Frida: The One tool to pwn them all (Android version)

@warlockk87

Quien soy?

Security Researcher para Infobyte Security

Pentester de WebApps y aplicaciones Mobile

Desarrollador Java

Aficionado a los CTF

Aficionado a los tabletop RPGs

Jugador Mobile Legends (y otros MOBAS)

Father in progress

@warlockk87

Resumen de contenidos

Contenidos

- Introduccion a Frida
- Ejemplos básicos de instrumentación dinámico con Frida
- Bypass de network security configuration
- Bypass de certificate pinning
- Bypass de controles de rooteo
- Bruteforcing de PIN
- Mocking con Frida
- Tools destacadas que utilizan el framework de Frida

FARADAY

Introduccion a Frida

- Creada por **@oleavr**
- Toolkit de instrumentación dinámica
- Inyecta el motor V8 (chrome) en el proceso objetivo y permite ejecutar Javascript en el mismo.
- Multiplataforma (Windows, mac, Linux, **Android**, iOS)
- Open-source
- Multiples tools creadas en base a Frida
- Casos de uso principales:
  - Reversing
  - Profiling
  - Agregar funcionalidades sin deployar nueva aplicación.
  - Pentesting (deshabilitar protecciones)
  - Generar mocks de servicios / clases / drivers
  - Automatización de pruebas???

Por que Frida?

- adb (android device bridge)
- Android Studio
- Emulador (genymotion / avd / ISO VirtualBox - VMware) o Celular
- Jadx-gui o dex2jar + jd-gui
- apktool
- jarsigner
- jdb
- BurpSuite / ZAP
- Wireshark
- Analizadores estaticos (MobSF / Androbugs / QARK / JAADAS)
- Frida
- Drozer
- XPosed (rooteo requerido)
- Objection
- Apkstudio

Modos de operación

- Inyectado
  - En el dispositivo hay un componente frida-server
  - A través del frida-server se inyecta el agente frida
  - Requiere el celular rooteado
  - Si el server crashea, hay que lanzarlo de nuevo

- Embebido
  - Usa una librería frida-gadget que se tiene que agregar al apk.
  - Se tiene que volver a firmar el aplicativo.
  - No es necesario usar el celular rooteado.
  - Usar **objection** para automatizar el proceso.
  - Se tiene que efectuar el proceso por cada aplicativo a probar.

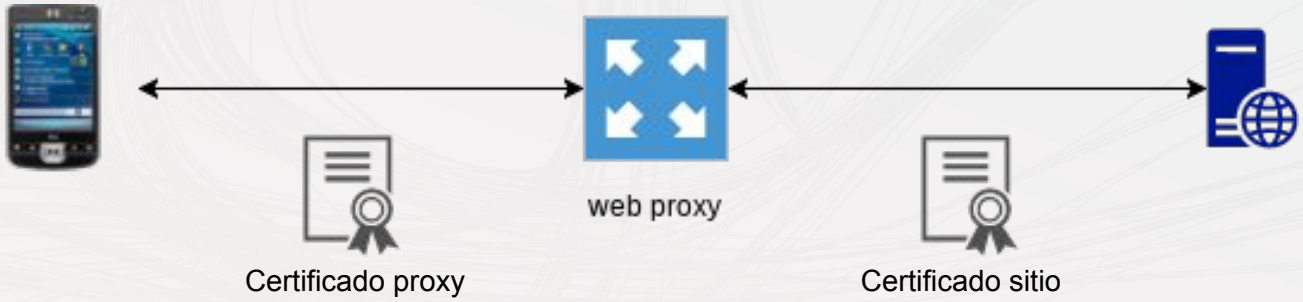
Ejemplos básicos de Frida

```
//Java.perform se asegura que la funcion que se pasa, corre en el thread de la VM
Java.perform( function () {
        //Java.use devuelve un objeto de javascript wrappeada en una clase cargada en la VM
        var activity_class = Java.use("com.example.ritesh.mybasiccalculator_riteshbhat.MainActivity");
        //forma de sobreescribir un metodo de una clase (en este caso el metodo add de la clase
        // MainActivity)
        activity_class.add.implementation = function (int1, int2) {
                //console.log muestra por pantalla lo que se pasa
                console.log("[+] se llama add");
                //this hace referencia al objeto del tipo MainActivity.
                return this.add(int1,int2);
        }
});
```

```
Java.perform(function(){
      Java.choose("com.example.ritesh.mybasiccalculator_riteshbhat.MainActivity",{
         onMatch: function(activity){
            //se hace cuando se encuentra un objeto del tipo MainActivity
            ...
         },
         onComplete:function() { /* cunao se terminaron de procesar las referencias */ }
      });
   });
```


