

btrisk

BİLGİ GÜVENLİĞİ VE BT YÖNETİŞİM HİZMETLERİ

SSL PINNING ATLATMA

İsmail Önder Kaya
OSCP



İçindekiler

I. GİRİŞ.....	2
II. ANDROİD'DE FRIDA KULLANARAK SSL PINNING ATLATMA.....	3
III.ANDROID'DE XPOSED FRAMEWORK KULLANARAK SSL PINNING ATLATMA.....	17
IV. IOS'DA SSL KILL SWITCH KULLANARAK SSL PINNING ATLATMA.....	27
V. IOS'DA BURP SUITE MOBILE ASSISTANT KULLANARAK SSL PINNING ATLATMA.....	39
VI. IOS'DA OBJECTİON KULLANARAK SSL PINNING ATLATMA.....	54
VII. BTRİSK Hakkında	62

I. GİRİŞ

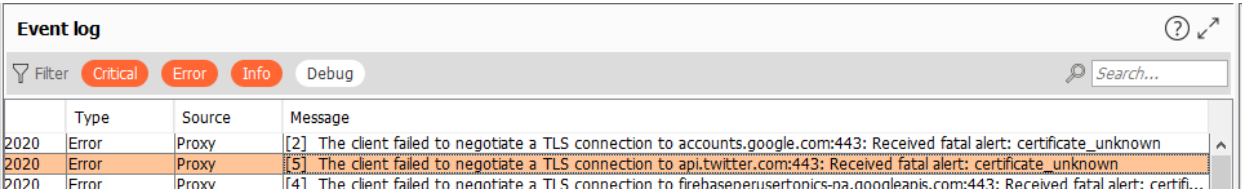
SSL Pinning(Sertifika Sabitleme) Nedir?

SSL(Secure Socket Layer), sunucu ile istemci arasında güvenli ve doğru server ile bağlantı kurmak amacıyla kullanılır. Bize doğrulama ve kriptolama sağlar. İletişim bu kanal üzerinden kurulduğunda güvenli olacaktır ve ortadaki adam saldırılarını(MiTM) engelleyecektir.

SSL Pinning, Certificate Pinning olarak da geçen bu yöntem ile programcı uygulamanın üzerinde çalıştığı cihazda hali hazırda yüklü olan Trusted sertifikalara güvenmeyerek, kendisi program içinde yine uygulama paketiyle birlikte gelmiş ve uygulamanın anlayacağı formattaki sertifika/sertifikaları programatik olarak hafızaya yükleyerek sadece bu sertifika otoriterin/otoritelerinin imzaladığı sertifikalara güvenmesiyle sağlamaktadır. Aslında ortadaki adam saldırılarını(MiTM) azaltmayı hedeflemiştir.

Bunun amacı ilk bakışta sadece güvenliđi hedefliyormuş gibi görünüyor, çünkü bu yapılmazsa cihaza aslında güvenilmeyen bir sertifika otoritesinin sertifikası da kullanıcı (veya bir saldırgan) tarafından yüklenmiş olabilir ve bu sertifika ile araya girme amaçlı olarak üretilmiş olan sahte sertifikalara da güvenilebilir. Bizim Burp Suite'den elde ettiğimiz CA sertifikasını yüklemek ile yaptığımız tam olarak bu. Böylece uygulama ve uygulamanın iletişim kurduğu HTTP sunucu arasına rahatlıkla girebiliyoruz. Ancak bu yöntem programcılar tarafından self signed sertifikaların kullanımı amacıyla da kullanılabilir. Yani aslında güvenilen bir sertifika otoritesi (CA) tarafından imzalanmamış (dolayısıyla para ödenmemiş) ve kendi ortamımızda imzaladığımız bir sertifikayı da bu şekilde kullanabiliriz. Araya girmeye çalıştığımızda uygulamamızın üzerinde çalıştığı platform bu sertifikaya güvenmeyeceğinden uygulama bir sertifika doğrulama hatası döndürerek uygulamayı durdurulacaktır. Programcının amacı ne olursa olsun SSL pinning biz güvenlik testi yapanlar için bir problem.

Burp üzerinde araya girmeye çalıştığımızda aşağıdaki gibi negotiation hatası almaktayız. Bu bize SSL iletişiminin doğrulanamadığı/anlaşamadığını gösteriyor.



The screenshot shows the 'Event log' window in Burp Suite. It has a filter bar with 'Critical', 'Error', 'Info', and 'Debug' buttons. Below the filter is a search bar. The log table has columns for 'Type', 'Source', and 'Message'. Three error entries are visible, all with 'Error' type and 'Proxy' source. The messages describe failures to negotiate a TLS connection due to a 'certificate_unknown' fatal alert.

Type	Source	Message
Error	Proxy	[2] The client failed to negotiate a TLS connection to accounts.google.com:443: Received fatal alert: certificate_unknown
Error	Proxy	[5] The client failed to negotiate a TLS connection to api.twitter.com:443: Received fatal alert: certificate_unknown
Error	Proxy	[4] The client failed to negotiate a TLS connection to firebasenuserionics-na.firebaseio.com:443: Received fatal alert: certifi...

Araya girme yöntemi olarak birçok yöntem bulabilirsiniz. Biz burada bazılarını göreceğiz.

II. ANDROİD'DE FRIDA KULLANARAK SSL PINNING ATLATMA

Frida Server Nedir?

Pentester olarak SSL pinning atlatmak bazı durumlarda zor olabilmektedir. Bunu aşmanın bir yolu da frida server kullanarak fonksiyona hook olma işlemidir. Frida; geliştiriciler, reverse-engineers ve güvenlik araştırmacıları için Dynamic instrumentation toolkit olarak tanımlanmaktadır. Android, Windows, IOS, macOS, GNU/Linux ve QNX platformları üzerinde çalışan uygulamalara, JavaScript kodlarını enjekte edilmesiyle frida JavaScript API sunmaktadır. Yazılan JavaScript kodları da frida-server tarafından runtime'da çalışan process'e enjekte olarak istediđiniz işlemi yapmayı sağlıyor. Biz burada proxy'ye(burp suite) yönlendirdiđimiz trafiđi belirlenen sertifikaya güvenmesini sağlayacağız.

Frida server size uygulama içinde gelen-giden paketleri yakalamanıza, process içinde istediđiniz deđişikliđi yapmanıza izin vermektedir. Biz de bu imkanı kullanarak SSL pinning devre dışı bırakarak attack proxy(burp suit) ile araya girme işlemi yapacağız.

Frida client-server mantıđı olan iki bileşenden oluşmaktadır. Android cihaza frida server kuracağız, bilgisayarımıza da client uygulamasını kurarak ilerleyeceğiz.

Frida Server ile SSL Pinning Atlatma

Yapacađımız işlemler için gerekli araçlar olacaktır. Bunlar;

Computer;

- python 3
- pip for python
- adb tools (Android Debug Bridge tools)
- local proxy (Burp suite)

android cihaz;

- android device rooted (desteklenen android sürümleri 4.4.4 to 8.1)

ya da

- android emülatör (desteklenen android sürümleri 4.4.4 to 8.1)

Biz burada emülatör kullanacağız. Kullanılan API 26 (Android 8.0)

İlk önce python kurulu bilgisayara şunları yükleyelim;

```
python -m pip install Frida
python -m pip install objection
python -m pip install frida-tools
```

Ya da

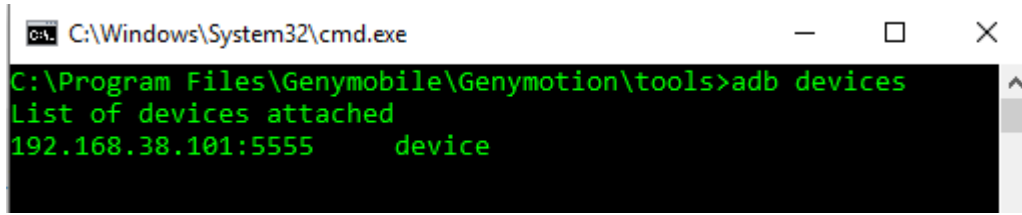
```
pip install Frida
pip install objection
pip install frida-tools
```

Frida server kurulması için <https://github.com/frida/frida/releases> linkten indirilmelidir.

Root'lu fiziksel cihaza adb bağlantısını sağlamak için şu adımları uygulamalısınız;

Ayarlar(Setting) → Geliştirici Seçenekleri(Developer Options) → USB Hata Ayıklama Modu(USB Debugging) → Enable(Aktif)

“adb devices” yazarak bağlı cihazları görelim.



```
C:\Windows\System32\cmd.exe
C:\Program Files\Genymobile\Genymotion\tools>adb devices
List of devices attached
192.168.38.101:5555 device
```

Burada server olana ve versiyonuna dikkat edilmelidir. Aşağıdaki sorgu ile açık olan emülatöre hangi işletim sistemi mimarisi olduğunu sorgulayalım;

ADB tool C:\Program Files\Genymobile\Genymotion\tools\ içerisinde ;

```
adb shell getprop ro.product.cpu.abi
#Output
x86
```

Windows, macOS, GNU / Linux, iOS, Android ve QNX işletim sistemlerinde çalışmaktadır. Bu sorgu için uygun olan frida server'ini indirelim. Burada x86 ve x64 olan ve farklı işletim sistemi için olanlar var bize en uygun olanını bulalım. Bazı versiyonlar bazı mimarilerde kurulum sıkıntısı çıkarabiliyor.

<https://github.com/frida/frida/releases/download/12.0.5/frida-server-12.0.5-android-x86.xz>

Sıkıştırılmış elf dosyasını çıkardıktan sonra cihaza kopyalayalım.

```
adb push frida-server-12.0.5-android-x86 /data/local/tmp
```

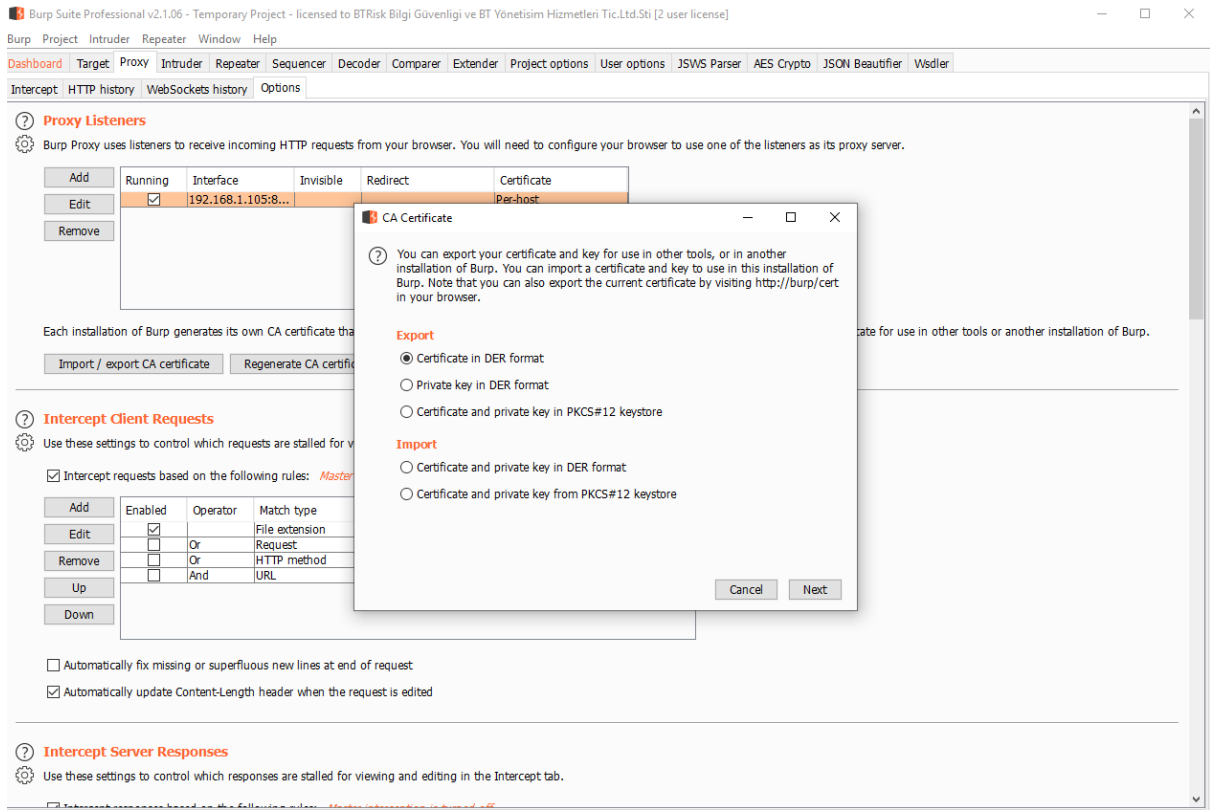
ismini değiştirelim(işimiz kolaylaşsın diye)

```
adb shell mv /data/local/tmp/frida-server-12.0.5-android-x86
/data/local/tmp/frida-server
```

Dosyaya tüm kullanıcıların ilgili dosya üzerinde okuma, yazma ve çalıştırma izni verelim.

```
adb shell chmod 777 /data/local/tmp/frida-server
```

Sonrasında burp sertifikası yüklemek için burp sertifikasını alalım. Proxy → Options sekmesinin altında bulunan Import/Export CA certificate butonuna tıklayarak karşınıza çıkan yerde Export kısmından Certificate in Der formatı seçip kaydedin.



“burp.cer” olarak kaydettiğimiz dosyayı cihaza kaydedelim. (cert-der.crt ismi vermek script için önemli olacak)

```
adb push burp.cer /data/local/tmp/cert-der.crt
```

Sonrasında script dosyasını kaydedelim;

İndirmek için şu adresten yararlanabilirsiniz. Bu script frida server ile SSL pinning geçersiz kılması için kullanacağız.

https://techblog.mediaservice.net/wp-content/uploads/2017/07/frida-android-repinning_sa-1.js

```
/*
Android SSL Re-pinning frida script v0.2 030417-pier
$ adb push burpca-cert-der.crt /data/local/tmp/cert-der.crt
$ frida -U -f it.app.mobile -l frida-android-repinning.js --no-pause
https://techblog.mediaservice.net/2017/07/universal-android-ssl-pinning-bypass-with-frida/
UPDATE 20191605: Fixed undeclared var. Thanks to @oleavr and @ehsanpc9999 !
*/

setTimeout(function () {
  Java.perform(function () {
    console.log("");
    console.log("[.] Cert Pinning Bypass/Re-Pinning");

    var CertificateFactory =
    Java.use("java.security.cert.CertificateFactory");
    var FileInputStream = Java.use("java.io.FileInputStream");
```

```
var BufferedInputStream = Java.use("java.io.BufferedInputStream");
var X509Certificate = Java.use("java.security.cert.X509Certificate");
var KeyStore = Java.use("java.security.KeyStore");
var TrustManagerFactory = Java.use("javax.net.ssl.TrustManagerFactory");
var SSLContext = Java.use("javax.net.ssl.SSLContext");

// Load CAs from an InputStream
console.log("[+] Loading our CA...")
var cf = CertificateFactory.getInstance("X.509");
try {
    var fileInputStream = FileInputStream.$new("/data/local/tmp/cert-
der.crt");
} catch (err) {
console.log("[o] " + err);
}

var bufferedInputStream = BufferedInputStream.$new(fileInputStream);
var ca = cf.generateCertificate(bufferedInputStream);
bufferedInputStream.close();

var certInfo = Java.cast(ca, X509Certificate);
console.log("[o] Our CA Info: " + certInfo.getSubjectDN());

// Create a KeyStore containing our trusted CAs
console.log("[+] Creating a KeyStore for our CA...");
var keyStoreType = KeyStore.getDefaultType();
var keyStore = KeyStore.getInstance(keyStoreType);
keyStore.load(null, null); keyStore.setCertificateEntry("ca", ca);

// Create a TrustManager that trusts the CAs in our KeyStore
console.log("[+] Creating a TrustManager that trusts the CA in our
KeyStore...");
var tmfAlgorithm = TrustManagerFactory.getDefaultAlgorithm();
var tmf = TrustManagerFactory.getInstance(tmfAlgorithm);
tmf.init(keyStore);
console.log("[+] Our TrustManager is ready...");

console.log("[+] Hijacking SSLContext methods now...")
console.log("[-] Waiting for the app to invoke SSLContext.init()...")

SSLContext.init.overload("[Ljavax.net.ssl.KeyManager;",
"[Ljavax.net.ssl.TrustManager;", "java.security.SecureRandom").implementation =
function (a, b, c) {
    console.log("[o] App invoked javax.net.ssl.SSLContext.init...");
    SSLContext.init.overload("[Ljavax.net.ssl.KeyManager;",
"[Ljavax.net.ssl.TrustManager;", "java.security.SecureRandom").call(this, a,
tmf.getTrustManagers(), c);
    console.log("[+] SSLContext initialized with our custom
TrustManager!");
}
});
}, 0);
-----
```

Javascript dosyasını oluşturduktan sonra cihazın içerisine atalım.

```
adb push fridascript.js /data/local/tmp/fridascript.js
```

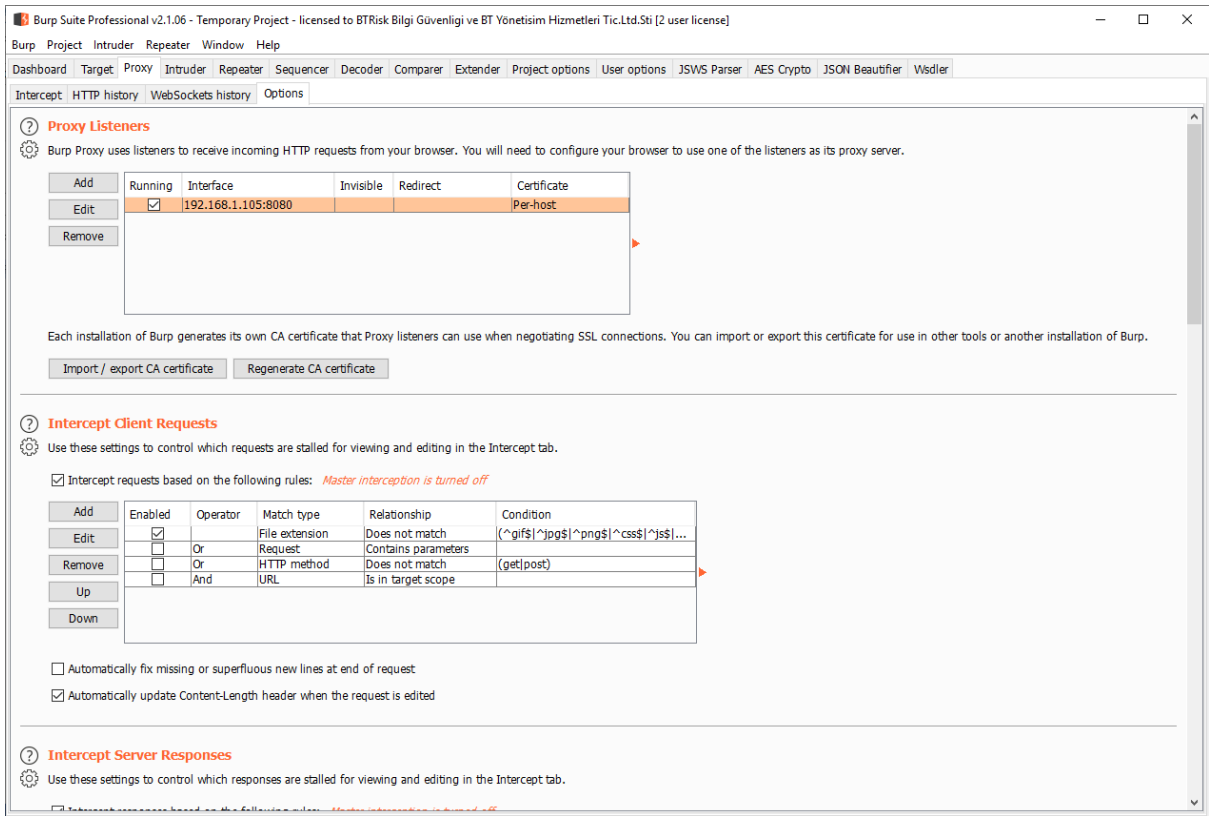
Dosyaların yüklendiđinden emin olalım.

```
adb shell ls -al /data/local/tmp/
```

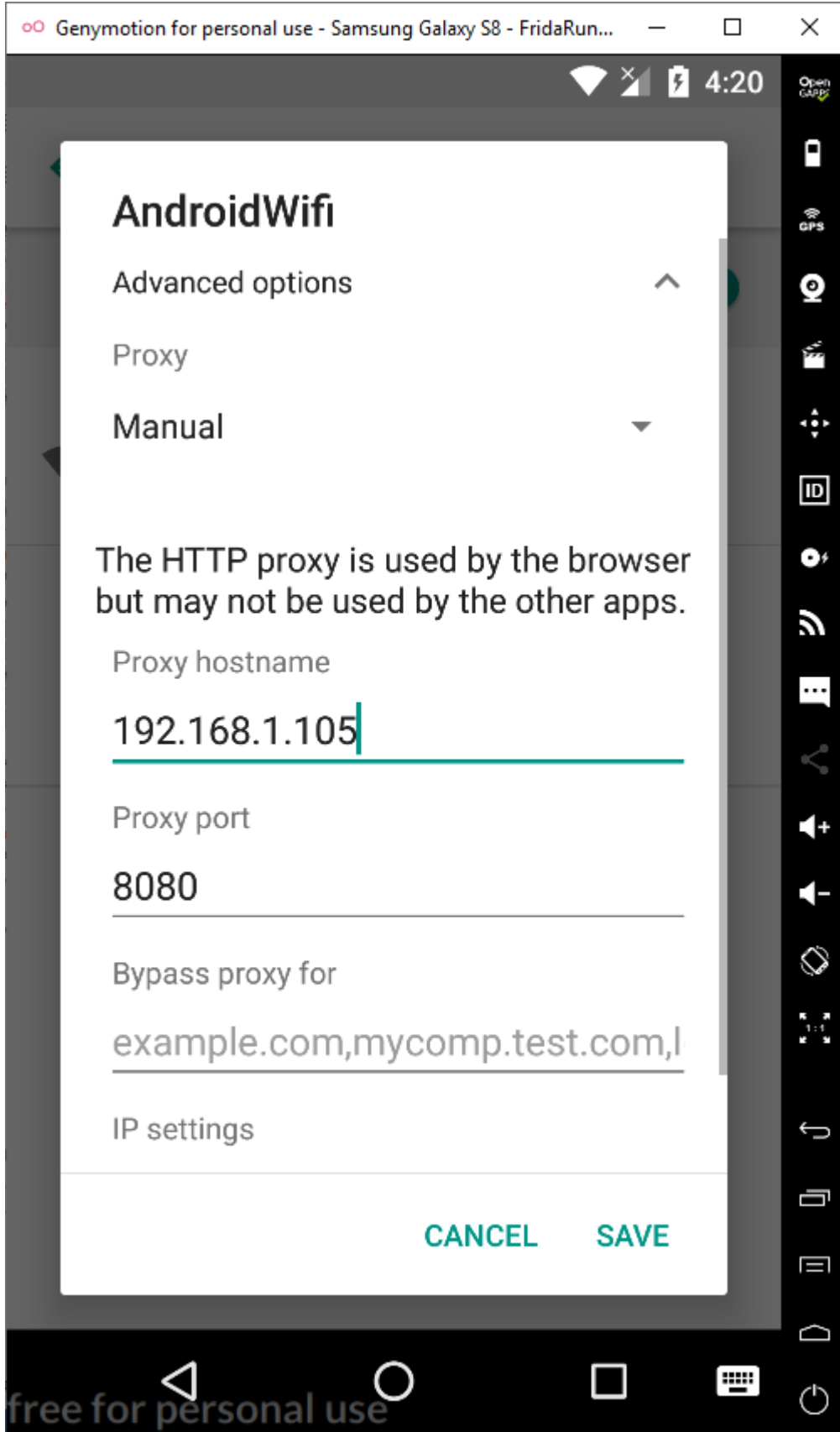
```
C:\Program Files\Genymobile\Genymotion\tools>adb shell ls -al /data/local/tmp/
total 25832
drwxrwx--x 2 shell shell      4096 2020-01-10 08:46 .
drwxr-x--x 3 root  root      4096 2020-01-10 08:29 ..
-rw-rw-rw- 1 root  root        973 2020-01-10 08:37 cert-der.crt
-rw-rw-rw- 1 root  root    26433300 2020-01-10 05:42 frida-server
-rw-rw-rw- 1 root  root      2945 2020-01-10 08:03 fridascript.js
```

Şimdi araya girebilmek için burp aracını dinleme moduna alalım.

