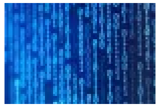
So Long, and Thanks for All the Fish

JUST SOME RANDOM THOUGHTS ABOUT THE MEANING OF LIFE, THE UNIVERSE, AND EVERYTHING

MENU

RECENT POSTS



Win32/StealthFalcon malware uses Windows Background Intelligent Transfer Service (BITS) to communicates to its C&C servers

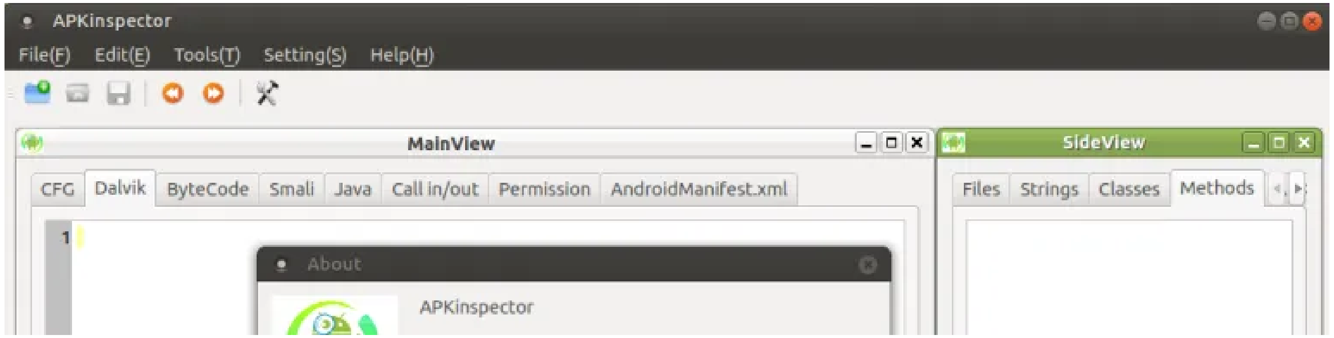
Reverse engineering and penetration testing on Android apps: my own list of tools

Written by Andrea Fortuna • on July 18, 2019 • in Cybersecurity, Penetration Testing

This list of tools is really useful both in penetration testing on an Android application and in reverse engineering of a suspicious application.
All tools are OSS and freely available: *so, enjoy*!

Reverse Engineering

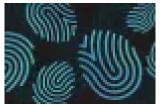
APKInspector



September 13, 2019

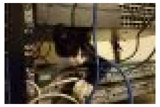
Simjacker: a brand new mobile vulnerability exploited by surveillance companies for espionage operation

September 12, 2019



Some thoughts about Browser Fingerprinting

September 12, 2019



NetCat attack (CVE-2019-11184): steal encrypted SSH keystrokes exploiting DDIO

September 11, 2019

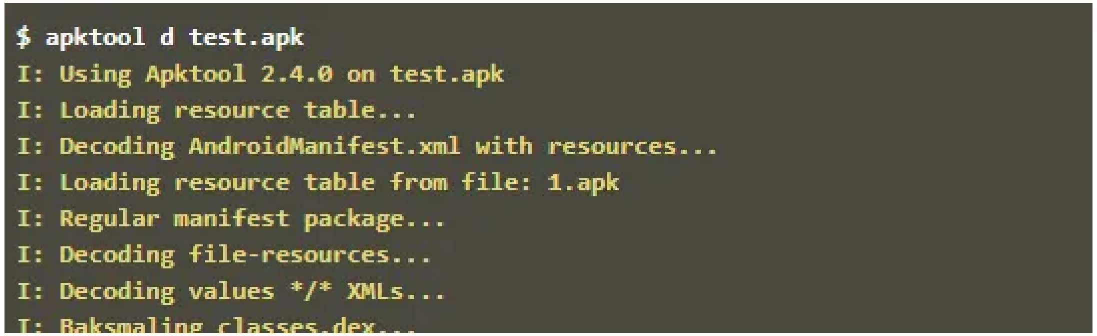
CATEGORIES

Select Category ▾

GUI tool for analysis of Android applications.

The goal of this project is to aide analysts and reverse engineers to visualize compiled Android packages and their corresponding DEX code.

APKTool



```
$ apktool d test.apk
I: Using Apktool 2.4.0 on test.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: 1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
```

A tool for reverse engineering 3rd party, closed, binary Android apps.