Demo 1

Interceptando el tráfico



Request

Raw | Params | Headers | Hex

```
POST
/safebrowsing/downloads?client=navclient-auto-ffox&appver=52.8&pver=2.2&key=no-google-api-key
HTTP/1.1
Host: safebrowsing.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Length: 503
Content-Type: text/plain
Connection: close
Cookie:
NID=129=Eb2HW8No97VJ3NZsPjyOnOFLUtLxJToutltGZHci6LkxrYGC3rwERKh_DQT-ucamLz1URdWwtf75pdovpafewU
K-8r1up4su9cGLhFv4OsAPnzC4YqeXk11sa5i4sge6
Pragma: no-cache
Cache-Control: no-cache

goog-badbinurl-shavar;a:149095-151157:s:154185-154664,154666-154669,154671-154678,154680-15468
2,154684-154687,154689-154695,154697-154720,154722-154776,154778-155706,155708-156049,156051-1
56080,156082-156106,156108-156242
goog-phish-shavar;a:498708-509651:s:710094-760528
goog-malware-shavar;a:276258-288842:s:272375-277907,277909-281580,281582-281800,281802-283603,
283605-283927,283929-284195,284197-286469
goog-unwanted-shavar;a:105118-116810:s:100486-106780,106782-108915,108917-114813,114815-115041
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Content-Type: application/vnd.google.safebrowsing-update
X-Content-Type-Options: nosniff
Date: Thu, 31 May 2018 11:36:32 GMT
Server: HTTP server (unknown)
Content-Length: 2173
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Alt-Svc: quic=":443"; ma=2592000; v="43,42,41,39,35"
Connection: close

n:1757
i:goog-badbinurl-shavar
ad:149095-149330
sd:154185-154410
u:safebrowsing-cache.google.com/safebrowsing/rd/ChVnb29nLWJhZGJpbnVybC1zaGF2YXI4AEACSgwIARCQyA
kY6sgJIAFKDAgBENTHCRiOyAkgAUoMCAEQxscJGNLHCSABSgwIARDExwkYxMcJIAFKDAgBELvHCRjCxwkgAUoMCAEQ-cY
JGLnHCSABSgwIARC7xgkY98YJIAFKDAgBE0LFCRi5xgkgAUoMCAEQ08QJG0DFCSABSgwIARDLwwkYy8MJIAFKRggAEPac
CRiQoQkgASo4EW2OAZABkwGeAaMBpgHSAdgB2gHeAd8B5wGIAooCnAKxArIC2QLhAuQC9QL3AqUDpgOtA64DrwM
i:goog-malware-shavar
ad:276258-277238
sd:272375-273440
u:safebrowsing-cache.google.com/safebrowsing/rd/ChNnb29nLW1hbHdhcmUtc2hhdmFyOABAAkoMCAEQhr4RGI
nAESAB
```

Network Security Config

Se referencia un archivo en AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest >
    <application android:networkSecurityConfig="@xml/network_security_config" >
        ...
    </application>
</manifest>
```

Se agrega archivo res/xml/network_security_config.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">example.com</domain>
        <pin-set expiration="2018-01-01">
            <pin digest="SHA-256">7HIpactkIAq2Y49orFOOQKurWxmmSFZhBCoQYcRhJ3Y=</pin>
            <!-- backup pin -->
            <pin digest="SHA-256">fwza0LRMXouZHRC8Ei+4PyuldPDcf3UKgO/04cDM1oE=</pin>
        </pin-set>
    </domain-config>
</network-security-config>
```