Proxy → Options sekmesinden dinleyeceğimiz IP ve Port numaralarını girelim.



Cihazın proxy ayarı yaparak burp aracına yönlendirelim. Settings → Network & Internet → Wi-Fi → Modify-Network → Proxy → Manuel



Gerekli dosyalar yükledikten sonra frida server'ı arka planda çalıştırmak için aşağıdaki komutumuzu giriyoruz.

```
adb shell /data/local/tmp/frida-server &
```

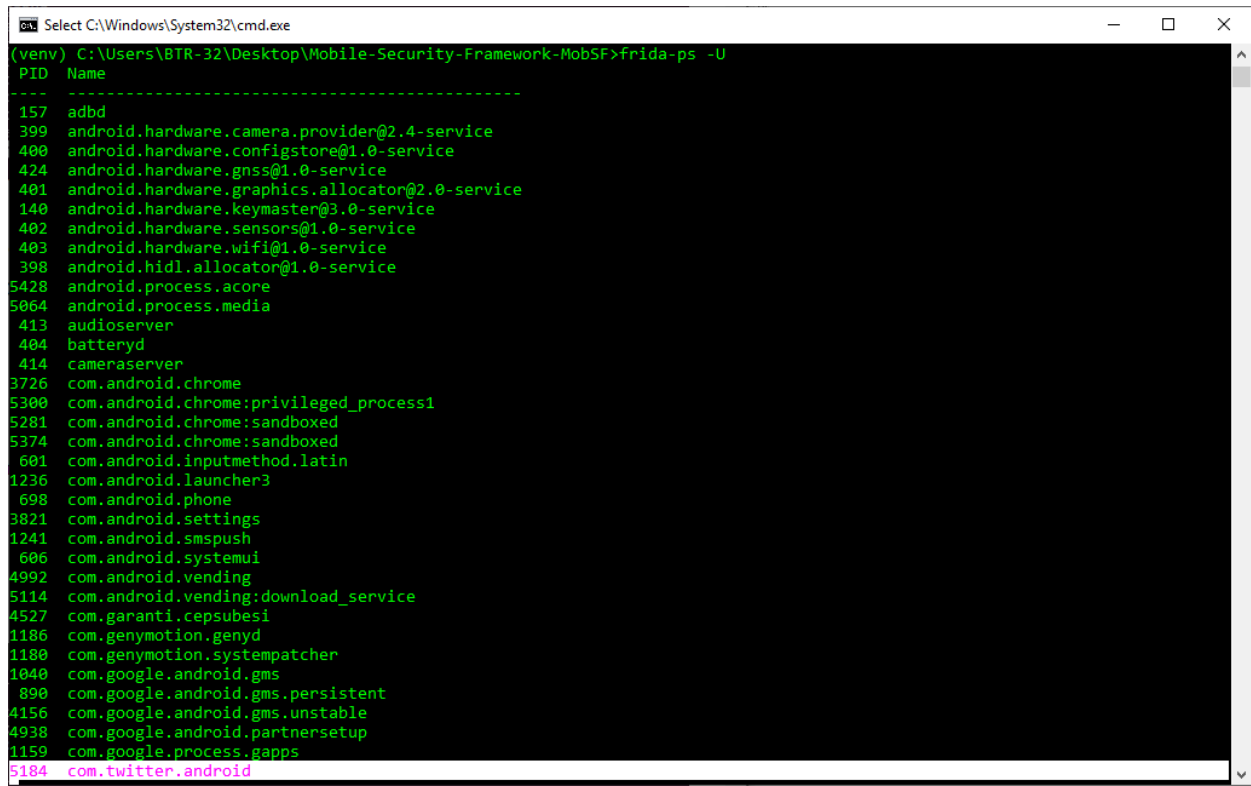
```
C:\Program Files\Genymobile\Genymotion\tools>adb shell /data/local/tmp/frida-server &
```

.\venv\Scripts\activate benim python3 bu dizinde.

```
C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>.\venv\Scripts\activate
(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>frida -U -f com.f
```

Şimdi uygulamayı açarak hook olmak istediğimiz process name bağlanalım. Bunun için frida kullanarak cihaz üzerinde çalışan servisleri listesini alalım.

```
frida-ps -U
```



```
Select C:\Windows\System32\cmd.exe
(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>frida-ps -U
PID Name
-----
157  adbd
399  android.hardware.camera.provider@2.4-service
400  android.hardware.configstore@1.0-service
424  android.hardware.gnss@1.0-service
401  android.hardware.graphics.allocation@2.0-service
140  android.hardware.keymaster@3.0-service
402  android.hardware.sensors@1.0-service
403  android.hardware.wifi@1.0-service
398  android.hidl.allocation@1.0-service
5428 android.process.acore
5064 android.process.media
413  audioserver
404  batteryd
414  cameracamera
3726 com.android.chrome
5300 com.android.chrome:privileged_process1
5281 com.android.chrome:sandboxed
5374 com.android.chrome:sandboxed
601  com.android.inputmethod.latin
1236 com.android.launcher3
698  com.android.phone
3821 com.android.settings
1241 com.android.smpush
606  com.android.systemui
4992 com.android.vending
5114 com.android.vending:download_service
4527 com.garanti.cepsubesi
1186 com.genymotion.genyid
1180 com.genymotion.systempatcher
1040 com.google.android.gms
890  com.google.android.gms.persistent
4156 com.google.android.gms.unstable
4938 com.google.android.partnersetup
1159 com.google.process.gapps
5184 com.twitter.android
```

Proses'i başlatan ve hook olarak SSL pinning devre dışı bırakan komutu başlatalım.

```
frida -U -f com.twitter.android -l fridascript.js --no-paus
```

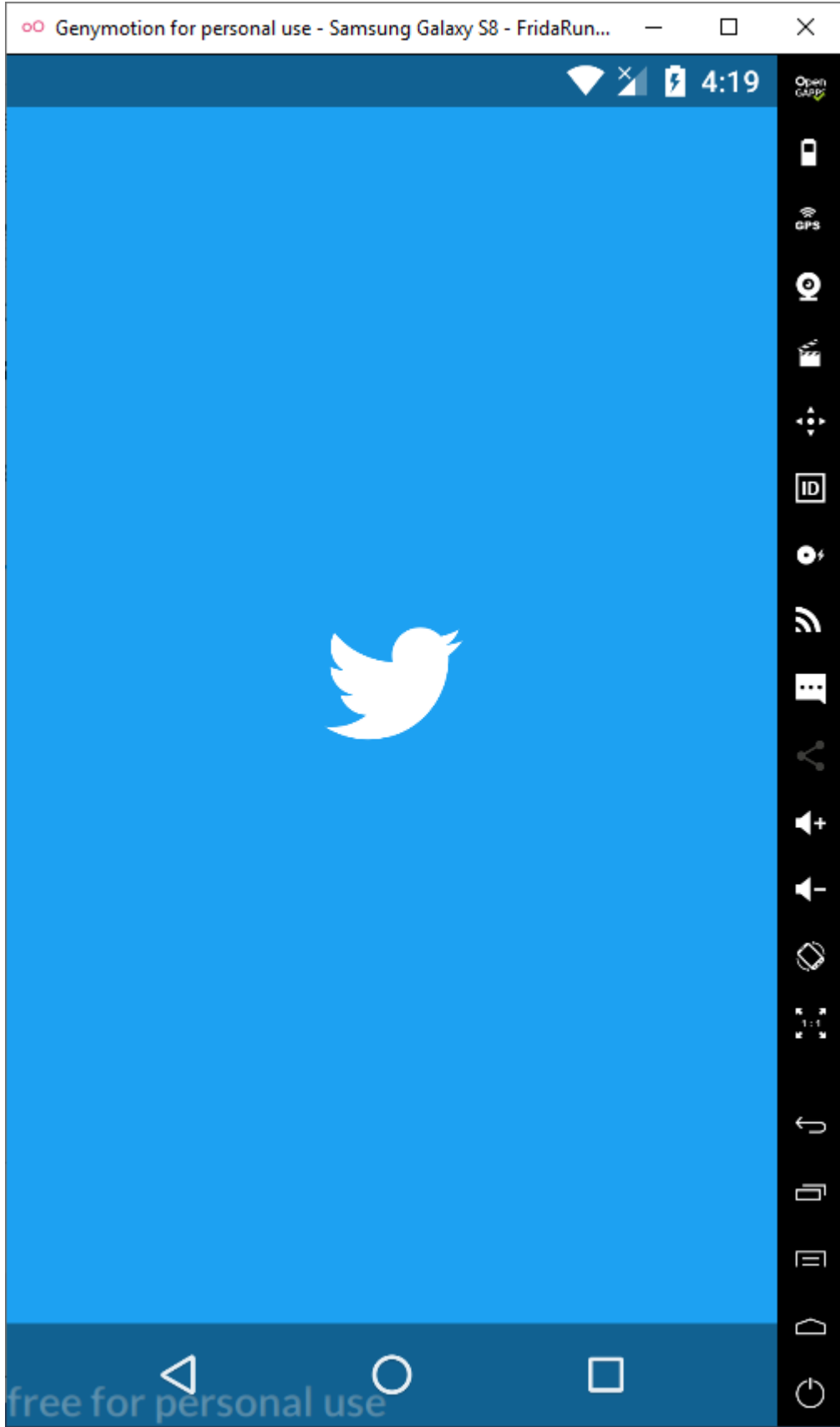
```
CA:\Windows\System32\cmd.exe - frida -U -f com.twitter.android -l fridascript.js --no-paus
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>.venv\Scripts\activate

(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>frida -U -f com.twitter.android -l fridascript.js --no-paus

┌───┐
│   │   Frida 12.8.6 - A world-class dynamic instrumentation toolkit
│   │
│   │   Commands:
│   │   help      -> Displays the help system
│   │   object?   -> Display information about 'object'
│   │   exit/quit -> Exit
│   │
│   │   More info at https://www.frida.re/docs/home/
└───┘
Spawned `com.twitter.android`. Resuming main thread!
[Genymotion Samsung::com.twitter.android]->
[.] Cert Pinning Bypass/Re-Pinning
[+] Loading our CA...
[o] Our CA Info: CN=PortSwigger CA, OU=PortSwigger CA, O=PortSwigger, L=PortSwigger, ST=PortSwigger, C=PortSwigger
[+] Creating a KeyStore for our CA...
[+] Creating a TrustManager that trusts the CA in our KeyStore...
[+] Our TrustManager is ready...
[+] Hijacking SSLContext methods now...
[-] Waiting for the app to invoke SSLContext.init()...
[o] App invoked javax.net.ssl.SSLContext.init...
[+] SSLContext initialized with our custom TrustManager!
[o] App invoked javax.net.ssl.SSLContext.init...
[+] SSLContext initialized with our custom TrustManager!
```

Cihazda uygulama açılacaktır.



Burp üzerinden istek ve yanıtları gözlemleyebiliriz.

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	SSL	IP
21	https://api.twitter.com	GET	/1.1/traffic/recommendations.json?...	✓		429	385	JSON	json			✓	104.244.42.194
22	https://api.twitter.com	GET	/1.1/help/settings.json?feature_set_...	✓		200	3055	JSON	json			✓	104.244.42.194
23	https://api.twitter.com	GET	/2/badge_count/badge_count.json...	✓		429	710	JSON	json			✓	104.244.42.194
24	https://api.twitter.com	GET	/robots.txt			200	705	text	txt			✓	104.244.42.194
25	https://pbs.twimg.com	GET	/robots.txt			200	506	text	txt			✓	93.184.220.70
26	https://video.twimg.com	GET	/robots.txt			200	398	text	txt			✓	68.232.34.217
27	https://api.twitter.com	GET	/1.1/traffic/recommendations.json?...	✓		429	385	JSON	json			✓	104.244.42.194
28	https://api.twitter.com	GET	/1.1/help/settings.json?feature_set_...	✓		200	3055	JSON	json			✓	104.244.42.194
29	https://api.twitter.com	GET	/2/badge_count/badge_count.json...	✓		429	710	JSON	json			✓	104.244.42.194
30	https://api.twitter.com	GET	/robots.txt			200	705	text	txt			✓	104.244.42.194
31	https://video.twimg.com	GET	/robots.txt			200	398	text	txt			✓	68.232.34.217
32	https://pbs.twimg.com	GET	/robots.txt			200	506	text	txt			✓	93.184.220.70

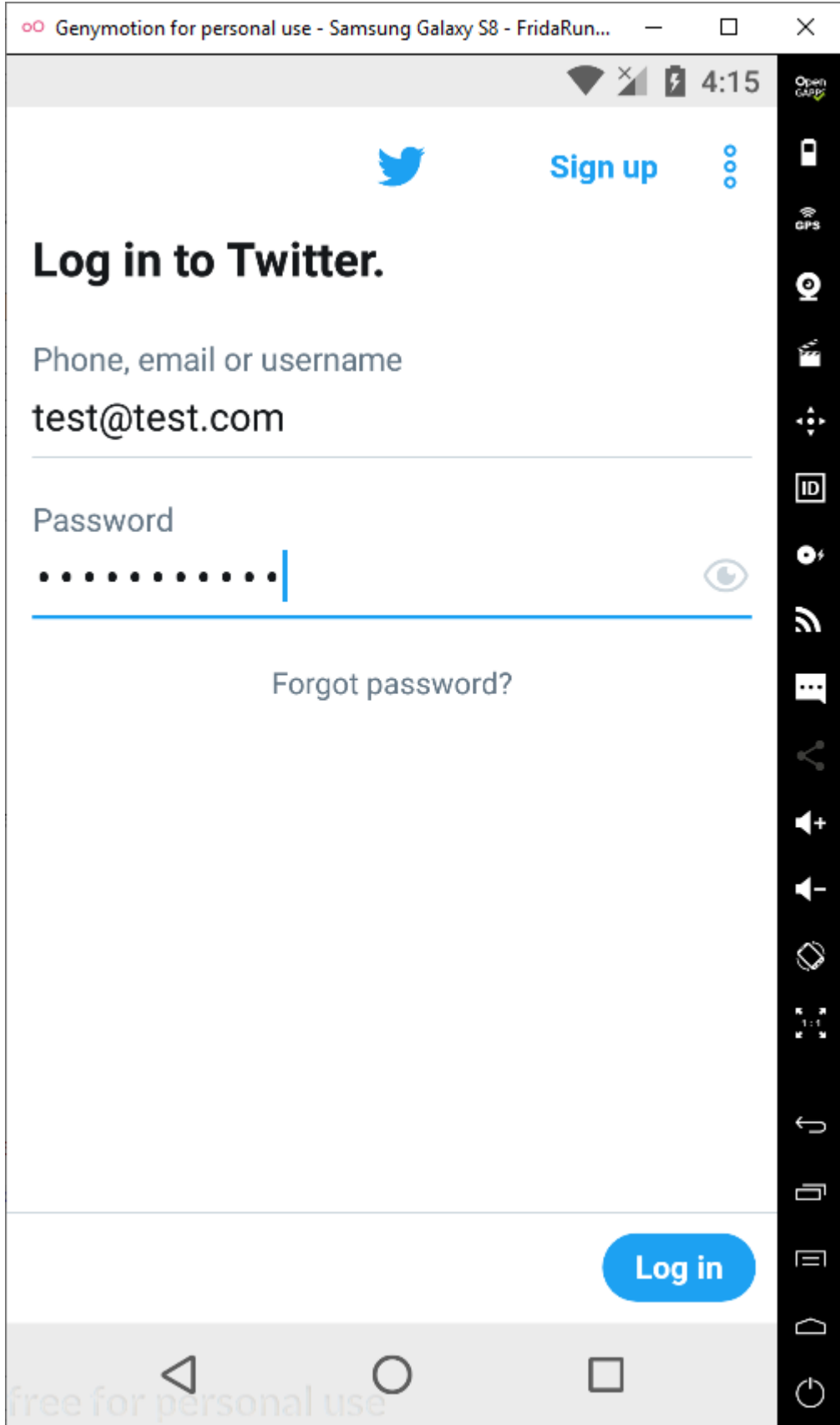
Request Response

Raw Params Headers Hex

```
GET /2/badge_count/badge_count.json?supports_ntab_urt=true HTTP/1.1
Timezone: GMT
Optimize-Body: true
Accept: application/json
X-Twitter-Client: TwitterAndroid
User-Agent: TwitterAndroid/8.25.1-release.01 (18251001-r-1) Samsung Galaxy S8/8.0.0 (Genyotion;Samsung Galaxy S8;Android;ybox86p,0;1;2015)
Accept-Encoding: gzip, deflate
X-Twitter-Client-Language: en-US
X-Client-UUID: ca42e8cf-ef39-4741-90f7-de23ae2b9474
X-Twitter-Client-DeviceID: a580dba36f0e187
Authorization: OAuth realm="http://api.twitter.com", oauth_version="1.0", oauth_nonce="40819917636854306504956121732519", oauth_timestamp="1578649156",
oauth_signature="wyuEn3UjHxe1bCVaxYw7FPLHg%3D", oauth_consumer_key="3nVuSoBZnx6U4vzUxt5w", oauth_signature_method="HMAC-SHA1"
X-Twitter-Client-Version: 8.25.1-release.01
Cache-Control: no-store
X-Twitter-Active-User: no
X-Twitter-API-Version: 5
X-B3-TraceId: 82249f03fe7a8cb
Accept-Language: en-US
X-Twitter-Client-Flavor:
Host: api.twitter.com
Connection: close
```

0 matches

Login olmaya çalışalım



Burp Suite'den baktığımızda gelen giden paketleri görebiliriz.

The screenshot shows the Burp Suite Professional v2020.4 interface. The top menu bar includes 'Burp Project', 'Intruder', 'Repeater', 'Window', and 'Help'. Below the menu is a 'Dashboard' with tabs for 'Target', 'Proxy', 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Extender', 'Project options', 'User options', 'JSWS Parser', 'AES Crypto', 'JSON Beautifier', and 'Wsdler'. The main area is divided into 'Intercept', 'HTTP history', 'WebSockets history', and 'Options'. The 'HTTP history' tab is active, showing a list of intercepted requests. The table below is a summary of these requests:

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
1824	https://api.twitter.com	POST	/1.1/jot/client_event	✓		200	600					✓	104.244.42.130		19:13:50 2 ...	8080
1823	https://api.twitter.com	POST	/1.1/jot/client_event	✓		200	600					✓	104.244.42.130		19:13:49 2 ...	8080
1822	https://api.twitter.com	POST	/auth/1/xauth_password.json	✓		401	1026	JSON	json			✓	104.244.42.130		19:13:01 2 ...	8080
1821	https://api.twitter.com	POST	/auth/1/xauth_password.json	✓		401	1026	JSON	json			✓	104.244.42.130		19:13:01 2 ...	8080
1820	https://api.twitter.com	POST	/1.1/guest/activate.json			200	777	JSON	json			✓	104.244.42.130		19:13:00 2 ...	8080
1819	https://api.twitter.com	POST	/auth/1/xauth_password.json	✓		401	979	JSON	json			✓	104.244.42.130		19:12:59 2 ...	8080

The 'Request' tab is selected, showing the raw HTTP request details:

```

Raw Params Headers Hex
1 POST /auth/1/xauth_password.json HTTP/1.1
2 Timesome: GMT
3 acc: 2qVdha325aBtFY1bD4vYhMYSh2ia5x3z4dDE33
4 Optimize-Body: true
5 Accept: application/json
6 X-Twitter-Client: TwitterAndroid
7 User-Agent: TwitterAndroid/8.4l.0-release.01 (18410001-r-1) Samsung/8.0.0 (Genyotion;Samsung;Android;wbox86p;0;1;2015)
8 X-Twitter-Client-AdID: ee31c6-1168-4cf1-9d70-7a7d6ae8074a
9 Accept-Encoding: gzip, deflate
10 X-Twitter-Client-Language: en-US
11 X-Client-UUID: ca42e8cf-ef38-4741-50e7-de23ae2b5474
12 X-Twitter-Client-DeviceID: a580ba3fE0e18f
13 Authorization: Basic AAAAAAAAAAAAAAAAAAFAwAAAAAAAAAMHCxpeSDG1gLNlghVe8d74h16k413DFUMF&kAQLseBhTSRrC1QpJxooGweyHrDb5te2jp6skWDFW8ZF
14 X-Twitter-Client-Version: 8.4l.0-release.01
15 Cache-Control: no-store
16 X-Quest-Token: 156618189048687616
17 X-Twitter-Active-User: yes
18 X-Twitter-API-Version: 5
19 X-B3-TraceId: 38d318b8cf97a2d
20 X-Twitter-Client-Limit-Ad-Tracking: 0
21 Accept-Language: en-US
22 X-Twitter-Client-Flavor:
23 Content-Type: application/x-www-form-urlencoded
24 Content-Length: 171
25 Host: api.twitter.com
26 Connection: close
27 Cookie: personalization_id=vl_jA3R0/J7vUgh20U5q4BvVw--; guest_id=vl331S8043605359911552
28
29 X_auth_identifier=test140test.com&x_auth_password=password123&send_error_codes=true&x_auth_login_challenge=1&x_auth_login_verification=1&x_auth_country_code=US&ui_metrics=
    
```

Kısaca çalıştırdığımız komutları listelersek;

```

adb shell getprop ro.product.cpu.abi
adb push frida-server-12.0.5-android-x86 /data/local/tmp
adb shell mv /data/local/tmp/frida-server-12.0.5-android-x86
/data/local/tmp/frida-server
adb shell chmod 777 /data/local/tmp/frida-server
adb push burp.cer /data/local/tmp/cert-der.crt
adb push fridascript.js /data/local/tmp/fridascript.js
adb shell /data/local/tmp/frida-server &
frida-ps -U
frida -U -f com.twitter.android -l fridascript.js --no-paus
    
```

Objection ile SSL Pinning Atlatma

Eđer android uygulaması OkHttp kütüphanesini kullanıyorsa, Objection bu OkHttp ve varsayılan TrustManagerImpl kütüphanelerine hook olarak SSL pinning atlatılabilir.

