RANDORISEC

Mobile Hacking Workshop – iOS

Dia dhuit !

March 27th 2021

Agenda

1. Testing environment
2. Basics
3. Static analysis – 2 labs
4. Data Security – 3 labs
5. Execution analysis – 2 labs
6. Transport Security – 1 lab

Legend

Lab: practical exercice

OWASP M2 Insecure data storage

OWASP Top10 item covered

A macOS is needed for this lab

Only for macOS

Testing environnement

Setup

- For this workshop, we are going to use
  - A **VM**: Mobexler **(credentials: Mobexler/12345)**
    - With **tools**:



  - **Please use VMWare Workstation Player or Fusion (DO NOT use Virtual Box)**

Basics

Main steps of a security iOS application assessment

1. Review the codebase or reverse engineer the binary
2. Run the app on a jailbroken device
3. Inspect and manipulate the app via instrumentation
4. Manipulate the runtime
5. MiTM all the network communications

OWASP Mobile Security Testing Project

Project = standard + checklist + guide

GitHub, Inc. [US] | https://github.com/OWASP/owasp-mstg

README.md

OWASP Mobile Security Testing Guide

Follow 542

owasp lab project build passing

This is the official GitHub Repository of the OWASP Mobile Security Testing Guide (MSTG). The MSTG is a comprehensive manual for mobile app security testing and reverse engineering. It describes technical processes for verifying the controls listed in the OWASP Mobile Application Verification Standard (MASVS). You can also read the MSTG on Gitbook or download it as an e-book.

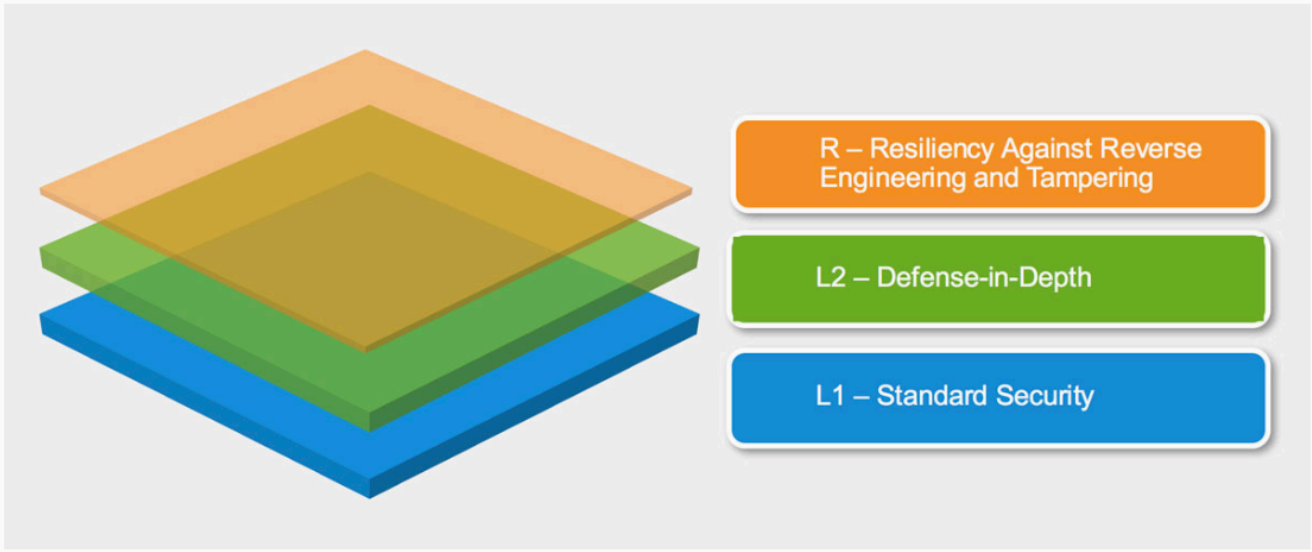
MASVS (Mobile AppSec Verification **Standard**)

Last version: version 1.2 (march 2020)

V1 Architecture, design and threat modelling
V2 Data Storage and Privacy
V3 Cryptography
V4 Authentication and Session Management
V5 Network Communication
V6 Platform Interaction
V7 Code Quality and Build Settings

OWASP Mobile AppSec **Checklist**

ID	MSTG-ID	Detailed Verification Requirement	Level 1	Level 2	Status	
V1		**Architecture, design and threat modelling**				
V2		**Data Storage and Privacy**				
2.1	MSTG-STORAGE-1	System credential storage facilities need to be used to store sensitive data, such as PII, user credentials or cryptographic keys.	✓	✓		Testing Loca
2.2	MSTG-STORAGE-2	No sensitive data should be stored outside of the app container or system credential storage facilities.	✓	✓		Testing Loca
2.3	MSTG-STORAGE-3	No sensitive data is written to application logs.	✓	✓		Checking Log
2.4	MSTG-STORAGE-4	No sensitive data is shared with third parties unless it is a necessary part of the architecture.	✓	✓		Determining
2.5	MSTG-STORAGE-5	The keyboard cache is disabled on text inputs that process sensitive data.	✓	✓		Finding Sens
2.6	MSTG-STORAGE-6	No sensitive data is exposed via IPC mechanisms.	✓	✓		Determining
2.7	MSTG-STORAGE-7	No sensitive data, such as passwords or pins, is exposed through the user interface.	✓	✓		Checking for
2.8	MSTG-STORAGE-8	No sensitive data is included in backups generated by the mobile operating system.		✓	N/A	Testing Back
2.9	MSTG-STORAGE-9	The app removes sensitive data from views when moved to the background.		✓	N/A	Testing Auto
2.15	MSTG-STORAGE-15	The app's local storage should be wiped after an excessive				
V3		**Cryptography**				
3.1	MSTG-CRYPTO-1	The app does not rely on symmetric cryptography with ha				



OWASP Mobile Security Testing **Guide**

The guide:

- 3 sections:
  - General (common to Android and iOS)
  - Android
  - iOS
- + 500 pages

OWASP - Top10 Mobile Risks

OWASP M1 Improper Platform Usage	OWASP M2 Insecure Data Storage	OWASP M3 Insecure Communication	OWASP M4 Insecure Authentication
OWASP M5 Insufficient Cryptography	OWASP M6 Insecure Authorization	OWASP M7 Client code quality	OWASP M8 Code Tampering
	OWASP M9 Reverse engineering	OWASP M10 Extraneous Functionality	

What is an IPA ?

- iOS App Store Package
- Archive (can be renamed to .zip and unzipped)
- Binary and static files (signed)

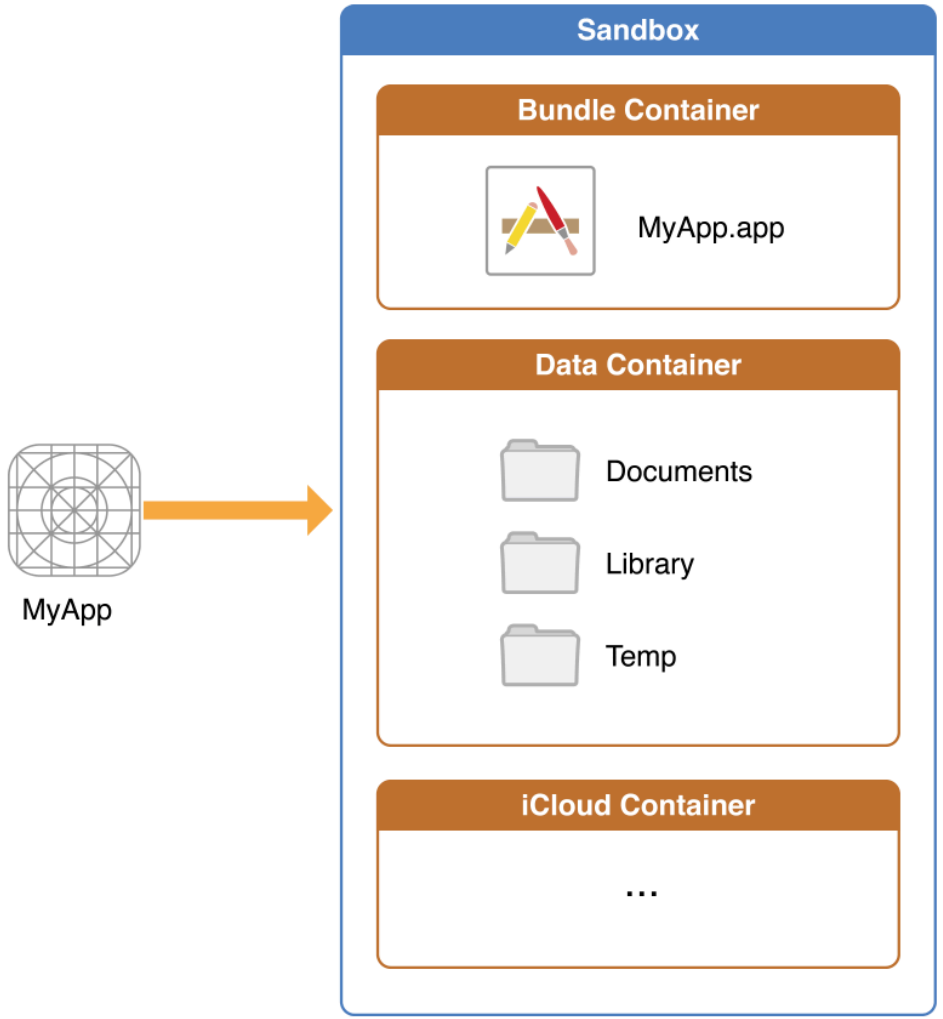
```
480B May 25  2018 Frameworks
1.1M May 25  2018 iGoat-Swift
 15K May 25  2018 embedded.mobileprovision
 96B May 25  2018 ..
 96B May 25  2018 _CodeSignature
372B May 25  2018 archived-expanded-entitlements.xcent
128B May 25  2018 Base.lproj
1.4K May 25  2018 Info.plist
  8B May 25  2018 PkgInfo
189K May 25  2018 Assets.car
128B May 25  2018 CrossSiteScriptingExerciseVC.nib
128B May 25  2018 HTMLViewController.nib
3.0K May 25  2018 AppIcon29x29@2x.png
5.3K May 25  2018 AppIcon29x29@3x.png
```

Filesystem: **System applications**

/Applications

```
iPhone:/Applications root# ls -al
total 0
drwxr-xr-x  76 root wheel 2432 May 31 06:20 ./
drwxr-xr-x  27 root wheel  864 May 31 06:18 ../
drwxrwxr-x  46 root admin 1472 May 21 20:47 AXUIViewService.app/
drwxrwxr-x   7 root admin  224 May 21 20:47 AccountAuthenticationDialog.app/
drwxrwxr-x   7 root admin  224 May 21 20:47 ActivityMessagesApp.app/
drwxrwxr-x   9 root admin  288 May 21 20:47 AdPlatformsDiagnostics.app/
drwxrwxr-x  52 root admin 1664 May 21 20:47 AppStore.app/
drwxrwxr-x  47 root admin 1504 May 21 20:47 AskPermissionUI.app/
drwxrwxr-x   7 root admin  224 May 21 20:47 BusinessExtensionsWrapper.app/
drwxrwxr-x  46 root admin 1472 May 21 20:47 CTCarrierSpaceAuth.app/
drwxrwxr-x  58 root admin 1856 May 21 20:47 Camera.app/
drwxrwxr-x  47 root admin 1504 May 21 20:47 CheckerBoard.app/
drwxrwxr-x  46 root admin 1472 May 21 20:47 CompassCalibrationViewService.app/
drwxrwxr-x   8 root admin  256 May 21 20:46 ContinuityCamera.app/
drwxrwxr-x  49 root admin 1568 May 21 20:46 CoreAuthUI.app/
drwxr-xr-x 124 root wheel 3968 May 31 01:37 Cydia.app/
drwxrwxr-x  46 root admin 1472 May 21 20:46 DDActionsService.app/
```