It can decode resources to nearly original form and rebuild them after making some modifications.

objection

RECENT COMMENTS

Cybersecurity: DNS Tunneling – Elite Homework on DNS tunneling techniques in cyberattacks

simjacker, anda bisa dimata-matai hanya dengan modal kartu sim - High3 Blog on Simjacker: a brand new mobile vulnerability exploited by surveillance companies for espionage operation

Andrea Fortuna on How to make a “Ultra-Geek” Linux Workstation

Juan on How to make a “Ultra-Geek” Linux Workstation

Process Injection Teknikleri - Process Hollowing ve Atombombing - Manalysiz on RunPE: a practical example of Process Hollowing technique

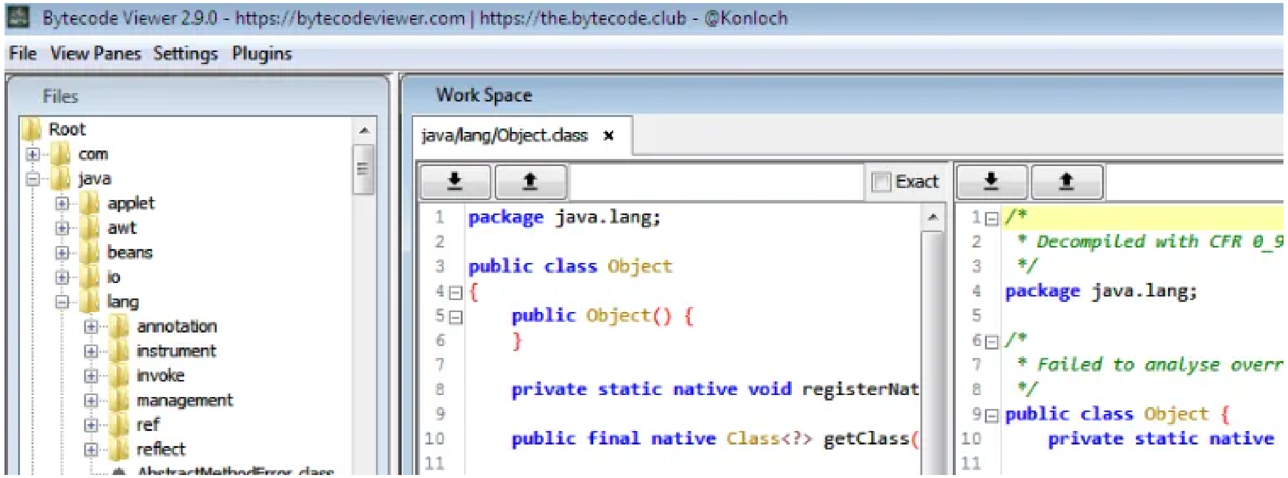
A runtime mobile exploration toolkit, powered by **Frida**.

> *It was built with the aim of helping assess mobile applications and their security posture without the need for a jailbroken or rooted mobile device.*

Sign.jar

Automatically signs an apk with the Android test certificate.

Bytecode Viewer

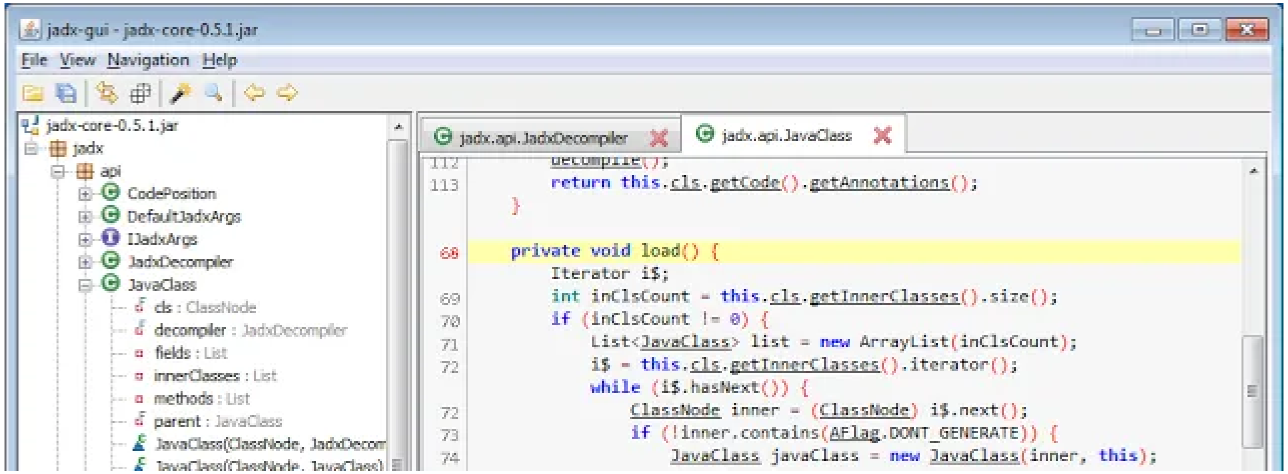

Bytecode Viewer 2.9.0 - https://bytecodeviewer.com | https://the.bytecode.club - @Konloch
File View Panes Settings Plugins
Files
Root
com
java
applet
awt
beans
io
lang
annotation
instrument
invoke
management
ref
reflect
AbstractMethodError.class
Work Space
java/lang/Object.class
Exact

```
package java.lang;

public class Object
{
    public Object() {
    }

    private static native void registerNat

    public final native Class<?> getClass(
```

```
/*
 * Decompiled with CFR 0_9
 */
package java.lang;

/*
 * Failed to analyse overr
 */
public class Object {
    private static native
```

Bytecode Viewer is an Advanced Lightweight Java Bytecode Viewer, It's written completely in Java, and it's open sourced.