Yine frida yardımıyla bu işlemleri yapabiliriz. Frida server cihaz üzerinde başlatalım.

```
C:\Windows\System32\cmd.exe - adb shell /data/local/tmp/frida-server
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\Genymobile\Genymotion\tools>adb
C:\Program Files\Genymobile\Genymotion\tools>adb shell /data/local/tmp/frida-server &
```

Objection kurulumunu yapalım.

```
pip3 install objection
```

Frida ya da adb kullanarak android uygulama adlarını öğrenelim.

```
frida-ps -U
```

Ya da

```
adb shell ps
```

```
C:\Windows\System32\cmd.exe
(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>frida-ps -U
PID Name
-----
154 adbd
398 android.hardware.camera.provider@2.4-service
399 android.hardware.configstore@1.0-service
423 android.hardware.gnss@1.0-service
400 android.hardware.graphics.allocation@2.0-service
137 android.hardware.keymaster@3.0-service
401 android.hardware.sensors@1.0-service
402 android.hardware.wifi@1.0-service
397 android.hidl.allocation@1.0-service
1428 android.process.acore
5107 android.process.media
412 audioserver
403 batteryd
413 cameracore
6724 com.android.chrome
6982 com.android.chrome:privileged_process0
6965 com.android.chrome:sandboxed
7085 com.android.chrome:webview_service
597 com.android.inputmethod.latin
1255 com.android.launcher3
701 com.android.phone
6094 com.android.printspooler
1507 com.android.providers.calendar
1231 com.android.smspush
611 com.android.systemui
6167 com.android.vending
```


Burp proxy ayarlarını yapalım. Hook işlemi yaparak sslpinnig disable edelim.

```
C:\Windows\System32\cmd.exe

(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>objection -g com.c... :c explore -q
Using USB device `Genymotion Samsung`
Agent injected and responds ok!
com.eurekosigorta on (Android: 8.0.0) [usb] # android sslpinning disable
(agent) Custom TrustManager ready, overriding SSLContext.init()
(agent) Found com.android.org.conscrypt.TrustManagerImpl, overriding TrustManagerImpl.verifyChain()
(agent) Found com.android.org.conscrypt.TrustManagerImpl, overriding TrustManagerImpl.checkTrustedRecursive()
(agent) Registering job 3hwehtjw2ep. Type: android-sslpinning-disable
com.eurekosigorta on (Android: 8.0.0) [usb] # (session detach message) process-terminated
com.eurekosigorta on (Android: 8.0.0) [usb] #
com.eurekosigorta on (Android: 8.0.0) [usb] # exit
Exiting...
Asking jobs to stop...
Unloading objection agent...
Unable to run cleanups: script is destroyed

(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>
```

Bundan sonra trafiđi burp üzerinden dinlenebilir.

III. ANDROID'DE XPOSED FRAMEWORK KULLANARAK SSL PINNING ATLATMA

Xposed Framework Nedir?

Xposed Framework, root'lu Android telefonunuz için geliştirilmiş bir framework'dur. Kendi başına pek bir şey yapmaz. Custom recovery üzerinden sistem seviyesindeki fonksiyonları kullanarak diğer uygulamaları/modları/ince ayarları yüklemenizi ve deđiştirmenize android işletim sistemi seviyesinde fonksiyonlara müdahale etmemize olanak sağlar. Xposed Framework, telefonunuzu özelleştirmenizi/deđiştirmenizi çok daha kolay hale getirir. Bizim için ise SSL class'larına müdahale etmemize olanak sağlayacak.

SSL pinning kullanan uygulamalarda araya girmek burp suite ile araya girmek gerekli ise, SSLUnpinning size yardımcı olabilir. SSLUnpinning Xposed Framework üzerinden sertifika doğrulamaları yapan belirli uygulamaları SSL classes'larını hook işlemi yapar. SSL pinning devre dışı bırakarak trafiđi dinleyebilmenize olanak sunar. Böylece istenilen sertifikaya güvenilerek trafik burp üzerinden geçirilebilir.

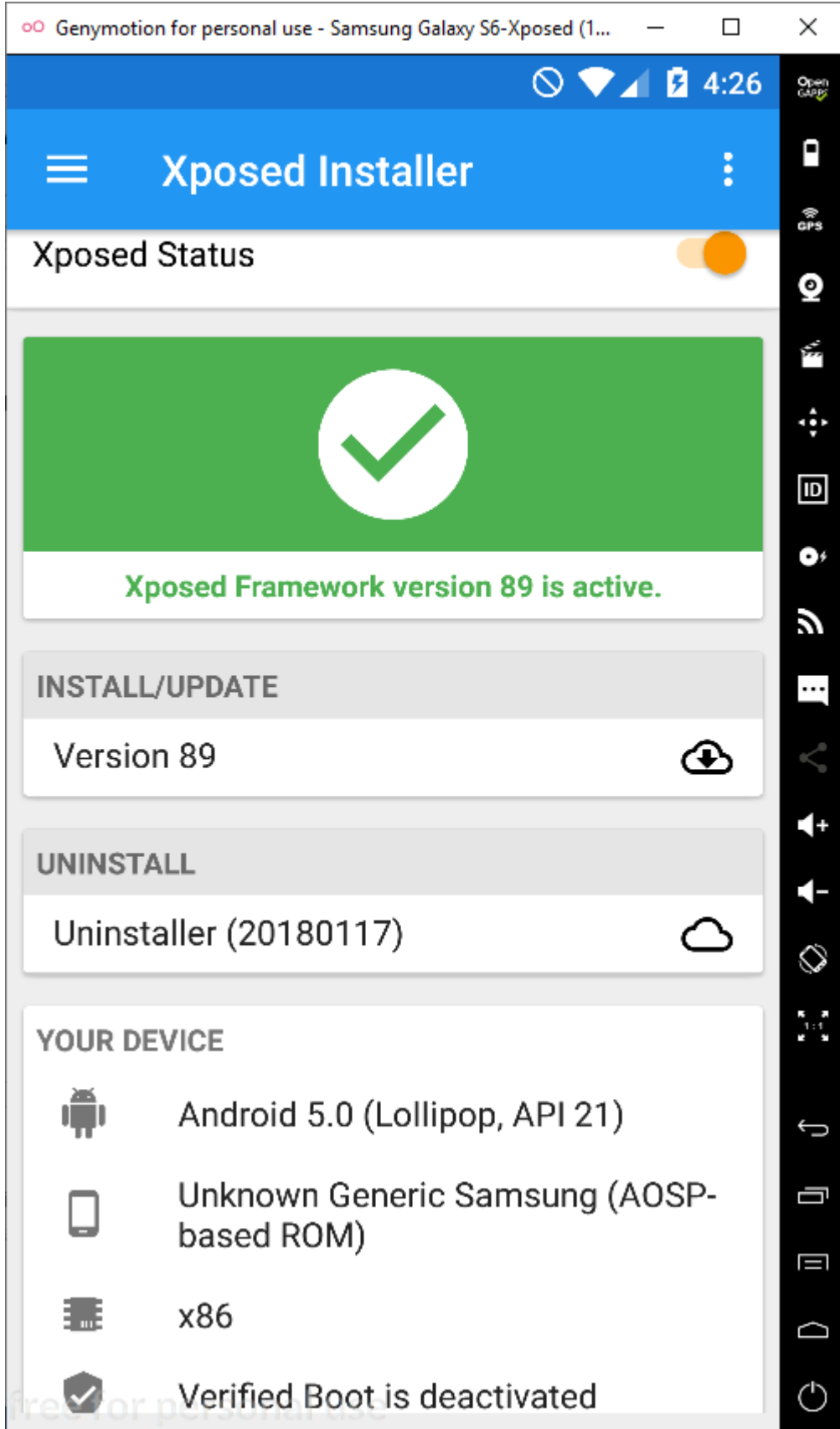
Xposed runtime'da müdahale edecektir. Xposed için root'lu cihaza ihtiyacımız bulunmaktadır. Android 5.0 ve üzeri bir versiyon olmalıdır.

Aşağıdaki adreslerden Xposed framework ve SSLUnpinning indirelim.

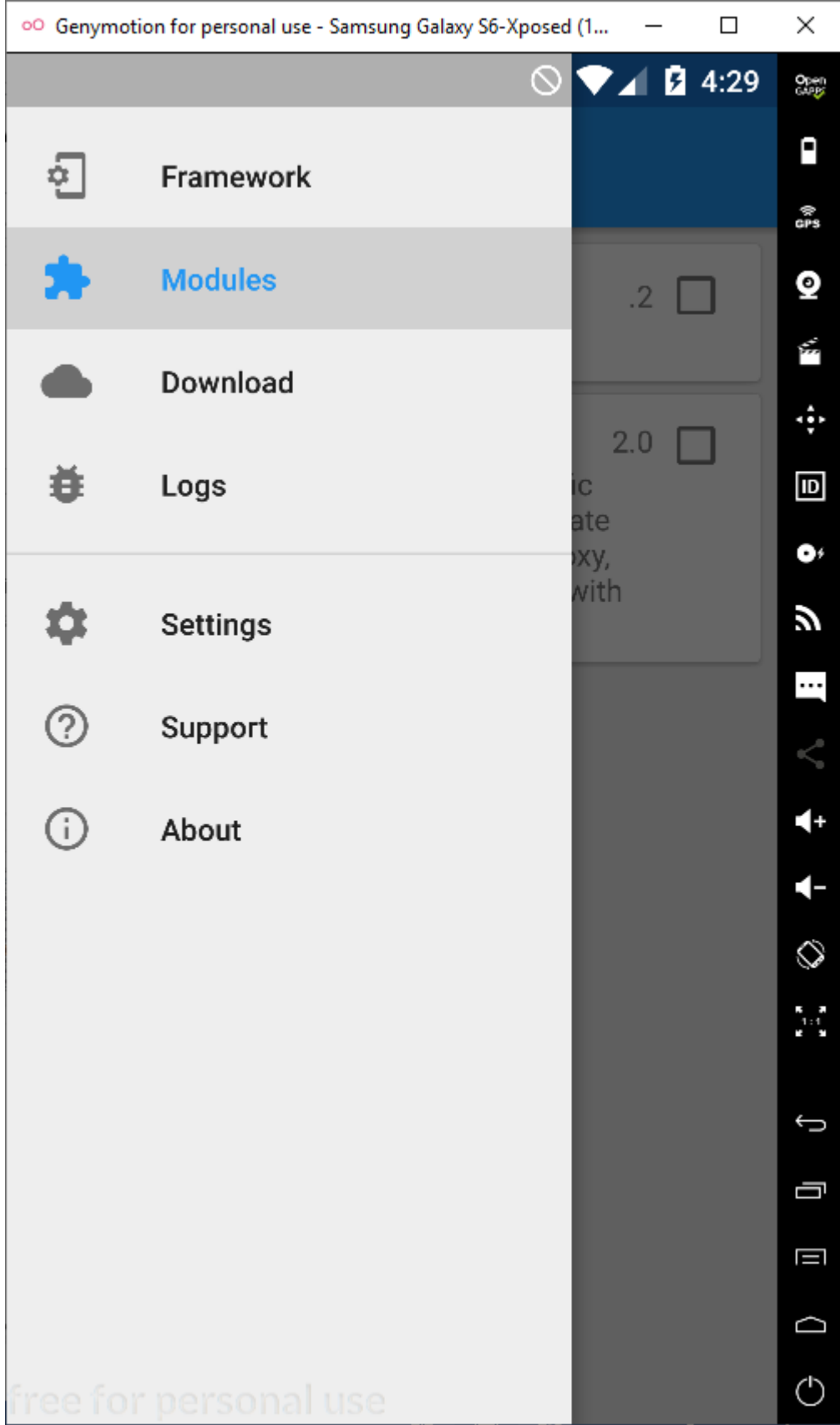
<https://repo.xposed.info/module/mobi.acpm.sslunpinning>

<https://repo.xposed.info/module/de.robv.android.xposed.installer>

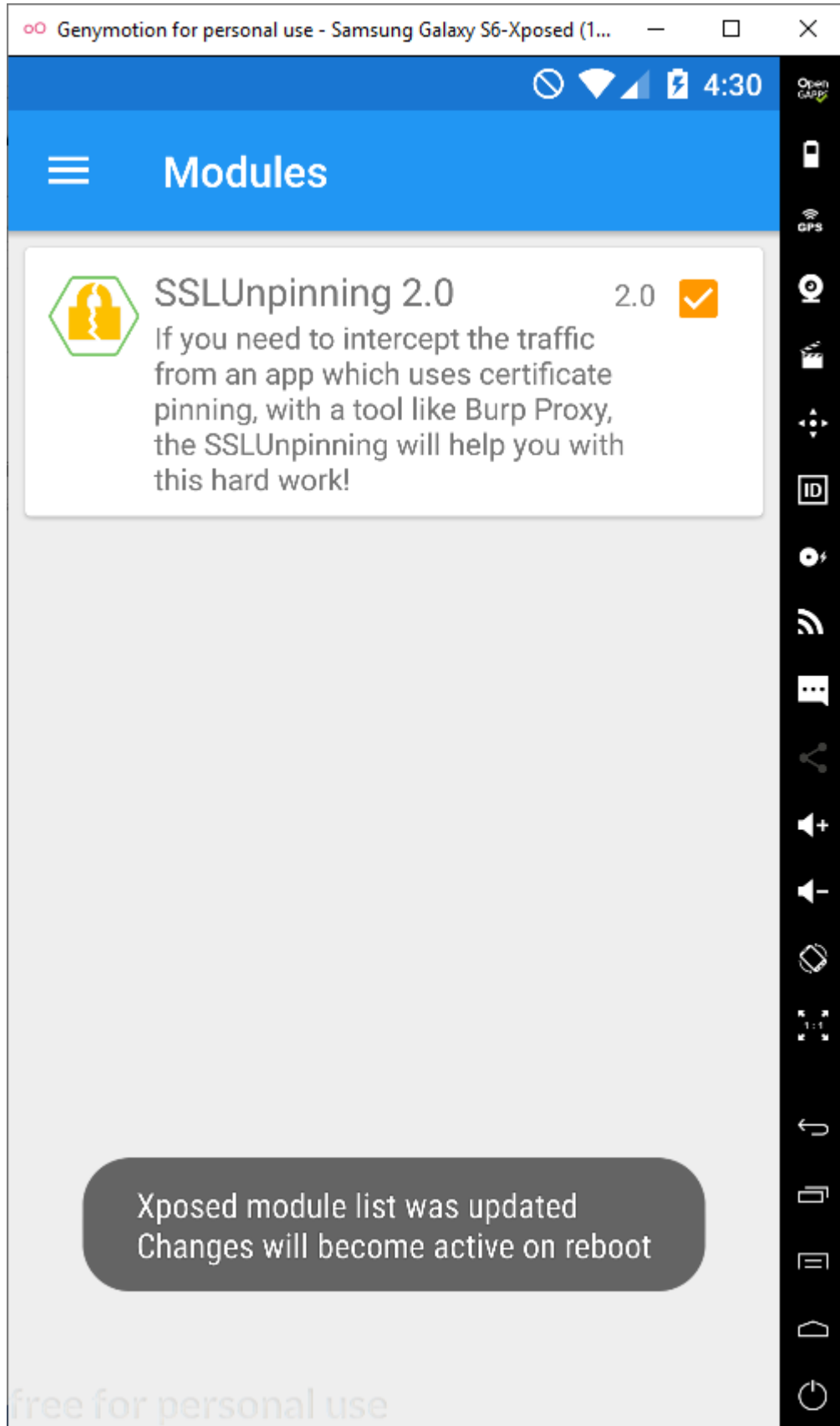
İndirilen dosyaları sürükleyerek kurabiliriz. İlk önce Xposed kuralım.



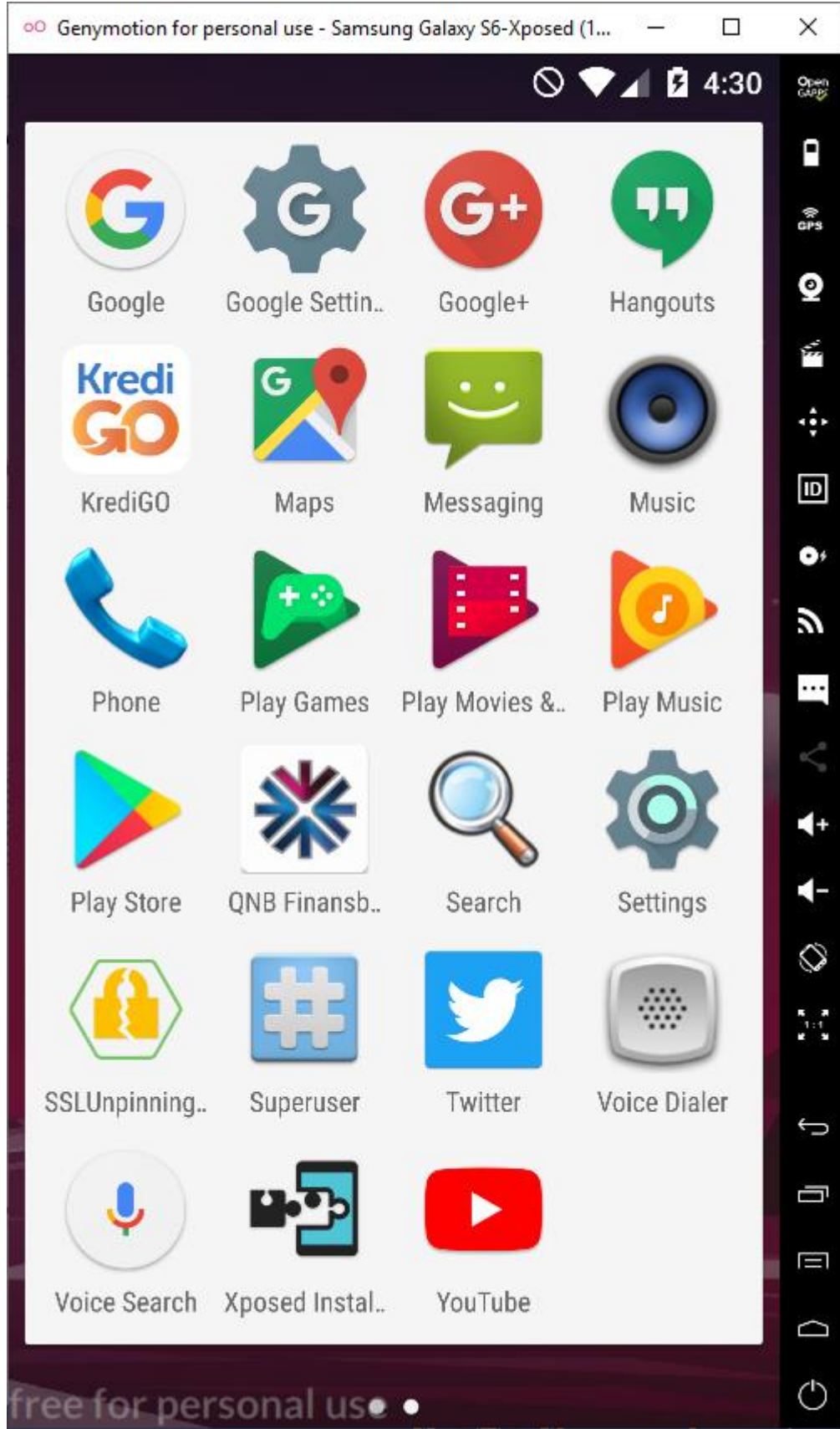
SSLUnpinning kuralım ve Modules bakalım.



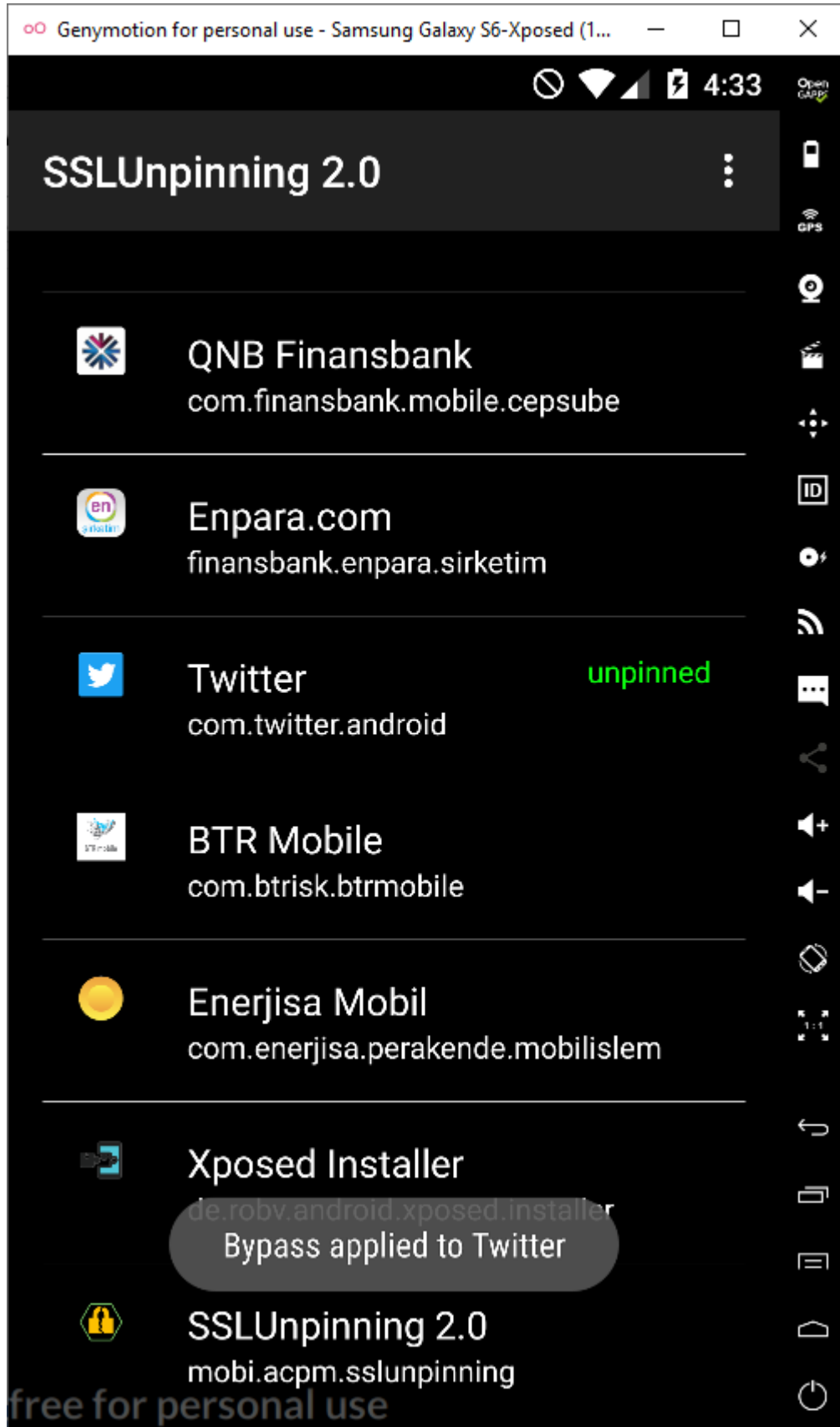
Şimdi radio button'una tıklayarak kuralım.



