Mobile Hacking 101 – A Cross-Platform Audit Journey

César Calderón

Whoami

Cesar Calderon

Senior Security Consultant– WebSec MX

Ex-Content Developer in Tryhackme

eJPT - eMAPT

@__stux

Marco Almaguer

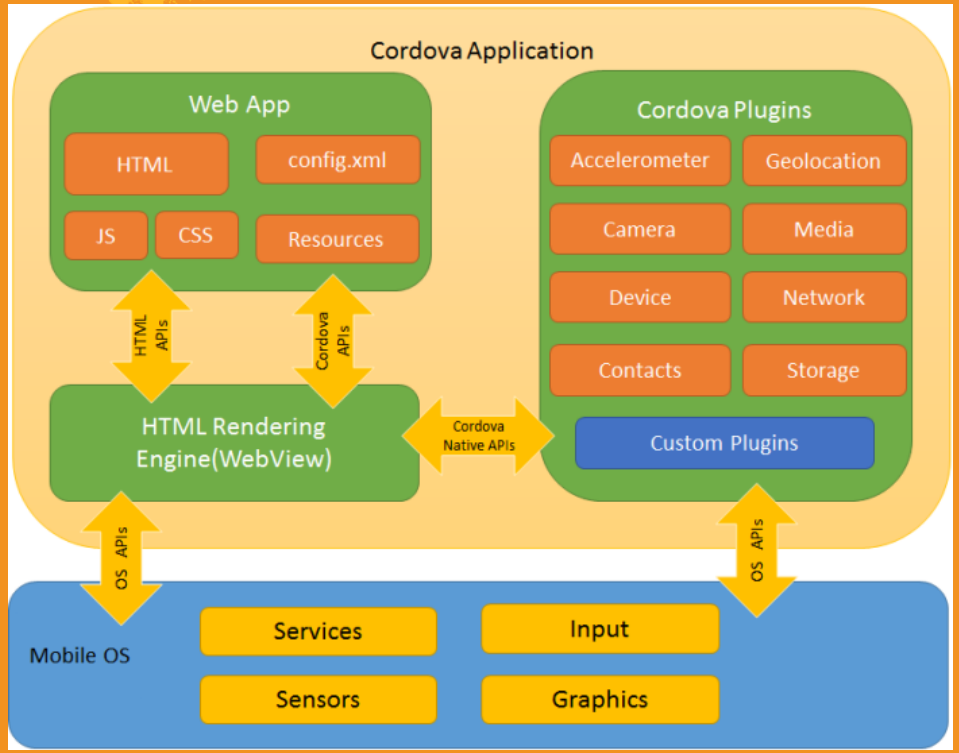
DISCLAIMER

Todos los contenidos expuestos, actividades realizadas y demostraciones de esta conferencia. taller son con fines 100% educativos, por lo cual los expositores y conferencia se deslindan de cualquier responsabilidad que los participantes a dicho evento llegaran a adquirir por el mal uso o daño que pudieran causar con la información o herramientas aquí expuestas.

Introducción

WEBSEC

Ionic/Cordova



Es un framework de desarrollo de aplicaciones móviles para crear aplicaciones híbridas utilizando las tecnologías web estándar como HTML, CSS y JavaScript. Permite el desarrollo multiplataforma, lo que evita la necesidad de desarrollar aplicaciones nativas para cada plataforma móvil y sistema operativo. Las aplicaciones desarrolladas con Apache Cordova utilizan contenedores para ejecutar las funciones nativas de la plataforma de destino.

Consideraciones de Análisis

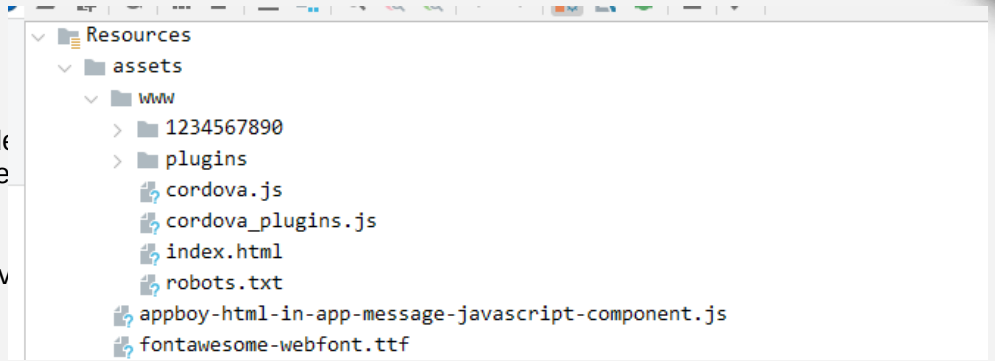
Análisis Estático Android/iOS

Mobsf:

- Para la identificación de los componentes de
- Posible identificación de posibles secretos e

Grep o Egrep:

- Búsqueda de información sensible en archiv



Análisis Dinámico Android/iOS

Android < 7. Instalar CA Burp a nivel de usuario y configurar el proxy en proxydroid y colocarlo como invisible en burpsuite.

Android >= 7. Instalar CA Burp a nivel de sistema y configurar el proxy en WiFi.

Scripts de Frida públicos para salto de root/jailbreak, sslpining, fingerprinting, funcionan adecuadamente

Scripts de Frida para extraer claves de cifrado en memoria no funcionan, esto debe identificar manualmente en los archivos .js.

En iOS solo es necesario instalar el certificado en el celular para interceptar el tráfico para esto no es necesario root, si la app tiene implementado ssl pinning se debe tener equipo con jailbreak.

Información sensible embebida en los aplicativos

¿Qué tipo de información es sensible?

Normalmente durante el análisis podemos encontrar diversos descuidos por parte de los desarrolladores o información almacenada de manera insegura dentro de las apps. Desde URL internas, direcciones que apuntan a entornos pre-productivos o de desarrollo o rutas sobre producción que no conocíamos hasta archivos relacionados a la puesta en producción (.gitignore, etc).

En ocasiones también he encontrado servicios utilizados por los desarrolladores (FTP, SFTP, SSH, etc.) dentro del código decompilado de la aplicación; archivos .XML dentro de la carpeta “asset” o dentro de otras carpetas internas que poseían archivos de configuración como “config.xml”, “routes_config.xml” o archivos de “test cases” de la aplicación con información sensible (usuarios, contraseñas, direcciones IP, servicio, etc).

Caso #1: Implementación insegura de CryptoJS

```
var CryptoJS = require("crypto-js" 4.1.1 );

var
mensaje='{"pSisCta":"TDB_CL","pCodigo":125,"pUsuario":"X","pSesionId":"X-
Google Nexus 5X - 6.0.0 - API 23 -
1080x1920","pIpUsuario":"X","pIp":"X","eUi":[{"pSmartPhone":"Google Nexus
5X - 6.0.0 - API 23 - 1080x1920","pTipoSmartPhone":"Google Nexus 5X -
6.0.0 - API 23 -
1080x1920","pSO":"Android","pUi":"X","pImei":"null","pVersionAPP":"3.0"}]}
';

var enc = CryptoJS.AES.encrypt(mensaje,
"CLAVENARREGLODEBYTES").toString();

var ciphertext = encodeURIComponent(enc.toString(CryptoJS.enc.Utf8));
```

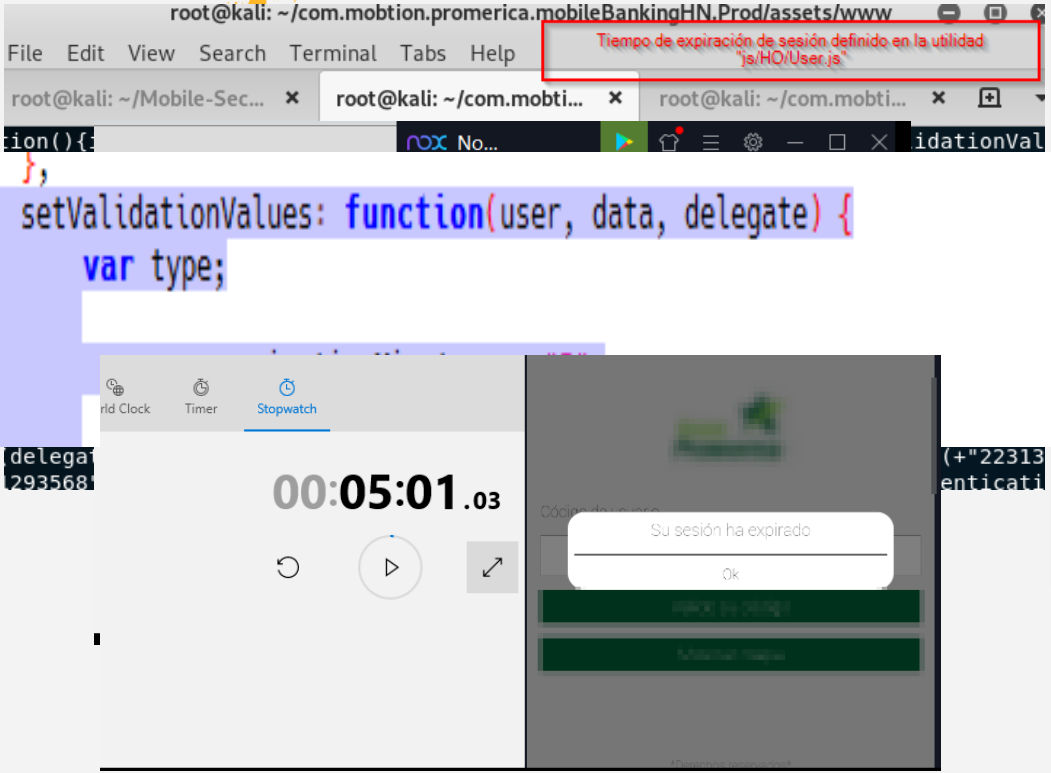
Save on RunKit | Node 18 | help | ▶ run

"U2FsdGVkX1%2BbzW7lzug7EiMC1H2BDMLhx75kpzhFB6z1u0cuig06JiTx…DolzSmB2FtZK8vmbbhsbWkmzs535HDEl65Dv7B7ojV8wzSW8rtg%3D%3D"