Filesystem: **User applications**



Filesystem: **User applications**

/private/var/containers/Bundle/Application

```
iPhone:/private/var/containers/Bundle/Application root# ls -al
total 0
drwxr-xr-x 29 _installd _installd 928 Jun 15 09:48 ./
drwxr-xr-x  6 _installd _installd 192 May 31 01:38 ../
drwxr-xr-x  5 _installd _installd 160 Jun 10 03:54 07B03D18-01A1-4F1E-A355-F000AC9B9F35/
drwxr-xr-x  5 _installd _installd 160 May 31 01:39 121F8420-4F80-4398-8C22-240CEFEF6F54/
drwxr-xr-x  5 _installd _installd 160 May 31 01:39 1E1ACAAC-CD00-47D2-BD0D-68A23A68A85A/
drwxr-xr-x  5 _installd _installd 160 May 31 02:19 277F19A6-D521-48ED-B75E-9D1E8FCA8C90/
drwxr-xr-x  5 _installd _installd 160 May 31 01:52 27FFB5C0-7872-4552-894D-D2374A0ACDD9/
drwxr-xr-x  5 _installd _installd 160 Jun 10 03:51 2E2449F1-AE2A-44A1-8A71-2FBF132852BD/
drwxr-xr-x  5 _installd _installd 160 May 31 01:38 43B016BC-5984-45A5-A4A5-B315E5046993/
drwxr-xr-x  5 _installd _installd 160 May 31 01:39 62784C25-81EB-4CCA-9BDE-C71AE162EA0A/
drwxr-xr-x  5 _installd _installd 160 May 31 01:39 65A879DB-CC9B-4AC6-8DB9-EFABC3DCDF90/
drwxr-xr-x  5 _installd _installd 160 May 31 01:39 7D73F79E-78CF-487B-8C9E-7A3B15CE13E0/
drwxr-xr-x  5 _installd _installd 160 May 31 01:38 81A56A73-B663-4566-8687-2EBE4C04ADD7/
```

Filesystem: User applications: **Bundle** Directory

/private/var/containers/Bundle/Application/UUID/App.app

```
iPhone:/private/var/containers/Bundle/Application/4E1CB17B-5A86-468D-AD57-F92C9F11A5B2/DamnVulnerableIOSApp.app root# ls -al
total 6456
drwxr-xr-x 38 _installd _installd    1216 Oct 26 18:49 ./
drwxr-xr-x  5 _installd _installd     160 Oct 26 18:49 ../
-rw-r--r--  1 _installd _installd   11553 Dec  2  2014 120x120.png
-rw-r--r--  1 _installd _installd   13907 Dec  2  2014 152x152.png
-rw-r--r--  1 _installd _installd    6525 Dec  2  2014 57x57.png
-rw-r--r--  1 _installd _installd  375699 Dec  2  2014 640_960_SplashScn.png
-rw-r--r--  1 _installd _installd  464522 Dec  2  2014 640x1136_SplashScn.png
-rw-r--r--  1 _installd _installd    7893 Dec  2  2014 72x72.png
-rw-r--r--  1 _installd _installd    8464 Dec  2  2014 76x76.png
-rw-r--r--  1 _installd _installd   11292 Dec  2  2014 AppIcon40x40\@2x.png
-rw-r--r--  1 _installd _installd   11553 Dec  2  2014 AppIcon60x60\@2x.png
drwxr-xr-x  3 _installd _installd      96 Oct 26 18:49 Base.lproj/
-rwxr-xr-x  1 _installd _installd 4483296 May 26 11:32 DamnVulnerableIOSApp*
-rw-r--r--  1 _installd _installd     423 May 26 11:32 DamnVulnerableIOSApp.entitlements
-rw-r--r--  1 _installd _installd    1410 Jan  1  1980 Info.plist
-rw-r--r--  1 _installd _installd  464522 Dec  2  2014 LaunchImage-700-568h\@2x.png
```

- Static files (signed):
- Binary
- Images
- Properties

Filesystem: User applications: Bundle Directory

Info.plist

```
plistutil -i Info.plist
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>CFBundleName</key>
        <string>DamnVulnerableIOSApp</string>
        <key>DTSDKName</key>
        <string>iphoneos8.1</string>
        <key>DTXcode</key>
        <string>0610</string>
        <key>DTSDKBuild</key>
        <string>12B411</string>
        <key>CFBundleDevelopmentRegion</key>
        <string>en</string>
        <key>CFBundleVersion</key>
        <string>1.0</string>
        <key>BuildMachineOSBuild</key>
        <string>14B25</string>
        <key>DTPlatformName</key>
        <string>iphoneos</string>
        <key>CFBundleShortVersionString</key>
        <string>1.3</string>
```

Filesystem: User applications: **Data** Directory

/private/var/mobile/Containers/Data/Application/UUID

```
iPhone:/private/var/mobile/Containers/Data/Application/AA7D5264-12B6-4109-A397-C007299AB958 root# ls -al
total 4
drwxr-xr-x  7 mobile mobile  224 May 31 01:52 ./
drwxr-xr-x 81 mobile mobile 2592 Jun 15 09:48 ../
-rw-r--r--  1 root   mobile  211 May 31 01:52 .com.apple.mobile_container_manager.metadata.plist
drwxr-xr-x  3 mobile mobile   96 Jun 11 10:45 Documents/
drwxr-xr-x  5 mobile mobile  160 Jun 11 02:14 Library/
drwxr-xr-x  2 mobile mobile   64 May 31 01:52 SystemData/
drwxr-xr-x 25 mobile mobile  800 Jun 15 16:25 tmp/
iPhone:/private/var/mobile/Containers/Data/Application/AA7D5264-12B6-4109-A397-C007299AB958 root# du -ah
12K     ./Documents/credentials.sqlite
12K     ./Documents
4.0K    ./.com.apple.mobile_container_manager.metadata.plist
52K     ./Library/Caches/com.swaroop.iGoat/Cache.db
0       ./Library/Caches/com.swaroop.iGoat/Cache.db-wal
32K     ./Library/Caches/com.swaroop.iGoat/Cache.db-shm
84K     ./Library/Caches/com.swaroop.iGoat
40K     ./Library/Caches/Snapshots/com.swaroop.iGoat/600E684E-7B24-4386-947A-FDF646E30248@2x.atx
36K     ./Library/Caches/Snapshots/com.swaroop.iGoat/downscaled/D29866C1-7955-4271-B34A-5F63AF19BB65@2x.atx
36K     ./Library/Caches/Snapshots/com.swaroop.iGoat/downscaled
76K     ./Library/Caches/Snapshots/com.swaroop.iGoat
76K     ./Library/Caches/Snapshots
160K    ./Library/Caches
0       ./Library/Preferences
4.0K    ./Library/Cookies/com.swaroop.iGoat.binarycookies
```

- User data
- Settings
- Cookies
- Cached files
- Temp files

Filesystem: to sum up

Bundle directory `/private/var/containers/Bundle/Application/UUID/App`

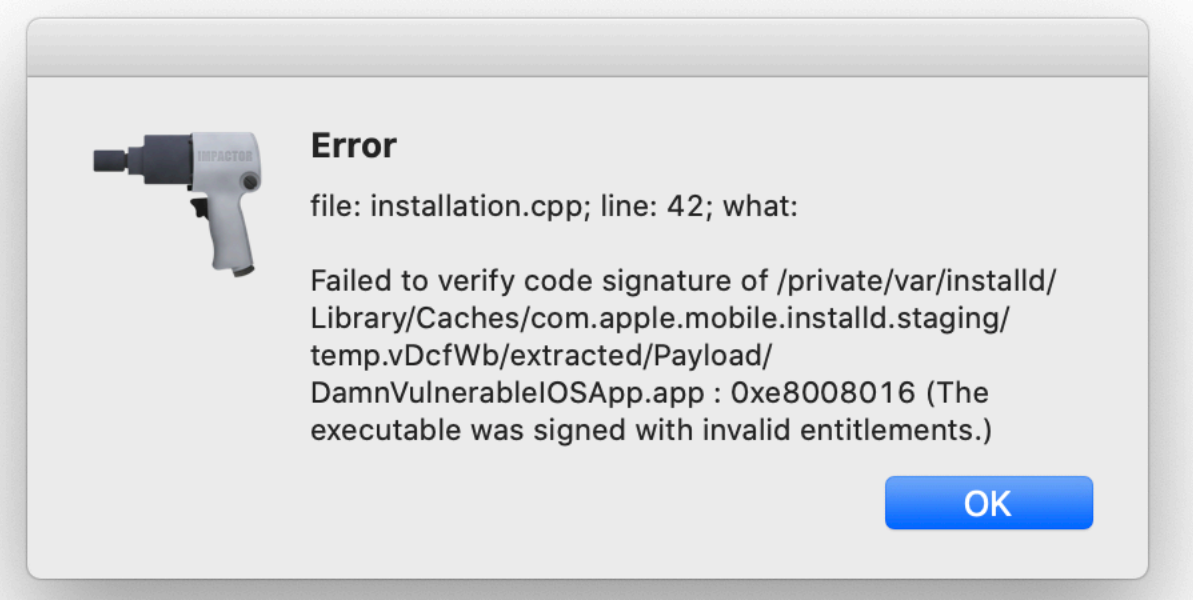
Data directory `/private/var/mobile/Containers/Data/Application/UUID`

Ex:

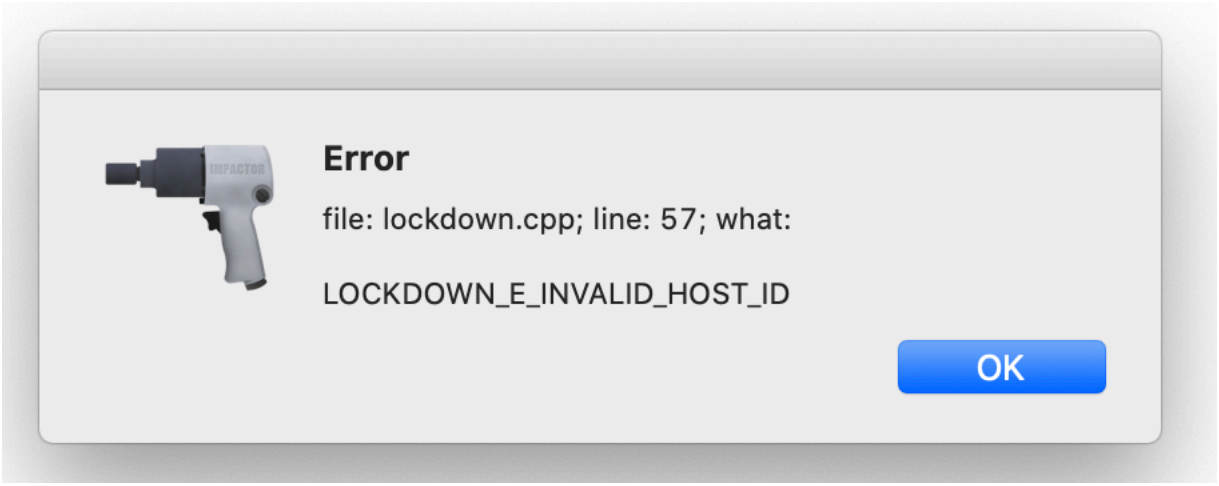
Bundle directory `/private/var/containers/Bundle/Application/E69C6843-CD63-41C7-9AA8-120177BACA73/Avatarify.app`