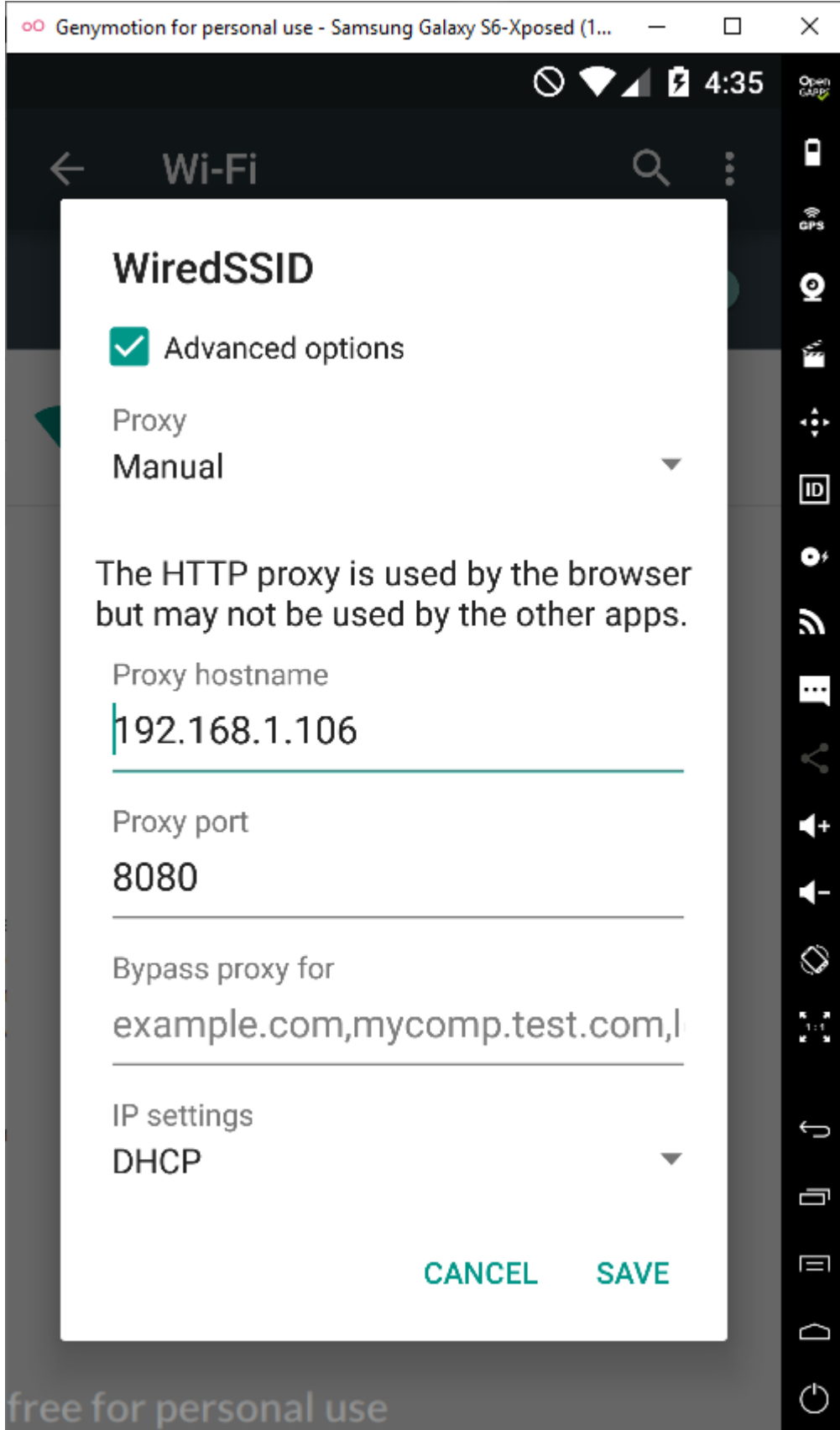
Reboot yaptıđında uygulama aktif olacaktır. SSLUnpinning uygulama olarak göreceđiz.



Araya gireceđimiz uygulamayı seçelim ve runtime'da sertifikaya güvenmesini engelleyerek tüm sertifikalara izin vermiş olacağız. Burada Burp suite sertifikasının cihaz üzerinde kurulu olması gerek yok.



Proxy ayarını yapalım. Setting → Wi-Fi → WiredSSID → Modify Network → Proxy → Advanced options
→ Manuel



Proxy → Options sekmesinden dinleyeceğimiz IP ve Port numaralarını girelim.

The screenshot shows the Burp Suite Professional v2.1.06 Options tab, specifically the Proxy Listeners section. The interface includes a menu bar (Burp, Project, Intruder, Repeater, Window, Help) and a toolbar (Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Project options, User options, JSWS Parser, AES Crypto, JSON Beautifier, Wsdler). The Proxy Listeners section has a sub-menu (Intercept, HTTP history, WebSockets history, Options) and a help icon. Below the help icon, there is a text box explaining that Burp Proxy uses listeners to receive incoming HTTP requests. A table lists the current listener configuration:

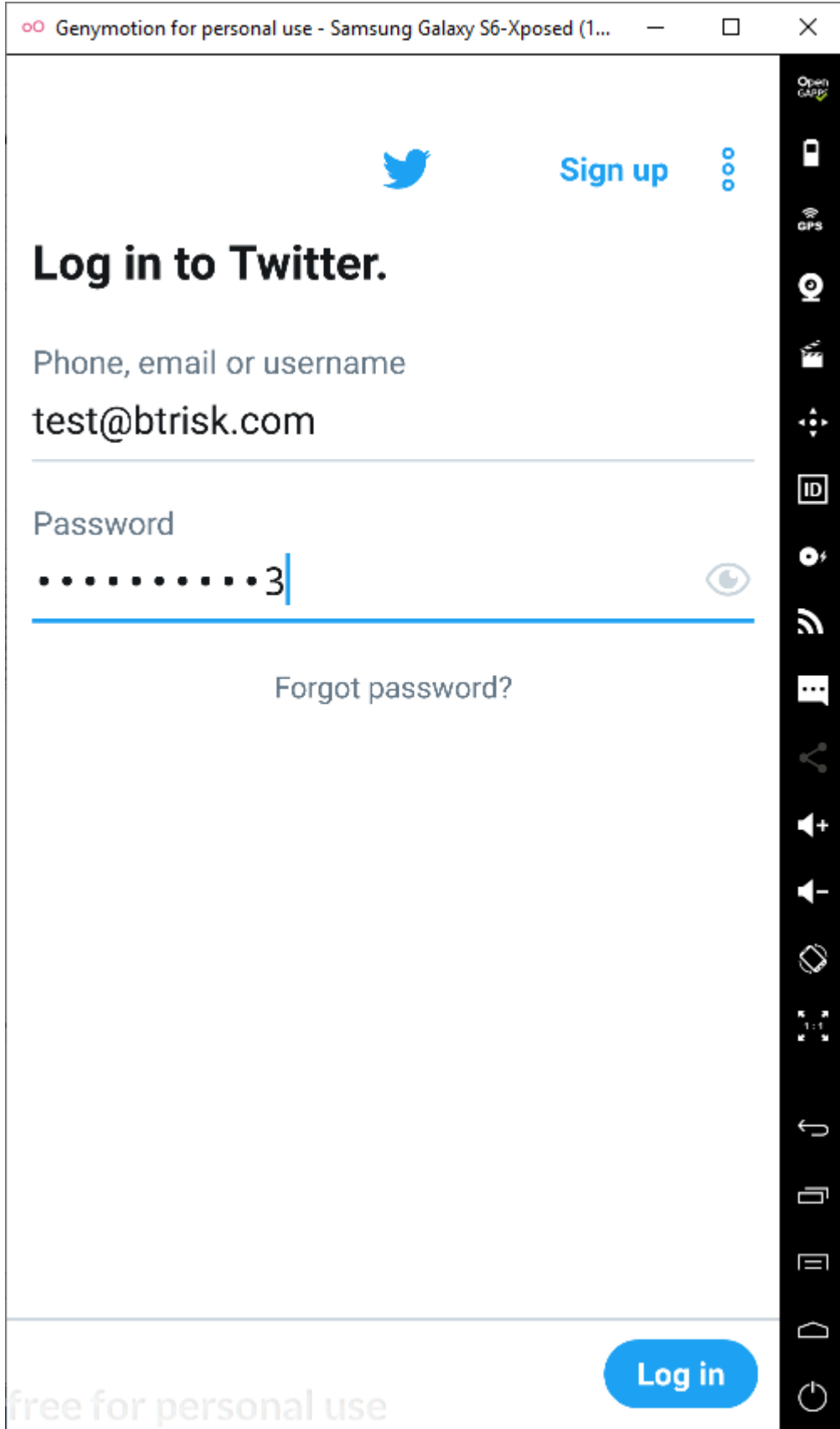
Running	Interface	Invisible	Redirect	Certificate
<input checked="" type="checkbox"/>	192.168.1.105:8080			Per-host

Buttons for 'Add', 'Edit', and 'Remove' are visible. Below the table, there is a text box explaining that each installation of Burp generates its own CA certificate, and buttons for 'Import / export CA certificate' and 'Regenerate CA certificate' are provided. The section is followed by 'Intercept Client Requests' and 'Intercept Server Responses' sections, each with a help icon and a text box explaining their purpose. The 'Intercept Client Requests' section has a checked checkbox for 'Intercept requests based on the following rules: Master interception is turned off' and a table of rules:

Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>		File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$...
<input type="checkbox"/>	Or	Request	Contains parameters	
<input type="checkbox"/>	Or	HTTP method	Does not match	(get post)
<input type="checkbox"/>	And	URL	Is in target scope	

Buttons for 'Add', 'Edit', 'Remove', 'Up', and 'Down' are visible. Below the table, there are checkboxes for 'Automatically fix missing or superfluous new lines at end of request' (unchecked) and 'Automatically update Content-Length header when the request is edited' (checked).

Uygulamayı açalım ve login almaya çalışalım.



Burp Suite'den dinlediđimizde login isteđini görebiliriz.

Burp Suite Professional v2020.4 - Temporary Project - licensed to BTRisk Bilgi Güvenliği ve BT Yönetişim Hizmetleri Tic.Ltd.Ski [2 user license]

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options JSWS Parser AES Crypto JSON Beautifier Wsdler

Intercept HTTP history WebSockets history Options

Filter: Showing all items

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
2877	https://api.twitter.com	GET	/robots.txt			200	427	text	txt			✓	104.244.42.194		23:26:19 3 ...	8080
2878	https://api.twitter.com	GET	/1.1/traffic/recommendations.json?...	✓		429	385	JSON	json			✓	104.244.42.194		23:26:20 3 ...	8080
2879	https://pbs.twimg.com	GET	/robots.txt			200	516	text	txt			✓	93.184.220.70		23:26:20 3 ...	8080
2880	https://video.twimg.com	GET	/robots.txt			200	335	text	txt			✓	151.101.240.158		23:26:20 3 ...	8080
2881	https://api.twitter.com	POST	/1.1/onboarding/task.json?flow_na...	✓		200	11608	JSON	json			✓	104.244.42.194		23:26:20 3 ...	8080
2882	https://api.twitter.com	POST	/auth/i/xauth_password.json	✓		401	979	JSON	json			✓	104.244.42.194		23:26:44 3 ...	8080
2883	https://api.twitter.com	POST	/1.1/guest/activate.json			200	777	JSON	json			✓	104.244.42.194		23:26:44 3 ...	8080
2884	https://api.twitter.com	POST	/auth/i/xauth_password.json	✓		401	979	JSON	json			✓	104.244.42.194		23:26:45 3 ...	8080
2885	https://api.twitter.com	POST	/auth/i/xauth_password.json	✓		401	979	JSON	json			✓	104.244.42.194		23:26:45 3 ...	8080

Request Response

Raw Params Headers Hex

```
1 POST /auth/i/xauth_password.json HTTP/1.1
2 Cache-Control: no-store
3 X-B3-Traceid: 55c24b6569f84addf
4 X-Twitter-Client-Flavor:
5 User-Agent: TwitterAndroid/8.26.0-release.00 (10260000-r-0) Samsung/5.0 (unknown;Samsung;generic;vbox86p;);;12014
6 Accept-Encoding: gzip, deflate
7 X-Twitter-Client-AdID: bffabdc5-03b4-4a04-9e67-a38247457d7
8 Timezone: America/New_York
9 X-Twitter-Client-Like-Ad-Tracking: 0
10 X-Twitter-Client-DeviceID: 812e4ade04e5300b
11 X-Twitter-Client-Language: en-US
12 X-Twitter-Client: TwitterAndroid
13 X-Twitter-API-Version: 5
14 Optimize-Body: true
15 X-Twitter-Active-Beer: yes
16 X-Twitter-Client-Version: 8.26.0-release.00
17 X-Guest-Token: 1257042c24793411595
18 X-Client-UUID: 4c740c0b-7665-4b71-a230-13b5147b5d0d
19 Accept: application/json
20 Authorization: Bearer AAAAAAAAAAAAAAAAAAAAFcAwAAAAAAAAAMHCXpeSDG1qLNLghVe8d74b16x4v3DpUMF&aQLsbeBHTSRc1QpJtxoGweyHdB5tce3jps6xWDFW62F
21 Accept-Language: en-US
22 Content-Type: application/x-www-form-urlencoded
23 Content-Length: 173
24 Host: api.twitter.com
25 Connection: close
26 Cookie: personalization_id=vi_EmP2ppWlhhixSPMApNSA==; guest_id=vi33a15865376977869165
27
28 X_auth_identifier=test40btrisk.com&x_auth_password=password123&send_error_codes=true&x_auth_login_challenge=1&x_auth_login_verification=1&x_auth_country_code=US&x_auth_metrics=
```

test@btrisk.com
Press F2 for focus

2FwACXp6oRzq1FRNOKUKkX7M0xspAH 0 matches Pretty

Tüm trafiği buradan görebiliriz.

IV. IOS'DA SSL KILL SWITCH KULLANARAK SSL PINNING ATLATMA

SSL Kill Switch Nedir?

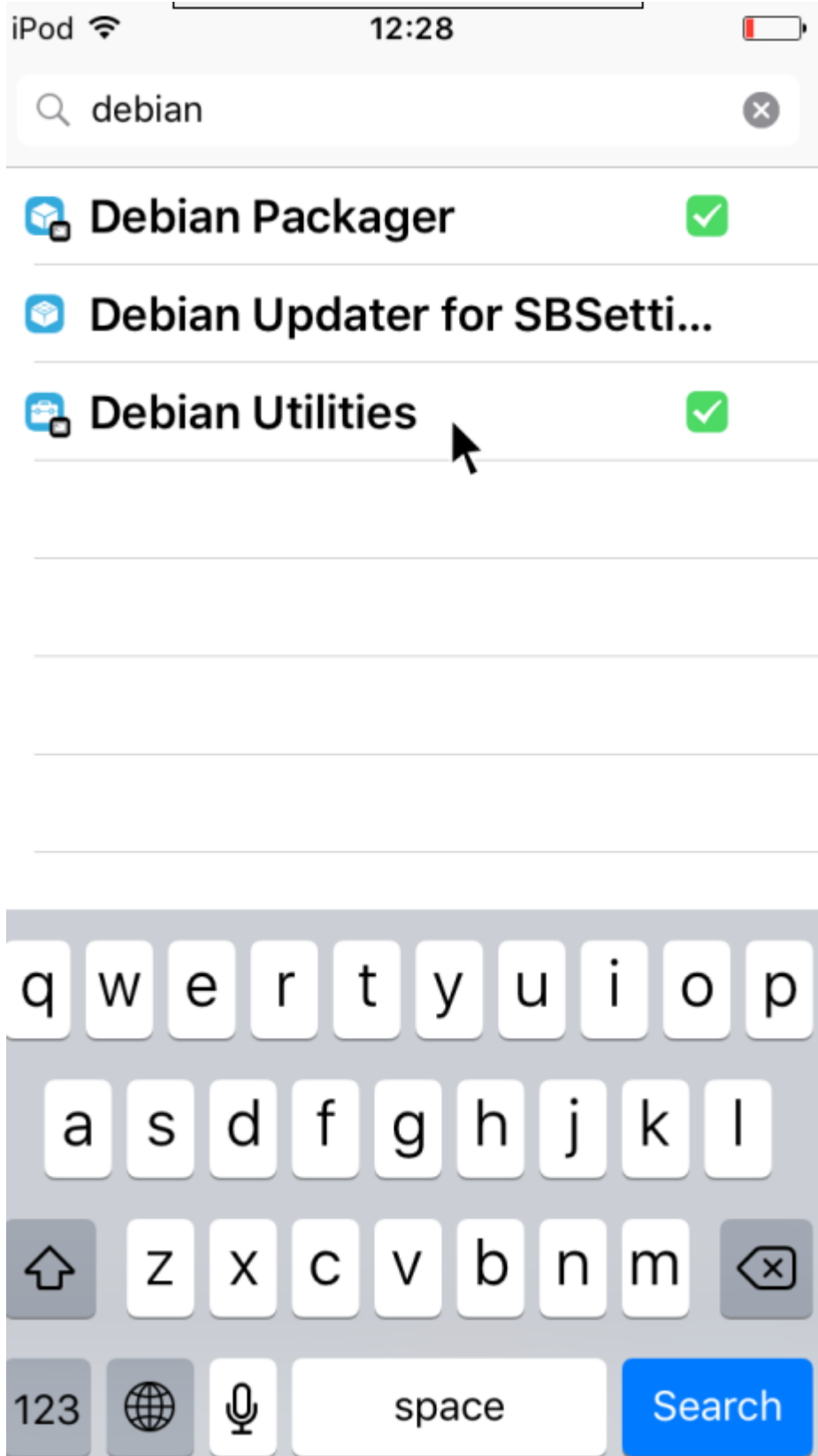
Jailbreak yapılmış IOS cihaza Secure Transport API içinde SSLSetSessionOption() ve SSLHandshake() fonksiyonlarını haklarını ezerek düşük seviyede SSL fonksiyonlarını deđiştirir. Sistemin varsayılan sertifika dođrulamasını ve her türlü özel sertifika dođrulamasını (SSL pinning) devre dışı bırakır. SSL Kill Switch 2 devamı niteliğinde ve daha güncel olanları desteklemektedir.

Eđer programcı standart platform API'leri yerine OpenSSL veya benzeri farklı bir kütüphaneyi kullanırsa SSL Kill Switch etkisiz olacaktır. Kullandığımız SSL Kill Switch IOS 2012 Blackhat'de tanıtıldı.

Kurulum için aşağıdakilerin cydia üzerinden kurulu olması gerekmektedir.

- Debian Packager
- Cydia Substrate
- PreferenceLoader
- dpkg

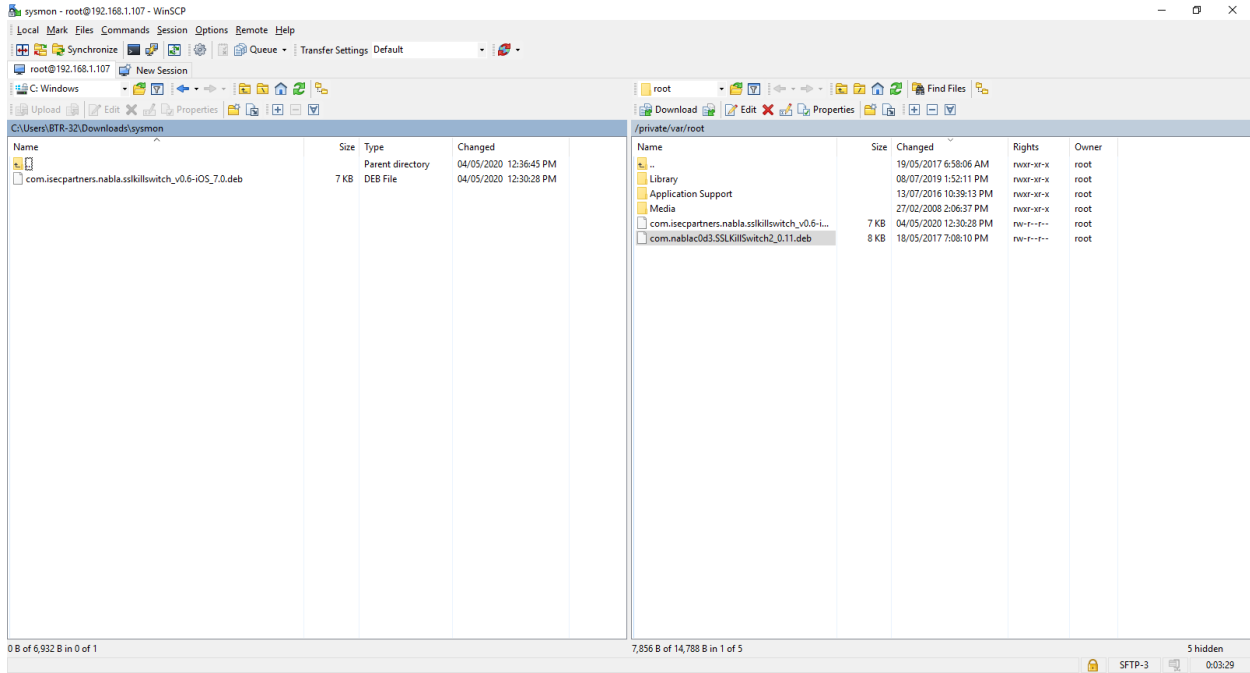
Kurmak için Cydia üzerinden dosyaları arayarak bulabilirsiniz. 3 gerekli paketi de arayarak yükleyelim.
Kullanılan iOS versiyon 9.3.2



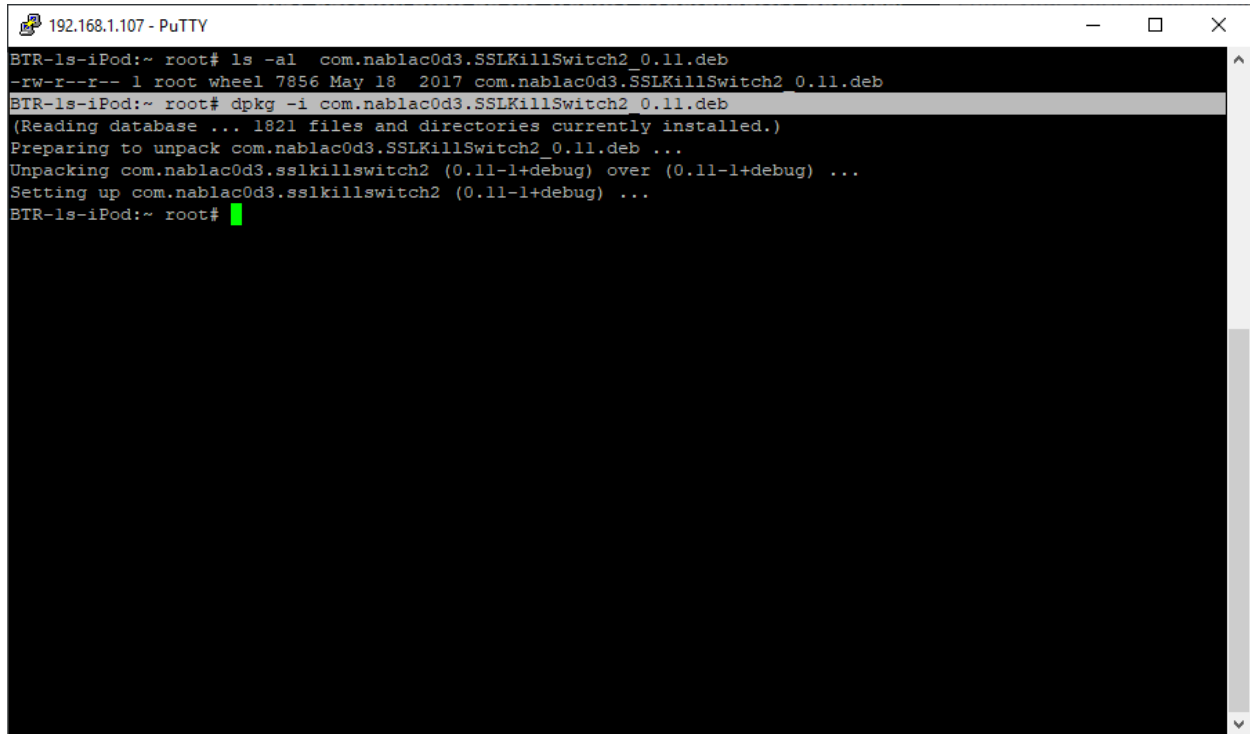
SSL Kill Switch'in GitHub sayfasının yayın sekmesinde bulunan en son derlenmiş paketi indirin. Aygıtı kopyalayın ve yükleyin.