U2FsdGVkX1%2BbzW7lzug7EiMC1H2BDMLhx75kpzhFB6z1u0cuig06JiTxWeHMkQYB%2FSjzrmr2qVXiIrSz5fkZuzMq8PLqdEHZjUitY13u9Q4WEZhkb5eIX722AvTBF281CQSaw3I%2F5r19w3CYZa8JhyU3xqdH4jkb3Xhp3T%2B6suQhE2yxPLB8RhrJzh6mNsYFumLTFgzdVG75PZSDLvZoueqFH9A3UxPpf%2F%2FTBgV7jBPLCaNDwT6WkByM%2Bpr9CcOEj0uGjOGGzjLTzlFLuGFtZJlxcZq4hbOMgiyGQ2ABRqgJv1T%2Fv8x5JmABQ3Fykl5cdbSPxOYS%2FYeXwiHuT1jm9ygrdjQtyo%2FlVpgo%2Bo9vRZ9prOeaJ%2FUFzkBBsJq3XVMD5WSrXLAL%2FvxDu4D20YhQ4fYIrnxCNvMjQaWDolzSmB2FtZK8vmbbhsbWkmzs535HDEl65Dv7B7ojV8wzSW8rtg%3D%3D

CryptoJS es una colección de algoritmos criptográficos implementados en JavaScript. Esta utilidad en aplicaciones móviles normalmente se utiliza para realizar el cifrado de las solicitudes/respuestas entre cliente y servidor.

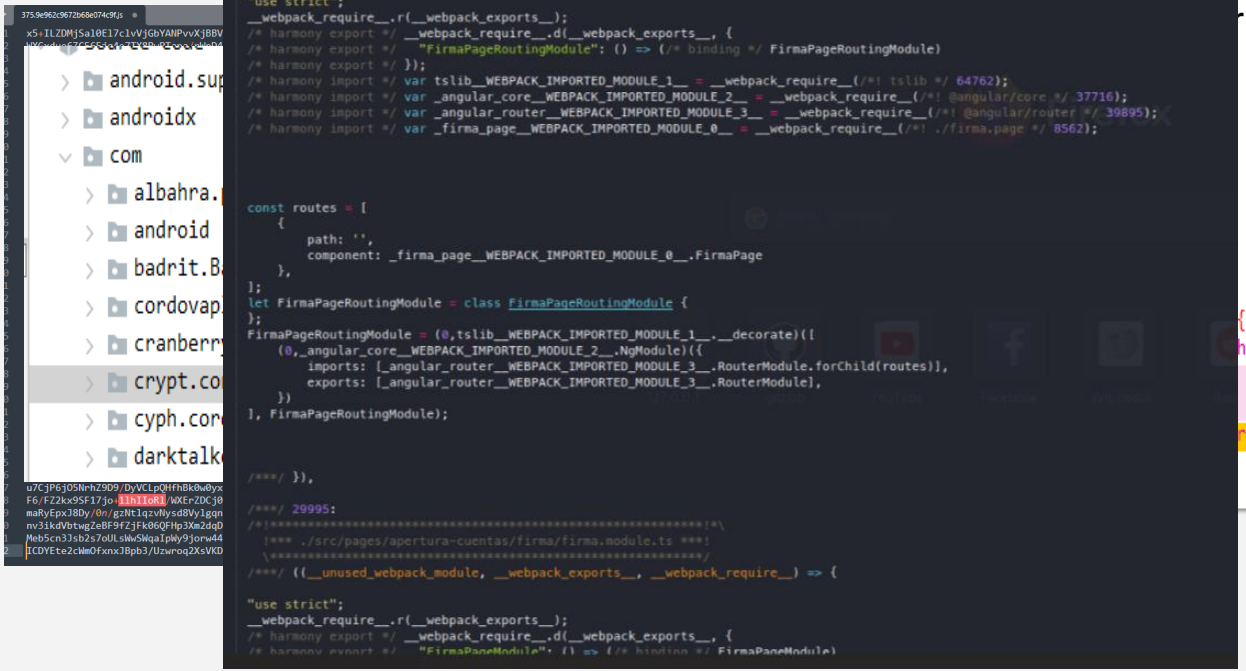
Caso #2: Validaciones a Nivel de cliente



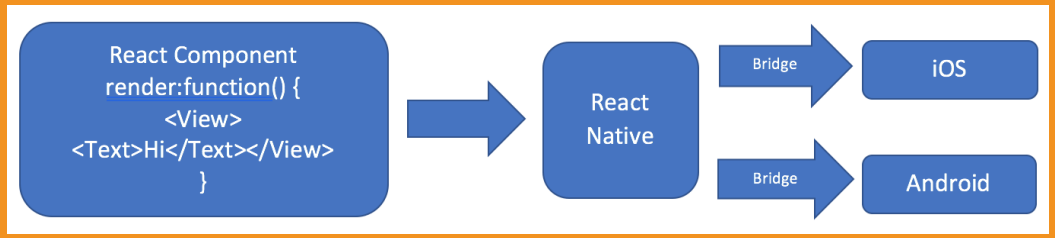
Durante el proceso de assesment se identificó que la aplicación de móvil space estaba realizando validaciones a nivel de cliente con lo cual era posible modificar la lógica de la aplicación.

Caso #3: Implementación in
re

te el proceso de
ría a la aplicación del
se
os de
aban
HTM,



React Native



Utiliza JavaScript, que es, con mucho, el lenguaje más dinámico, popular y de alto rendimiento. React Native combina los beneficios de JavaScript y React.JS y también está patrocinado por Facebook.

La mejor parte de trabajar con React Native es que entre los tres marcos, permite al desarrollador escribir las piezas de código en Swift, Obj-C o Java cuando sea necesario. Si desea manejar operaciones pesadas en su aplicación, obtiene acceso a módulos y bibliotecas nativas en aplicaciones basadas en React Native.

Consideraciones de Análisis

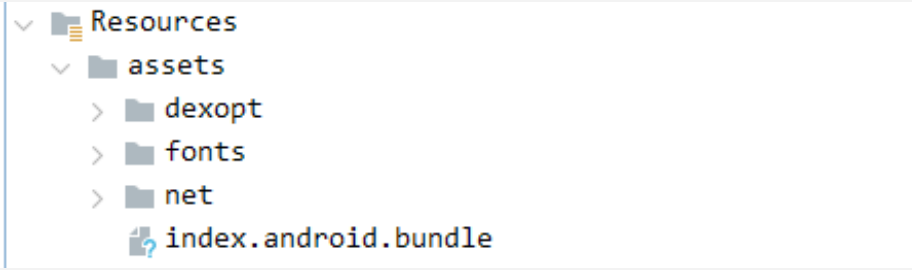
Análisis Estático Android/iOS

Mobsf:

- Para la identificación de los componentes de aplicación en Ar
- Posible identificación de posibles secretos en otras partes de

Grep o Egrep:

- Búsqueda de información sensible en archivos asociada al ap



Análisis Dinámico Android/iOS

Android < 7. Instalar CA Burp a nivel de usuario y configurar el proxy en proxydroid y colocarlo como invisible en burpsuite.