Data directory `/private/var/mobile/Containers/Data/Application/31D5446F-A320-4C83-B592-84783B3F68FF`

Can’t install this IPA 🤬
(aka signing problems)

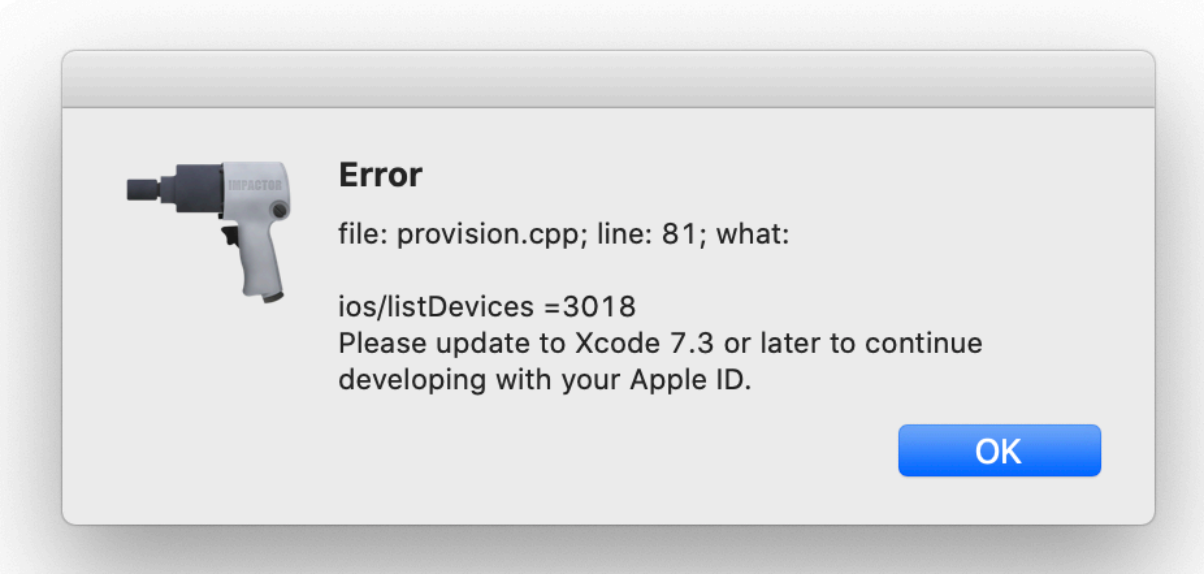


Signing problems: *file: lockdown.cpp; line: 57*



- Trust This Computer ?
- **Select « Trust »**

Signing problems: *file: provision.cpp; line: 81*



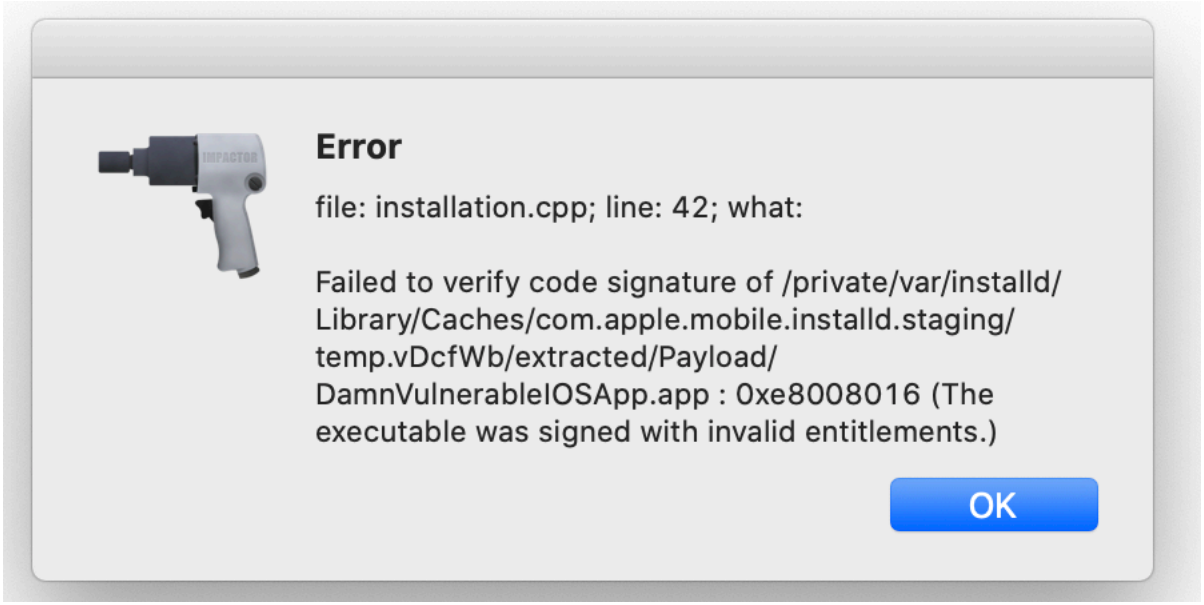
- Go to: https://developer.apple.com/programs/enroll/
- **Enroll on the Apple Developer Program**

Signing problems: *file: provision.cpp; line: 173*

- Go to: https://appleid.apple.com/account/manage
- Enable 2FA
- **Generate app-specific password**

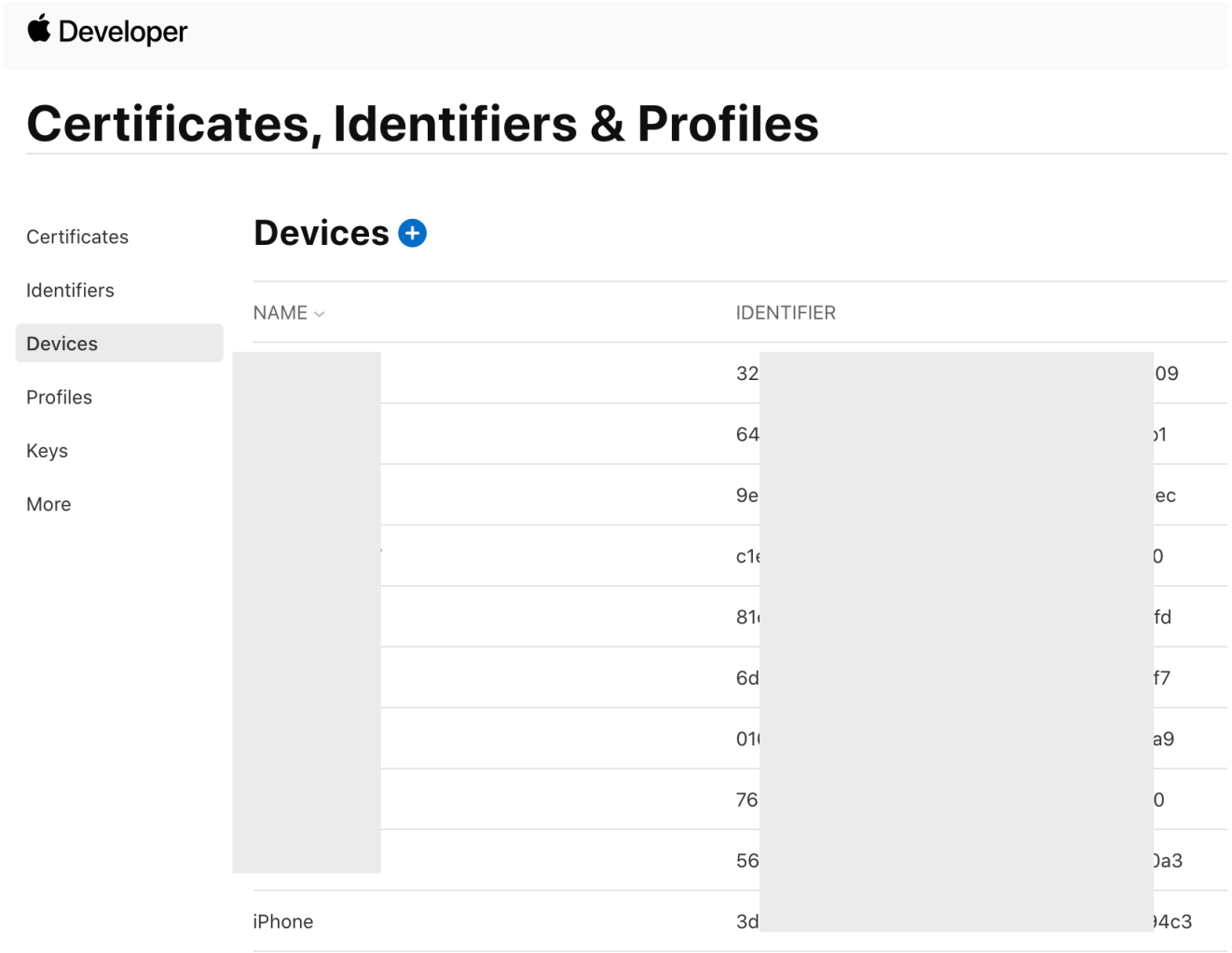
Signing problems: *file: installation.cpp; line: 42*

Need a macOS



- Go to: https://developer.apple.com/
- Create a **provisioning profile**
- **Sign** your IPA with it

Signing problems: *sign your IPA*



Need a macOS

Signing problems: *sign your IPA*

Need a macOS

Developer

Certificates, Identifiers & Profiles

‹ All Profiles

Register a New Provisioning Profile

Development

- **iOS App Development**
  Create a provisioning profile to install development apps on test devices.
- **tvOS App Development**
  Create a provisioning profile to install development apps on tvOS test devices.
- **macOS App Development**
  Create a provisioning profile to install development apps on test devices.

Distribution

- **Ad Hoc**
  Create a distribution provisioning profile to install your app on a limited number of registered devices.

