Tales of Practical Android Penetration Testing
(Mobile Pentest Toolkit)

Alexander Subbotin

OWASP Bucharest AppSec 2018
About Me

• IT Security Consultant (https://subbotin.de)
• Penetration Tester/Ethical Hacker with 5 years experience
• Working for enterprise (banking industry, telecommunication companies, wholesale, etc.)
• Trainer for Android and Web Pentesting
• Author and Maintainer of Awesome Pentest Cheatsheets project https://github.com/coreb1t/awesome-pentest-cheat-sheets
• Bug Hunter
Yahoo on HackerOne https://hackerone.com/coreb1t
• Setup Pentest Environment
• Requirements:
  – Kali like distribution for mobile penetration testing
  – Updates for most used tools
  – Extensibility
• **Setup Pentest Environment - Current status**

• [https://github.com/tanprathan/MobileApp-Pentest-Cheatsheet](https://github.com/tanprathan/MobileApp-Pentest-Cheatsheet)

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Notes</th>
<th>Last Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobiSec</td>
<td>Last update 3 years ago</td>
<td>3 years ago</td>
</tr>
<tr>
<td>Santoku</td>
<td>Based on Ubuntu 14.04</td>
<td></td>
</tr>
<tr>
<td>Vezir Project</td>
<td>Based on Ubuntu 15.04</td>
<td>2,5 years ago</td>
</tr>
<tr>
<td>Apple</td>
<td>For Window only</td>
<td>2018-05-08</td>
</tr>
<tr>
<td>Android Tamer</td>
<td>Manually updated to last versions of platform-tools, Android SDK, Android Studio and much more</td>
<td></td>
</tr>
</tbody>
</table>
• Setup Pentest Environment
• Do we really need to use separated environment/VM?
• 95 % of time we are using the same (few) tools
  • adb
  • Java Decompiler
  • Tools for static analysis
  • Tools for dynamic analysis
  • Debugger
  • Tools allowing runtime modification
• That is how the idea for Mobile-Pentest-Toolkit (MPT) was born

• For each category of tools use just one tool

F R I D A   apktool   drozer

pidcat   signapk

JD-GUI
• Can you remember all the command line parameters for the mentioned tools?
  • Example:
  • jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore <name> <apk><alias> -storepass <pw>
  • frida -R -f <package-name> -l file.js --no-pause

• You have to specify **what** to do and not **how**.

• MPT provides a simplest interface to your tooling related to android security testing.
• Setup Pentest Environment - **Tools**
• MPT implements a simple package manager
• Currently supported git, http, and zip installation

```python
ANDROID_TOOLS = {
    'pidcat': {
        'url': 'https://github.com/JakeWharton/pidcat',
        'bin': os.path.join(MOBILE_FOLDER + 'pidcat/pidcat.py'),
        'install': 'git'
    },
    'adus': {
        'url': 'https://github.com/coreblt/adus',
        'bin': os.path.join(MOBILE_FOLDER + 'adus/adus.sh'),
        'install': 'git'
    },
    'mobSF': {
        'url': 'https://github.com/ajinabraham/Mobile-Security-Framework-MobSF',
        'bin': '',  # use run_mobile_security_framework() to run the mobSF
        'install': 'git'
    }
}
```
• Setup Pentest Environment - **Device**
• Install Pentest tools
  • XposedFramework
  • Drozer
  • JustTrustMe (xposed plugin)
  • Inspeckage (xposed plugin)
  • ...

• Setup Pentest Environment - Device

• Install Pentest tools
• Setup Pentest Environment - **Pentest**
• Install the app
• Create configuration
• Allows to use MPT from everywhere
• Setup Pentest Environment - **Pentest**
• Install the app
• Create configuration
• Allows to use MPT from everywhere

```bash
python git:(master) × mpt --setup test/root-checker-201810.apk
[00:25:12] [I] Installing apk file: test/root-checker-201810.apk
[00:25:14] [I] Folder for security assessment pentest-2018-10-30 created
python git:(master) × tree pentest-2018-10-30
pentest-2018-10-30
  └── app
        └── root-checker-201810.apk
  └── backup
```
• Starting your favorite tools
• jd-gui (source code review)
• Drozer (android app analysis)
• mobSF (static analysis)
• frida
• and more ...
## OWASP testing methodology – Insecure Data Storage