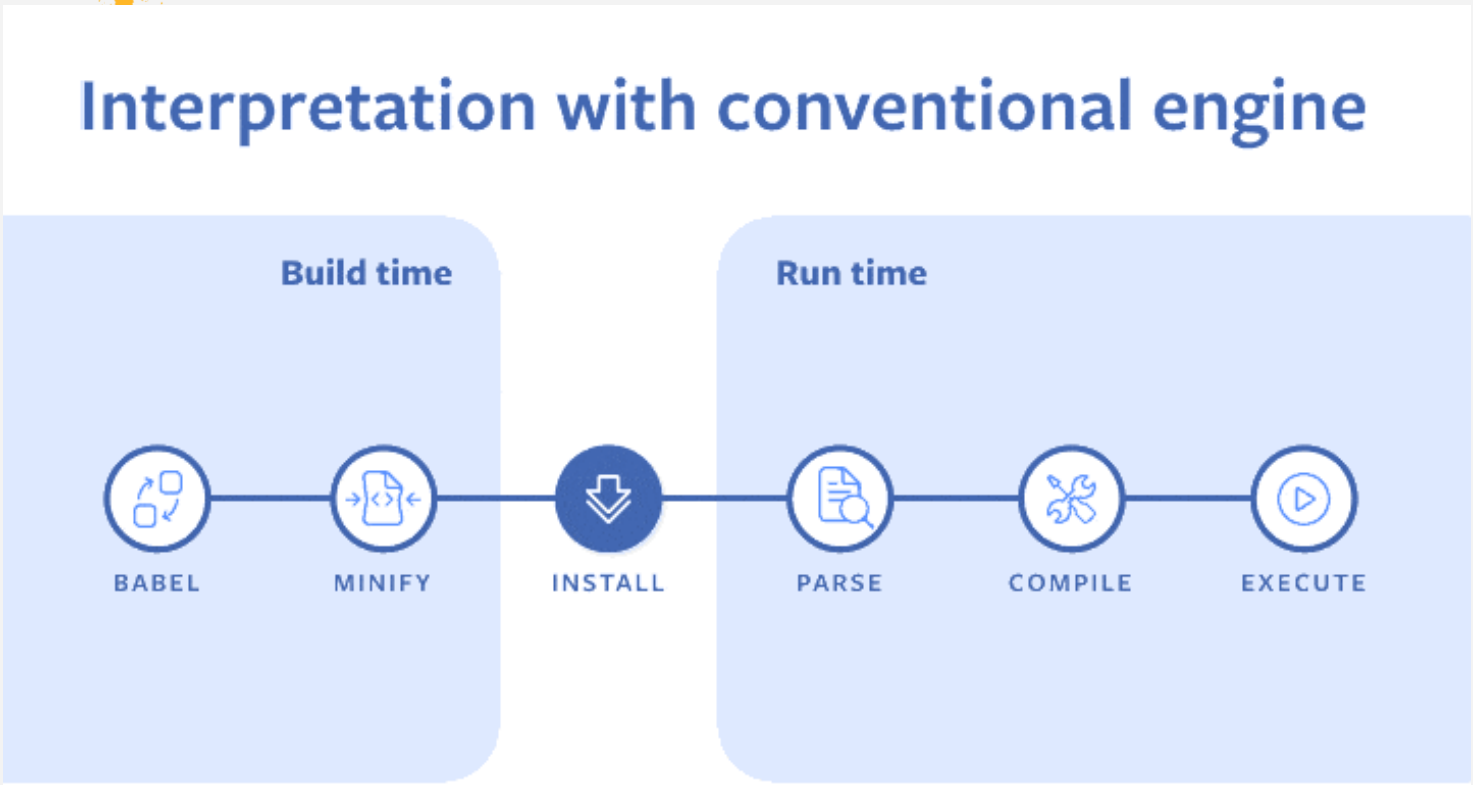
Android >= 7. Instalar CA Burp a nivel de sistema y configurar el proxy en WiFi.

Scripts de Frida públicos para salto de root/jailbreak, sslpining, fingerprinting, funcionan adecuadamente

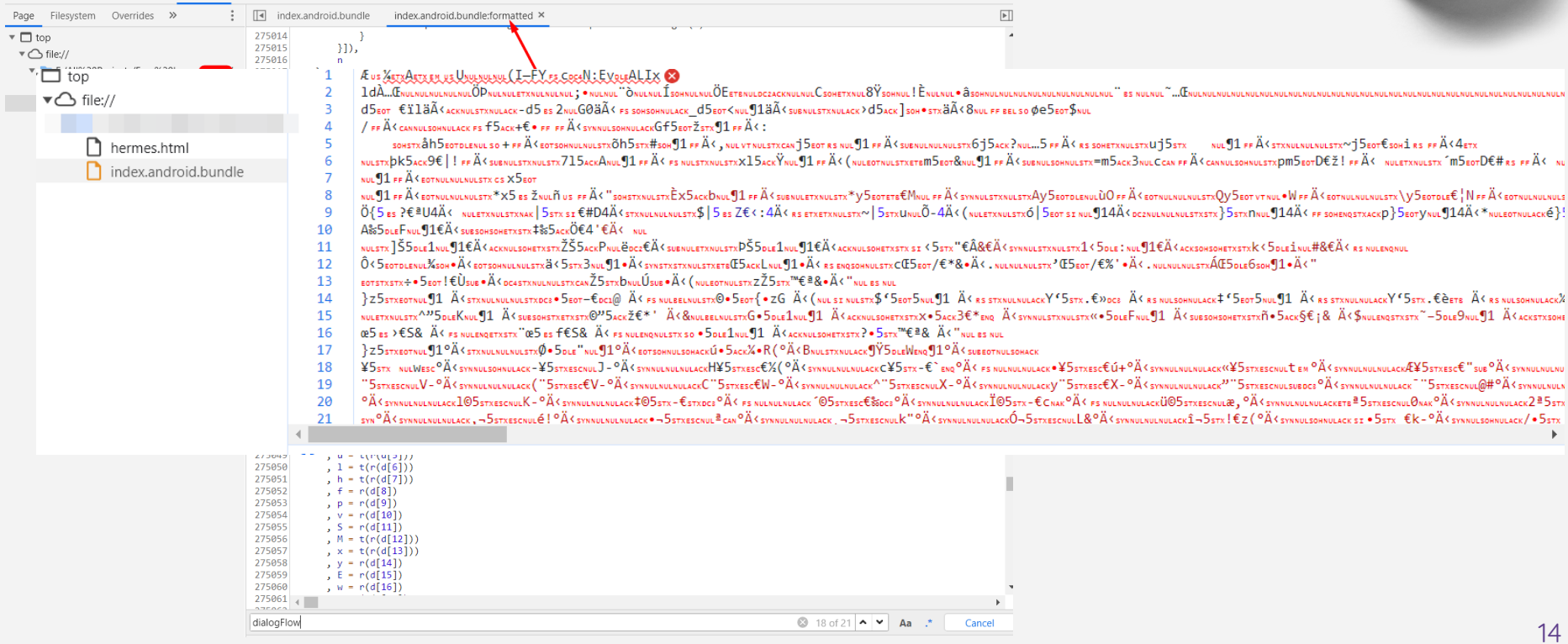
Scripts de Frida para extraer claves de cifrado en memoria no funcionan, esto debe identificar manualmente en los archivos .js.

En iOS solo es necesario instalar el certificado en el celular para interceptar el tráfico para esto no es necesario root, si la app tiene implementado ssl pinning se debe tener equipo con jailbreak.

Optimización de aplicativos con Hermes



Optimización de aplicativos con Hermes - Ejemplo



Disassembling and assembling the Hermes Bytecode

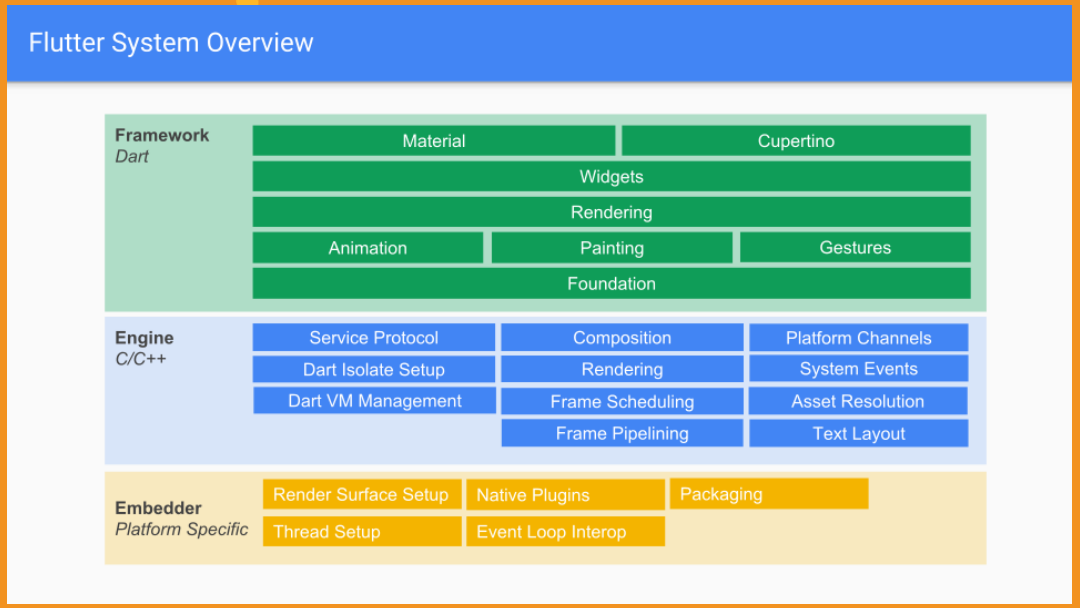
https

1 instruction.hasm

```
182874      GetById                 Reg8:5, Reg8:4, UInt8:2, UInt16:2795
182875      ; Oper[3]: String(2795) 'counter'
182876
182877      LoadConstUInt8          Reg8:4, UInt8:1
182878      Add                     Reg8:4, Reg8:5, Reg8:4
182879      PutNewOwnById           Reg8:1, Reg8:4, UInt16:2795
182880      ; Oper[2]: String(2795) 'counter'
182881
182882      Call2                   Reg8:1, Reg8:2, Reg8:3, Reg8:1
182883      LoadFromEnvironment     Reg8:1, Reg8:0, UInt8:0
182884      GetByIdShort            Reg8:1, Reg8:1, UInt8:1, UInt8:151
182885      ; Oper[3]: String(151) 'state'
182886
182887      GetById                 Reg8:2, Reg8:1, UInt8:2, UInt16:2795
182888      ; Oper[3]: String(2795) 'counter'
182889
182890      LoadConstInt            Reg8:1, Imm32:336
182891      JNotGreaterEqual        Addr8:43, Reg8:2, Reg8:1
182892      GetGlobalObject         Reg8:1
182893      TryGetById              Reg8:2, Reg8:1, UInt8:3, UInt16:3716
182894      ; Oper[3]: String(3716) 'alert'
182895
182896      LoadFromEnvironment     Reg8:4, Reg8:0, UInt8:0
182897      GetById                 Reg8:3, Reg8:4, UInt8:5, UInt16:4072
182898      ; Oper[3]: String(4072) 'decrypt'
182899
182900      LoadConstString         Reg8:1, UInt16:1724
182901      ; Oper[1]: String(1724) 'ZXxZt3UWNXVYadJ2XJZzm25vJFX93ZXnX2f
182902
182903      LoadConstString         Reg8:0, UInt16:219
182904      ; Oper[1]: String(219) 'onPress'
182905
182906      Call3                   Reg8:1, Reg8:3, Reg8:4, Reg8:1, Reg8:0
```

Optimización de aplicativos co...

