Android (in)Security

Having fun with Android

OWASP
The Open Web Application Security Project

Sarantis Makoudis
About Me

- BSc in Digital Systems, University of Piraeus, 2010
- MSc in Information Security, Royal Holloway, University of London, 2012
- Penetration Tester at 7Safe, part of PA Consulting, since September 2012
- Pretty much a geek, with a great love for IT, movies and boardgames.
Agenda

- Android Basics
- Data Interception by malicious Keyboard
- Malicious I/O Capture
- Authentication Bypass
- Malicious Code injection
- Phishing attacks
- Defences
Android Basics
Android Architecture
Android Debug Bridge

- ... Or adb for short.
- Part of the Android SDK.
- Client, Server, daemon
- Used during development
- The main tool used for Android debugging (and hacking!)
ADB Commands

- **Adb (dis)connect <IP>:** connects a device (or VM) to the host machine.
- **Adb devices:** List of all currently connected devices.
- **Adb shell:** Opens a shell on the host machine for the connected device.
- **Adb shell –c <command>:** Executes directly a shell command on the connected device.
- **Adb am start <>:** Start an activity of an already installed application.
- **Adb tcpip <port>:** Opens an adb daemon listener on the given port.
Demos

- Let’s pray to the demo gods that they will be nice with me today...

- **Android Emulator**: AndroVM, produced by Daniel Fages (@madCdAn) [androvm.org](http://androvm.org)

- **Android Version**: 4.1.1 Jelly Bean

- **Merchant Application**: produced by Matthew Seaward, really thankful that he borrowed it to me for this presentation.
Data Interception by Malicious Keyboard
- Google Play store or third party sites
- Anyone can upload their applications on the store
- Are you sure about their origin?
- Keyboards are one of those applications...
- Google shows a warning when you try to install it.
- But I want those cool emoticons! I am sure that nothing will be wrong!
- ....Or, is it?
Live Demo
Malicious I/O Capture
- A second way of capturing user input.
- Less visible than a third party keyboard.
- Manipulating the devices’ display drivers to send also the input to the attacker.
- Represents the position of a touch or swipe on an x, y base.
- **Down-side:** Physical access to the phone, the make and model of phone must be known to interpret the data.
The victim enters his credentials which are stolen by low-level IO reader and sent to the remote attacker's server.
Live Demo
Analysis of data on screen
As you have seen, our application contains a Login screen.

Android provides the developer with a variety of different tools

Being so open and friendly can also be the downside for Android application Developers.

Android allows us to bypass the authentication of the application in more than one ways.

Here we will see two different scenarios: A simple SQL injection or the use of `am` command to bypass the authentication (rooted phone needed)

More on Authentication bypass later on...
Live Demo
Malicious Code Injection
Android applications are Java at heart
Any tools that work for Java work for them plus many more....

Dex2jar
- Transforms .dex files to .jar files
- Jar files can be then decompiled with any Java Decompiler
- But...
- The code produced isn’t complete and as a result can not be recompiled

Smali & baksmali
- An assembler/dissassembler for the dex files
- Generates a disassembled code in smali format, which is close to Java
- We can inject code, reassemble it and install it normally!

APKManager
- A script/tool that utilises the smali/baksmali tools, signs and install apks plus much more....
Live Demo
Phishing Attacks
This time we will do the things different...

**Live Demo**
1. The Malicious Application has started monitoring processes in the system (the malicious functionality is invisible for the end-user).

2. The Merchant Application has started.

The legitimate application is running its 1st activity:
the login screen.
3. The Malicious Application detects that the legitimate application has started and pops-up its own fake login screen.
4. The victim enters his credentials which are stolen and sent to the remote attackers server.
Defences
Defences

- OWASP Top 10
- Code obfuscation
- Don’t download third-party applications from suspicious sources!
- If you have to do it, at least check the manifest.xml for anything “phishy” or even decompile the app (yeah, it’s that easy....)
- Close applications that you don’t trust before using your e-banking App.
- Don’t hand your phone to suspicious looking guys (like me or anyone in this room!) :P
I would like to thank:

- Steven van der Baan
- Aleksander Gorkowienko
- Matthew Seaward

For their help during the research and development of these demos, as well as their innovative ideas!
Questions?
THANK YOU!!!

Sarantis Makoudis
sarantis.makoudis@gmail.com
@SMakoudis
www.7safe.com