsiniz.

Dosya Sistemi İncelemesi:

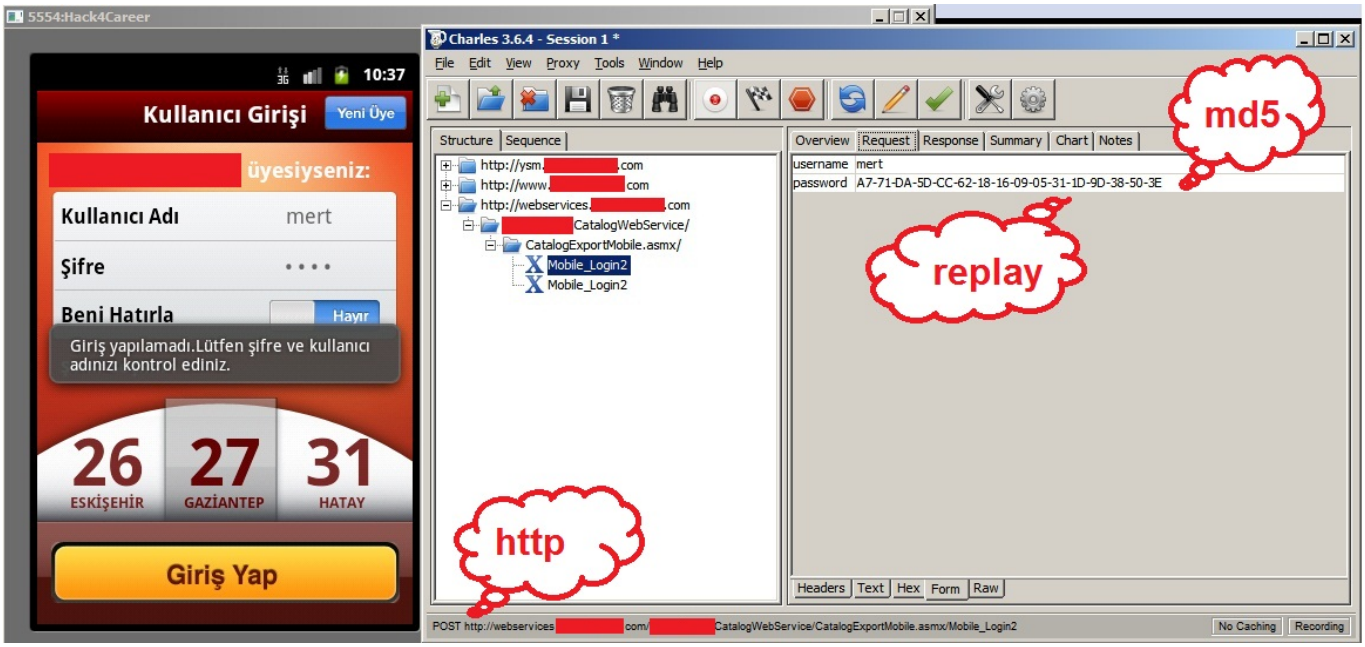
- Android işletim sistemi uygulamalara verilerini saklamak için 4 seçenek sunmaktadır; Dahili Depolama (özel verileri saklamak için), Shared Preferences (basit depolama için), Harici Depolama (özel olmayak büyük veri setlerini saklamak için) ve SQLite veritabanı (yapısal depolama)
- Harici depolama hafıza kartını kullandığı ve buraya kopyalanan her dosya diğer tüm uygulamalar tarafından okunabildiği için buraya yazılan hassas verilerin mutlaka ama mutlaka şifreli olarak yazılması gerekmektedir.