<table>
<thead>
<tr>
<th>V2</th>
<th>Data Storage and Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Verify that system credential storage facilities are used appropriately to store sensitive data, such as PII, user credentials or cryptographic keys.</td>
</tr>
<tr>
<td>2.2</td>
<td>Verify that no sensitive data is stored outside of the app container or system credential storage facilities.</td>
</tr>
<tr>
<td>2.3</td>
<td>Verify that no sensitive data is written to application logs.</td>
</tr>
<tr>
<td>2.4</td>
<td>Verify that no sensitive data is shared with third parties unless it is a necessary part of the architecture.</td>
</tr>
</tbody>
</table>

adb logcat

```
10-25 12:45:08.590 1833 1833 W System.err: at de.roby.android.xposed.XposedBridge.main(XposedBridge.java:1)
10-25 12:45:08.590 1833 1833 I info : saveLoginInfo: username = test | password = secretpass
10-25 12:45:08.592 1833 1833 I saveLoginInfo: Saving to file /data/user/0/com.htbridge.pivaa/cache/cache182113048F
10-25 12:45:08.609 1833 1833 I hbridge: saveLoginInfoExternalStorage: writable, all ok!
10-25 12:45:08.611 1833 1833 I hbridge: getExternalStorageDirectory = /storage/emulated/0
10-25 12:45:08.615 1833 1833 I hbridge: saveLoginInfoExternalStorage: username = test | password = secretpass
10-25 12:45:10.620 599 1173 I ActivityManager: START u0 {cmp=com.htbridge.pivaa/.WebviewActivity} from uid 1
```
Solution: use pidcat
→ colored output for only on process
Backup Option
→ Compare two states of application
OWASP testing methodology – Insecure Data Storage

Using --backup option create 2 backups for different states

After login the /data/data/<app> folder states differ from each other
Other challenges

• Dynamic analysis
• Dynamic instrumentation and runtime hooking (Frida)
• Root Detection Bypass
• SSL Pinning Bypass
Other challenges

• Dynamic analysis - Inspeckage
Other challenges

- Dynamic instrumentation and runtime hooking (Frida)
- Download a proper Frida version and execute Frida on the device (`--frida` option)
Other challenges

- Dynamic instrumentation and runtime hooking (Frida)
- Use frida to hook cryptographic functions

```javascript
// public static String encryptAES_ECB_PKCS5Padding(String value) {

var enc = Java.use('com.htbridge.pivaa.handlers.Encryption')
enc.encryptAES_ECB_PKCS5Padding.overload('java.lang.String').implementation = function(arg1){
  console.log('enc.encryptAES_ECB_PKCS5Padding() hook')
  console.log('value to encrypt: ' + arg1)
  var a = this.encryptAES_ECB_PKCS5Padding(arg1)
  return a;
}
```
Other challenges

• Dynamic instrumentation and runtime hooking (Frida)
• Use frida to hook cryptographic functions
Other challenges

- Root Detection Bypass

Disable root detection at runtime using frida
Other challenges

- SSL Pinning Bypass
• **Other helpful tools**

  • **Objection** - is a runtime mobile exploration toolkit, powered by [Frida](https://github.com/sensepost/objection) working on not rooted and jailbroken devices.
  
  • [https://github.com/sensepost/objection](https://github.com/sensepost/objection)

  • **AppMon** - automated framework for monitoring and tampering system API calls of native iOS and android apps
  
  • [https://github.com/dpnishant/appmon](https://github.com/dpnishant/appmon)

  • **House** - runtime mobile application analysis toolkit with a Web GUI, powered by [Frida](https://github.com/nccgroup/house)
  
  • [https://github.com/nccgroup/house](https://github.com/nccgroup/house)
• MPT - Overview
  • Setup Pentest Environment
    – Tools
    – Device
    – Config
  • Simple Interface to interact with pentest tools
  • Allows to perform static, dynamic analysis
  • Support to bypass SSL certificate pinning and root detection
  • Supports zsh autocompletion
• Further Ideas
  • Automatically rebuild apk with backup and debug flags enabled (in progress)
  • Automatically generate PoCs for sending broadcast messages and start activities and services (in progress)
  • Integrate file explorer for files on the devices
  • Generate Frida hooks for selected code (method) on the fly
  • Implement anti-debugging bypass (in progress)
Thank you for your attention!

Alexander Subbotin

@coreb1t

@coreb1t