NATIVE APPS

OVERVIEW

- Access and Theft Protection
- Certificates, Licenses and Profiles on iOS
- Attacks on iOS
- iOS development in a nutshell

ACCESS AND THEFT PROTECTION

ACCESS PROTECTION

Todays phones typically offer protection against unauthorized access by providing:

- an entry field for a pin code
- a fingerprint scanner
- a security pattern
- an iris scanner (only prototypes)

THEFT PROTECTION

- iOS: Activation Look or sometimes called iCloudLock.
- Android: Factory Reset Protection

CERTIFICATES, LICENSES AND PROFILES ON IOS

SANDBOXING I

- On iOS, only signed apps are executed. Apple has full power and can reject an app.
- An executed app can only access their own files and settings.
- Most global device settings cannot be changed by an app (i.e. enable/disable Wifi/Bluetooth,...).

SANDBOXING II

- An app signature is typically checked online before the app is started. This enables Apple to revoke already installed apps.

THINK ABOUT / DISCUSS

What are the main advantages and disadvantages of this approach?

JAILBREAKING I

There has been a large effort in finding ways to remove this signature checking so that also non signed apps can be executed. The term Jailbreaking is used for this task. Large sums are paid for an undisclosed jailbreak.

CERTIFICATES

When developing for iOS, different entities are identified:

- All developers have a certificate containing the private key of at least one developer. You can form teams where multiple developers share the same certificate.
- Each application you develop has a unique App ID.
- Each device you use for running your application has a unique identifier (check Window->Devices in Xcode).

↓ more ↓

iPhone 6s Plus

Kapazität: 55,36 GB
Telefonnummer: +41 79 86
UDID: C158F4EADAD13E278

iOS 9.1
Ihre iPhone-Software ist auf dem neuesten Stand. iTunes sucht wieder automatisch nach einem Update am 27.10.15.

Nach Update suchen	iPhone wiederherstellen ...

Backups

Automatisch sichern

🔘 **iCloud**
Die wichtigsten Daten auf Ihrem iPhone in iCloud sichern.

⚪ **Dieser Computer**
Ein vollständiges Backup Ihres iPhone wird auf diesem Computer gespeichert.

☐ **iPhone-Backup verschlüsseln**
Dadurch können Passwörter für Accounts und die Daten von Heath und HomeKit gesichert werden.

Passwort ändern ...

Backup manuell erstellen und wiederherstellen

Sichern Sie Ihr iPhone manuell auf diesen Computer oder stellen Sie ein auf diesem Computer gespeichertes Backup wieder her.