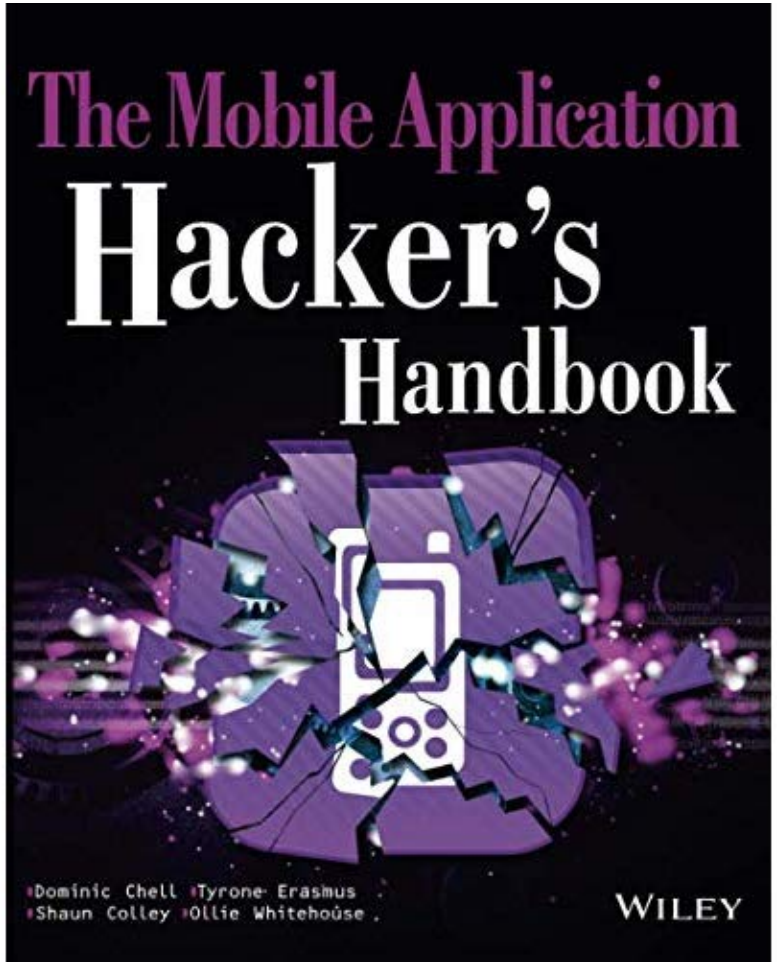
Signing problems: *sign your IPA*

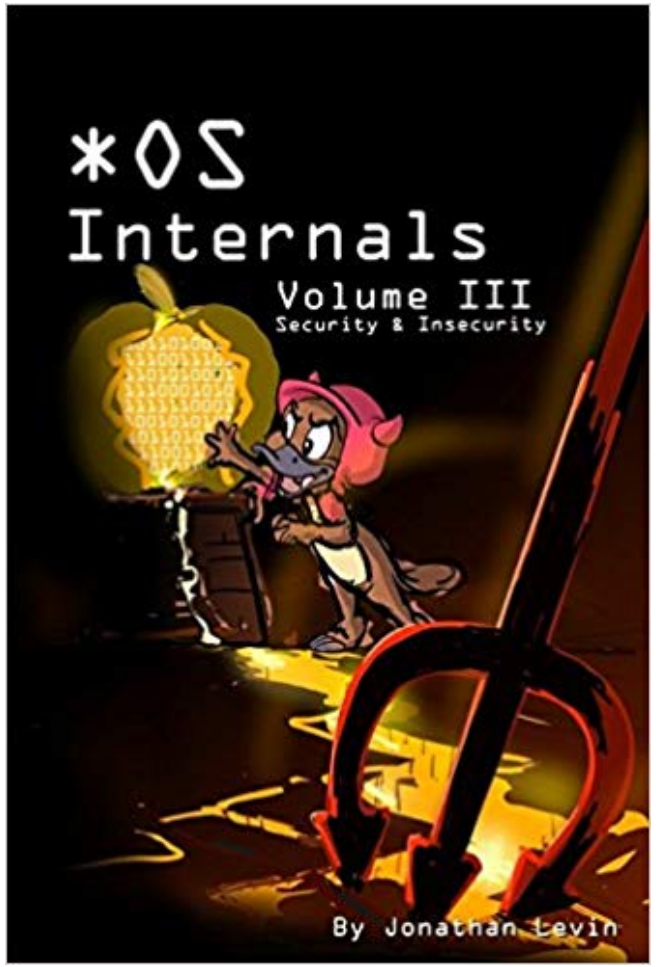
Need a macOS

```
applesign -i <identity> -m <provision_profile> -o <signed.ipa> <original.ipa>
```

```
Unzipping /Users/davydouhine/Documents/iOS/ipa/dvia.ipa
Payload found
Main IPA executable is not encrypted
Embedding new mobileprovision
{"application-identifier":"T9T8LG2CVR.*","keychain-access-groups":["T9T8LG2CVR.*"],"get-task-allow":false,"com.apple.developer.team-identifier":"T9T8LG2CVR"}
Updated binary entitlements/Users/davydouhine/Documents/iOS/ipa/dvia.ipa.ae9b005e-61f9-4a59-84ce-67f61334c2c0/Payload/DamnVulnerableIOSApp.app/DamnVulnerableIOSApp.entitlements
Signing libraries and frameworks
Executable found at /Users/davydouhine/Documents/iOS/ipa/dvia.ipa.ae9b005e-61f9-4a59-84ce-67f61334c2c0/Payload/DamnVulnerableIOSApp.app/DamnVulnerableIOSApp
Resolving signing order using layered list
Signed /Users/davydouhine/Documents/iOS/ipa/dvia.ipa.ae9b005e-61f9-4a59-84ce-67f61334c2c0/Payload/DamnVulnerableIOSApp.app/DamnVulnerableIOSApp
Verifying /Users/davydouhine/Documents/iOS/ipa/dvia.ipa.ae9b005e-61f9-4a59-84ce-67f61334c2c0/Payload/DamnVulnerableIOSApp.app/DamnVulnerableIOSApp
Zipifying into /Users/davydouhine/Documents/iOS/ipa/dvia_signed.ipa ...
Cleaning up /Users/davydouhine/Documents/iOS/ipa/dvia.ipa.ae9b005e-61f9-4a59-84ce-67f61334c2c0
Target is now signed: dvia_signed.ipa
```

To go further 📚





Get the **materials** 

Inside your VM (Mobexler):

```
git clone https://github.com/randorisec/BSidesDublin2021-Workshops
```

```
Headbook-v1.0.ipa
Mobile_Hacking_iOS_cheatsheet_v0.1.pdf
dvia.ipa
frida.js
iGoat.ipa
iOS_cheatsheet.txt
```

Install/update tools on Mobexler

```
sudo apt-get install libplist-utils
wget
https://github.com/corellium/usbfluxd/releases/do
wnload/v1.0/usbfluxd-x86_64-libc6-
libdbus13.tar.gz
(…)
```

Check `iOS_cheatsheets.txt` for commands

Corellium 😎 first steps

Inside your VM (Mobexler):

1. Log in (https://xxx.enterprise.corellium.com/)
2. Download OpenVPN config file (Connect / OVPN FILE)
3. Launch the VPN

```
sudo openvpn ovpn-conf-file
```

4. Connect on the device using SSH (root/alpine) – <IP> is at the top right of the screen

```
ssh root@<IP>
```

Corellium 😎 usbfluxd

Inside your VM (Mobexler):

5. Get **usbfluxd** (https://github.com/corellium/usbfluxd)


```
wget https://github.com/corellium/usbfluxd/releases/download/v1.0/usbfluxd-x86_64-libc6-libdbus13.tar.gz
```

6. Launch **usbfluxd**

```
sudo usbfluxd -r <IP> -v -f -n
```

7. Launch **usbfluxctl** to get the UDID of the device

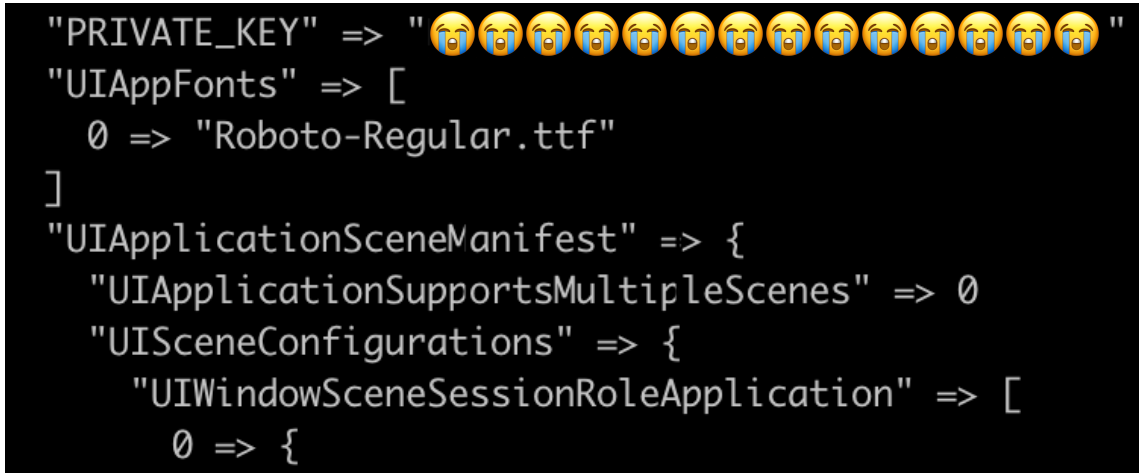
```
./usbfluxctl list
```

```
Mobexler@Mobexler ~/usbfluxd-x86_64-libc6-libdbus13 ./usbfluxctl list
1: usbmuxd\06410-11-1-2.local:5000 (1)
        322a1b7ca439c99c564b954d4a5d841fe0112c09
```

Static analysis

Example of requirement: **MSTG-CRYPTO-1**

The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption.



OWASP M5
Insufficient
Cryptography

Static analysis

Needs:

- full source code !
- tools (Checkmarx , Sonar, Clang Static Analyzer, …) !

But

- a simple “grep” can help
- even **without the full source code** a lot can be done by **reverse engineering** the static files (binary, etc.)

OWASP M2 Insecure Data Storage

OWASP M3 Insecure Communication

OWASP M5 Insufficient Cryptography

OWASP M7 Client code quality

OWASP M10 Extraneous Functionality

Static analysis

To detect:

- Caching of resources or keystrokes
- Cleartext credentials storage or in unprotected files
- Back-end servers
- Bad pasteboard management
- SQL injection
- Code injection
- Verbose logging
- Vulnerable C function calls like strcat, strcpy, etc.
- Custom URL schemes aka deeplinks (e.g: cydia://)

OWASP M2 Insecure Data Storage

OWASP M3 Insecure Communication

OWASP M5 Insufficient Cryptography

OWASP M7 Client code quality

OWASP M10 Extraneous Functionality

Lab1: App metadata

OWASP M2 Insecure data storage

Goal: analyze Info.plist

Steps:

1. Unzip `Headbook.ipa` on your VM (you don't need to install it on your Corellium device)
2. Analyze Info.plist
   ```
   plistutil -i Info.plist
   ```
3. Submit the flag to https://ctf.ivrodriguez.com

Lab2: Custom URL Scheme (Deep link)

OWASP M7 Client code quality

Goal: find and use DVIA deep link

Testing Custom URL Schemes (MSTG-PLATFORM-3)

Overview

Custom URL schemes allow apps to communicate via a custom protocol. An app must declare support for the schemes and handle incoming URLs that use those schemes.

Apple warns about the improper use of custom URL schemes in the Apple Developer Documentation:

> URL schemes offer a potential attack vector into your app, so make sure to validate all URL parameters and discard any malformed URLs. In addition, limit the available actions to those that do not risk the user's data. For example, do not allow other apps to directly delete content or access sensitive information about the user. When testing your URL-handling code, make sure your test cases include improperly formatted URLs.