Network Security Config Bypass

Modificar la configuración del apk.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" and
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="25"/>
    <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="
        <activity android:label="@string/app_name" android:name="sg.vantagepoint.uncrackable1.Ma
            <intent-filter>
                <action android:name="andro
                <category android:name="and
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config>
        <trust-anchors>
            <certificates src="system"/>
            <certificates src="user"/>
        </trust-anchors>
    </base-config>
</network-security-config>
```

Requiere decompilar y volver a compilar.

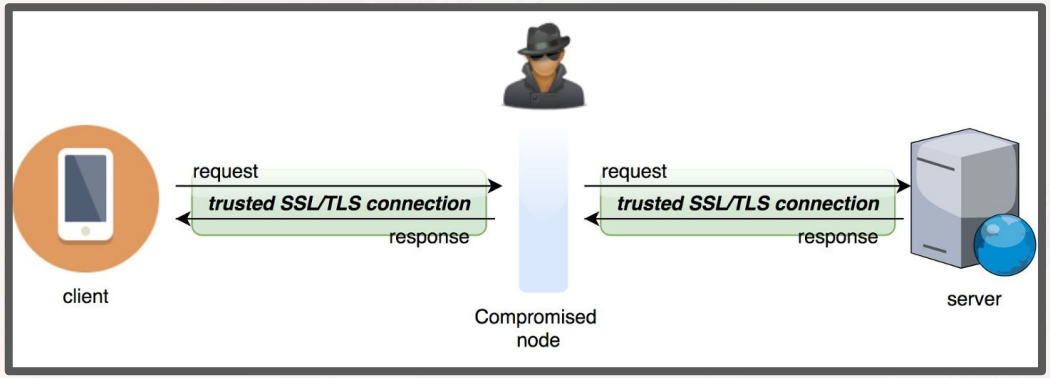
Modo Max Power

FARADAY

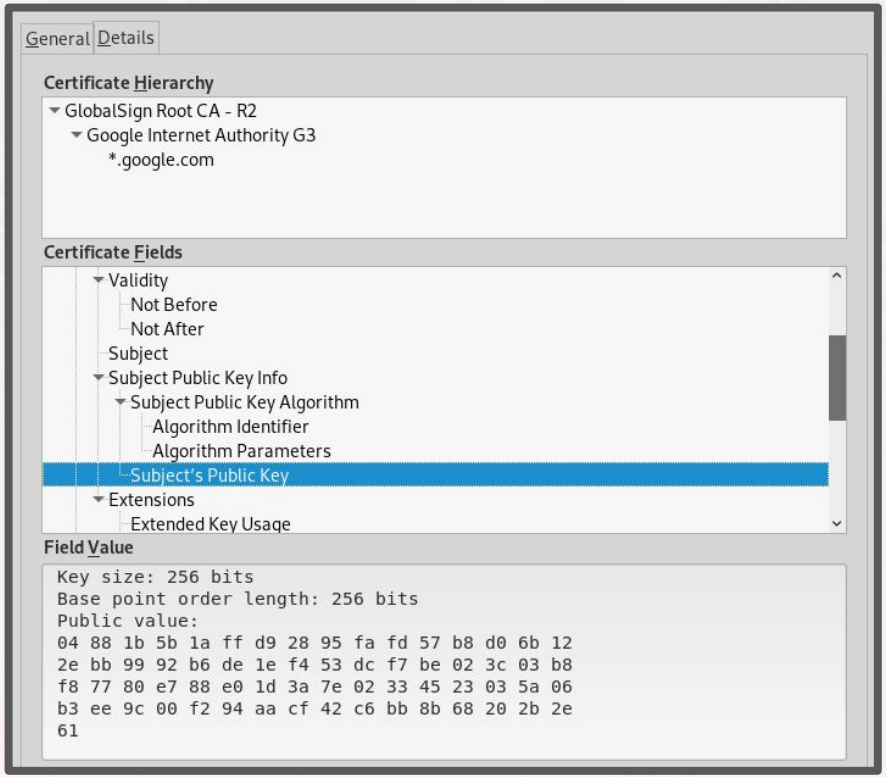
```
Java.perform(function(){
    NetworkSecurityConfig_Builder =Java.use("android.security.net.config.NetworkSecurityConfig$Builder");
        console.log("NetworkSecurityConfig_Builder: " + NetworkSecurityConfig_Builder);
    CertificatesEntryRef = Java.use("android.security.net.config.CertificatesEntryRef");
        console.log("CertificatesEntryRef: " + CertificatesEntryRef);
    CertificateSource = Java.use("android.security.net.config.CertificateSource");
        console.log("CertificateSource: " + CertificateSource);
    UserCertificateSource = Java.use("android.security.net.config.UserCertificateSource");
        console.log("UserCertificateSource: " + UserCertificateSource);