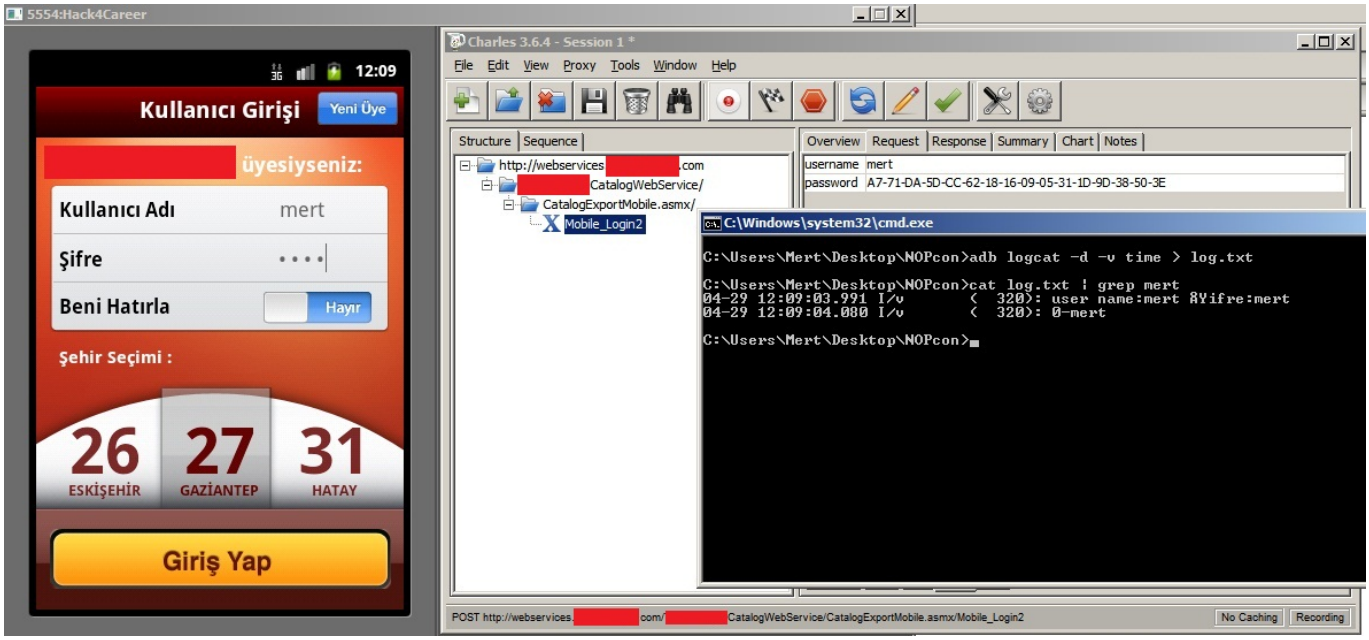


Yukarıda kısaca bahsettiğim adımları izleyerek çok sık kullandığım Android uygulamalarına kısaca göz attığımda şans eseri bir kaç güvenlik zafiyeti ile karşılaştım.

İlk uygulamanın kullanıcı tarafından girilen kullanıcı adı ve şifreyi, şifreli kanal (SSL) üzerinden gerçekleştirmediğini farkettim. Her ne kadar sunucuya gönderilen şifre, MD5 ile hashlenmiş olsa da tuz değeri (salt) kullanılmadığı için üretilen MD5 olduğu gibi kullanılabilmektedir. Durum böyle olunca da kablosuz ağ üzerinde ARP zehirlemesi gerçekleştirerek trafiğinizi kayıt eden art niyetli bir kişi hesabınıza bu bilgiler ile yetkisiz olarak erişebilir.



Ayrıca yine aynı uygulamanın işletim sistemi kayıtlarına uygulamaya girmek için kullanılan şifreyi açık (clear) olarak yazdığını farkettim.



Tüm şifrelerinizi ana bir şifre ile AES ile güvenli bir şekilde dosya sistemi üzerinde saklayan başka bir uygulamanın ana şifrenin bir kopyasını uygulama içinde gömülü olan başka bir şifre ile şifreleyerek dosya sistemi üzerine kayıt ettiğini farkettim. Durum böle olunca da bu gömülü şifre ile ana şifreyi deşifre etmek, ardından bu ana şifre ile de şifrelenmiş diğer tüm şifreleri deşifre etmek mümkündür.