Flutter



WEBSEC

El proyecto Flutter comenzó como un experimento de Google para ver si podían facilitar la codificación para los desarrolladores sin ningún conocimiento o experiencia previa. Permite a los desarrolladores crear aplicaciones y juegos desde sus hogares utilizando una base de código en dispositivos iOS y Android.

La mejor parte de este marco es que hace posible el desarrollo de aplicaciones incluso para pequeñas empresas y nuevas empresas. No se requiere tarifa por adelantado. En cambio, lo que paga se basa en los ingresos que genera su aplicación.

Consideraciones de Análisis

Análisis Estático Android/iOS

Mobsf:

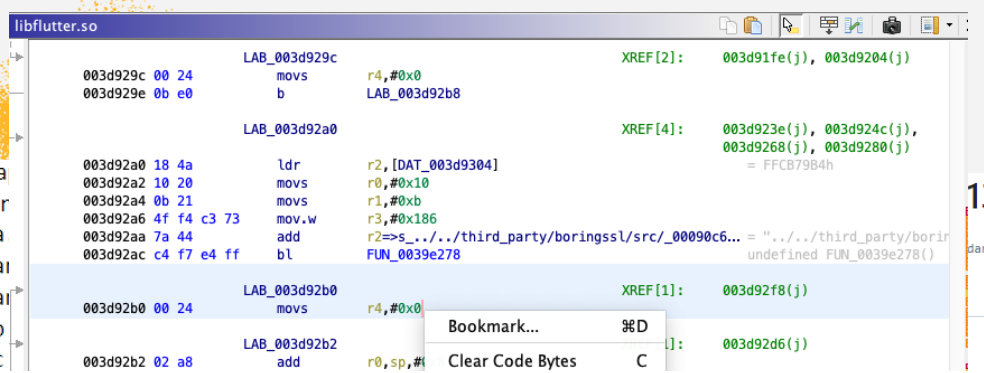
- Para la identificación de los componentes de aplicación en AndroidManifest.xml y Info.plist.
- Posible identificación de posibles secretos en otras partes de código.

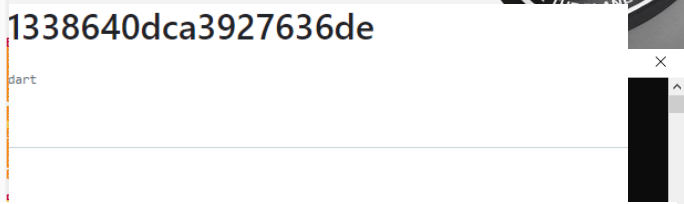
reFlutter:

- Crear dump .dart sobre el proceso de funciones

Análisis Dinámico Android/iOS

Android/iOS utilizar reFlutter para el parcheo de apps, Dart utiliza su propia base de certificados y los instalados por el usuario/sistema serán ignorados.



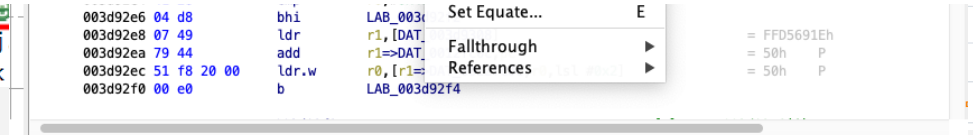


`Split APKs` is a technique used by Android to reduce the size of an application and allow users to download and use only the necessary parts of the application.

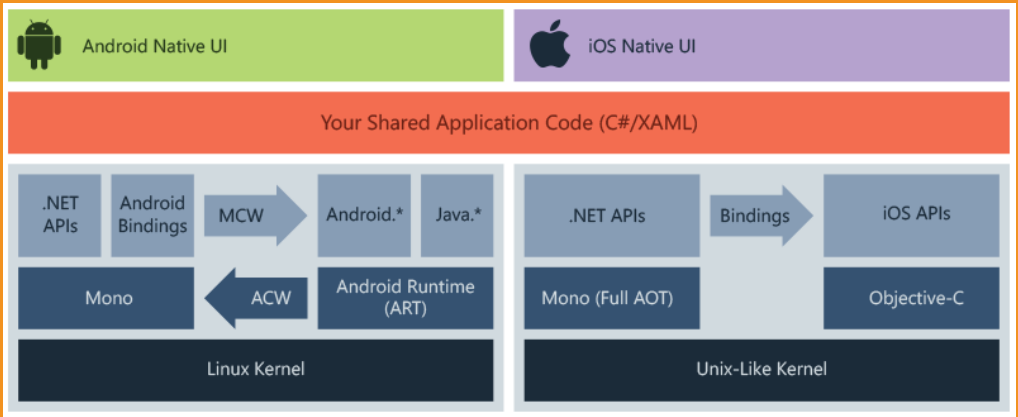
Instead of downloading a complete application in a single APK file, Split APKs divide the application into several smaller APK files, each of which contains only a part of the application such as resources, code libraries, assets, and configuration files.

```
adb shell pm path com.package
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/base.apk
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/split_config.arm64_v8a.apk
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/split_config.en.apk
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/split_config.xxhdpi.apk
```

For example, in Flutter if the application is a Split it's necessary patch split_config.arm64_v8a.apk, this file contains libflutter.so



Xamarin



Cuando se trata de React Native frente a Xamarin, Xamarin combina lo mejor de C# y .NET para desarrollar aplicaciones para Android, iOS y Mac. Mientras compila algo con lenguajes nativos, un desarrollador puede lograrlo con C# y Xamarin.

Los desarrolladores no pueden usar bibliotecas nativas de código abierto accesibles para iOS, Android y Xamarin. Para cumplir con estos requisitos, los desarrolladores pueden usar varias bibliotecas .NET y llenar el vacío.

Consideraciones de Análisis

Análisis Estático Android/iOS

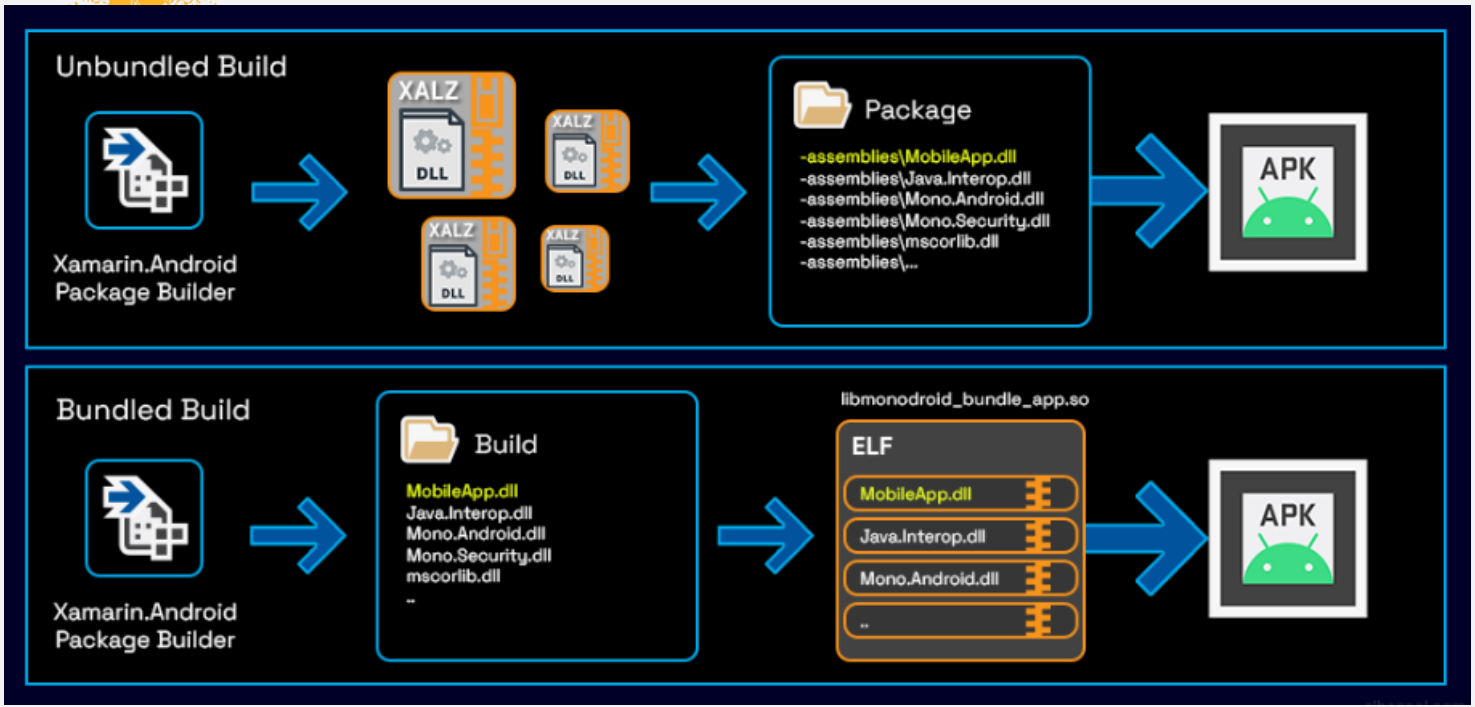
Mobsf:

- Para la identificación de los componentes de aplicación en AndroidManifest.xml y Info.plist.
- Posible identificación de posibles secretos en otras partes de código.