    NetworkSecurityConfig_Builder.getEffectiveCertificatesEntryRefs.implementation = function(){
        console.log("entra");
        origin = this.getEffectiveCertificatesEntryRefs()

        source = UserCertificateSource.getInstance()
        userCert = CertificatesEntryRef.$new(source,true)
        origin.add(userCert)

        return origin
    }
})
```

Certificate Pinning

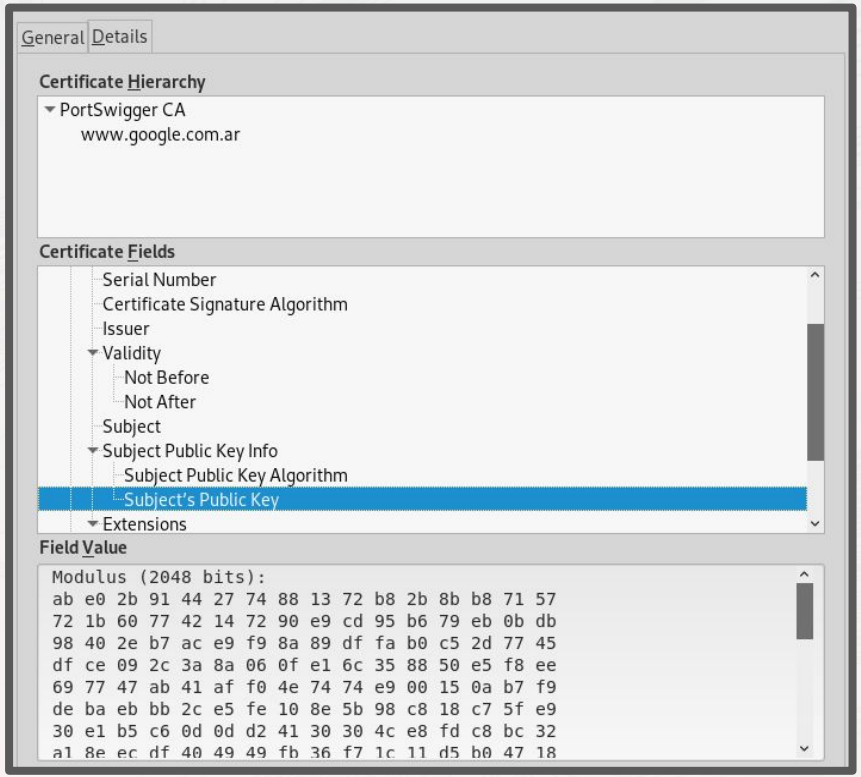
- Modo de validar el certificado entregado por el servidor.
- Control de seguridad para evitar ataques del tipo MitM
- Se puede pinnear:
  - un conjunto de certificados (archivos)
  - PKI - Subject Public Key
- El método más recomendado es el de Subject Public Key (administración, instalación, control)



Certificate Pinning (PKI)



Certificado original



Certificado proxy

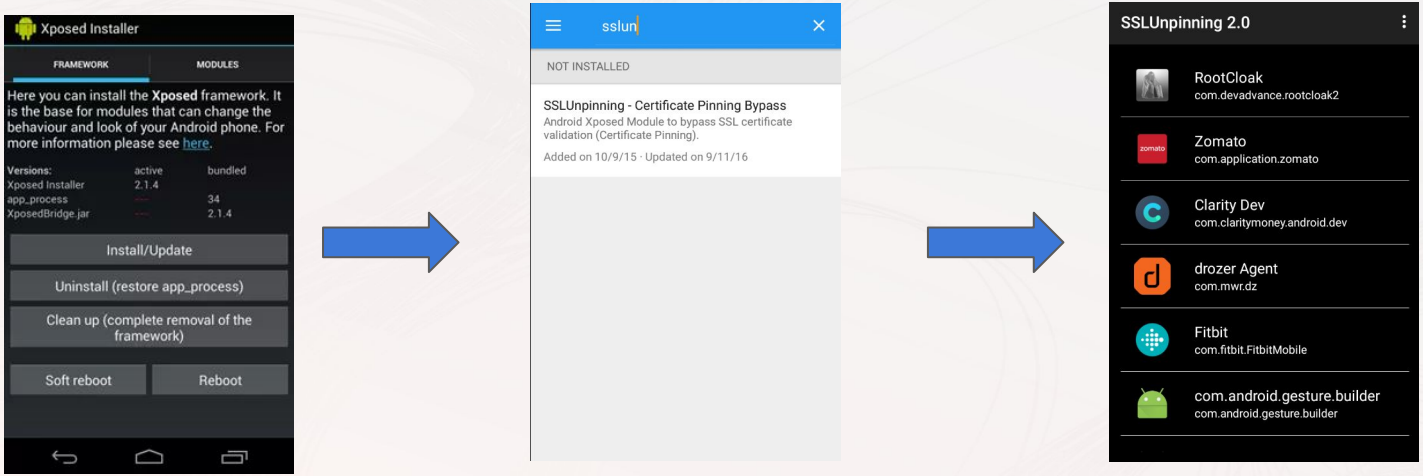
Múltiples formas de hacerlo

- CertificatePinner (libreria OkHttp)
- Retrofit (OkHttp v3)
- TrustManagerImpl(en sdk de android)
  - Mediante certificados en dispositivo.
  - Mediante PKI en codigo o parametros.
- NetworkSecurityConfig (Usa TrustManagerImpl de fondo)
- Custom-made

