



Tales of Practical Android Penetration Testing (Mobile Pentest Toolkit)

Alexander Subbotin

OWASP Bucharest AppSec 2018



OWASP

The Open Web Application Security Project



OWASP

The Open Web Application Security Project

• About Me

- IT Security Consultant
- 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>



**Alexander
Subbotin**
IT-Security Consulting



OWASP

The Open Web Application Security Project

- Requirements:
 - Kali like distribution for mobile penetrations testing
 - Updates for most used tools
 - extensibility



OWASP

The Open Web Application Security Project

- Setup Pentest Environment
- Requirements:
 - Kali like distribution for mobile penetrations testing
 - Updates for most used tools
 - extensibility



OWASP

The Open Web Application Security Project

- Setup Pentest Environment - Current status
- <https://github.com/tanprathan/MobileApp-Pentest-Cheatsheet>

Distribution	Notes	Last Update
MobiSec	Last update 3 years ago	3 years ago
Santoku	Based on Ubuntu 14.04	
Vezir Project	Based on Ubuntu 15.04	2,5 years ago
Apple	For Window only	2018-05-08
Android Tamer	Manually updated to last versions of platform-tools, Android SDK, Android Studio and much more	



OWASP

The Open Web Application Security Project

- 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



OWASP

The Open Web Application Security Project

- That is how the idea for Mobile-Pentest-Toolkit (MPT) was born
- For each category of tools use just one tool

FRIDA

apktool

abe

drozer



pidcat



signapk



JD-GUI



OWASP

The Open Web Application Security Project

- 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.



OWASP

The Open Web Application Security Project

- Setup Pentest Environment - **Tools**
- MPT implements a simple package manager
- Currently supported git, http, and zip installation

```
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'
    }
}
```



OWASP

The Open Web Application Security Project

- Setup Pentest Environment - **Tools**
- MPT implements a simple package manager
- Currently supported git, http, and zip installation

DEMO

```
ANDROID_TOOLS =
{
    'pidcat': {
        'url': 'https://github.com/akeelmar/pidcat',
        'bin': os.path.join(MOBILE_FOLDER + 'pidcat/pidcat.py'),
        'install': 'git'
    },
    'adus': {
        'url': 'https://github.com/corebit/adus',
        'bin': os.path.join(MOBILE_FOLDER + 'adus/adus.py'),
        'install': 'git'
    },
    'mobSF': {
        'url': 'https://github.com/ajinabraham/Mobile-Security-Framework-MobSF',
        'bin': '', # use run_mobile_security_framework() to run the mobSF
        'install': 'git'
    }
}
```



OWASP

The Open Web Application Security Project

- Setup Pentest Environment - **Device**
- Install Pentest tools
- XposedFramework
- Drozer
- JustTrustMe (xposed plugin)
- Inspeckage (xposed plugin)
- ...



OWASP

The Open Web Application Security Project

- Setup Pentest Environment - **Device**
- Install Pentest tools
- XposedFramework
- Drozer
- JustTrustMe (xposed plugin)
- Inspeckage (xposed plugin)
- ...

DEMO



OWASP

The Open Web Application Security Project

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



OWASP

The Open Web Application Security Project

- Setup Pentest Environment - **Pentest**
- Install the app
- Create configuration
- Allows to use IMPI from everywhere

DEMO



OWASP

The Open Web Application Security Project

- Starting your favorite tools
- jd-gui (source code review)
- Drozer (android app analysis)
- mobSF (static analysis)
- frida
- more



OWASP

The Open Web Application Security Project

- Starting your favorite tools
- jd-gui (source code review)
- Drozer (android app analysis)
- mobSF (static analysis)
- frida
- more