Source: https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06h-Testing-Platform-Interaction.md

Lab2: Custom URL Scheme (Deep link)

OWASP M7
Client code
quality

Goal: find DVIA deep link

Steps:

1. Unzip **`dvia.ipa`** on your VM (you don't need to install it on your Corellium device)
2. Analyze **`Info.plist`**

```
<key>CFBundleURLTypes</key>
<array>
        <dict>
                <key>CFBundleTypeRole</key>
                <string>None</string>
                <key>CFBundleURLSchemes</key>
                <array>
                        <string>dvia</string>
                </array>
        </dict>
</array>
```

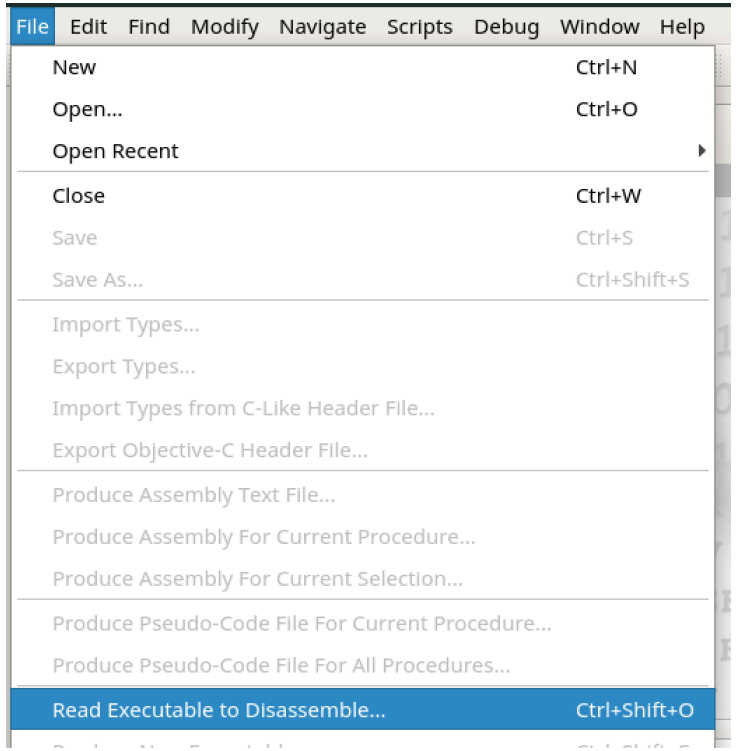
Introducing Hopper

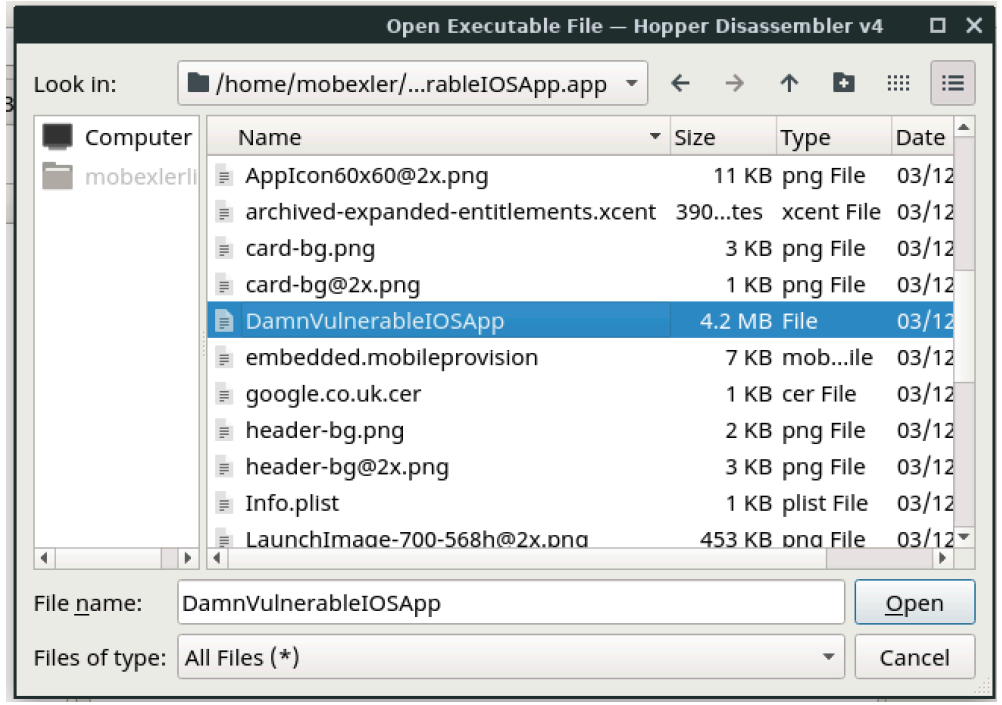
Goal: find DVIA deep link

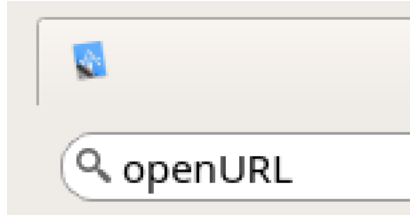
OWASP M7
Client code
quality

Steps:

3. Analyze **DVIA** binary (`DamnVulnerableIOSApp`) with Hopper







RANDORISEC

Lab2: Custom URL Scheme (Deep link)

OWASP M7
Client code quality

Goal: find DVIA deep link

```
/* @class AppDelegate */
-(char)application:(void *)arg2 openURL:(void *)arg3 sourceApplication:(void *)arg4 annotation:(void *)arg5 {
    r7 = (sp - 0x14) + 0xc;
    sp = sp - 0x38;
    r6 = self;
    r11 = [arg3 retain];
    r7 = r7;
    r5 = [[r11 absoluteString] retain];
    if (r5 != 0x0) {
            [sp + 0x10 rangeOfString:r5, @"/call_number/"];
            if (var_20 == (0x80000000 ^ 0xffffffff)) {
                    r4 = 0x0;
            }
            else {
                    r10 = [[r6 getParameters:r11] retain];
                    r0 = [r10 objectForKey:@"phone"];
                    r7 = r7;
                    r0 = [r0 retain];
                    [r0 release];
                    if (r0 != 0x0) {
                            var_24 = [UIAlertView alloc];
                            r8 = [[r10 objectForKey:@"phone"] retain];
                            r4 = [[NSString stringWithFormat:@"Calling %@ without validation. Ring Ring !"] retain];
                            r6 = [var_24 initWithTitle:@"Success" message:r4 delegate:0x0 cancelButtonTitle:@"OK" otherButtonTitles:0x0];
                            [r6 show];
                            [r6 release];
                            [r4 release];
                            [r8 release];
                    }
                    [r10 release];
                    r4 = 0x1;
            }
    }
```

Lab2: Custom URL Scheme (Deep link)

OWASP M7
Client code quality

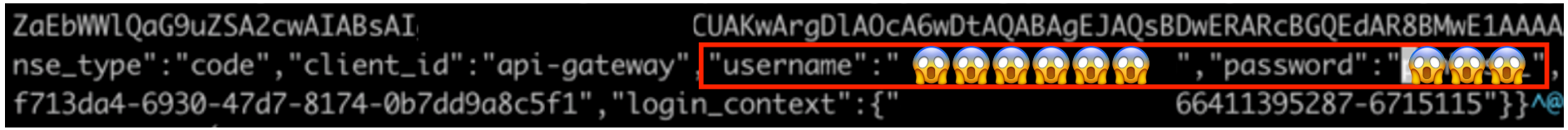
Goal: find DVIA deep link

```
/* @class AppDelegate */
-(char)application:(void *)arg2 openURL:(void *)arg3 sourceApplication:(void *)arg4 annotation:(void *)arg5 {
    r7 = (sp - 0x14) + 0xc;
    sp = sp - 0x38;
    r6 = self;
    r11 = [arg3 retain];
    r7 = r7;
    r5 = [[r11 absoluteString] retain];
    if (r5 != 0x0) {
            [sp + 0x10 rangeOfString:r5, @"/call_number/"];
            if (var_20 == (0x80000000 ^ 0xffffffff)) {
                    r4 = 0x0;
            }
            else {
                    r10 = [[r6 getParameters:r11] retain];
                    r0 = [r10 objectForKey:@"phone"];
                    r7 = r7;
                    r0 = [r0 retain];
                    [r0 release];
                    if (r0 != 0x0) {
                            var_24 = [UIAlertView alloc];
                            r8 = [[r10 objectForKey:@"phone"] retain];
                            r4 = [[NSString stringWithFormat:@"Calling %@ without validation. Ring Ring !"] retain];
                            r6 = [var_24 initWithTitle:@"Success" message:r4 delegate:0x0 cancelButtonTitle:@"OK" otherButtonTitles:0x0];
                            [r6 show];
                            [r6 release];
                            [r4 release];
                            [r8 release];
                    }
                    [r10 release];
                    r4 = 0x1;
            }
    }
```

Data security

Example of requirement: **MSTG-STORAGE-1**

System credential storage facilities are used appropriately to store sensitive data, such as PII, user credentials or cryptographic keys.



OWASP M2 Insecure data storage

Lab3: Cleartext Property List Files (Plist)

OWASP M2 Insecure data storage

Goal: find credentials stored in cleartext

Steps:

1. Open **iGoat**
2. Go to “Data Protection (Rest) / **Plist Storage** ”
3. Click “Start”
4. Enter credentials
5. Click “Verify”
6. The credentials are stored in `Documents/Credentials.plist`

Lab4: fsmon-ios

OWASP M2 Insecure data storage

Goal: find files used by an app

Steps:

1. Open **iGoat**
2. Go to “Data Protection (Rest) / **Cookie Storage** ”
3. Click “Start”
4. Enter info
5. Click “Verify”
6. Where is stored the answer ?

Use `fsmon-ios` to find it:

```
/var/root/fsmon-ios -P iGoat
```

Syslog

```
Jan  6 17:35:45 iPad-de-davy kernel[0] <Notice>: xpcproxy[80834] Builtin profile: container (sandbox)
Jan  6 17:35:45 iPad-de-davy kernel[0] <Notice>: xpcproxy[80834] Container: /private/var/mobile/Containers/Data/Application/237ED97E-6F71-4650-B215-55C8E3831738
Jan  6 17:35:46 iPad-de-davy mytalkingtom[80834] <Notice>: MS:Notice: Injecting: com.outfit7.mytalkingtom [mytalkingtom] (1280.38)
Jan  6 17:35:46 iPad-de-davy mytalkingtom[80834] <Error>: MS:Error: unable to open() binary file
Jan  6 17:35:46 iPad-de-davy mytalkingtom[80834] <Notice>: MS:Notice: Loading: /Library/MobileSubstrate/DynamicLibraries/SSLKillSwitch2.dylib
Jan  6 17:35:46 iPad-de-davy mytalkingtom[80834] <Warning>: === SSL Kill Switch 2: Preference set to 1.
Jan  6 17:35:46 iPad-de-davy mytalkingtom[80834] <Warning>: === SSL Kill Switch 2: Subtrate hook enabled.
Jan  6 17:35:46 iPad-de-davy mytalkingtom[80834] <Error>: MS:Error: binary does not support this cpu type
Jan  6 17:35:46 iPad-de-davy mytalkingtom[80834] <Error>: MS:Error: failure to check trustme.dylib
Jan  6 17:35:46 iPad-de-davy gamecontrollerd[80836] <Notice>: MS:Notice: Injecting: (null) [gamecontrollerd] (1280.38)
Jan  6 17:35:47 iPad-de-davy gamecontrollerd[80836] <Error>: MS:Error: unable to open() binary file
Jan  6 17:35:47 iPad-de-davy gamecontrollerd[80836] <Warning>: === SSL Kill Switch 2: Preference set to 1.
Jan  6 17:35:47 iPad-de-davy gamecontrollerd[80836] <Warning>: === SSL Kill Switch 2: Subtrate hook enabled.
Jan  6 17:35:47 iPad-de-davy gamecontrollerd[80836] <Error>: MS:Error: binary does not support this cpu type
Jan  6 17:35:47 iPad-de-davy gamecontrollerd[80836] <Error>: MS:Error: failure to check trustme.dylib
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Warning>: -> registered mono modules 0x10208f9f0
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Warning>: You've implemented -[<UIApplicationDelegate> application:didReceiveRemoteNotification:fetchCompletion
ote-notification" to the list of your supported UIBackgroundModes in your Info.plist.
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Notice>: <FIRAnalytics/INFO> Firebase Analytics v.3404000 started
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Notice>: <FIRAnalytics/INFO> Successfully created Firebase Analytics App Delegate Proxy automatically. To disab
egateProxyEnabled to NO in the Info.plist
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Warning>: AppsFlyer SDK version 4.6.3 started build (521)
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Warning>: AppInit: r=1
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Warning>: O7FunNetworkLib Version: 5.10.0
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Warning>: *** -[NSKeyedUnarchiver initForReadingWithData:]: data is NULL
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Warning>: -canOpenURL: failed for URL: "fb://" - error: "(null)"
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Notice>: <FIRAnalytics/INFO> Firebase Analytics enabled
Jan  6 17:35:47 iPad-de-davy mytalkingtom[80834] <Notice>: <FIRAnalytics/INFO> Firebase Analytics enabled
```