```
String hostname = "yourdomain.com";
 CertificatePinner certificatePinner = new CertificatePinner.Builder()
     .add(hostname, "sha256/47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=")
     .build();
 OkHttpClient client = OkHttpClient.Builder()
     .certificatePinner(certificatePinner)
     .build();

 Request request = new Request.Builder()
     .url("https://" + hostname)
     .build();
 client.newCall(request).execute();
```

Bypass de Certificate Pinning

- Usar XPosed Framework
- Instalar SSLUnpinning/JustTrustMe
- Habilitar la app en el nuevo módulo
- Volver a abrir la app



Bypass de Certificate Pinning

```
public void check(String hostname, List<Certificate> peerCertificates)
throws SSLPeerUnverifiedException {
    List<Pin> pins = findMatchingPins(hostname);
    if (pins.isEmpty()) return;
 …
      for (int p = 0, pinsSize = pins.size(); p < pinsSize; p++) {
        Pin pin = pins.get(p);
        if (pin.hashAlgorithm.equals("sha256/")) {
          if (sha256 == null) sha256 = sha256(x509Certificate);
          if (pin.hash.equals(sha256)) return; // Success!
        } else if (pin.hashAlgorithm.equals("sha1/")) {
          if (sha1 == null) sha1 = sha1(x509Certificate);
          if (pin.hash.equals(sha1)) return; // Success!
        } else {
          throw new AssertionError();
        }
      }
    }
…
}
```

```
const-string v11, "sha256/"
invoke-virtual {v10, v11}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z
move-result v10
if-eqz v10, :cond_5
.line 162
if-nez v8, :cond_3
invoke-static {v9}, Lokhttp3/CertificatePinner;
        ->sha256(Ljava/security/cert/X509Certificate;)Lokio/ByteString;
move-result-object v8
.line 163
:cond_3
iget-object v10, v4, Lokhttp3/CertificatePinner$Pin;->hash:Lokio/ByteString;
invoke-virtual {v10, v8}, Lokio/ByteString;->equals(Ljava/lang/Object;)Z
move-result v10
if-nez v10, :cond_0    --> Reemplazar if-nez pof if-eqz
.line 159
:cond_4
add-int/lit8 v3, v3, 0x1
goto :goto_2
.line 164
:cond_5
iget-object v10, v4, Lokhttp3/CertificatePinner$Pin;
        ->hashAlgorithm:Ljava/lang/String;
 const-string v11, "sha1/"
```

Modo Max Power

```
try {
    var CertificatePinner = Java.use('okhttp3.CertificatePinner');
    console.log("[+] OkHTTP 3.x Found");
    CertificatePinner.check.overload('java.lang.String', 'java.util.List').implementation = function() {
        console.log("[+] OkHTTP 3.x check() called. Not throwing an exception.");
    };
} catch (err) {
    console.log("[-] OkHTTP 3.x Not Found")
}
```

```
Java.perform(function() {

    var array_list = Java.use("java.util.ArrayList");
    var ApiClient = Java.use('com.android.org.conscrypt.TrustManagerImpl');

    ApiClient.checkTrustedRecursive.implementation = function(a1,a2,a3,a4,a5,a6) {
            // console.log('Bypassing SSL Pinning');
            var k = array_list.$new();
            return k;
    }
});
```

Modo Max Power (2)