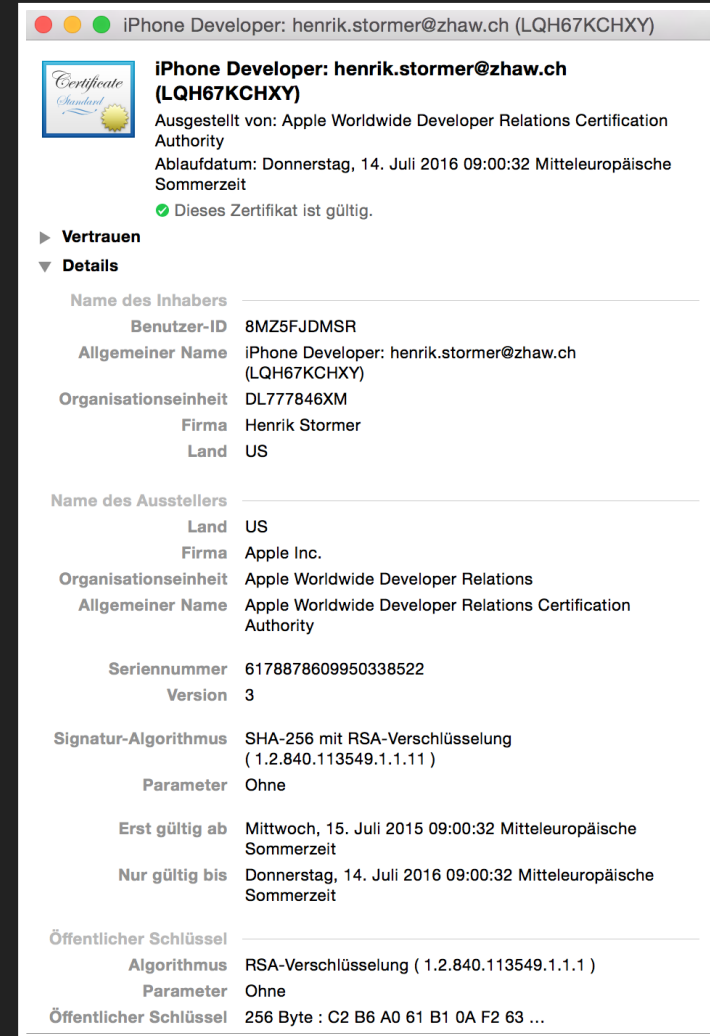| Jetzt sichern |
|---|

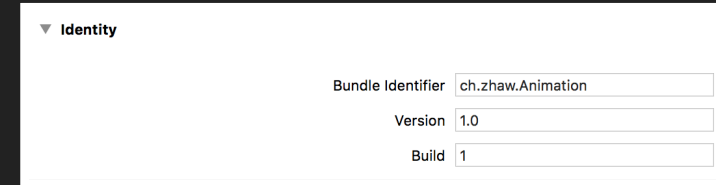| Backup wiederherstellen ... |
|---|

Letztes Backup:
Gestern 19:59 auf iCloud

DEVELOPER CERTIFICATE

- Created for each developer
- Contains the name, the e-mail and the public key
- Is usually valid for one year

APP ID

Identity

Bundle Identifier ch.zhaw.Animation
Version 1.0
Build 1

PROVISIONING PROFILE I

- Is defined in your project preferences as Bundle Identifier.

- Should follow a reverse URL scheme (com.mydomain.myappname) to make sure the App ID is not yet used.

- Will be checked as soon as a provisioning profile for this App ID is generated.

- Is generated in your project by Xcode (iOS development IDE) when you first run the app (development).

- For submission to the app store, you need a distribution profile.

- Contains references to the developer(s) and the App ID. Development profiles have references to the testing devices.

PROVISIONING PROFILE II

- Most developers will create two provisioning profiles. One for testing on a real device (development profile), the other for uploading to the app store (distribution profile).
- In Xcode, you can define two schemes (development and distribution) and attach the corresponding profile.
- As development profiles contain the device id, be careful to set this correctly to all testing devices.

PROVISIONING PROFILE CONTENT (XML PART) I

```xml
<?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>AppIDName</key>
    <string>Xcode iOS App ID ch zhaw Connect4</string>
    <!--This is usually your team identifier-->
    <key>ApplicationIdentifierPrefix</key>
    <array><string>DL777846XM</string></array>
    <!--By default, this is the date when you first ran the app-->
    <key>CreationDate</key><date>2015-07-15T13:37:27Z</date>
    <key>Platform</key><array><string>iOS</string></array>
```

PROVISIONING PROFILE CONTENT (XML PART) II

```
<!--The developer(s) certificate as base64 string -->
<key>DeveloperCertificates</key>
<array><data>MIIFpDCCBIygAw...</data></array>
<key>Entitlements</key>
<dict>
<!--By default, the whole team has access-->
    <key>keychain-access-groups</key>
    <array><string>DL777846XM.*</string></array>
    <key>get-task-allow</key><true/>
<!--Here is the App ID reference-->
    <key>application-identifier</key>
    <string>DL777846XM.ch.zhaw.Connect4</string>
    <key>com.apple.developer.team-identifier</key>
    <string>DL777846XM</string>
</dict>
```

PROVISIONING PROFILE CONTENT (XML PART) III

```
    <!--By default, 90 days after creation, see TimeToLive below-->
    <key>ExpirationDate</key><date>2015-10-13T13:37:27Z</date>
    <key>Name</key><string>iOSTeam Provisioning Profile: ch.zhaw.Connect4</string>
    <!--And here is the link to the device(s). The app will not run on other ones-->
    <key>ProvisionedDevices</key><array><string>9377641e3cc98e...</string></array>
    <key>TeamIdentifier</key><array><string>DL777846XM</string></array>
    <key>TeamName</key><string>Henrik Stormer</string>
    <key>TimeToLive</key><integer>90</integer>
    <key>UUID</key><string>28ee7b5e-96a7-4954-a0de-d90ce458031d</string>
    <key>Version</key><integer>1</integer>
</dict>
<!--A binary part is following-->
```

ENTERPRISE PROVISIONING PROFILE

- A special company profile for in-house applications.
- Your app can be distributed internally and circumvent the app store (sometimes called sideloading).
- If your app is generated with an enterprise profile it will run on all devices.