<https://github.com/iSECPartners/ios-ssl-kill-switch/releases>

Yüklenen dosyayı WinSCP ile atabiliriz.



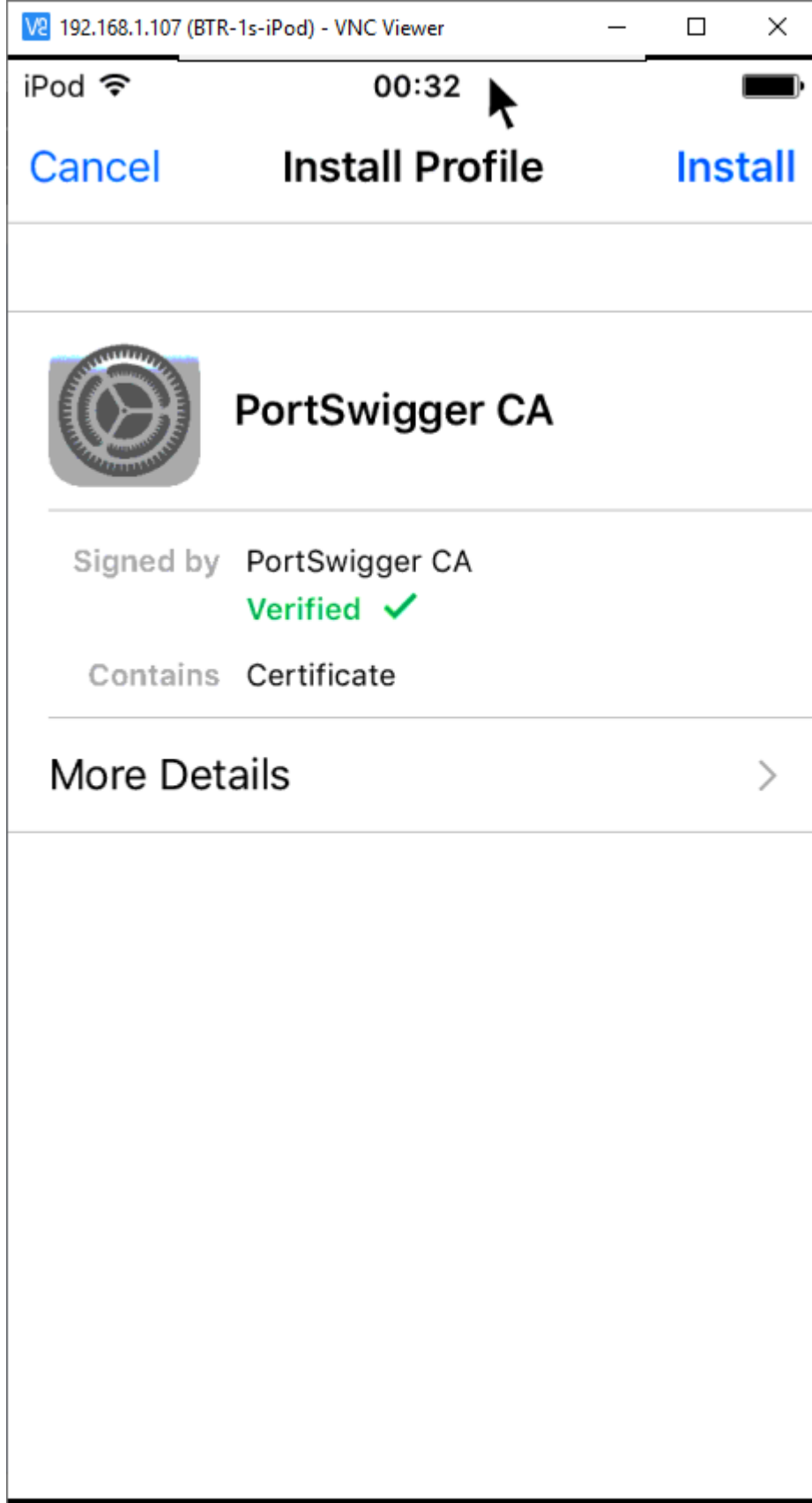
Upload ettiğimiz paketi kuralım.



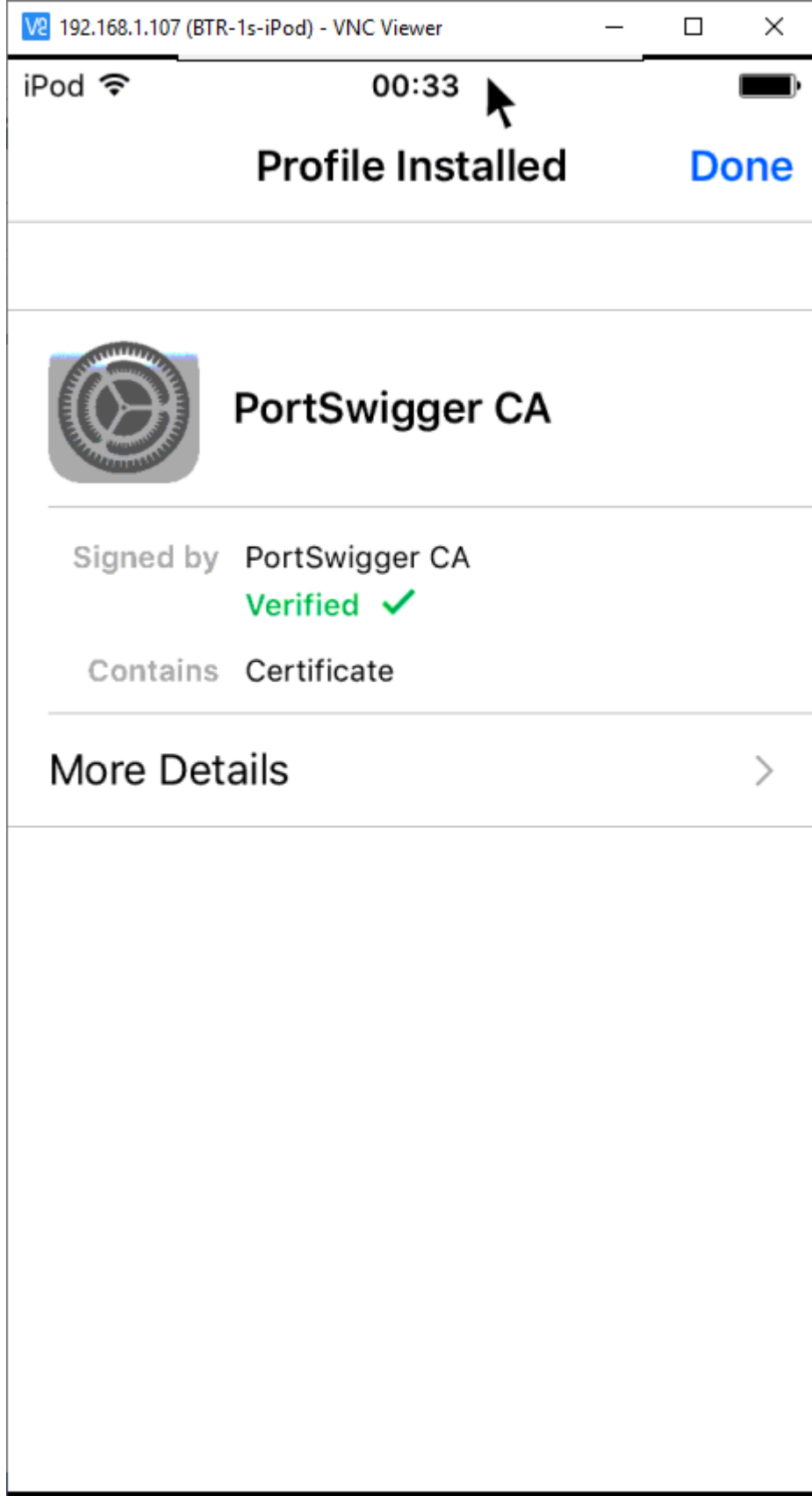
Uygulamanın burp sertifikasına güvenmesi için burp'un servis verdiği ip:port adresine browser üzerinden erişelim.



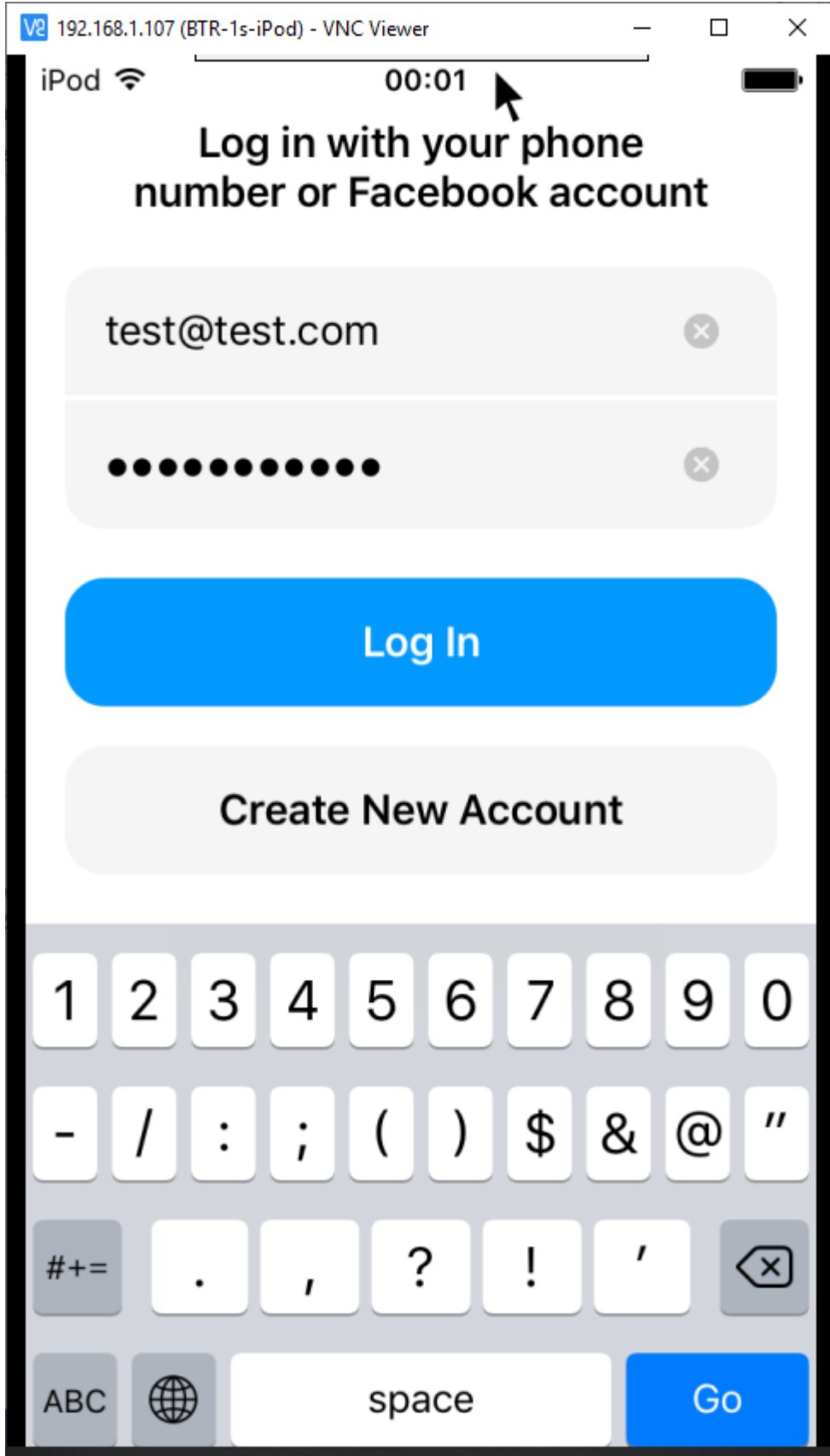
“CA Certificate” tıklayarak sertifikayı yükleyelim.



Yüklemeniz başarılıysa, doğrulama mesajını göreceksiniz.



Uygulamayı bir kez giriş yapmaya çalıştığımızda



SSL hatası aldık.

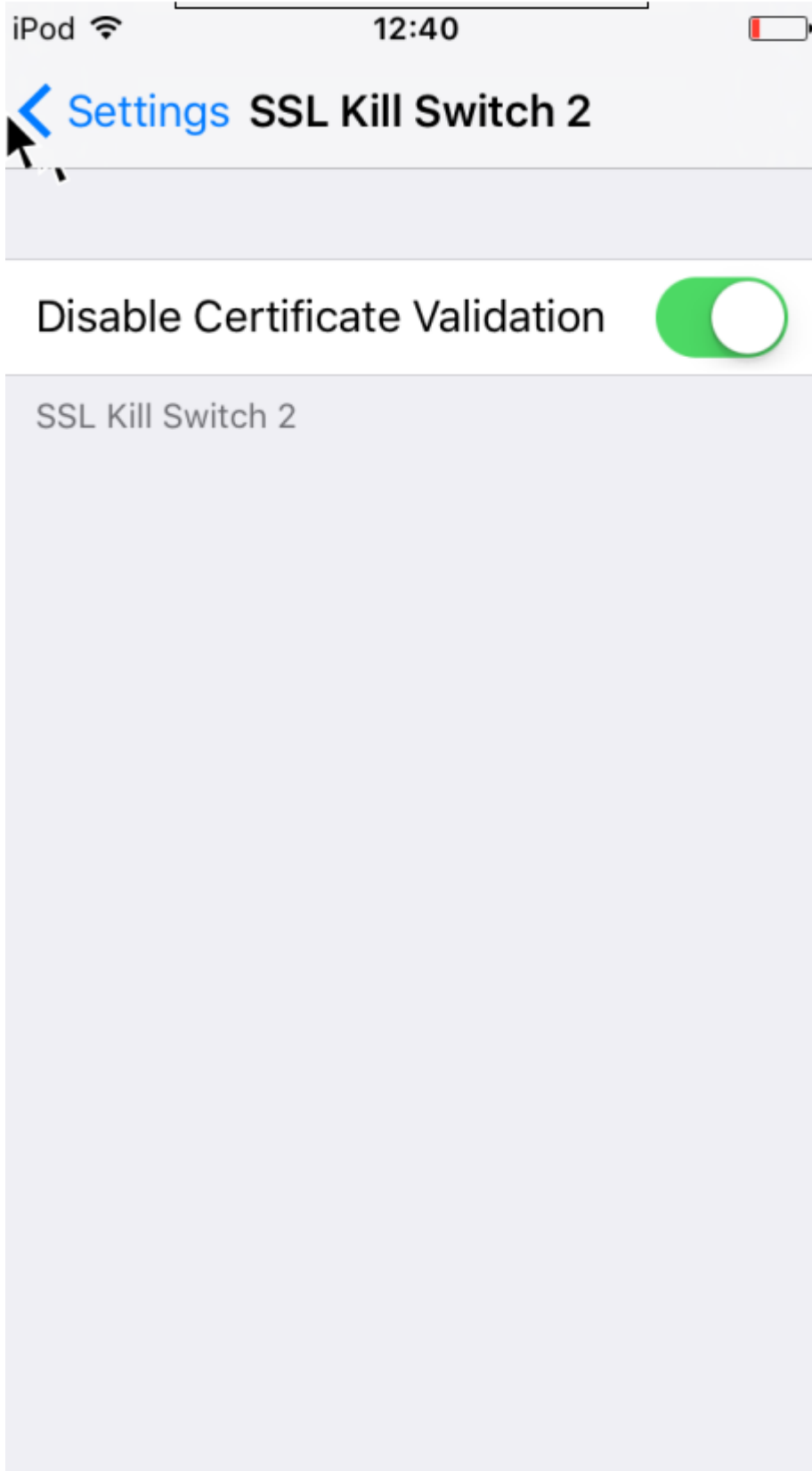
Event log ? ↗

Filter **Critical** **Error** **Info** **Debug** Search...

Time	Type	So...	Message
00:02:51 5 May 2020	Error	Proxy	The client failed to negotiate a TLS connection to graph.facebook.com:443: Remote host terminated the handshake
23:54:11 4 May 2020	Info	Proxy	Proxy service stopped on 127.0.0.1:8080
23:54:20 4 May 2020	Info	Proxy	Proxy service started on 192.168.1.106:8080
23:54:06 4 May 2020	Info	Proxy	Proxy service started on 127.0.0.1:8080

Şimdi SSL Kill Switch izin vererek araya girmeye çalışalım.

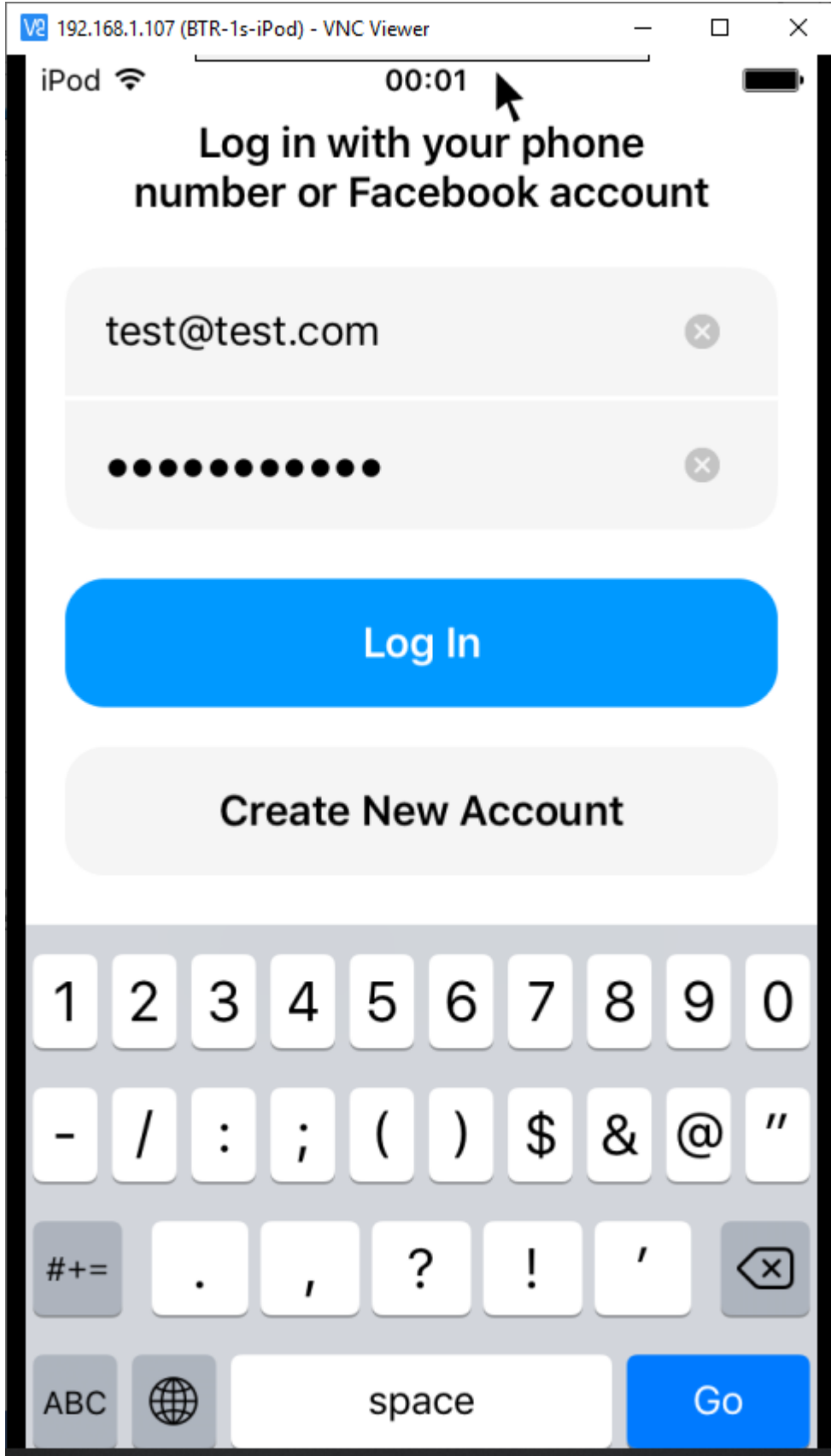
Settings → SSL Kill Switch 2



Bu işlem yapıldığında uygulamaya enjekte olduğunu gösteren bir örnek aşağıda bulabilirsiniz.

```
Oct 28 15:05:45 Prateeks-IPad kernel[0] <Debug>: launchd[387] Container: /private/var/mobile/  
Applications/0F2A7AB7-1E95-47B1-A7D3-33F0DB76B7C3 (sandbox)  
Oct 28 15:05:46 Prateeks-IPad Twitter[387] <Notice>: MS:Notice: Injecting: com.atebits.Tweetie2  
[Twitter] (847.24)  
Oct 28 15:05:46 Prateeks-IPad Twitter[387] <Notice>: MS:Notice: Loading: /Library/  
MobileSubstrate/DynamicLibraries/SSLKillSwitch.dylib  
Oct 28 15:05:46 Prateeks-IPad Twitter[387] <Warning>: SSL Kill Switch - Hook Enabled.  
Oct 28 15:05:46 Prateeks-IPad backboardd[33] <Error>: HID: The 'Passive' connection 'Twitter'  
access to protected services is denied.  
Oct 28 15:05:47 Prateeks-IPad Twitter[387] <Error>: Could not successfully update network info  
during initialization.
```

Tekrar login olmaya çalışalım.



Burp'den paketleri izleyebiliriz. Parola için bir hashleme yapmış olabilirler fakat kullanıcı adını açık bir şekilde görebiliyoruz.

The screenshot shows the Burp Suite Professional interface. The top menu bar includes Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Project options, User options, JSWS Parser, AES Crypto, JSON Beautifier, Wsdler, and SQLPy. The main window displays a table of intercepted items. The selected item is a POST request to https://graph.facebook.com/v2.10/auth/login. The request details are shown in the lower pane, including headers and form data.