```
var TrustManager = Java.registerClass({
    name: 'com.sensepost.test.TrustManager',
    implements: [X509TrustManager],
    methods: {
        checkClientTrusted: function(chain, authType) {},
        checkServerTrusted: function(chain, authType) {},
        getAcceptedIssuers: function() {
            return [];
        }
    }
});

// Prepare the TrustManagers array to pass to SSLContext.init()
var TrustManagers = [TrustManager.$new()];

// Get a handle on the init() on the SSLContext class
var SSLContext_init = SSLContext.init.overload(
    '[Ljavax.net.ssl.KeyManager;', '[Ljavax.net.ssl.TrustManager;', 'java.security.SecureRandom');

try {
    // Override the init method, specifying our new TrustManager
    SSLContext_init.implementation = function(keyManager, trustManager, secureRandom) {
        console.log("[+] Overriding SSLContext.init() with the custom TrustManager android < 7");
        SSLContext_init.call(this, keyManager, TrustManagers, secureRandom);
    };
} catch (err) {
    console.log("[-] TrustManager Not Found");
}
```

Detección de root

1. Existencia de paquetes o archivos particulares como
    - /system/app/Superuser.apk
    - Eu.chainfire.supersu

2. Existencia de “su”Buscar en directorios (/sbin/su, /system/su, etc)

3. Ejecutar mediante Runtime.getRuntime().exec()

4. Revisar los procesos que corren en /proc

5. Ver permisos de diferentes directorios

Hay muchas soluciones custom-made porque la complejidad de desarrollo es baja

Bypass de control de rooteo

```
.method public static a()Z
    .locals 7

    const/4 v0, 0x0

    const-string v1, "PATH"

    invoke-static {v1}, Ljava/lang/System;->getenv(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v1

    const-string v2, ":"

    invoke-virtual {v1, v2}, Ljava/lang/String;->split(

    move-result-object v2

    array-length v3, v2

    move v1, v0

    :goto_0
    if-ge v1, v3, :cond_0

    aget-object v4, v2, v1

    new-instance v5, Ljava/io/File;

    const-string v6, "su"

    invoke-direct {v5, v4, v6}, Ljava/io/File;-><init>(

    invoke-virtual {v5}, Ljava/io/File;->exists()Z

    move-result v4

    if-eqz v4, :cond_1

    const/4 v0, 0x1

    :cond_0
    return v0

    :cond_1
    add-int/lit8 v1, v1, 0x1

    goto :goto_0
.end method
```

```
.method public static a()Z

    .registers 1
    const v0, 0
    return v0
.end method
```

```
public static boolean a() {
    for (String file : System.getenv("PATH").split(":")) {
        if (new File(file, "su").exists()) {
            return true;
        }
                                                .contains("test-keys");
    for (String file : new String[]{"/system/app/Superuse
        if (new File(file).exists()) {
            return true;
        }
    }
    return false;
}
```

Modo Max Power

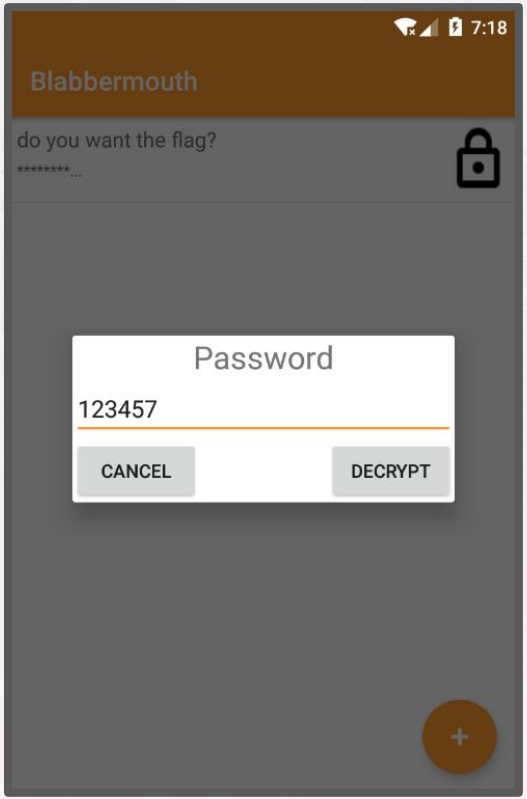
Ver https://codeshare.frida.re/@dzonerzy/fridantiroot/

Solución de uncrackable Lvl 1
(https://github.com/OWASP/owasp-mstg/tree/master/Crackmes)