OVERVIEW | USER REVIEWS | WHAT'S NEW | PERMISSIONS

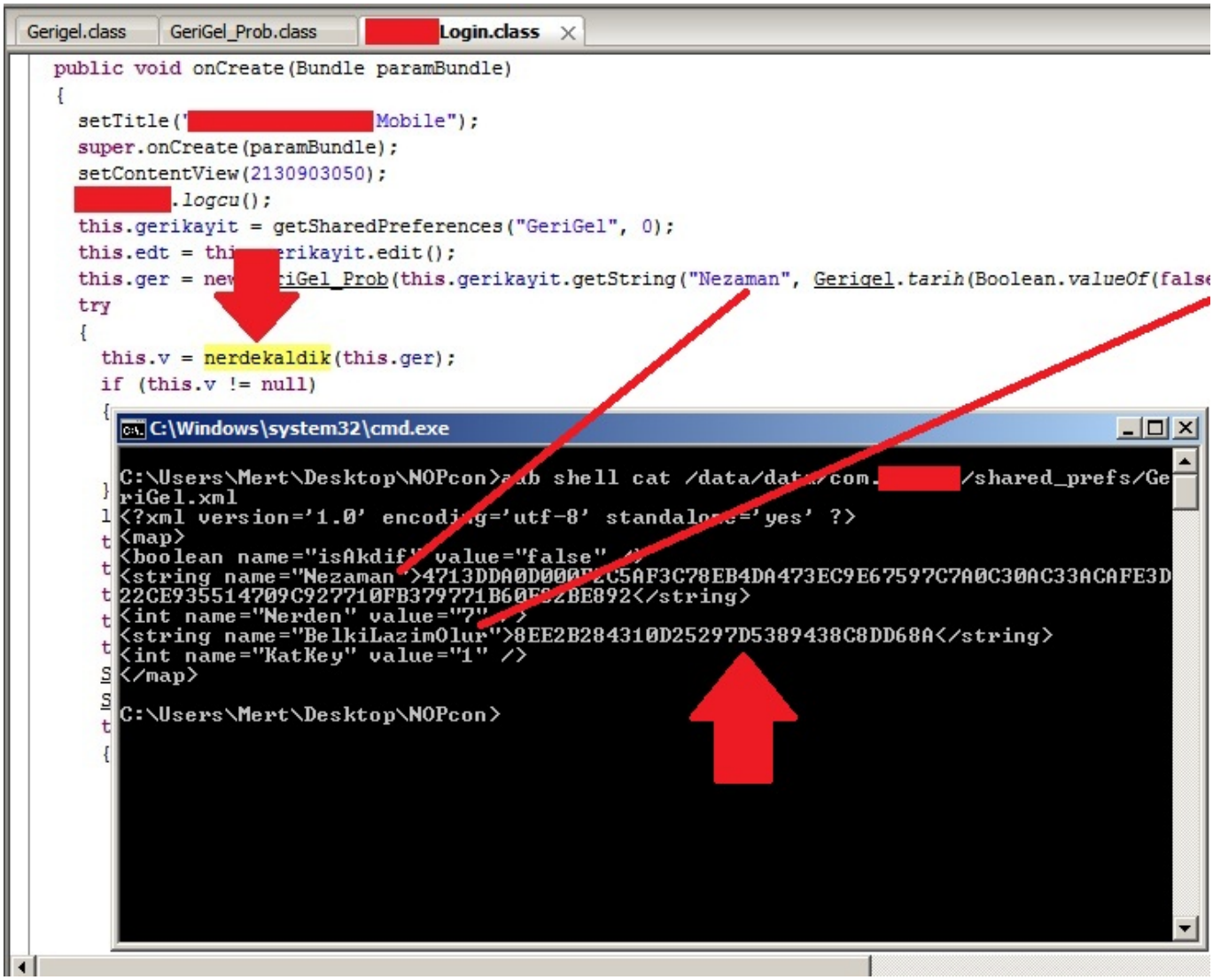
Description

[redacted] stores your passwords securely. All passwords protected by master password and access them all with single password. It has user friendly interface and easy to use. It uses the AES encryption for store your passwords. You can copy and paste your password whereever you want.