Lab5: Syslog

OWASP M2 Insecure data storage

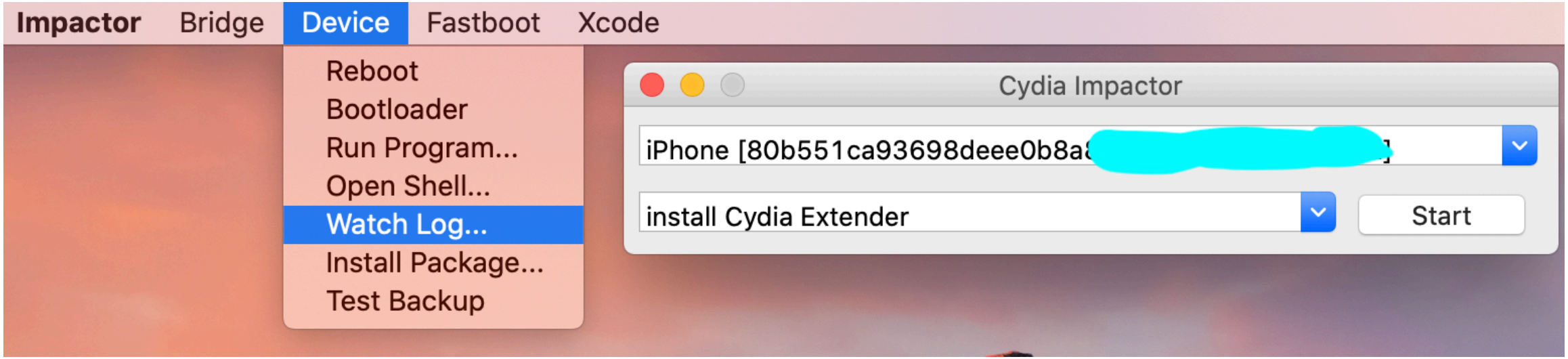
Goal: get sensitive info stored in the syslog by iGoat

Steps:

1. Open **iGoat**
2. Go to “Key Management / **Random Key Generation**”
3. Find the encryption key
4. Get the syslog with `Impactor`

Introducing Impactor

OWASP M2
Insecure data storage



Syslog

OWASP M2
Insecure data storage

To use Impactor using the command line you first need to get the device UDID (serial number)

Get device UDID:
Use **`usbfluxctl`** (or launch **`Impactor`** without parameters):

Mobexler: **`/home/mobexler/iOSZone/Impactor64_0.9.52/Impactor`**

```
Mobexler@Mobexler ~ /home/mobexler/iOSZone/Impactor64_0.9.52/Impactor
I:0x7fb6a0001908:6d0f                                          cff7
```

Mac: **`/Applications/Impactor.app/Contents/MacOS/Impactor`**

```
davys-macbook-pro$/Applications/Impactor.app/Contents/MacOS/Impactor
I:0x100f08ad8:6d0f                                          cff7
```

Lab5: Syslog

Goal: get sensitive info stored in the syslog by iGoat

OWASP M2
Insecure data storage

Launch Impactor:

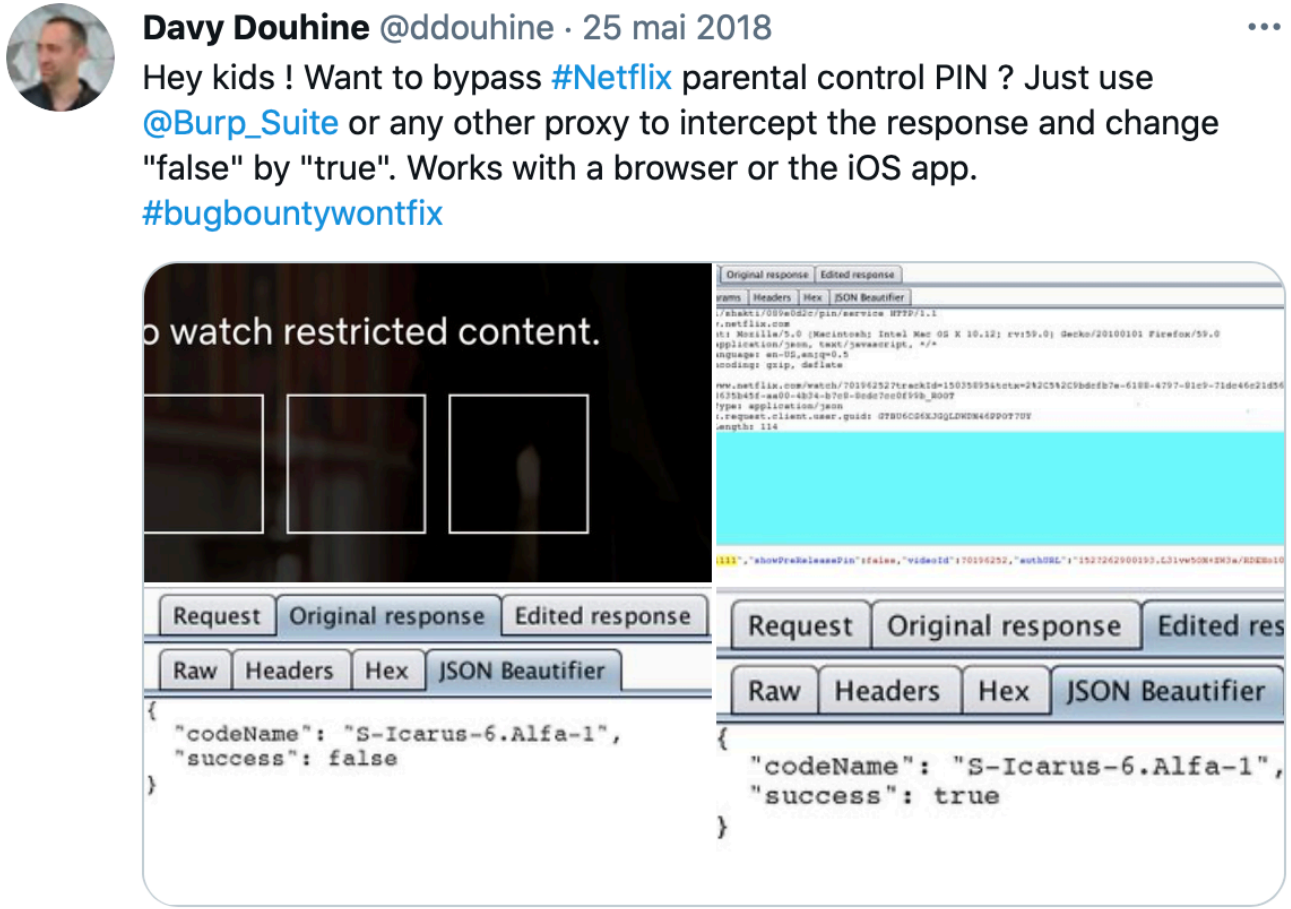
```
./Impactor idevicesyslog -u <UDID>
```

```
Mobexler@Mobexler ~/iOSZone/Impactor64_0.9.52 ./Impactor idevicesyslog -u 322a1b7ca439c99c564b954d4a5d841fe0112c09 ✔ 123 14:36:29

[connected]
Mar 23 02:06:45 iPhone runningboardd(RunningBoard)[26] <Notice>: Invalidating assertion 26-289-118 (target:application<com.swaroop.iGoat>) from origi
nator 289
Mar 23 02:06:45 iPhone runningboardd(RunningBoard)[26] <Notice>: Calculated state for application<com.swaroop.iGoat>: running-active (role: UserInter
activeNonFocal)
Mar 23 02:06:45 iPhone powerd[35] <Notice>: Process runningboardd.26 Released SystemIsActive "application<com.swaroop.iGoat>26-289-118:FBApplicationP
rocess" age:00:00:01  id:51539640687 [System: SysAct]
Mar 23 02:06:45 iPhone runningboardd(RunningBoard)[26] <Notice>: Released power assertion with ID 33135
```

Execution analysis

Example of requirement: **MSTG-ARCH-2**

Security controls are never enforced only on the client side



OWASP M8 code tampering

Lab6: PIN bypass with Frida

OWASP M8 code tampering

Goal: bypass a security check using Frida

Steps:

1. Open **iGoat**
2. Go to “Runtime Analysis” / “**Runtime Brute force attack**” and “Start”
3. Find the class and the method used to check the PIN code
4. Instrument iGoat using `frida` to bypass the PIN code check

Hooking

cycript

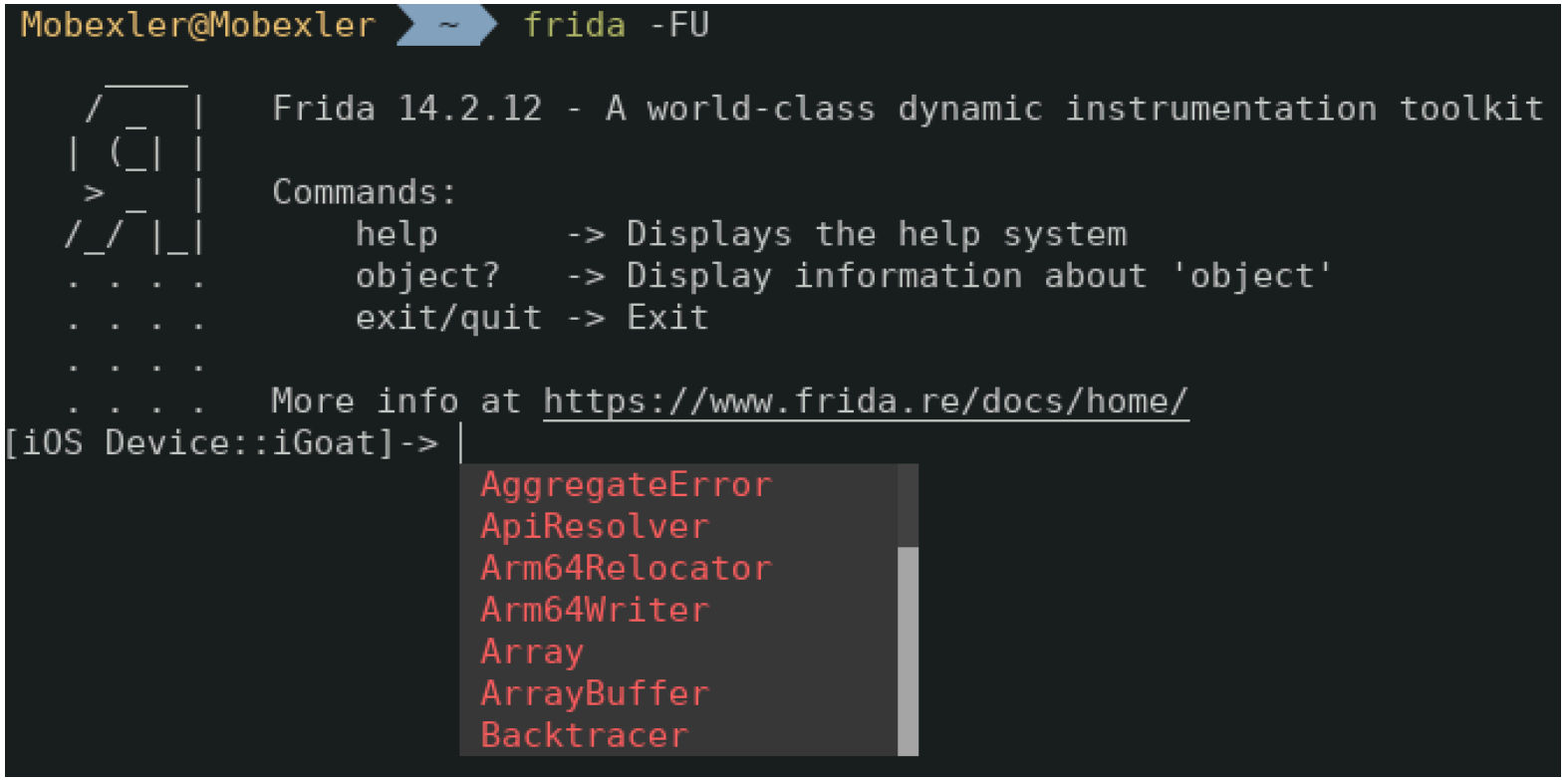
Cycript allows developers to explore and modify running applications on either iOS or Mac OS X using a hybrid of Objective-C++ and JavaScript syntax through an interactive console that features syntax highlighting and tab completion.
(It also runs standalone on Android and Linux and provides access to Java, but without injection.)