```
public static boolean a() {
    for (String file : System.getenv("PATH").split(":")) {
        if (new File(file, "su").exists()) {
            return true;
        }
    }
    return false;
}

public static boolean b() {
    String str = Build.TAGS;
    return str != null && str.contains("test-keys");
}

public static boolean c() {
    for (String file : new String[]{"/system/app/Superuse
        if (new File(file).exists()) {
            return true;
        }
    }
    return false;
}
```

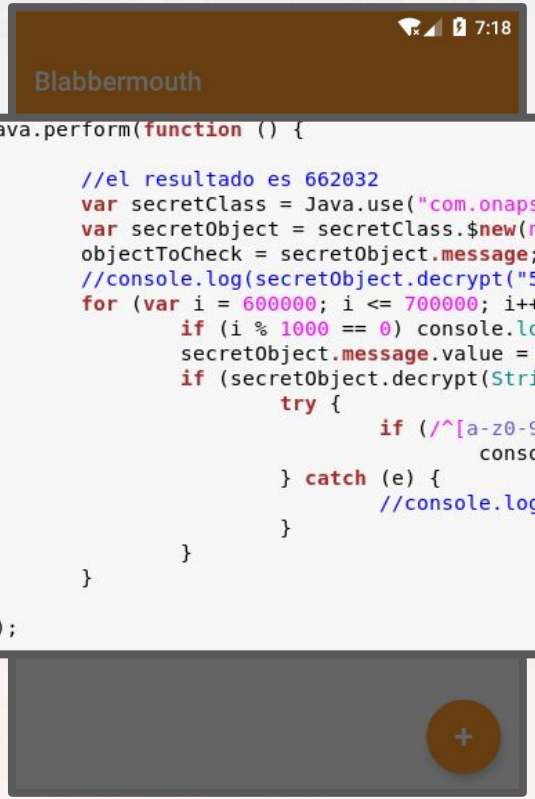
```
Java.perform(function x() {

        console.log("Se llama la funcion adecuada");
        var my_root_control = Java.use("sg.vantagepoint.a.c");
        my_root_control.a.implementation = function() {
                console.log("control root a");
                return false;
        }

        my_root_control.b.implementation = function() {
                console.log("control root b");
                return false;
        }

        my_root_control.c.implementation = function() {
                console.log("control root c");
                return false;
        }

});
```

Ataques de fuerza bruta




```java
public boolean decrypt(String key) {
    try {
        byte[] bytes = Base64.decode(this.message.getBytes(), 0);
        byte[] keyBytes = key.getBytes("UTF-8");
        Log.d("DECRYPT", "Long KEY: " + keyBytes.length);
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        md.update(keyBytes);
        keyBytes = md.digest();
        Log.d("DECRYPT", "Long KEY: " + keyBytes.length);
        byte[] ivector = Arrays.copyOfRange(bytes, 0, 16);
        Log.d("DECRYPT", "IV: " + new String(ivector));
        byte[] enc_msg = Arrays.copyOfRange(bytes, 16, bytes.length);
        AlgorithmParameterSpec ivSpec = new IvParameterSpec(ivector);
        SecretKeySpec newKey = new SecretKeySpec(keyBytes, "AES");
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        cipher.init(2, newKey, ivSpec);
        this.message = new String(cipher.doFinal(enc_msg), "UTF-8");
        this.password = key;
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        Log.e("WRONG PASSWORD: ", key);
        return false;
    }
}
```

Ataques de fuerza bruta




```
public boolean decrypt(String key) {
    try {
        byte[] bytes = Base64.decode(this.message.getBytes(), 0);
        byte[] keyBytes = key.getBytes("UTF-8");
```

```
Java.perform(function () {

        //el resultado es 662032
        var secretClass = Java.use("com.onapsis.ekochallenge2017.Secret");
        var secretObject = secretClass.$new(null);
        objectToCheck = secretObject.message;
        //console.log(secretObject.decrypt("550113"));
        for (var i = 600000; i <= 700000; i++) {
                if (i % 1000 == 0) console.log("Corriendo: " + i);
                secretObject.message.value = "zv5q1QJTJtkcx/OJUgl+i2ZxJ8OFW7/ig9ColAgi89xNMtoMMhCrqVKRjYkADzYzDAHYLkKVU3tM+/RCcYhNuw==";
                if (secretObject.decrypt(String(i))) {
                        try {
                                if (/^[a-z0-9!"#$%&'()*+,.\/:;<=>?@\[\] ^_`{|}~-]*$/i.test(secretObject.message.value))
                                        console.log(String(i) + ":" + secretObject.message.value);
                        } catch (e) {
                                //console.log("Se imprimen los errores");
                        }
                }
        }