ENTERPRISE PROVISIONING PROFILE: CHANGES TO STANDARD PROFILE

```xml
    <!-- The key ProvisionsAllDevices is set, making the profile run on all devices-->
<key>ProvisionsAllDevices</key>
<true/>

    <!-- Time to live of an enterprise profile is usually 1 year-->
<key>TimeToLive</key>
<integer>365</integer>
```

ENTITLEMENTS

MOTIVATION FOR ENTITLEMENTS

To further control the usage of APIs, Apple has added so called entitlements. These are basically permissions granted to applications. Of course, the permissions are checked (an App cannot use APIs without permission).

▶ ⊗ **Maps**

▶ ⊶ **Keychain Sharing**

▶ ⧉ **Background Modes**

▶ ⧉ **Inter-App Audio**

▶ ⊟⊟ **App Groups**

▶ 🔒 **Data Protection**

▶ 🏠 **HomeKit**

▶ ♡ **HealthKit**

▶ 📶 **Wireless Accessory Configuration**

APP SUBMITTING TO THE APP STORE

- The app store is the primary method for deploying apps to consumers
- Each submitted app is reviewed
- You need to be a registered developer in order to be able to submit apps to the app store ($99 per year). You can then create a distribution profile and upload your app.
- For uploading, you need to create an IPA package, which is basically a ZIP file with a defined structure.
- The IPA package contains the distribution profile.

ATTACKS ON IOS

ATTACKS IN THE PAST

In the past, a number of attacks have been made to get malicious code on the users device:

1. Attacks on the sandbox by calling unneccessary or private APIs.
2. Attacks on the developers by changing Xcode.

SANDBOX ATTACKS (1)

You install an app, the app reads your contacts and sends them to a server.

Apple is using entitlements to prevent your app from calling certain APIs. Additionally, when accessing other APIs like contacts or localization, the user has to give his/her permission.

DEVELOPER ATTACKS (2)

A developer downloads a malicious Xcode from the internet, when building an app and submitting it to the app store, malicious code is contained in the app.

Developers should not download Xcode or other libraries from dubiuos sources.

IOS DEVELOPMENT IN A NUTSHELL

DEVELOPING NATIVE APPS

- SDK to develop for iOS
- IDE called Xcode (to create both iOS and OSX applications)
- Xcode is a set of development, performance testing and measuring tools
- Interface Builder is a visual design tool for the GUI
- iOS Simulator allows developers to test their apps on all current devices

↓ more ↓

FirstProject 〉 FirstProject 〉 ViewController.swift 〉

▼ FirstProject
  ▼ FirstProject
    AppDelegate.swift
    ViewController.swift
    Main.storyboard
    Assets.xcassets
    LaunchScreen.storyboard
    Info.plist
  ▶ FirstProjectTests
  ▶ FirstProjectUITests
  ▶ Products

```swift
//
//  ViewController.swift
//  FirstProject
//
//  Created by Henrik Stormer on 08.07.15.
//  Copyright © 2015 Henrik Stormer. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
```

```
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated
    }

}
```

INTERFACE BUILDER

- iOS8 is supporting five different screen sizes
- Concepts such as Auto-Layout and Adaptive Layout are used to aid the developer in supporting all screen sizes

PROGRAMMING LANGUAGE: SWIFT

- Announced 2014
- Influenced by other languages such as C#, Ruby, Haskell, Python and countless others
- Easier to learn than Objectice-C
- Inferred data types, data structure declarations, tuples, closures, optional semicolons, no pointers

DEV ENVIRONMENT

- Free developer account
    - Access to Xcode, sample code, videos, and documentation
    - Create and test iOS apps to run in the iOS Simulator
- Upgraded deloper account at a cost of $99 a year
    - Submit apps to the App Store
    - Access to betas of future versions of Xcode and iOS

ONLINE

developer.apple.com

RayWenderlich.com

ios.devtools.me

iosdevweekly.com

www.galloway.me.uk

merowing.info

ashfurrow.com