FRIDA

OVERVIEW DOCS NEWS CODE CONTACT

Inject JavaScript to explore native apps on Windows, macOS, Linux, iOS, Android, and QNX.

Introducing Frida



Frida-trace: native API tracing & hooking

```
bibi:~ root# frida-trace -U -p 82924 -i "*URL*"
Instrumenting functions...
CaptiveCopyWiFiLandingPageURL: Auto-generated handler at "/private/var/root/__handlers__/CaptiveNetwork/CaptiveCopyWiFiLandingPageURL.js"
CFURLCreateBookmarkData: Auto-generated handler at "/private/var/root/__handlers__/Foundation/CFURLCreateBookmarkData.js"
_CFURLCreateDisplayPathComponentsArray: Auto-generated handler at "/private/var/root/__handlers__/Foundation/_CFURLCreateDisplayPathComponentsArray.js"
_CFBundleCreateWithExecutableURLIfMightBeBundle: Auto-generated handler at "/private/var/root/__handlers__/Foundation/_CFBundleCreateWithExecutableURL_3b8f9203.js"
_CFURLCreateByResolvingAliasFile: Auto-generated handler at "/private/var/root/__handlers__/Foundation/_CFURLCreateByResolvingAliasFile.js"
_CFURLCreateFromComponents: Auto-generated handler at "/private/var/root/__handlers__/Foundation/_CFURLCreateFromComponents.js"
CFBundleCopyExecutableURL: Auto-generated handler at "/private/var/root/__handlers__/Foundation/CFBundleCopyExecutableURL.js"
_CFURLIsItemPromiseAtURL: Auto-generated handler at "/private/var/root/__handlers__/Foundation/_CFURLIsItemPromiseAtURL.js"
CFCopyHomeDirectoryURL: Auto-generated handler at "/private/var/root/__handlers__/Foundation/CFCopyHomeDirectoryURL.js"
CFURLCreateFilePathURL: Auto-generated handler at "/private/var/root/__handlers__/Foundation/CFURLCreateFilePathURL.js"
```

(...)

```
CMByteStreamCreateForFileURL: Auto-generated handler at "/private/var/root/__handlers__/CoreMedia/CMByteStreamCreateForFileURL.js"
FigNote_FlushRunningLogAndCopyURLContainingLogs: Auto-generated handler at "/private/var/root/__handlers__/CoreMedia/FigNote_FlushRunningLogAndCopyUR_0e23347a.js"
Started tracing 1165 functions. Press Ctrl+C to stop.
           /* TID 0xc07 */
 19170 ms  CFURLCopyScheme()
 19171 ms  CFURLCopyAbsoluteURL()
 19171 ms  CFURLCopyFileSystemPath()
 19182 ms  CFURLCopyScheme()
 19184 ms  CFURLCopyAbsoluteURL()
 19184 ms  CFURLCopyFileSystemPath()
 19214 ms  CFBundleCopyResourceURL()
 19215 ms     | CFURLCreateWithFileSystemPath()
 19215 ms     |    | CFURLCreateWithFileSystemPathRelativeToBase()
```

Frida: CodeShare

iOS DataProtection

7 | 4K

Uploaded by: @ay-kay

List iOS file data protection classes (NSFileProtectionKey) of an app

PROJECT PAGE

aesinfo

7 | 5K

Uploaded by: @dzonerzy

Show useful info about AES encryption/decryption at application runtime

PROJECT PAGE

frida-multiple-unpinning

5 | 4K

Uploaded by: @akabe1

Another Android ssl certificate pinning bypass script for various methods (https://gist.github.com/akabe1/5632cbc1cd49f0237cbd0a93bc8e4452)

PROJECT PAGE

ObjC method observer

4 | 8K

Uploaded by: @mrmacete

Observe all method calls to a specific class (e.g. observeClass('LicenseManager')) , or dynamically resolve methods to observe using ApiResolver (e.g. observeSomething('*[* *Password:*]')). The script tries to do its best to resolve and display input parameters and return value. Each call log comes with its stacktrace.

PROJECT PAGE

Frida: CodeShare: ObjC-method-observer

```
Mobexler@Mobexler ~ frida -FU -l frida.js

     ____
    / _  |   Frida 14.2.12 - 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/
[iOS Device::iGoat]-> observeSomething('*[* *fileExists*]');
Observing  -[NSURL cls_fileExists]
Observing  -[PFUbiquityLocation fileExistsAtLocation]
Observing  -[PFUbiquityLocation fileExistsAtLocationWithLocalPeerID:error:]
Observing  +[CIRedEyeRepair2 fileExistsAtPath:]
Observing  -[NSFileManager fileExistsAtPath:]
Observing  -[NSFileManager fileExistsAtPath:isDirectory:]
Observing  -[NSFileManager _web_fileExistsAtPath_nowarn:isDirectory:traverseLink:]
Observing  -[MBFileManager fileExistsAtPath:]
[iOS Device::iGoat]->
```

Frida: CodeShare: ObjC-method-observer

```
[Remote::iGoat]-> (0x28387c060)  -[NSFileManager fileExistsAtPath:]
fileExistsAtPath: /var/mobile/Containers/Data/Application/87057B91-EC0F-4EDB-917C-B237619AFBF1/Documents/Credentials.plist (NSPathStore2)
0x102aaeff0 iGoat!0xd6ff0
0x1a41f1448 UIKitCore!-[UIViewController _sendViewDidLoadWithAppearanceProxyObjectTaggingEnabled]
0x1a41f5f58 UIKitCore!-[UIViewController loadViewIfRequired]
0x1a41f6360 UIKitCore!-[UIViewController view]
0x102a6a074 iGoat!0x92074
0x102a69dbc iGoat!0x91dbc
0x102a5eb6c iGoat!0x86b6c
0x1a4965e3c UIKitCore!-[UIStoryboardSegueTemplate _performWithDestinationViewController:sender:]
0x1a4965d4c UIKitCore!-[UIStoryboardSegueTemplate _perform:]
0x1a4966018 UIKitCore!-[UIStoryboardSegueTemplate perform:]
0x1a48119ac UIKitCore!-[UIApplication sendAction:to:from:forEvent:]
0x1a3f0e318 UIKitCore!-[UIBarButtonItem(UIInternal) _sendAction:withEvent:]
0x1a48119ac UIKitCore!-[UIApplication sendAction:to:from:forEvent:]
0x1a4247fbc UIKitCore!-[UIControl sendAction:to:forEvent:]
0x1a4248320 UIKitCore!-[UIControl _sendActionsForEvents:withEvent:]
0x1a4248450 UIKitCore!-[UIControl _sendActionsForEvents:withEvent:]
RET: 0x1
```

Frida basics

OWASP M8 code tampering

Goal: discover the tool

List running processes on a **device connected on the network**:

```
frida-ps -H <IP>
```

List running processes on a **device connected with USB**:

```
frida-ps -U
```

Inject Frida in appName's process:

```
frida -U <appName>
```

Inject Frida in appName's process and load `frida.js` file:

```
frida -U <appName> -l frida.js
```

Inject Frida in the foreground app process and load `frida.js` file:

```
frida -FU -l frida.js
```

Lab6: PIN bypass with Frida

OWASP M8 code tampering

Steps (detailed):

1. Go to “Runtime Analysis” / “**Runtime Brute force attack**” and “Start”

2. Identify class/method used for the PIN check with **`frida`** and **`ObjC-Method-Observer:`**

```
frida -FU -l frida.js observeSomething('*[* *Pin:*]');
```

3. Confirm the class/method Use **frida-trace** to add “backtrace”:

```
frida-trace -FU -m "-[class method]"
```

4. Modify the method result with the JS handler created by **frida-trace:**

Add **`retval.replace(1);`** to the **`onLeave`** function)

Introducing Objection

```
davy@kali:~$ objection -g "DVIA" -N -h 192.168.1.25 explore
Using networked device @`192.168.1.25:27042`
Agent injected and responds ok!

     _   _         _   _
 ___| |_|_|___ ___| |_|_|___ ___
| . | . | | -_|  _|  _| | . |   |
|___|___| |___|___|_| |_|___|_|_|
      |___|(object)inject(ion) v1.9.2

     Runtime Mobile Exploration
        by: @leonjza from @sensepost

[tab] for command suggestions
com.highaltitudehacks.dvia on (iPhone: 13.6.1) [net] #
```

Introducing Objection

```
[tab] for command suggestions
com.highaltitudehacks.dvia on (iPhone: 13.6.1) [net] # env

Name              Path
----------------  ------------------------------------------------------------------------------------------
BundlePath        /private/var/containers/Bundle/Application/44EE06A7-3510-40B9-B17E-61741D2D7627/DamnVulnerableIOSApp.app
CachesDirectory   /var/mobile/Containers/Data/Application/C737E3BA-961A-4752-B322-05526770743A/Library/Caches
DocumentDirectory /var/mobile/Containers/Data/Application/C737E3BA-961A-4752-B322-05526770743A/Documents
LibraryDirectory  /var/mobile/Containers/Data/Application/C737E3BA-961A-4752-B322-05526770743A/Library
com.highaltitudehacks.dvia on (iPhone: 13.6.1) [net] # ios plist
                                                      pasteboard   Work with the iOS pasteboard
                                                      plist        Work with iOS Plists
                                                      sslpinning   Work with iOS SSL pinning
```

Introducing Objection

To launch objection (on a device connected with USB):

```
objection -g "iGoat" explore
```

To launch objection (on a device connected on the network):

```
objection -g "iGoat" -N -h <IP> explore
```

To display Info.plist:

```
ios plist cat Info.plist
```

Lab7: PIN bypass with Objection

OWASP M8 code tampering

Goal: bypass a security check using Objection

Steps:

1. Open **iGoat**
2. Go to “Runtime Analysis” / “**Runtime Brute force attack**” and “Start”
3. Find the class and the method used to check the PIN code
4. Instrument iGoat using `Objection` to bypass the PIN code check

Lab7: PIN bypass with Frida

OWASP M8 code tampering

Steps (detailed):