});
```

```
        Log.e("WRONG PASSWORD: ", key);
        return false;
    }
}
```


Demo 2

Mocking con Frida

Reemplazar funcionamiento para simular casos de error particulares

Probar funciones para las que no tenemos hardware

a. BLE
b. NFC
c. Servidores web inexistentes (o caidos)
d. Dispositivos específicos con formatos de conexión particulares

Fuzzear librerías

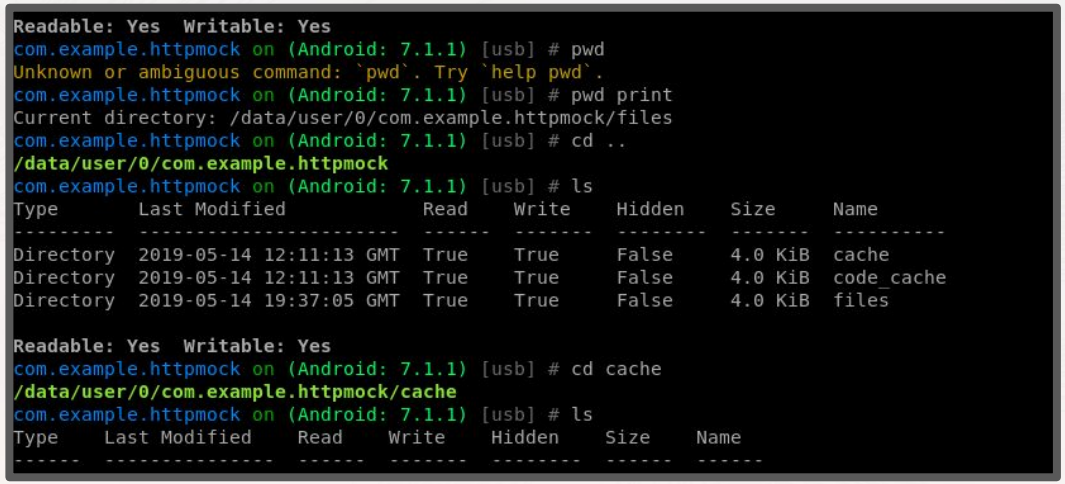
Modificar valores de instancias para probar lógica difícil de replicar.


Demo 3

Tools mas utilizadas - Objection

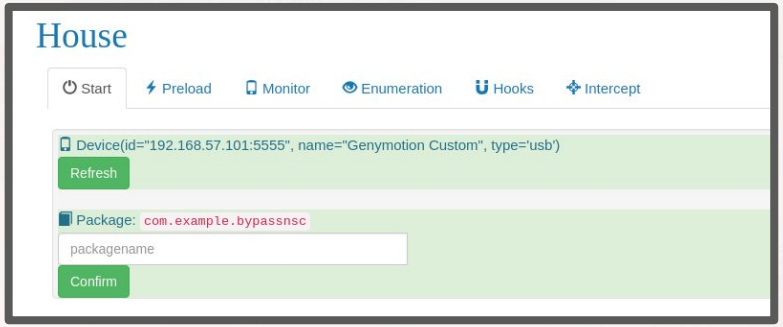
Objection

- Instalar frida-gadget de manera automatica
- Listar archivos en carpeta del proyecto
- Certificate pinning bypass
- Bypass de control de rooteo
- Subir y bajar archivos



Tools mas utilizadas - House

- Permite trackear llamadas
  - input/output archivos
  - Html
  - Sqllite3
  - IPC
- Inyectar seguimiento de funciones
- Management de scripts de Frida
- Management de hooks a funciones particulares



Demo 4

Referencias

https://www.frida.re/

https://github.com/dweinstein/awesome-frida

https://github.com/sensepost/objection

https://codeshare.frida.re/

Repo de la charla:

https://github.com/CesarMRodriguez/owasp_2019

Gracias,

Preguntas?

Cesar Rodriguez crodriguez@faradaysec.com @warlockk87