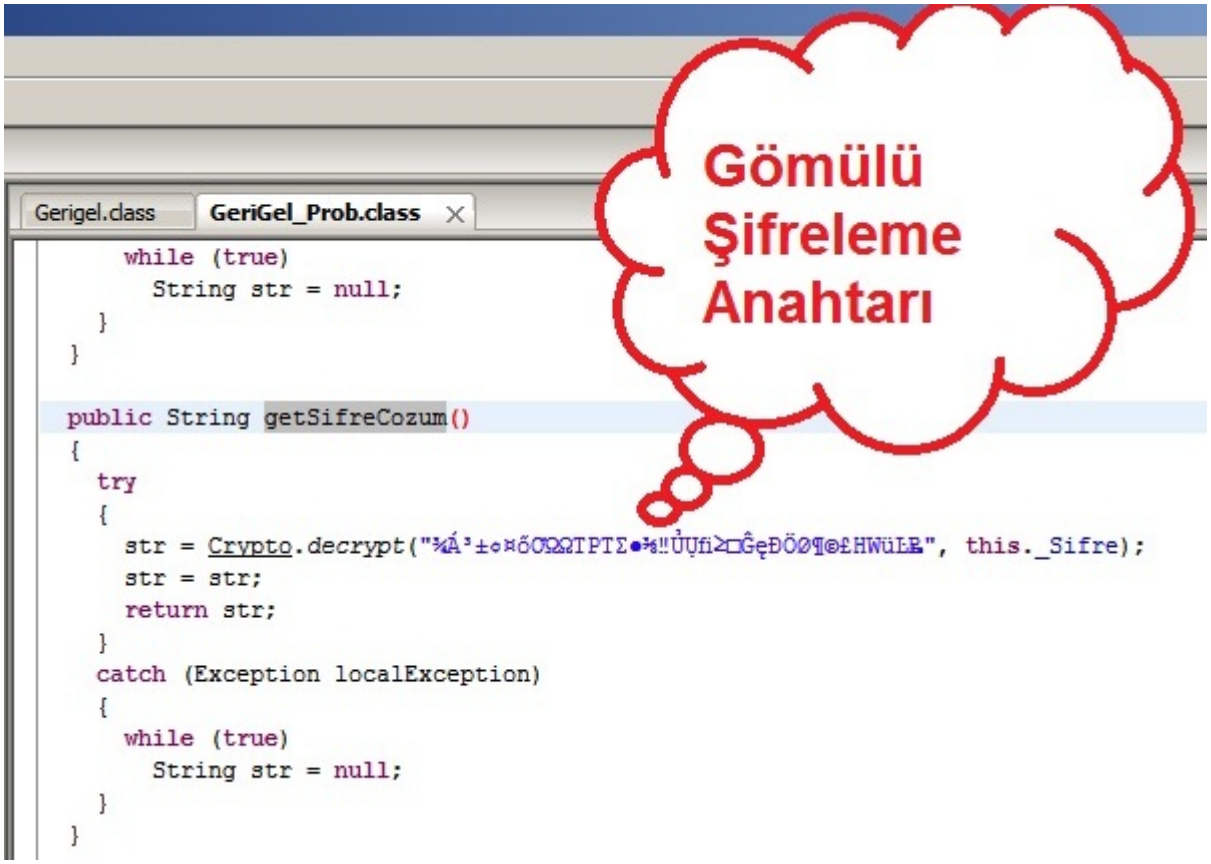
[redacted] tüm şifrelerinizi güvenli bir şekilde saklar. Tüm şifrelerinize sadece bir ana şifre ile erişirsiniz. Kullanıcı dostu bir arayüze sahiptir ve kullanımı kolaydır. Gelişmiş bir şifreleme yöntemi olan AES şifreleme sistemini kullanır. Şifrelerinizi ve diğer bilgileri kopyalayıp istediğiniz uygulamaya yapıştırabilirsiniz.

Visit Developer's Website › Email Developer ›

App Screenshots


Gerigel.class
GeriGel_Prob.class
[redacted]Login.class
public void onCreate(Bundle paramBundle)
{
setTitle("[redacted]Mobile");
super.onCreate(paramBundle);
setContentView(2130903050);
[redacted].logcu();
this.gerikayit = getSharedPreferences("GeriGel", 0);
this.edt = thi[redacted]erikayit.edit();
this.ger = new [redacted]riGel_Prob(this.gerikayit.getString("Nezaman", Gerigel.tarih(Boolean.valueOf(false
try
{
this.v = nerdekaldik(this.ger);
if (this.v != null)
C:\Windows\system32\cmd.exe
C:\Users\Mert\Desktop\NOPcon>adb shell cat /data/data/com.[redacted]/shared_prefs/GeriGel.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<boolean name="isAkdif" value="false" />
<string name="Nezaman">4713DDA0D000[redacted]C5AF3C78EB4DA473EC9E67597C7A0C30AC33ACAFE3D22CE935514709C927710FB379771B60[redacted]2BE892</string>
<int name="Nerden" value="7" />
<string name="BelkiLazimOlur">8EE2B284310D25297D5389438C8DD68A</string>
<int name="KatKey" value="1" />
</map>
C:\Users\Mert\Desktop\NOPcon>


Gerigel.class
GeriGel_Prob.class
[redacted]Login.class
private Boolean nerdekaldik(GeriGel_Prob paramGeriGel_Prob)
{
Boolean localBoolean;
try
{
Statix.Criptorix = paramGeriGel_Prob.getSifreCozum();
UserMainMetot.Giris(this, paramGeriGel_Prob.getSifreCozum());
localBoolean = new Gerigel().ZamanNeAlemde(paramGeriGel_Prob, this);
if (localBoolean == null)
{
UserMainMetot.GirisYapti = null;
localBoolean = null;
}
}
catch (Exception localException)
{
UserMainMetot.GirisYapti = null;
localBoolean = null;
}
return localBoolean;
}



Sonuç olarak Google Play'e yüklemiş olduğunuz tüm mobil uygulamalarınız art niyetli kişiler tarafından indirilebilir ve analiz edilebilir olduğun unutmayın bu nedenle mobil uygulamalarınızı hizmete sunmadan önce mutlaka ama mutlaka güvenlik testinden geçirin veya geçirtin.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.