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
14	https://graph.facebook.com	GET	/pwd_key_fetch?access_token=43...	✓		200	774	JSON				✓	185.60.218.19
15	https://graph.facebook.com	POST	/v2.10/auth/login	✓		400	1006	JSON				✓	185.60.218.19

```
Request
Response
Raw Params Headers Hex
1 POST /v2.10/auth/login HTTP/1.1
2 Host: graph.facebook.com
3 Content-Type: multipart/form-data; boundary=0FFe53AD9CC2461C977F9D2CE3BF57C0
4 Accept-Encoding: gzip, deflate
5 Connection: close
6 Accept: */*
7 User-Agent: LightSpeed [FBAN/MessengerLiteForiOS;FBAV/262.1.0.71.117;FBBV/211898162;FBDV/iPod7,1;FBMD/iPod touch;FBSN/iPhone
8 OS;FBSV/9.3.2;FBSS/2;FBCR;/;FBID/phone;FBLC/en_US;FBOP/0]
9 Content-Length: 1742
10 request_token: F905A09A-05ED-4FEE-80FF-4EA664300B30
11 Accept-Language: en-gb
12
13 --0FFe53AD9CC2461C977F9D2CE3BF57C0
14 Content-Disposition: form-data; name="access_token"
15
16 43762631e97378813e1a7033ae7883bfb31f35375bad9c7a
17 --0FFe53AD9CC2461C977F9D2CE3BF57C0
18 Content-Disposition: form-data; name="app_id"
19
20 43762631e973788
21 --0FFe53AD9CC2461C977F9D2CE3BF57C0
22 Content-Disposition: form-data; name="credentials_type"
23
24 password
25 --0FFe53AD9CC2461C977F9D2CE3BF57C0
26 Content-Disposition: form-data; name="device_id"
27
28 A52AAFB0-8A00-48e5-B965-F2E0D71B700B
29 --0FFe53AD9CC2461C977F9D2CE3BF57C0
30 Content-Disposition: form-data; name="email"
31
32 test@test.com
33 --0FFe53AD9CC2461C977F9D2CE3BF57C0
34 Content-Disposition: form-data; name="error_detail_type"
35
36 button_with_disabled
37 --0FFe53AD9CC2461C977F9D2CE3BF57C0
```

V.IOS'DA BURP SUITE MOBILE ASSISTANT KULLANARAK SSL PINNING ATLATMA

Burp Suite Mobile Assistant Nedir?

Burp Suite Mobile Assistant low-level system API seviyesinde uygulamaya hook olarak sertifika pinning mekanizmasını atlatarak test uygulamasına enjekte olmakta ve böylece Burp Suite ile trafiği izlememize olanak sunmaktadır. Jailbreak yapılmış cihazda çalışmaktadır. Jailbreak genellikle Mobile Assistant'ı yüklemek için kullanılabilen popüler paket yöneticisi Cydia'yı kullanır.

MobileAssistant uygulaması 28 Nisan 2017 tarihinde duyurulmuş. iPhones, iPods ve iPads'da versiyon 8.0'den 10'a kadar desteklemektedir.

Burp Suite bize bir servis sağlamaktadır. Bunu aktif edelim.

The screenshot shows the Burp Suite Professional v2020.4 interface. The main window is titled "Proxy Listeners" and contains a table with the following columns: Add, Edit, Remove, Running, Interface, Invisible, Redirect, and Certificate. The "Running" column has a checked checkbox, and the "Interface" column shows the IP address "192.168.1.106:8080". The "Certificate" column shows "Per-host". Below the table, there are buttons for "Import / export CA certificate" and "Regenerate CA certificate".

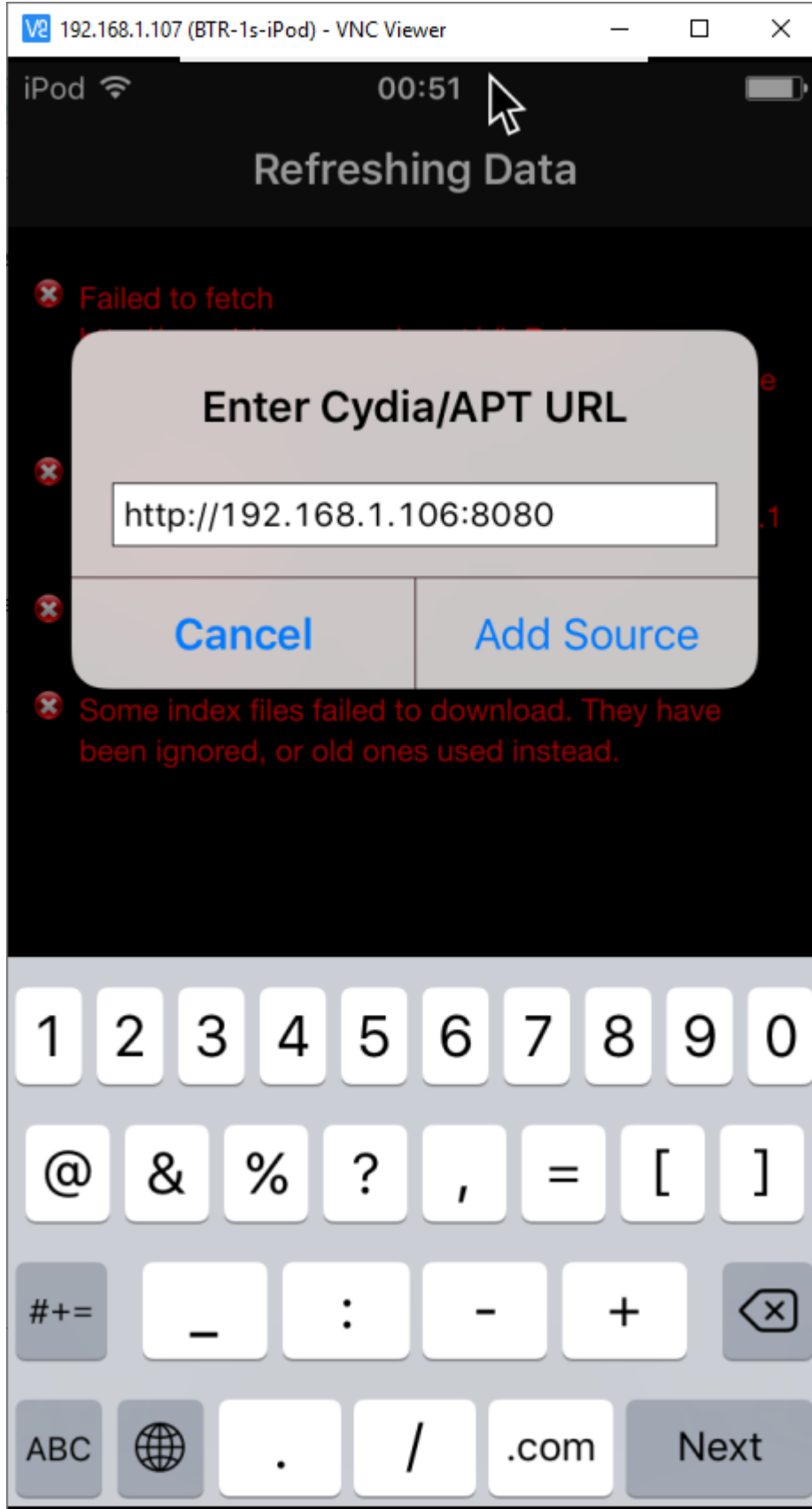
The "Intercept Client Requests" section is also visible, showing a table with columns: Add, Edit, Remove, Up, Down, Enabled, Operator, Match type, Relationship, and Condition. The "Enabled" column has a checked checkbox. The "Match type" column shows "File extension", "Request", and "URL". The "Relationship" column shows "Does not match", "Contains parameters", and "Does not match". The "Condition" column shows "(!gif|!jpg|!png|!css|!js|!...)", "Contains parameters", and "Is in target scope".

Jailbreakli cihazda Cydia üzerinden MobileAssistant paketini yükleyelim. Cihaz üzerinde Cydia uygulamasını açalım.

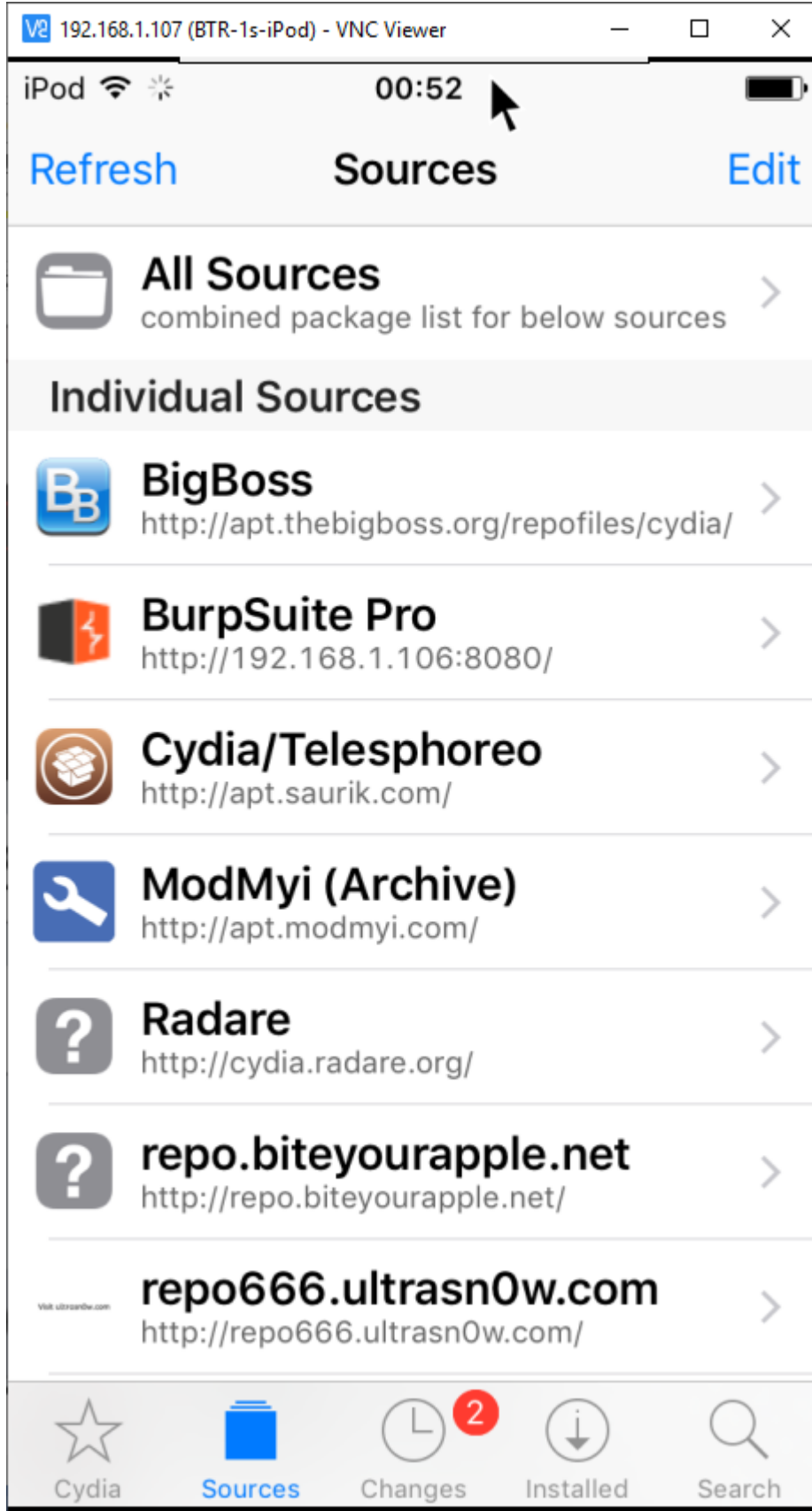


Repo URL yüklememiz için Burp Suite'in hizmet verdiği 8080 portunu ekleyelim.

Cydia → Sources → Edit → Add Ekledikten sonra kaynak dođrulanacaktır.

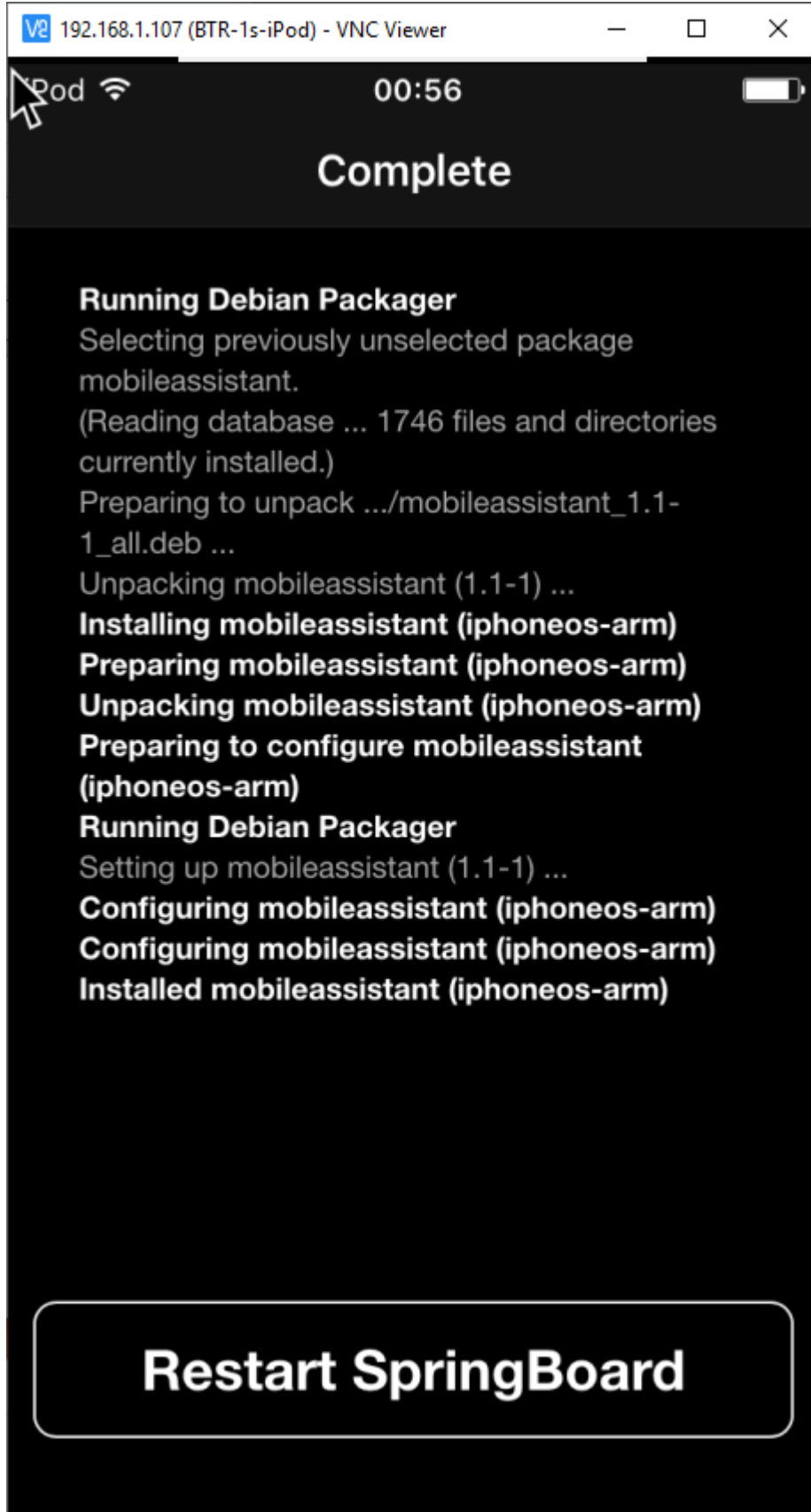


Sources listesinde eklediđimiz Burp Suite reposunu göreceđiz.



Burp Suite reposuna tıklayarak kuruluma devam edelim.

BurpSuite Pro → MobileAssistant → Install

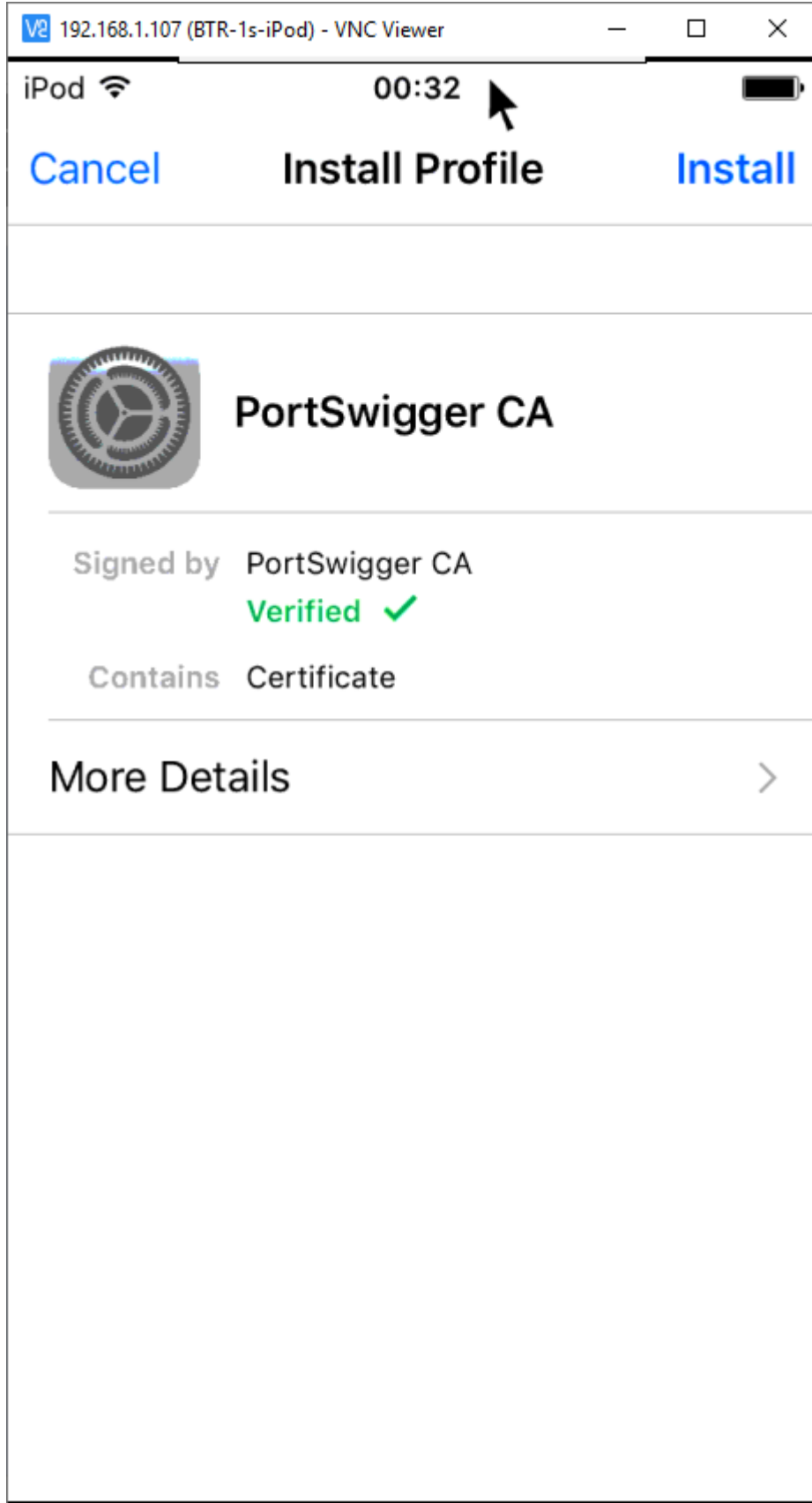


Restart yaptıktan sonra uygulama olarak iconunu göreceđiz. Sertifikayı yüklemeyi unutmayalım.

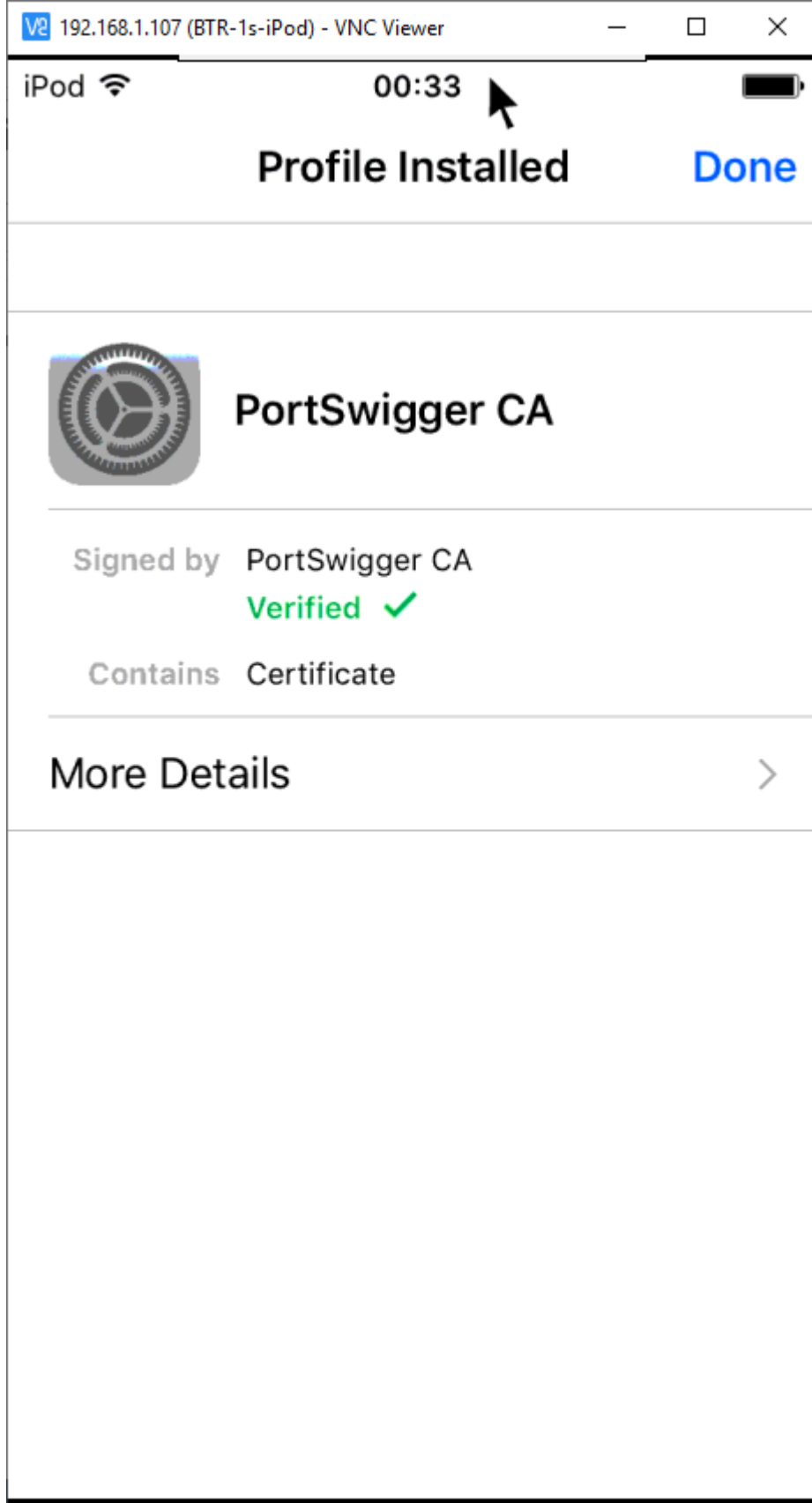
Uygulamanın burp sertifikasına güvenmesi için burp'un servis verdiği ip:port adresine browser üzerinden erişelim.