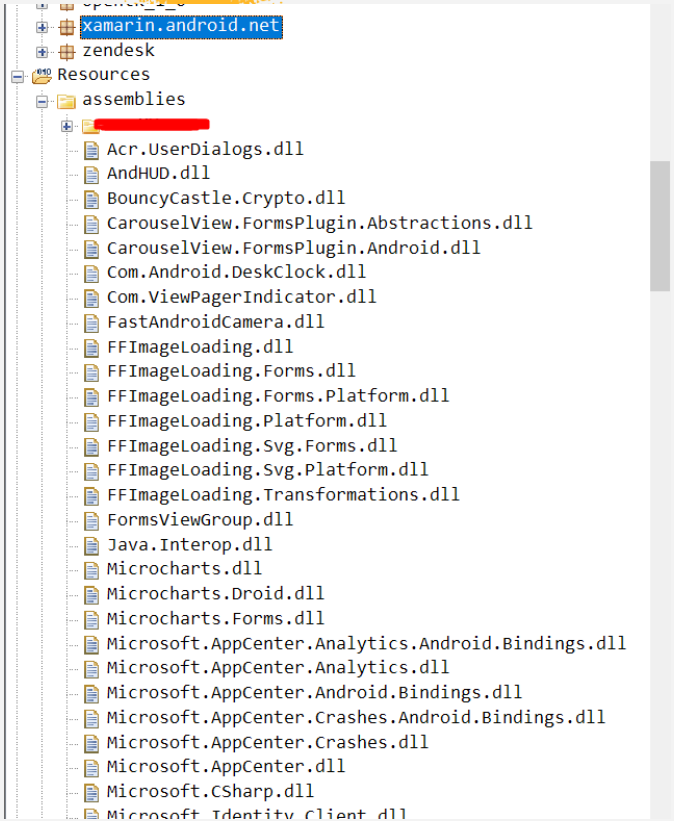
ILSpy:

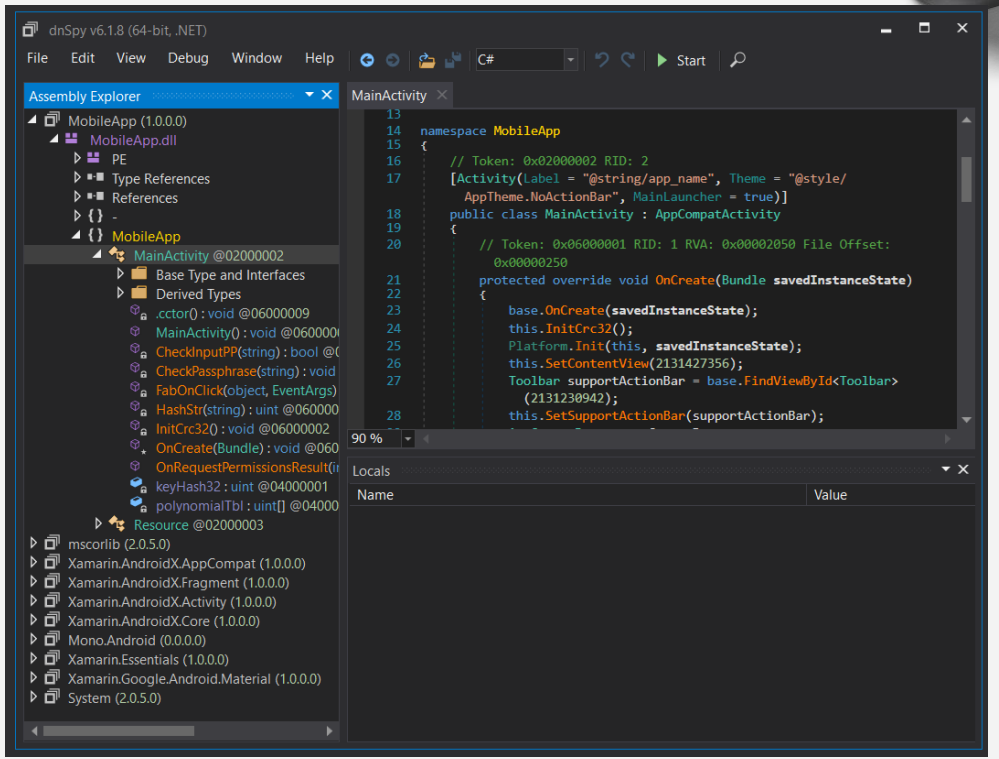
- Explorador y descompilador de ensamblados NET

Métodos de compilación en Xamarim



Análisis de Aplicativos - Unbundled Build V1





Análisis de Aplicativos - Unbundled Build V2

Add support for compressed assemblies in APK #4686

Merged jonpryor merged 2 commits into `xamarin:master` from `grendello:compress-assemblies` on May 26, 2020

Conversation 36 · Commits 2 · Checks 0 · Files change

grendello commented on May 13, 2020 • edited

Currently, `Xamarin.Android` supports managed assembly compression in the APK archive if application is bundled (with Mono's `mkbundle`) into a native shared library. Managed assemblies are compressed using gzip compression and placed in an array inside the data section of the shared library. However, support for `mkbundle` is possibly going to be removed and we realized it is a feature some developers appreciate since the produced APKs are smaller and the impact on startup time isn't big enough to worry.

This commit aims to be a replacement for `mkbundle` with a handful of improvements thrown in. First of all, the compression is performed using the managed implementation of the excellent LZ4 algorithm. This gives us a decent compression ratio and a much faster (de)compression speed than gzip/zlib offer. Also, assemblies are stored directly in the APK in their usual directory, which allows us to `mmap` them on the runtime directly from the APK. The build process calculates the size required to store the decompressed assemblies and adds a data section to `libxamarin-app.so` which makes Android allocate all the required memory when the DSO is loaded, thus removing the need of dynamic memory allocation and making the startup faster.

Compression is supported only in `Release` builds and is enabled by default, but it can be turned off by setting the `$(AndroidEnableAssemblyCompression)` MSBuild property to `False`. If there's a need to turn compression off for an individual assembly by adding the `AndroidSkipCompression` metadata item to the assembly in question using code similar to this, in the application's project file:

ILSpy

File View Window Help

(Default) C# C# 9.0 / VS 2019.8

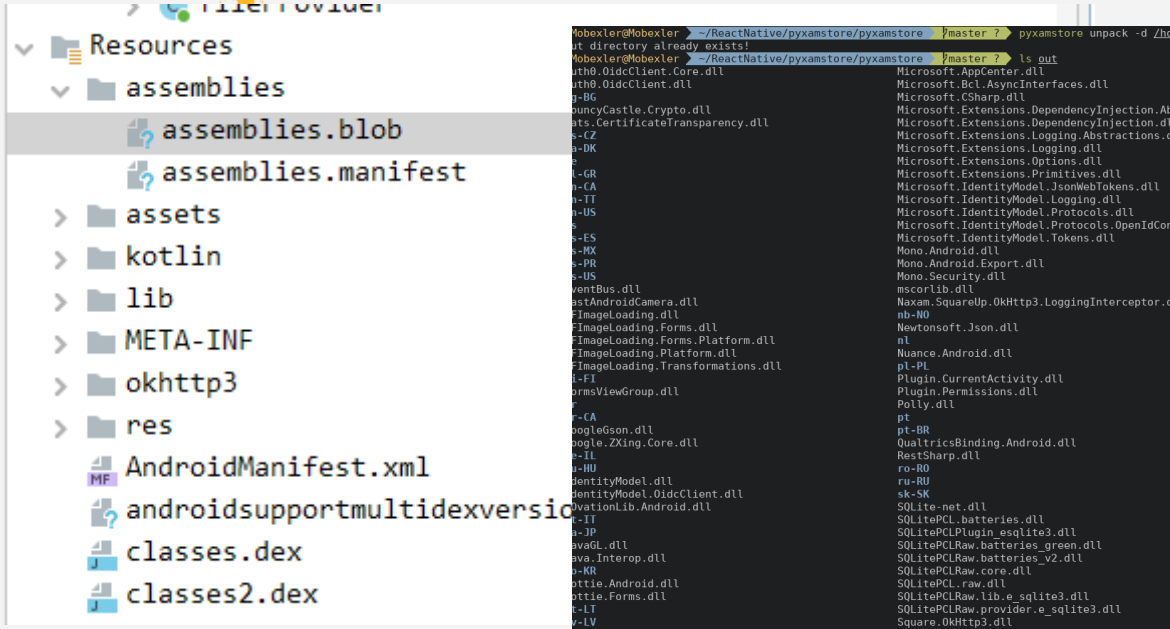
Assemblies

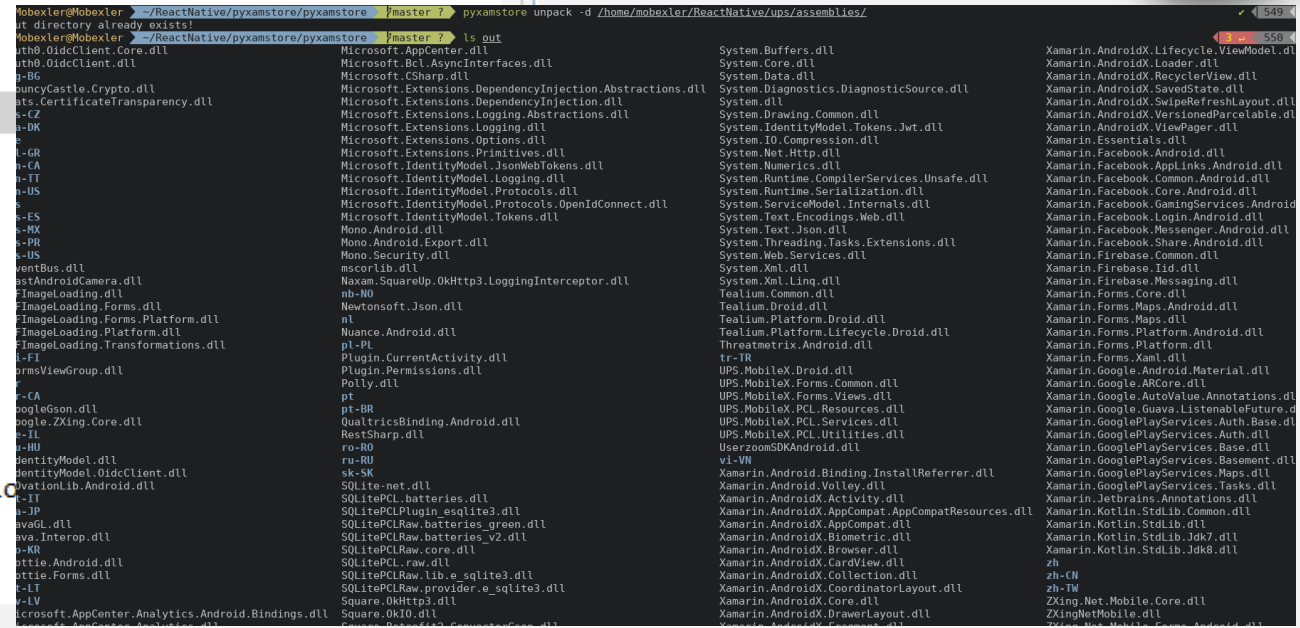
SeeTheSharpFlag

```
// This file does not contain a managed assembly.

System.BadImageFormatException: Image is too small.
   en ICSharpCode.ILSpy.LoadedAssembly.<LoadAsync>d__50.MoveNext()
--- Fin del seguimiento de la pila de la ubicación anterior donde se produjo la excepción ---
   en System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
   en System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
   en ICSharpCode.ILSpy.TreeNodes.AssemblyTreeNode.<Init>d__24.MoveNext()
```

https://github.com/x41sec/tools/blob/master/Mobile/Xamarin/Xamarin_XALZ_decompress.py

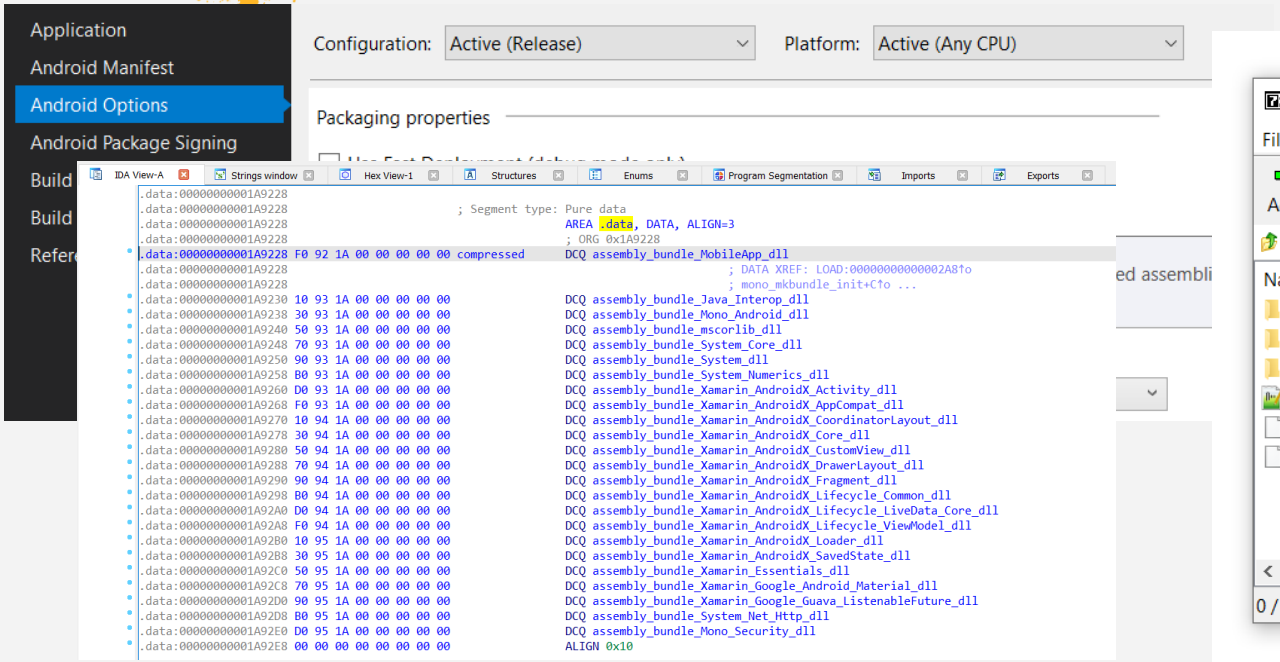
Análisis de Aplicativos - Unbundled Build V3





https://github.com/jakev/pyxamstore/

Análisis de Aplicativos - Bundled Build



https://cihansol.com/blog/index.php/2021/08/09/unpacking-xamarin-android-mobile-applications/

Análisis de Aplicativos - Bundled Build

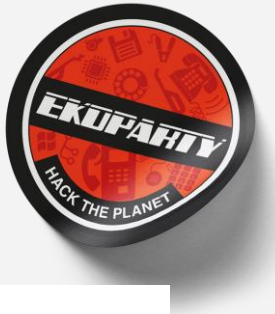
```
Z>XamAsmUnZ.exe -elf                    com.cihansol.mobileapp-packed\lib\arm64-v8a\libmonodroid_bundle_app.so
XamAsmUnZ
Author: Cihan

ELF file is of type: Bit64

Finding assembly bundles.

Bundle: [MobileApp.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [47358]
GZData Size Uncompressed: [131584]

Bundle: [Java.Interop.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [54778]
GZData Size Uncompressed: [162304]

Bundle: [Mono.Android.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [266626]
GZData Size Uncompressed: [973824]

Bundle: [mscorlib.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [679247]
GZData Size Uncompressed: [1870848]

Bundle: [System.Core.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [27538]
GZData Size Uncompressed: [54784]

Bundle: [System.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [167501]
GZData Size Uncompressed: [386048]

Bundle: [System.Numerics.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [12096]
GZData Size Uncompressed: [25600]

Bundle: [Xamarin.AndroidX.Activity.dll]
GZData Offset: [XamAsmUnZ.AssemblyBundle]
GZData Size Compressed: [2412]
GZData Size Uncompressed: [6144]
```

extracted_assemblies64

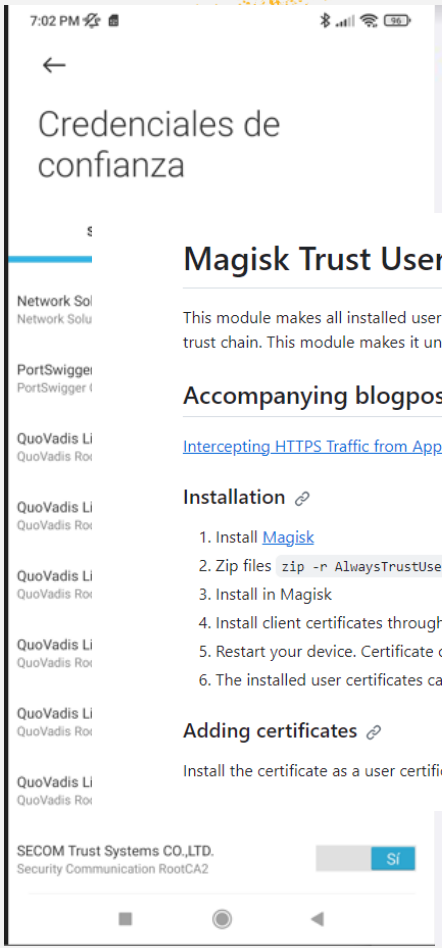
Nombre	Fecha de modificación	Tipo	Tamaño
Java.Interop.dll	6/10/2023 01:14	Extensión de la ap...	159 KB
MobileApp.dll	6/10/2023 01:14	Extensión de la ap...	129 KB
Mono.Android.dll	6/10/2023 01:14	Extensión de la ap...	951 KB
Mono.Security.dll	6/10/2023 01:14	Extensión de la ap...	107 KB
mscorlib.dll	6/10/2023 01:14	Extensión de la ap...	1,827 KB
System.Core.dll	6/10/2023 01:14	Extensión de la ap...	54 KB
System.dll	6/10/2023 01:14	Extensión de la ap...	377 KB
System.Net.Http.dll	6/10/2023 01:14	Extensión de la ap...	208 KB
System.Numerics.dll	6/10/2023 01:14	Extensión de la ap...	25 KB
Xamarin.AndroidX.Activity.dll	6/10/2023 01:14	Extensión de la ap...	6 KB
Xamarin.AndroidX.AppCompat.dll	6/10/2023 01:14	Extensión de la ap...	320 KB
Xamarin.AndroidX.Core.dll	6/10/2023 01:14	Extensión de la ap...	154 KB
Xamarin.AndroidX.CustomView.dll	6/10/2023 01:14	Extensión de la ap...	9 KB
Xamarin.AndroidX.DrawerLayout.dll	6/10/2023 01:14	Extensión de la ap...	40 KB
Xamarin.AndroidX.Fragment.dll	6/10/2023 01:14	Extensión de la ap...	149 KB
Xamarin.AndroidX.Lifecycle.Common.dll	6/10/2023 01:14	Extensión de la ap...	15 KB
Xamarin.AndroidX.Lifecycle.LiveData.Cor...	6/10/2023 01:14	Extensión de la ap...	16 KB
Xamarin.AndroidX.Lifecycle.ViewModel.dll	6/10/2023 01:14	Extensión de la ap...	17 KB
Xamarin.AndroidX.Loader.dll	6/10/2023 01:14	Extensión de la ap...	36 KB
Xamarin.AndroidX.SavedState.dll	6/10/2023 01:14	Extensión de la ap...	13 KB
Xamarin.Essentials.dll	6/10/2023 01:14	Extensión de la ap...	26 KB
Xamarin.Google.Guava.ListenableFuture.dll	6/10/2023 01:14	Extensión de la ap...	18 KB