DEMO



OWASP

The Open Web Application Security Project

OWASP testing methodology – Insecure Data Storage

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



adb logcat Is the output really readable?

```
10-25 12:45:08.590 1833 1833 W System.err: at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:776)
10-25 12:45:08.590 1833 1833 W System.err: at de.robv.android.xposed.XposedBridge.main(XposedBridge.java:1
10-25 12:45:08.590 1833 1833 I info : saveLoginInfo: username = test | password = secretpass
10-25 12:45:08.590 1833 1833 I info :
10-25 12:45:08.592 1833 1833 I saveLoginInfo: Saving to file /data/user/0/com.htbridge.pivaa/cache/cache18211
3D48F
10-25 12:45:08.609 1833 1833 I htbridge: saveLoginInfoExternalStorage: writable, all ok!
10-25 12:45:08.611 1833 1833 I htbridge: getExternalStorageDirectory = /storage/emulated/0
10-25 12:45:08.614 1833 1833 I htbridge: getExternalStoragePublicDirectory = /storage/emulated/0/Android/data
10-25 12:45:08.615 1833 1833 I htbridge: saveLoginInfoExternalStorage: username = test | password = secretpas
10-25 12:45:08.615 1833 1833 I htbridge:
10-25 12:45:08.615 1833 1833 I htbridge: saveLoginInfoExternalStorage: opening for writing /storage/emulated/
ials.dat
10-25 12:45:08.617 1833 1833 I htbridge: saveLoginInfoExternalStorage: opening for reading /storage/emulated/
ials.dat
10-25 12:45:10.620 599 1173 I ActivityManager: START u0 {cmp=com.htbridge.pivaa/.WebViewActivity} from uid 1
```



OWASP

The Open Web Application Security Project

OWASP testing methodology – Insecure Data Storage



Solution: use pidcat

→ colored output for only on process

```
W      at de.robv.android.xposed.XposedBridge.main(XposedBridge.java:107)
info    I saveLoginInfo: username = test | password = secretpass
saveLoginInfo I Saving to file /data/user/0/com.htbridge.pivaa/cache/cache1007723920 md5 content = 0B120EC357E51EEC0
B31D0A9F
htbridge I saveLoginInfoExternalStorage: writable, all ok!
I getExternalStorageDirectory = /storage/emulated/0
I getExternalStoragePublicDirectory = /storage/emulated/0/Android/data/com.htbridge.pivaa/files
I saveLoginInfoExternalStorage: username = test | password = secretpass
I saveLoginInfoExternalStorage: opening for writing /storage/emulated/0/Android/data/com.htbridge.pivaa
/credentials.dat
I saveLoginInfoExternalStorage: opening for reading /storage/emulated/0/Android/data/com.htbridge.pivaa
/credentials.dat
chromium I [INFO:CONSOLE(1)] "portal.check()", source: https://www.htbridge.com/ssl/assets/app.js?v=1539773887
cr_AwContents W onDetachedFromWindow called when already detached. Ignoring
```



OWASP

The Open Web Application Security Project

OWASP testing methodology – Insecure Data Storage

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



Backup Option

→ Compare two states of application



OWASP

The Open Web Application Security Project

OWASP testing methodology – Insecure Data Storage

V2	Data Storage and Privacy	
	Verify that system credential storage facilities are used appropriately to store sensitive data, such as PII, user credentials or cryptographic keys.	✓
2.1	Verify that no sensitive data is stored outside of the app container or system credential storage facilities.	
2.2	Verify that no sensitive data is written to application logs.	✓
2.3	Verify that no sensitive data is shared in third parties, unless it is necessary part of the architecture.	✓
2.4		



Backup Option

→ Compare two states of application



OWASP

The Open Web Application Security Project

Other challenges

- Dynamic analysis
- Dynamic instrumentation and runtime hooking (Frida)
- Root Detection Bypass
- SSL Pinning Bypass



OWASP

The Open Web Application Security Project

Other challenges

- Dynamic analysis
- Dynamic instrumentation and runtime hooking (Frida)
- Root Detection Bypass
- SSL Pinning Bypass

DEMO



OWASP

The Open Web Application Security Project

- Other helpful tools
- **Objection** - is a runtime mobile exploration toolkit, powered by [Frida](#) working on not rooted and jailbroken devices.
- <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>
- **House** - runtime mobile application analysis toolkit with a Web GUI, powered by [Frida](#)
- <https://github.com/nccgroup/house>



OWASP

The Open Web Application Security Project

- 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



OWASP

The Open Web Application Security Project

- 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)