Yükleyelim.

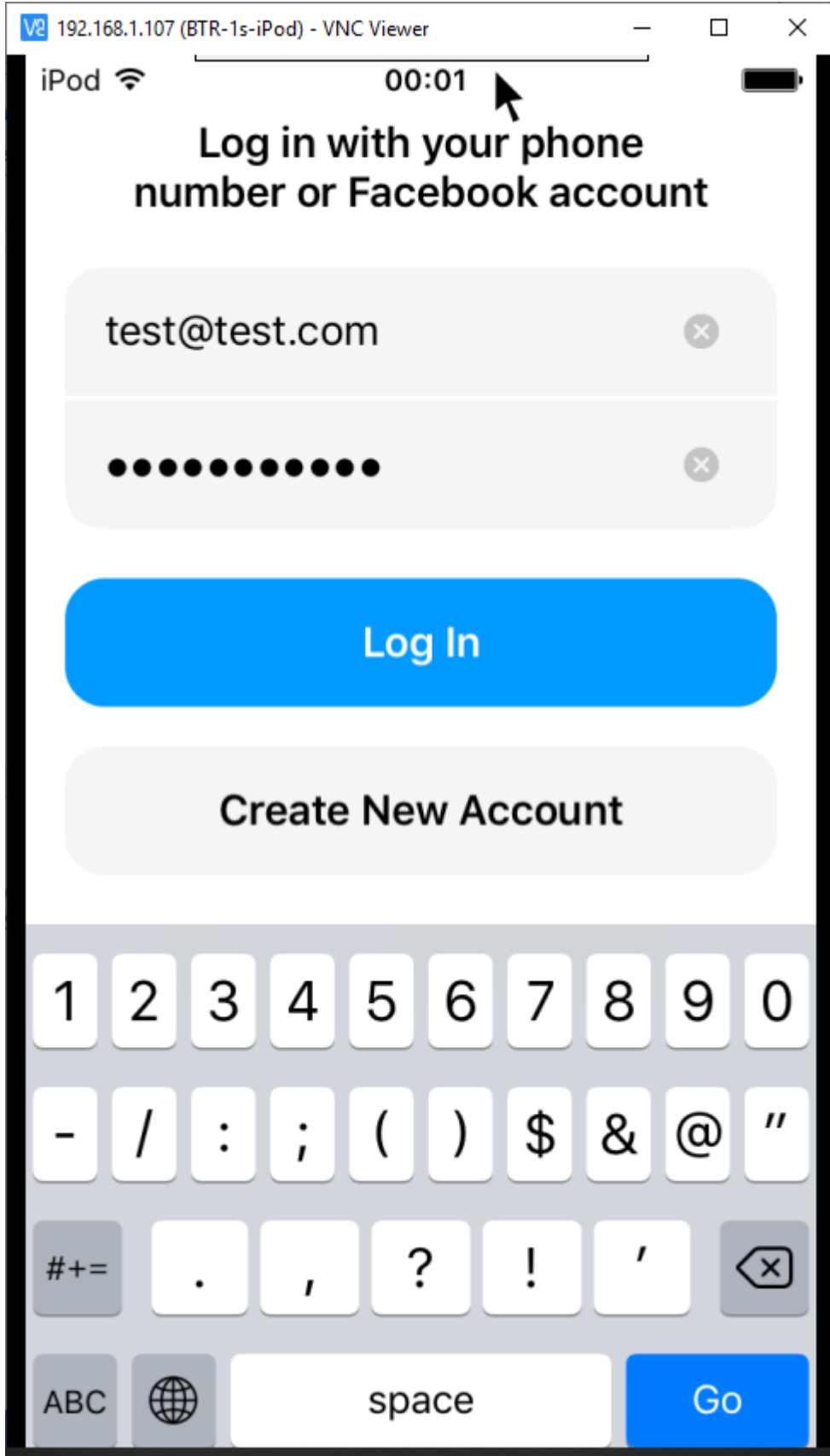


Yüklemeniz başarılıysa, doğrulama mesajını göreceksiniz.



Sertifika yükleme işlemini MobileAssistant üzerinden de kurabiliriz. “CA Certificate” Install diyerek kurabiliriz.

Uygulamayı bir kes giriş yapmaya çalıştığımızda



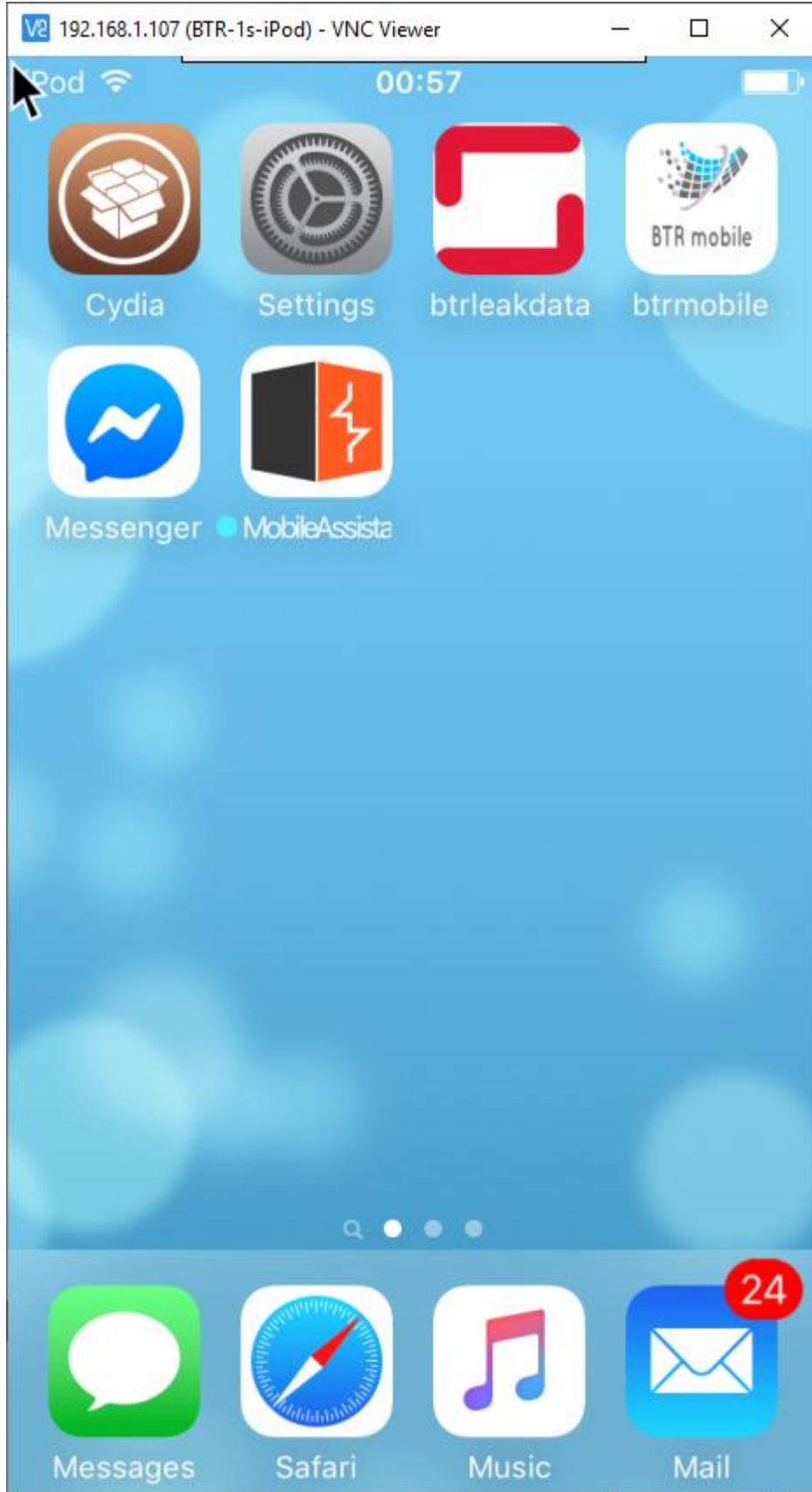
SSL hatası aldık.

Event log ? ↗

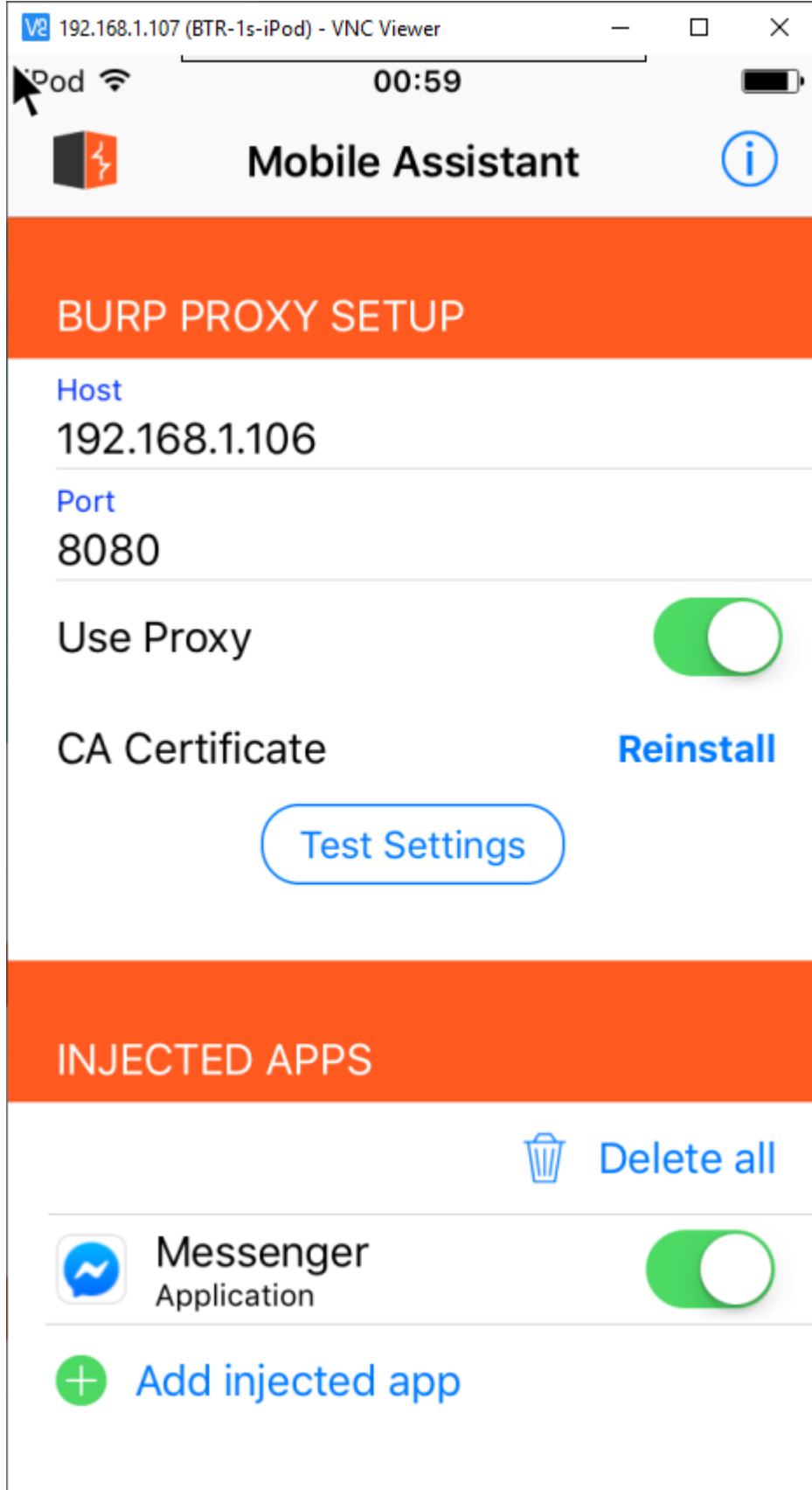
Filter **Critical** **Error** **Info** **Debug** Search...

Time	Type	So...	Message
00:02:51 5 May 2020	Error	Proxy	The client failed to negotiate a TLS connection to graph.facebook.com:443: Remote host terminated the handshake
23:54:11 4 May 2020	Info	Proxy	Proxy service stopped on 127.0.0.1:8080
23:54:20 4 May 2020	Info	Proxy	Proxy service started on 192.168.1.106:8080
23:54:06 4 May 2020	Info	Proxy	Proxy service started on 127.0.0.1:8080

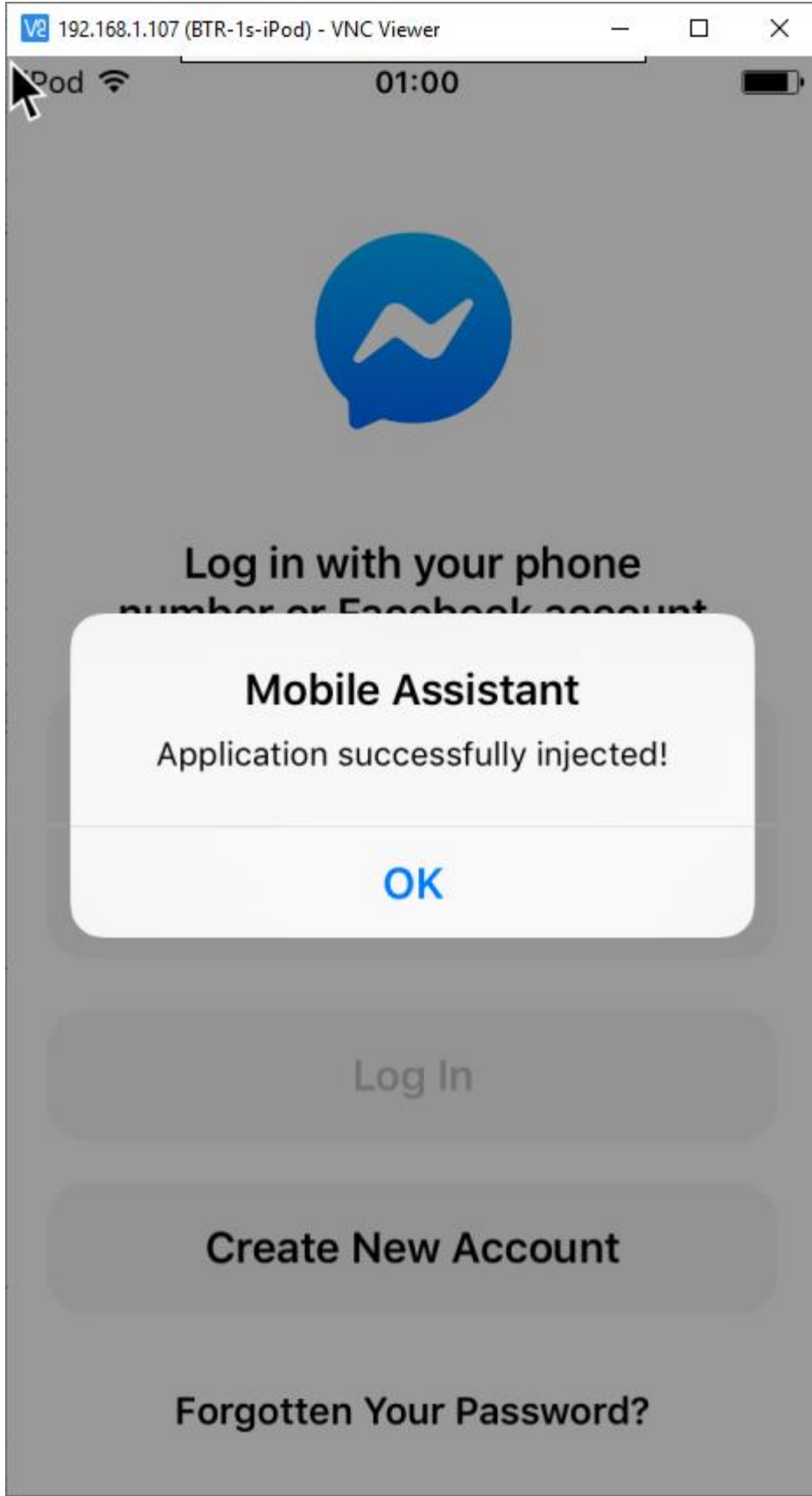
Şimdi MobileAssistant uygulamasını açalım.



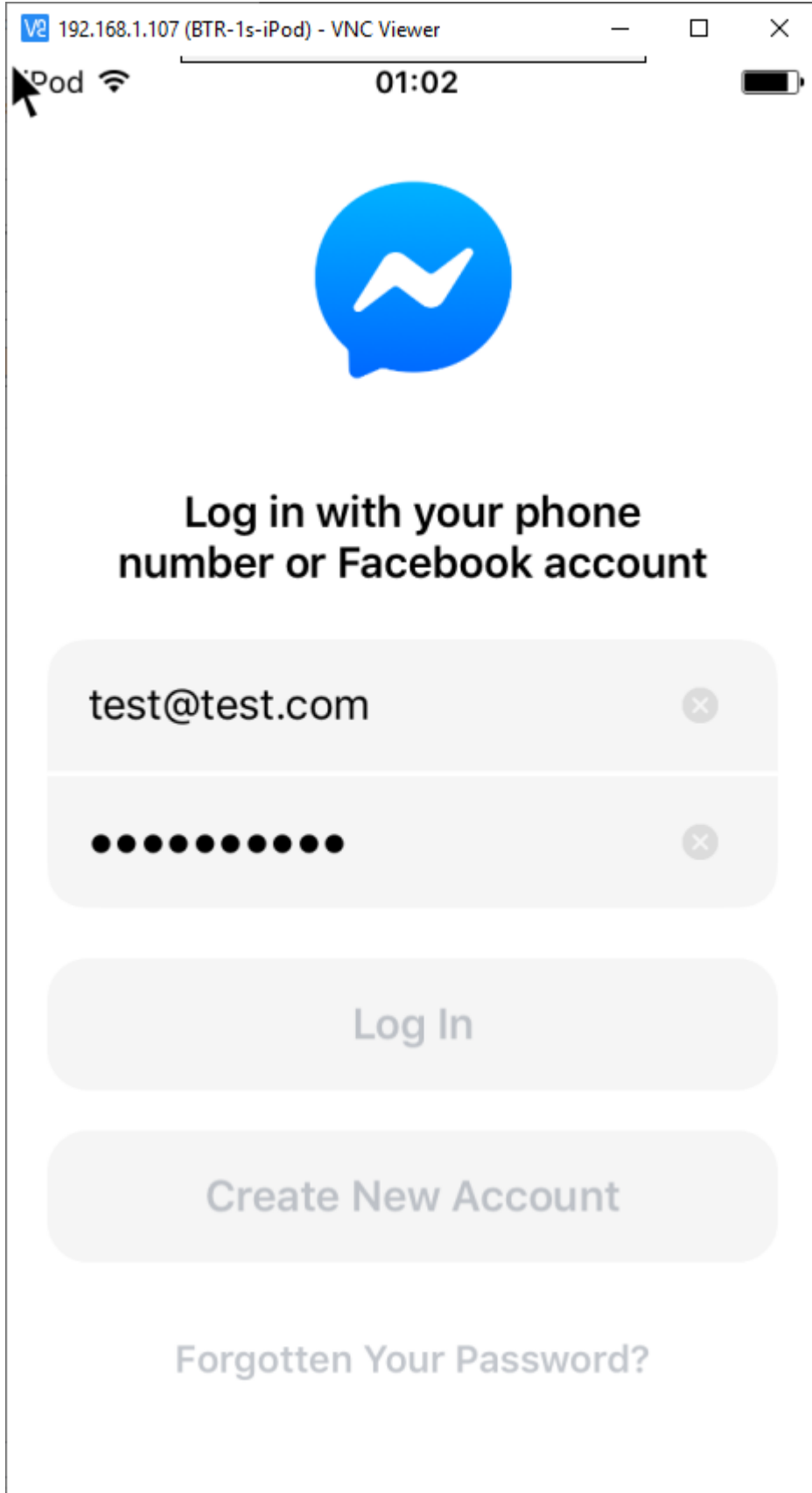
Proxy ayarlarını ve runtime'da hook olması için sertifika kontrolünü disable edeceğimiz uygulamayı seçelim. İşlemlerin doğru olduğunu da "Test Settings" ile kontrol edebiliriz.



Uygulamayı açtığımızda bir pop-up ile bizi başarılı olduğunu belirtiyor.



Login bilgilerini girerek login olalım.



Burpden trafiđi gözlemlediđimizde kullanıcı adını görebildik.

The screenshot displays the Burp Suite Professional v2020.4 interface. At the top, there is a menu bar with options like 'Dashboard', 'Target', 'Proxy', 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Extender', 'Project options', 'User actions', 'JSWS Parser', 'AES Crypto', 'JSON Beautifier', 'Wsdler', and 'SQLPy'. Below the menu, there is a 'Filter: Showing all items' section. The main area shows a table of intercepted requests with columns for #, Host, Method, URL, Params, Edited, Status, Length, MIME type, Extension, Title, Comment, TLS, IP, Cookies, Time, and Listener port. Three requests are listed, all from https://graph.facebook.com. The first is a POST to /setup/ikcloud.com, the second is a GET to /setup/family/getFamilyDetails, and the third is a POST to /v2.10/auth/login. The third request is selected, and its raw content is displayed in the 'Raw' tab below. The raw content shows a multipart form-data request with various fields including 'request_token', 'password', 'device_id', 'error_detail_type', and 'family_device_id'.

Mobile Assistant SSL pinning atlattı ve uygulama verileri sunucu ve istemci arasında iletmek için Burp Suite'in imzaladığı sertifikaya güvendi.

Mobile Assistant'ı edinmek isterseniz, web tarayıcısını BurpSuite'in tarayıcı içi ara yüzüne, yani <http://192.168.1.6:8080/mobileassistant.deb>'ye yönlendirerek yapabilirler.