https://github.com/cihansol/XamAsmUnZ/releases

WEBSEC

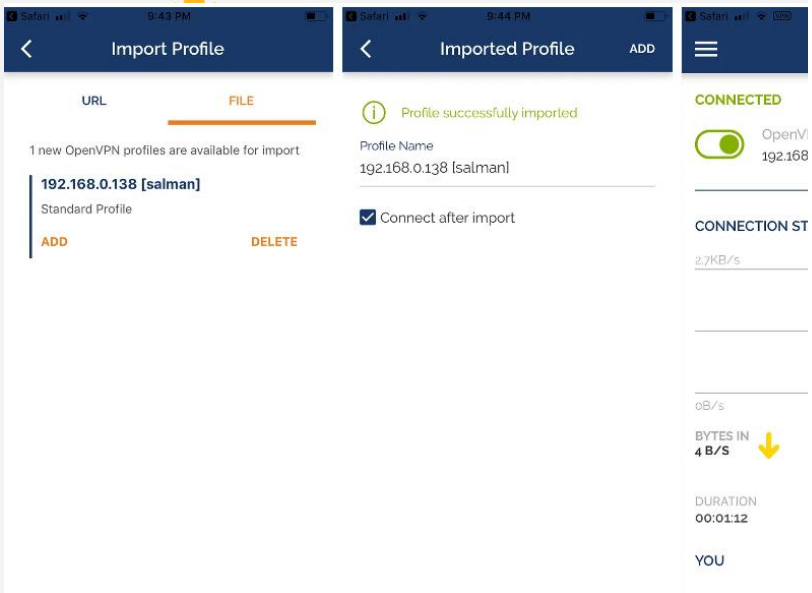


nálisis



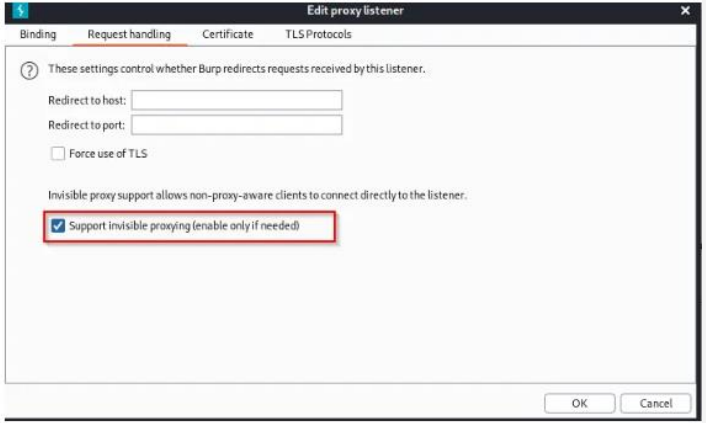
entado

Consideraciones de Análisis



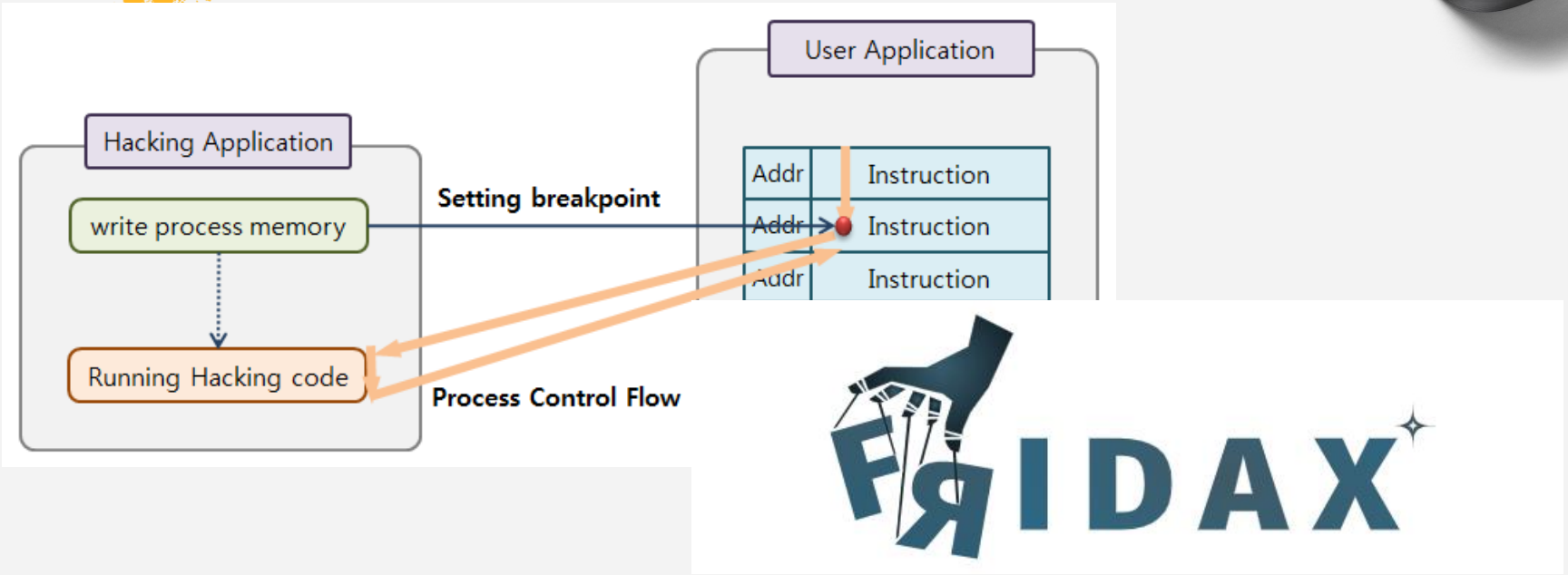
Listening on burpsuite

Enable invisible proxy



https://slmnsd552.medium.com/how-to-capture-non-proxy-aware-mobile-application-traffic-ios-android-xamarin-flutter-924fe044facf

Consideraciones de Análisis



Mobile Helper Framework

```python
class FrameWork:
    REACT_NATIVE = "React Native"
    REACT_NATIVE_IOS = "React Native iOS"
    CORDOVA = "Cordova"
    FLUTTER = "Flutter"
    FLUTTER_IOS = "Flutter iOS"
    XAMARIN = "Xamarin"
    XAMARIN_IOS = "Xamarin iOS"
    NATIVESCRIPT = "NativeScript"
    NATIVE = "Native (Java/Kotlin)"
    NATIVE_IOS = "Native (Objective-C/Swift)"
    XAMARINV2 = "Xamarin"


class Technology:
    def __init__(self, framework, directories):
        self.framework = framework
        self.directories = directories


tech_list = [
    Technology(
        framework=FrameWork.REACT_NATIVE,
        directories=[
            "libreactnativejni.so",
            "index.android.bundle",
        ]
    ),
    Technology(
        framework=FrameWork.REACT_NATIVE_IOS,
        directories=[
            "main.jsbundle",
        ]
    ),
    Technology(
        framework=FrameWork.CORDOVA,
        directories=[
            "index.html",
            "cordova.js",
            "cordova_plugins.js"
        ]
    ),
```

```python
            "index.android.bundle",
        ]
    ),
    Technology(
        framework=FrameWork.REACT_NATIVE_IOS,
        directories=[
            "main.jsbundle",
        ]
    ),
    Technology(
        framework=FrameWork.CORDOVA,
        directories=[
            "index.html",
            "cordova.js",
            "cordova_plugins.js"
        ]
    ),
    Technology(
        framework=FrameWork.FLUTTER,
        directories=[
            "libflutter.so"
        ]
    ),
    Technology(
        framework=FrameWork.FLUTTER_IOS,
        directories=[
            "Flutter.framework/Flutter"
        ]
    ),
    Technology(
        framework=FrameWork.XAMARIN,
        directories=[
            "Mono.Android.dll",
            "libmonodroid.so",
            "libmonosgen-2.0.so",
        ]
    ),
    Technology(
        framework=FrameWork.XAMARINV2,
        directories=[
            "assemblies.blob",
            "assemblies.manifest",
        ]
    ),
    Technology(
        framework=FrameWork.XAMARIN_IOS,
        directories=[
            "Xamarin.iOS.dll"
```

WEBSEC

EKOPARTY HACK THE PLANET

Es una herramienta que automatiza el proceso de identificar el marco de trabajo/tecnología utilizada para crear una aplicación móvil. Además, ayuda a encontrar información sensible o proporciona sugerencias para trabajar con la plataforma identificada.

https://github.com/stuxctf/mhf

Funcionalidades

Features

This tool uses Apktool for decompilation of Android applications.