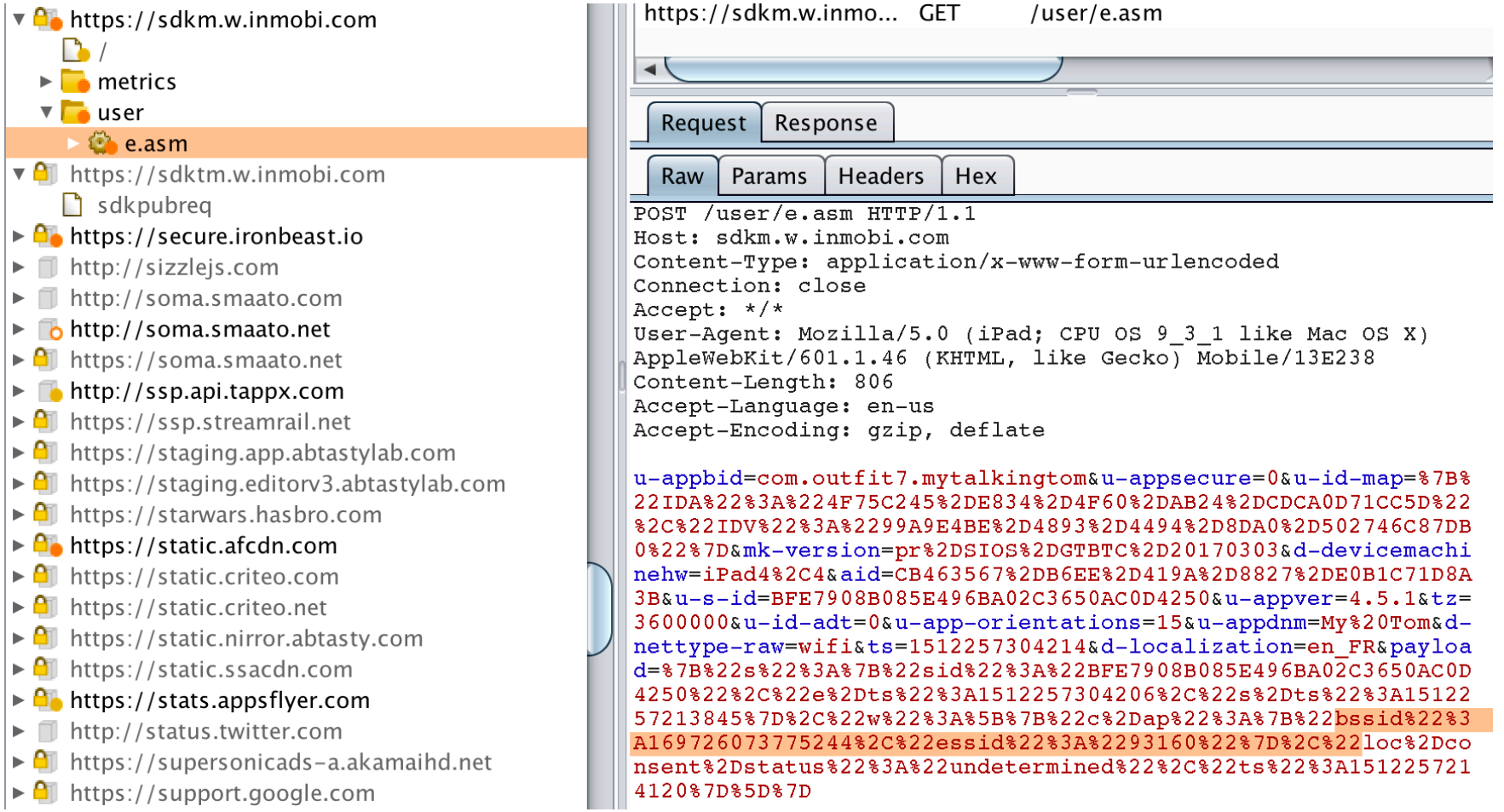
Use **objection** and:

- `ios hooking list class_methods` to list class/method used for the login
- `ios hooking watch method` to add “backtrace” and confirm the class/method
- `ios hooking set return_value` to modify the method result

Transport security

Example of requirement: **MSTG-STORAGE-4**

No *sensitive data is shared third parties*



OWASP M3
Insecure Communication

Lab8: network capture with **rvictl**

Only for macOS - sorry

OWASP M3
Insecure Communication

Goal: analyze network communications

Steps:

1. Connect a device using USB and click on “Trust”
2. Get device UDID (serial number)
3. Create redirection:

```
rvictl -s <UDID>
```

4. Capture network traffic with tcpdump or wireshark

```
tcpdump -Xni rvi0
```

5. Delete redirection:

```
rvictl -x <UDID>
```

Lab8: network capture with **rvi_capture**

OWASP M3
Insecure Communication

Goal: analyze network communications

For Linux :)
But doesn't work with iOS >= 13

Steps:

1. Connect a device using USB and click on "Trust"
2. Get device UDID (serial number)
3. Launch capture:

```
./rvi_capture.py --udid <UDID> iPhone.pcap
```

4. Stop capture and analyze network traffic with tcpdump or wireshark

```
tcpdump -Xnr iPhone.pcap
```

Lab8: network capture with **named pipes**

OWASP M3
Insecure
Communication

Goal: analyze network communications

For Linux :)

Steps:

1. Connect a Corellium device using VPN, then redirect USB using SSH and redirect the network traffic using a named pipe.

As root type:

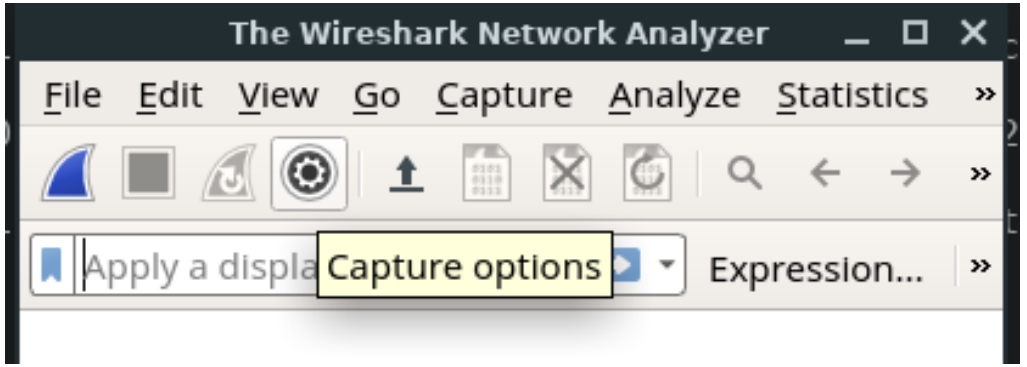
```
usbfluxd -r <IP> -v -f -n

mkfifo /tmp/remote

ssh root@<IP> "tcpdump -s0 -U -n -w - -i en0 not port
22" > /tmp/remote
```

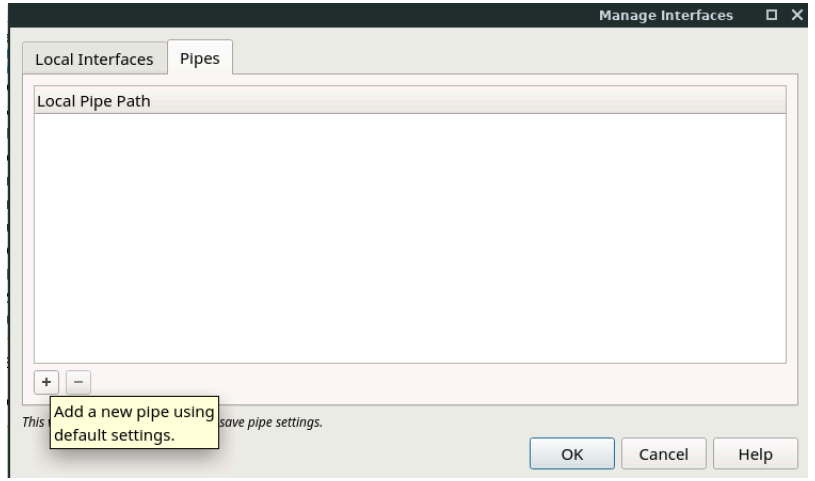
Lab8: network capture with **named pipes**

OWASP M3
Insecure
Communication

Steps:

2. Launch wireshark, click on “Manage Interfaces”, create a named pipe (eg: /tmp/remote)



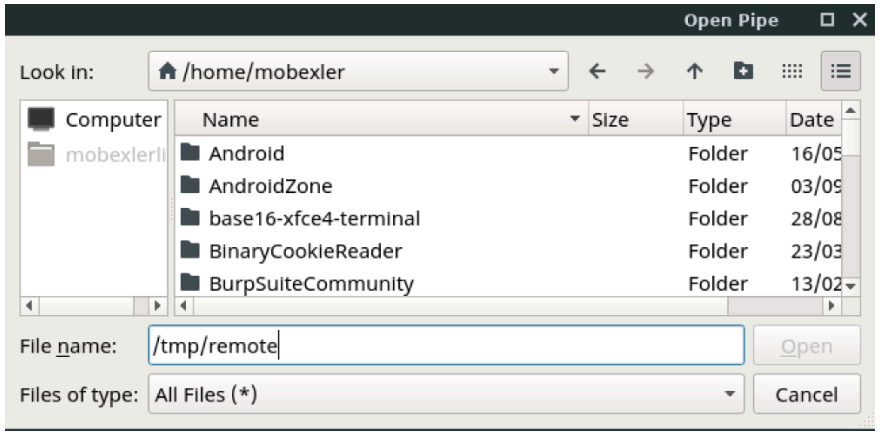


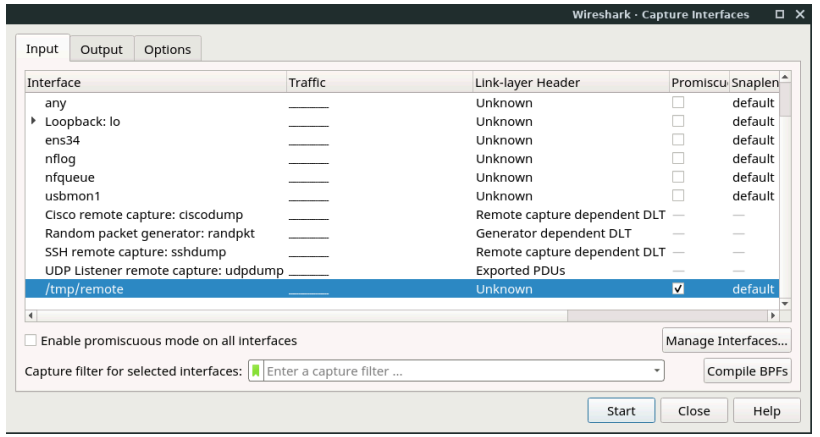
Lab8: network capture with **named pipes**

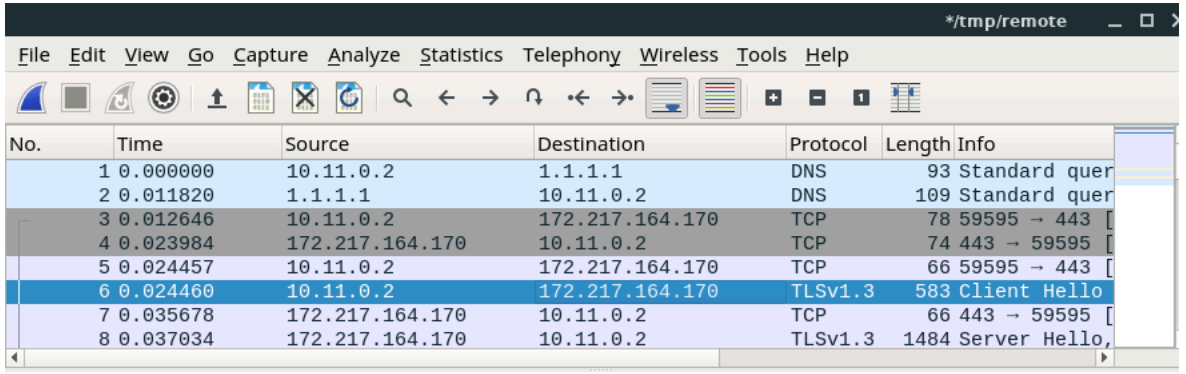
OWASP M3
Insecure
Communication

Steps:

3. Set the input of Wireshark on the named pipe and click on Start !







Questions