VI. IOS'DA OBJECTION KULLANARAK SSL PINNING ATLATMA

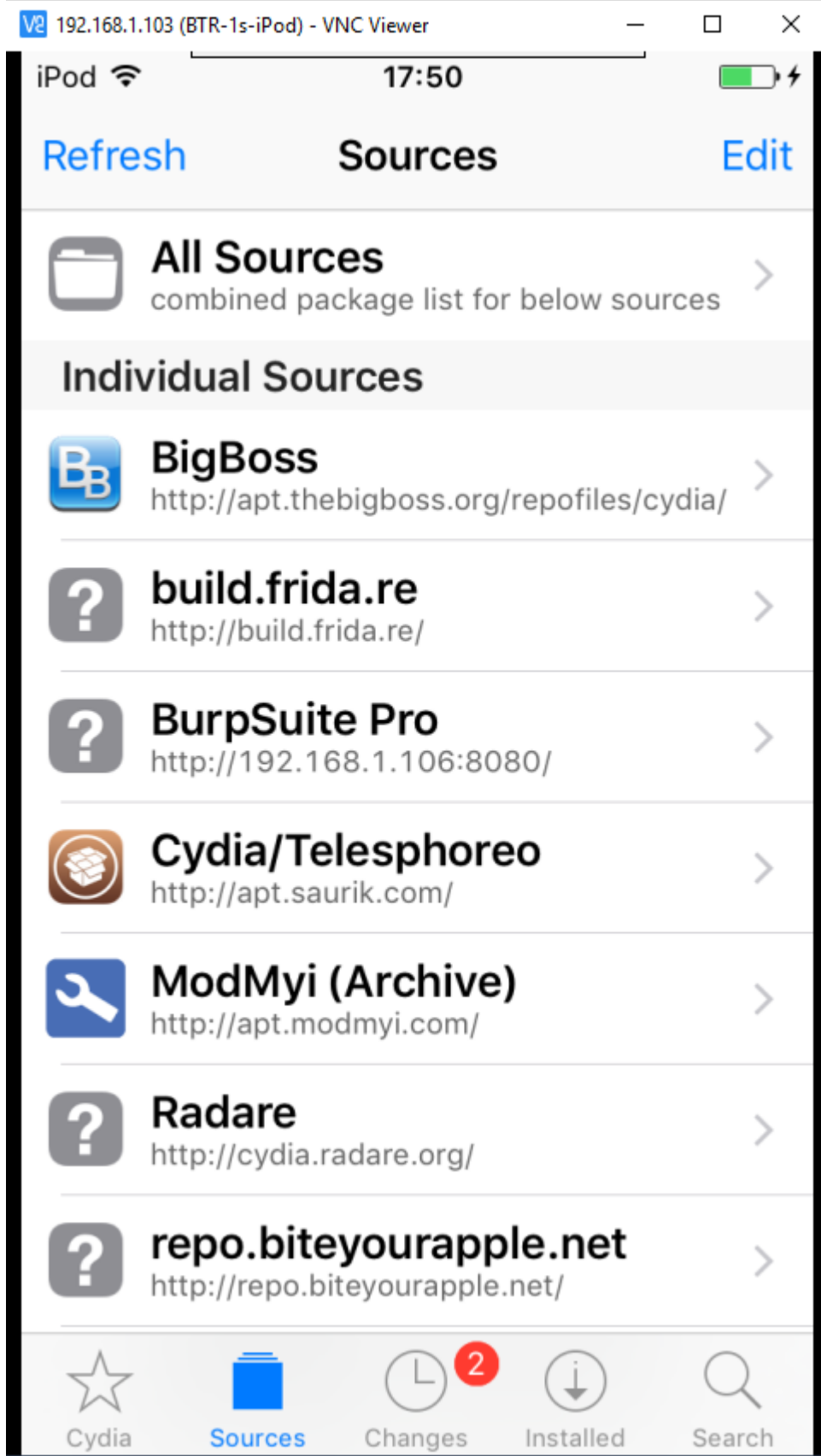
Objection nedir?

Objection frida tarafından desteklenen mobil uygulamalara runtime anında işlemler yapmanıza yarayan runtime mobil keşif aracıdır. iOS ve Android desteklemektedir. Runtime root kontrolünde atlatabildiğiniz gibi SSL pinning de atlatabilirsiniz, dosyaları görüntüleme, TouchID atlatma ve daha birçok şey yapılabilir. Runtime'da deđişiklikler yapmaya izin verdiğiinden beklenen deđerleri deđiştirerek istenilen şeyler yapılabilir.

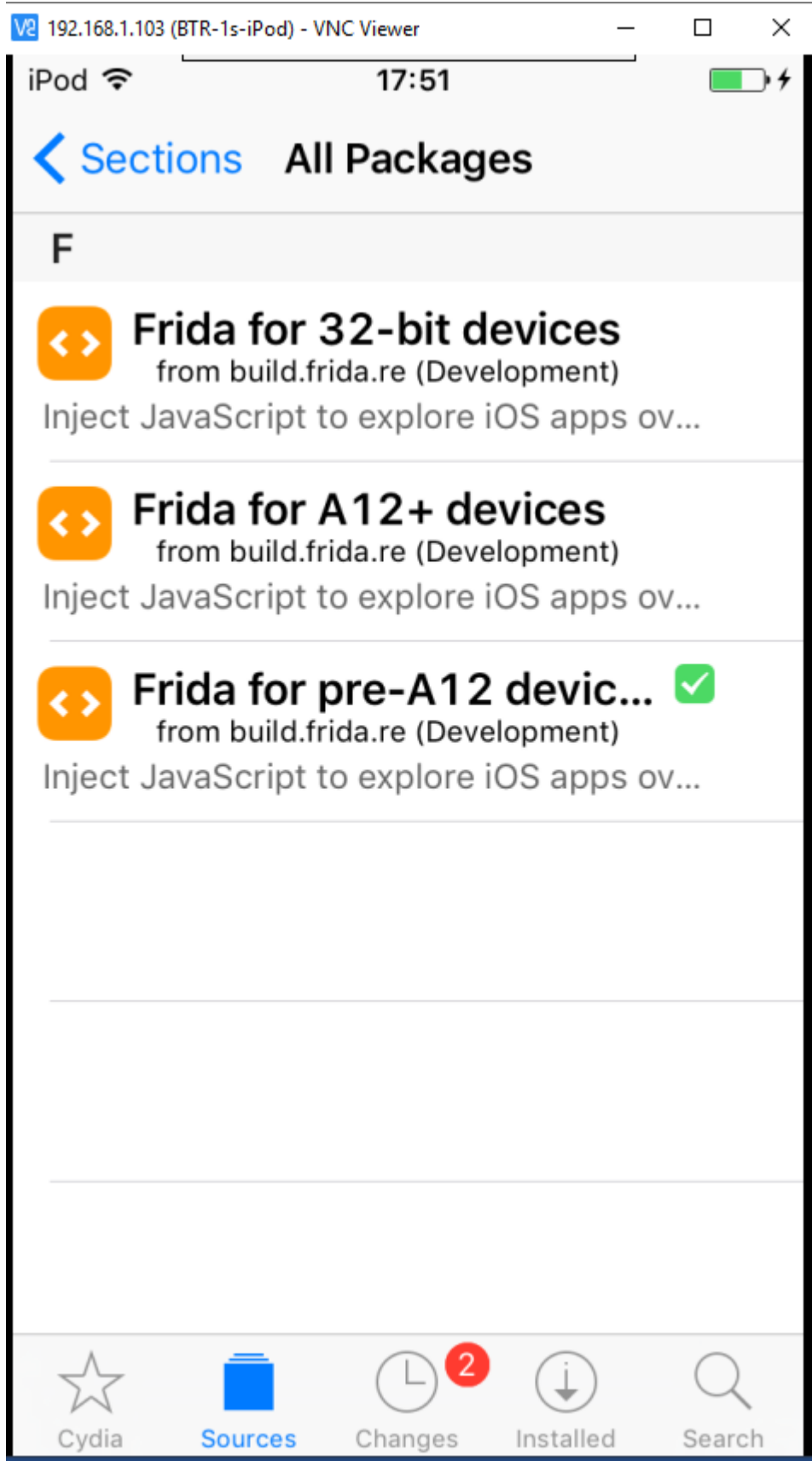
Biz burada SSL pinning atlatmak için kullanacağız. Kurulum için python3 kurulu olmalıdır.

```
pip install objection
```

Cydia başlatın Frida repository'ni ekleyelim Manage -> Sources -> Edit -> Add



Arayarak ya da package içinden seçelim ve kuralım.



Çalıştıđından emin olalım ve hook edeceğimiz uygulamayı bulalım.

```
frida-ps -Uai
```

```
Select C:\Windows\System32\cmd.exe - objection -g 952 explore
(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>frida-ps -Uai
PID Name Identifier
-----
988 Mail com.apple.mobilemail
952 Messenger com.facebook.Messenger
- App Store com.apple.AppStore
- BIP com.turkcell.bip
- Calculator com.apple.calculator
- Calendar com.apple.mobilecal
- Camera com.apple.camera
- Clock com.apple.mobletimer
- Contacts com.apple.MobileAddressBook
- Cydia com.seurik.Cydia
- Edas com.edastest.xform
- Elevate com.netmera.elevateDemoApp
- Epsas tr.com.aksaelektrik.com.mobile.ios
- FaceTime com.apple.facetime
- File Browser net.ddns.mobileapps.filebrowser
- Find Friends com.apple.mobileme.fmfi
- Find iPhone com.apple.mobileme.fmfi
- Game Center com.apple.gamecenter
- Health com.apple.Health
- Maps com.apple.Maps
- Messages com.apple.MobileSMS
- Mobile Assistant net.portswigger.MobileAssistant
- Music com.apple.Music
- News com.apple.news
- Notes com.apple.mobilenotes
- Pangu io.pangu.nvvastone
- Photos com.apple.mobileslideshow
- Podcasts com.apple.podcasts
- Reminders com.apple.reminders
- Safari com.apple.mobilesafari
- Settings com.apple.Preferences
- Stocks com.apple.stocks
- Telegram ph.telegra.Telegraph
- TestFlight com.apple.TestFlight
- TheosTutorials com.reverseeffect.theostuts
- Tips com.apple.tips
- TurkBankasI10S com.turkbankasi.retail
```

```
objection --gadget "com.facebook.Messenger" explore
```

uygulamaya hook olabilir. Fakat isimle gidince hata aldım. Hatanın nedenini bulamadım.

```
Select C:\Windows\System32\cmd.exe - objection -g 952 explore
(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>objection --gadget "com.facebook.Messenger" explore
Using USB device "Apple iPod"
Agent injected and responds ok!
Traceback (most recent call last):
  File "C:\Users\BTR-32\AppData\Local\Programs\Python\Python36\lib\runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "C:\Users\BTR-32\AppData\Local\Programs\Python\Python36\lib\runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\Scripts\objection.exe\__main__.py", line 9, in <module>
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\click\core.py", line 722, in __call__
    return self.main(*args, **kwargs)
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\click\core.py", line 697, in main
    rv = self.invoke(ctx)
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\click\core.py", line 1066, in invoke
    return process_result(sub_ctx.command.invoke(sub_ctx))
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\click\core.py", line 895, in invoke
    return ctx.invoke(self.callback, **ctx.params)
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\click\core.py", line 535, in invoke
    return callback(*args, **kwargs)
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\objection\console\cli.py", line 156, in explore
    device_info = get_device_info()
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\objection\commands\device.py", line 27, in get_device_info
    package_info = api.env_ios()
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\frida\core.py", line 401, in method
    return script_rpc_request('call', js_name, args, **kwargs)
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\frida\core.py", line 26, in wrapper
    return f(*args, **kwargs)
  File "C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF\venv\lib\site-packages\frida\core.py", line 333, in _rpc_request
    raise result[2]
frida.InvalidOperationError: script is destroyed
Waiting JPOS to stop...
Unloading objection agent...
Unable to run cleanup: script is destroyed

(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>objection --gadget "com.facebook.Messenger" explore
Using USB device "Apple iPod"
Agent injected and responds ok!
Traceback (most recent call last):
  File "C:\Users\BTR-32\AppData\Local\Programs\Python\Python36\lib\runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
```

Bunun yerine PID değeri ile bağlanmayı desteklemektedir. Bununla deneyelim

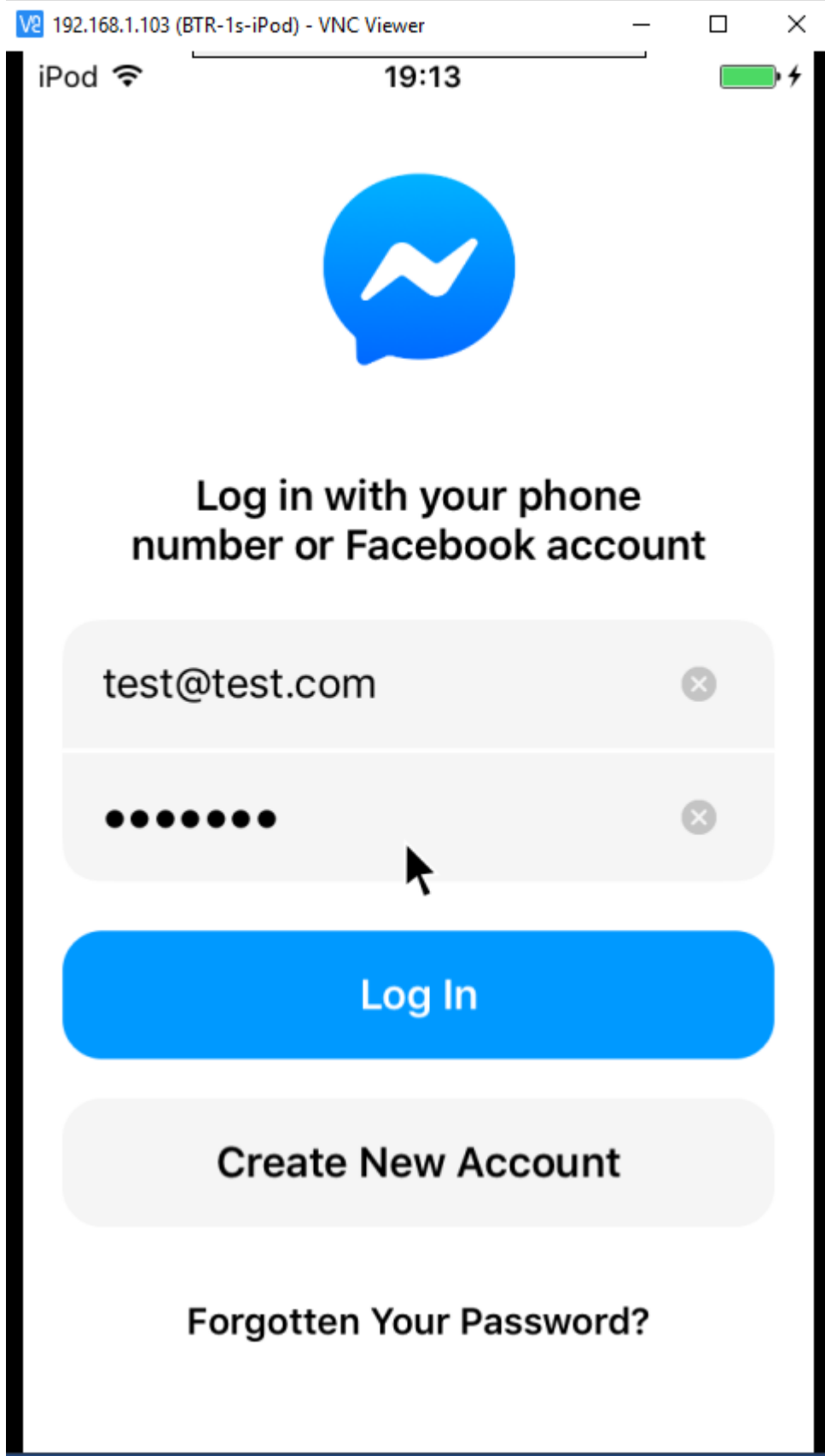
```
Select C:\Windows\System32\cmd.exe - objection -g 952 explore
(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>frida-ps -Ual
PID Name Identifier
-----
388 Mail com.apple.mobilemail
952 Messenger com.facebook.Messenger
- App Store com.apple.AppStore
- BIP com.turkcell.bip
- Calculator com.apple.calculator
- Calendar com.apple.mobilecal
- Camera com.apple.camera
- Clock com.apple.mobiletimer
- Contacts com.apple.MobileAddressBook
- Cydia com.saunik.Cydia
- Edas com.edastest.xform
- Elevate com.netmera.elevateDemoApp
- Epsas tr.com.aksaelektrik.com.mobile.ios
- FaceTime com.apple.facetime
- File Browser net.ddns.mobileapps.filebrowser
- Find Friends com.apple.mobileme.fmf1
- Find iPhone com.apple.mobileme.fmip1
- Game Center com.apple.gamecenter
- Health com.apple.Health
- Maps com.apple.Maps
- Messages com.apple.MobileSMS
- Mobile Assistant net.portswigger.MobileAssistant
- Music com.apple.Music
- News com.apple.news
- Notes com.apple.mobilenotes
- Pangu io.pangu.nvwastone
- Photos com.apple.mobileslideshow
- Podcasts com.apple.podcasts
- Reminders com.apple.reminders
- Safari com.apple.mobilesafari
- Settings com.apple.Preferences
- Stocks com.apple.stocks
- Telegram ph.telegram.Telegraph
- TestFlight com.apple.TestFlight
- TheosTutorials com.reverseeffect.theostuts
- Tips com.apple.tips
- TurkBankasIOS com.turkbankasi.retail
```

ssl pinning devre dışı bırakalım.

```
ios sslpinning disable
```

```
Select C:\Windows\System32\cmd.exe - objection -g 952 explore
(venv) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>objection -g 952 explore
Using USB device "Apple iPod"
Agent injected and responds ok!
[+] (object)inject(1on) v1.8.3
Runtime Mobile Exploration
by: @leonjza from @sensepost
[tab] for command suggestions
com.Facebook.Messenger on (iPod touch: 9.3.2) [usb] # ios sslpinning disable
```

Login olmaya çalıştığımızda



Objection işlemin başarılı olduğunu gösteriyor.

```
C:\Windows\System32\cmd.exe - objection -g 952 explore
(vrem) C:\Users\BTR-32\Desktop\Mobile-Security-Framework-MobSF>objection -g 952 explore
Using USB device "Apple iPod"
Agent injected and responds ok!

Runtime Mobile Exploitation
by: @leonjza from @sensepost

[tab] For command suggestions
com.facebook.Messenger on (iPod touch: 9.3.2) [usb] # ios sslpinning disable
(agent) Hooking common framework methods
(agent) Found NSURLSession based classes. Hooking known pinning methods.
(agent) Hooking lower level SSL methods
(agent) Hooking lower level TLS methods
(agent) Hooking BoringSSL methods
(agent) SSL_set_custom_verify not found, trying SSL_CTX_set_custom_verify
(agent) Registering job jv0yzfggdtb, type: ios-sslpinning-disable
com.facebook.Messenger on (iPod touch: 9.3.2) [usb] # (agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), setting kSSLSessionOptionBreakOnServerAuth to disable cert validation.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
(agent) [jv0yzfggdtb] Called SSLSetSessionOption(), removing ability to modify kSSLSessionOptionBreakOnServerAuth.
```

Burp suite ile dinlediğimizde trafiğin üzerinden geçtiğini görebiliriz.

Burp Suite Professional v2020.4 - Temporary Project - licensed to BTRisk Bilgi Güvenliği ve BT Yönetişim Hizmetleri Tic.Ltd.Sti (2 user license)

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options JSWS Parser AES Crypto JSON Beautifier Wsdler SQLPy

Intercept HTTP history WebSockets history Options

Filter: Showing all items

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
2347	https://scontent.fst7-2.fna...	GET	/v/t1.0-1/cp0/p60x60/34458520_...	✓		200	2862	JPEG	jpg			✓	195.175.95.84
2348	https://scontent.fst7-2.fna...	GET	/v/t1.0-1/cp0/p60x60/90556225_...	✓		200	2580	JPEG	jpg			✓	195.175.95.84
2349	https://scontent.fst7-2.fna...	GET	/v/t1.0-1/cp0/p60x60/10419500_...	✓		200	2235	JPEG	jpg			✓	195.175.95.84
2350	https://graph.facebook.com	GET	/pwd_key_fetch?access_token=43...	✓		200	742	JSON				✓	185.60.218.19
2351	https://graph.facebook.com	POST	/v2.10/auth/login	✓		500	614	JSON				✓	185.60.218.19
2352	https://graph.facebook.com	GET	/pwd_key_fetch?access_token=43...	✓		200	742	JSON				✓	185.60.218.19
2353	https://graph.facebook.com	POST	/v2.10/auth/login	✓		500	614	JSON				✓	185.60.218.19
2354	https://setup.icloud.com	POST	/setup/family/getFamilyDetails	✓		401	685					✓	17.248.147.14
2355	https://graph.facebook.com	GET	/pwd_key_fetch?access_token=43...	✓		200	742	JSON				✓	185.60.218.19
2356	https://graph.facebook.com	POST	/v2.10/auth/login	✓		500	614	JSON				✓	185.60.218.19

Request Response

Raw Params Headers Hex

```
12
13 --B2E7B7580C0D4291902877050667B69D
14 Content-Disposition: form-data; name="access_token"
15
16 43762631e97370813e1a7033ae7083fb31f35375bad9c7a
17 --B2E7B7580C0D4291902877050667B69D
18 Content-Disposition: form-data; name="app_id"
19
20 43762631e973708
21 --B2E7B7580C0D4291902877050667B69D
22 Content-Disposition: form-data; name="credentials_type"
23
24 password
25 --B2E7B7580C0D4291902877050667B69D
26 Content-Disposition: form-data; name="device_id"
27
28 A52AAFB0-8A00-4065-B965-F2E0D71B700B
29 --B2E7B7580C0D4291902877050667B69D
30 Content-Disposition: form-data; name="email"
31
32 test@test.com
33 --B2E7B7580C0D4291902877050667B69D
34 Content-Disposition: form-data; name="error_detail_type"
35
36 button with disabled
37 --B2E7B7580C0D4291902877050667B69D
38 Content-Disposition: form-data; name="family_device_id"
39
40 A52AAFB0-8A00-4065-B965-F2E0D71B700B
```

0 matches Pretty

Referans;

<https://ninadmathpati.com/all-about-ssl-pinning-bypass/>

<https://omespino.com/tutorial-universal-android-ssl-pinning-in-10-minutes-with-frida/>

<https://blog.netspi.com/four-ways-bypass-android-ssl-verification-certificate-pinning/>

VII. BTRISK Hakkında

2009 yılında kurulmuş ve sadece bilgi güvenliđi hizmetlerine odaklanmış olan BTRisk Bilgi Güvenliđi ve BT Yönetişim Hizmetleri bilgi güvenliđi probleminde yönetim kurulu seviyesinden sistem odası uygulamasına kadar uzanan alanda çözüm üretmektedir.

BTRisk bilgi güvenliđi problemini görünür hale getirerek algılanmasını, anlaşılmasını ve dolayısıyla ele alınmasını mümkün hale getirmektedir.

BTRisk bilgi güvenliđi probleminde karşı geliştirdiđi yaklaşımları gerçek hayat koşullarında test etmiş ve uygulanabilir hale getirmiştir.

Bilgi güvenliđi ve BT yönetim hizmet alanlarımız aşağıdaki gibidir:

- Pentest Hizmetleri
- Bilgi Güvenliđi ve BT Yönetişim Hizmetleri
- Bilgi Güvenliđi Operasyon Hizmetleri
- Bilgi Güvenliđi Eğitimleri

Özgün ürünlerimiz aşağıdaki gibidir:

- BTRWATCH Bilgi Güvenliđi Risk Analizi ve Denetim Uygulaması
- BTRMON 5651 Uyumlu Wi-Fi ve Kablolu Ağ Hotspot Çözümü
- BTROTP Tek Kullanımlık Parola Çözümü

Pentest & BT
Denetimi

ISO27001
Danışmanlık
Hizmetleri

BG Operasyon
Hizmetleri

btrwatch

btr^{ot}p

btr^{mon}

btrisk
OKULU