This tool renames the .ipa file of iOS applications to .zip and extracts the contents.

Feature	Note	Cordova	React Native	Native JavaScript	Flutter	Xamarin
JavaScript beautifier	Use this for the first few occasions to see better results.	✅	✅	✅		
Identifying multiple sensitive information	IPs, Private Keys, API Keys, Emails, URLs	✅	✅	✅	❌	
Automaticall						
Extracts a						
Ex						
[						
Detect if the resources are compressed.		❌	Hermes ✅	❌	❌	XALZ ✅
Detect if the app is split		❌	❌	❌	❌	❌

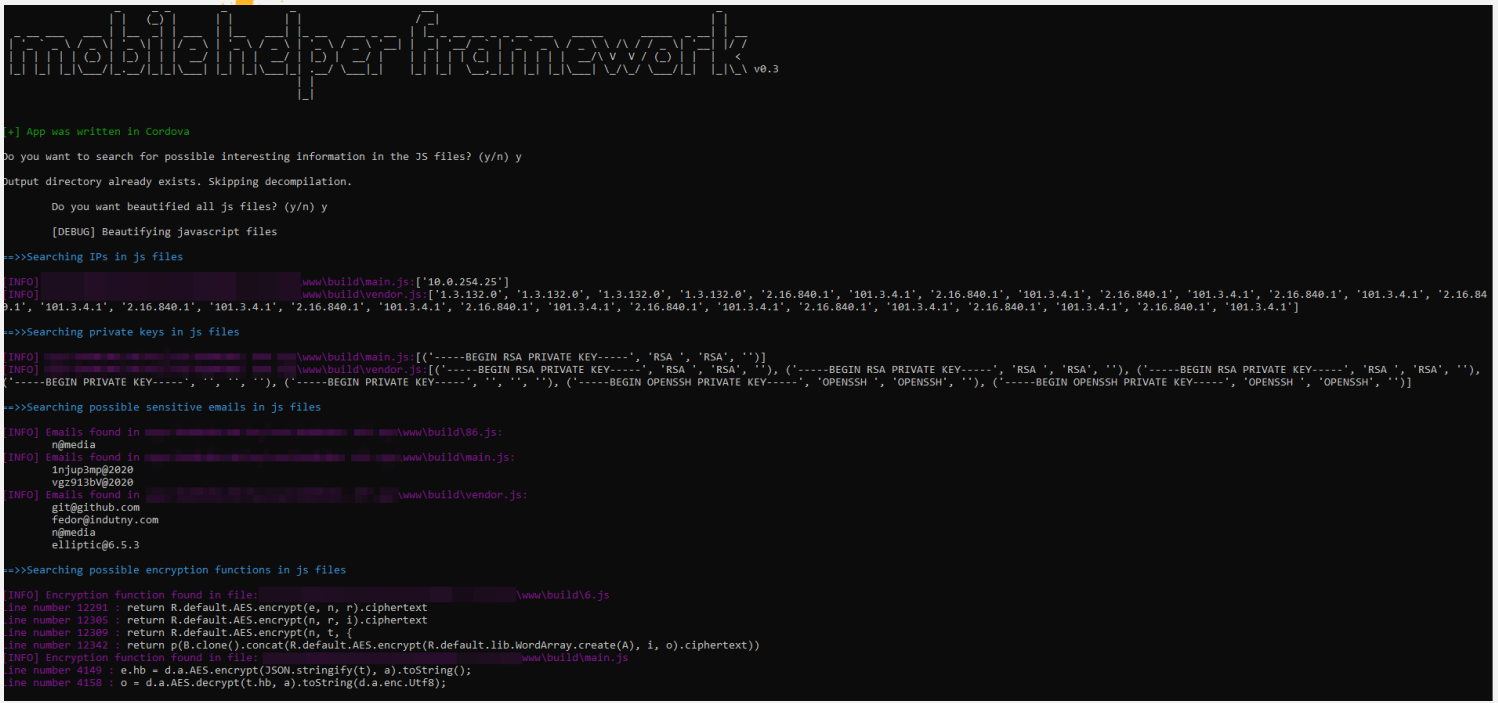
`Split APKs` is a technique used by Android to reduce the size of an application and allow users to download and use only the necessary parts of the application.

Instead of downloading a complete application in a single APK file, Split APKs divide the application into several smaller APK files, each of which contains only a part of the application such as resources, code libraries, assets, and configuration files.

```
adb shell pm path com.package
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/base.apk
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/split_config.arm64_v8a.apk
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/split_config.en.apk
package:/data/app/com.package-NW8ZbgI5VPzvSZ1NgMa4CQ==/split_config.xxhdpi.apk
```

For example, in Flutter if the application is a Split it's necessary patch split_config.arm64_v8a.apk, this file contains libflutter.so

Ejemplo - Codova



Ejemplo – React Native

```
[+] App was written in React Native

Do you want to look for possible interesting information in the bundle file associated with the application? (Y/n): Y

Output directory already exists. Skipping decompilation.

Do you want beautified the react code? (Y/n): n

Continue with the operation? (Y/n): Y

==>>Searching possible internal IPs in the file


==>>Searching possible emails in the file

[INFO] Emails found: {'rd

==>>Searching possible encryption functions in the file

[INFO] Encryption function found in line 224203:
CryptoJS.enc;

==>>Searching Private Keys in the file


==>>Searching possible interesting words in the file

accessToken

==>>Searching high confidential secrets

[INFO] Generic API Key identify C:\Users\ccalderon\Downloads\mcdonals_REACT\assets\index.android.bundle:["
```

Ejemplo - Xamarin

```
Do you want get DLL information? (Y/n): Y

Decompiling application [redacted]com.cihansol.mobileapp-nopack.apk

      [!] APK compiled in mode Unbundled Build
      The package builder will utilize LZ4 Compression and create files with the .dll extension

[!] To reverse
      Windo
      Windo
      Linux,

Possible main

DLL Compresed
```

```
[+] App was written in Xamarin

Do you want get DLL information? (Y/n): Y

Output directory already exists. Skipping decompilation.

      [!] APK compiled in mode Bundled Build
      The package builder bundles .dll assemblies into native code, in the file:

      Bundle path:
                  com.cihansol.mobileapp-packed (1)\lib\arm64-v8a\libmonodroid_bundle_app.so
                  com.cihansol.mobileapp-packed (1)\lib\armeabi-v7a\libmonodroid_bundle_app.so

      For extraction of the .dll assemblies use: https://github.com/cihansol/XamAsmUnZ
```

```
\com.cihansol.mobileapp-nopack\unknown\assemblies\System.Numerics.dll
\com.cihansol.mobileapp-nopack\unknown\assemblies\Xamarin.AndroidX.Activity.dll
\com.cihansol.mobileapp-nopack\unknown\assemblies\Xamarin.AndroidX.AppCompat.dll
\com.cihansol.mobileapp-nopack\unknown\assemblies\Xamarin.AndroidX.Core.dll
\com.cihansol.mobileapp-nopack\unknown\assemblies\Xamarin.AndroidX.CustomView.dll
\com.cihansol.mobileapp-nopack\unknown\assemblies\Xamarin.AndroidX.DrawerLayout.dll
\com.cihansol.mobileapp-nopack\unknown\assemblies\Xamarin.AndroidX.Fragment.dll
\com.cihansol.mobileapp-nopack\unknown\assemblies\Xamarin.AndroidX.Lifecycle.Common.dll
```

Referencias

https://slmnsd552.medium.com/how-to-capture-non-proxy-aware-mobile-application-traffic-ios-android-xamarin-flutter-924fe044facf

https://swarm.ptsecurity.com/fork-bomb-for-flutter/

https://blog.tst.sh/reverse-engineering-flutter-apps-part-1/

https://cihansol.com/blog/index.php/2021/08/09/unpacking-xamarin-android-mobile-applications/

https://suam.wtf/posts/react-native-application-static-analysis-en/

https://book.hacktricks.xyz/mobile-pentesting/cordova-apps

https://www.thecobraden.com/posts/unpacking_xamarin_assembly_stores/

https://book.hacktricks.xyz/mobile-pentesting/android-app-pentesting/react-native-application

https://www.youtube.com/watch?v=ovW3E09gWjY&list=PLZsnFVE_qTjB5FRgECU8gOQeVCPBoQOz9


EKOPARTY
HACK THE PLANET
¡GRACIAS!
@_websec
@__stux
HACKING EVERYTHING
EP 2023
EKOPARTY