Jadx



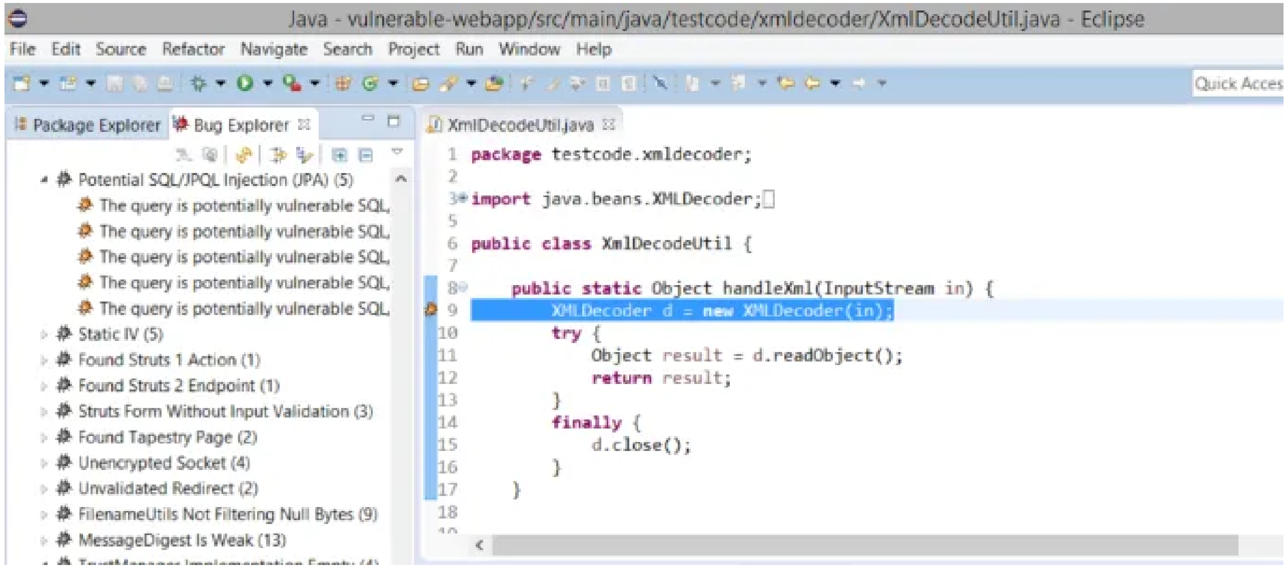
```
        decompile();
        return this.cls.getCode().getAnnotations();
    }

    private void load() {
        Iterator i$;
        int inClsCount = this.cls.getInnerClasses().size();
        if (inClsCount != 0) {
            List<JavaClass> list = new ArrayList(inClsCount);
            i$ = this.cls.getInnerClasses().iterator();
            while (i$.hasNext()) {
                ClassNode inner = (ClassNode) i$.next();
                if (!inner.contains(AFlag.DONT_GENERATE)) {
                    JavaClass javaClass = new JavaClass(inner, this);
```

Dex to Java decompiler: Command line and GUI tools for produce Java source code from Android Dex and Apk files.

Oat2dex

A tool for converting .oat file to .dex files.

FindSecurityBugs



Java - vulnerable-webapp/src/main/java/testcode/xmldecoder/XmlDecodeUtil.java - Eclipse

- Potential SQL/JPQL Injection (JPA) (5)
  - The query is potentially vulnerable SQL
  - The query is potentially vulnerable SQL
  - The query is potentially vulnerable SQL
  - The query is potentially vulnerable SQL
  - The query is potentially vulnerable SQL
- Static IV (5)
- Found Struts 1 Action (1)
- Found Struts 2 Endpoint (1)
- Struts Form Without Input Validation (3)
- Found Tapestry Page (2)
- Unencrypted Socket (4)
- Unvalidated Redirect (2)
- FilenameUtils Not Filtering Null Bytes (9)
- MessageDigest Is Weak (13)

```
package testcode.xmldecoder;

import java.beans.XMLDecoder;

public class XmlDecodeUtil {

    public static Object handleXml(InputStream in) {
        XMLDecoder d = new XMLDecoder(in);
        try {
            Object result = d.readObject();
            return result;
        }
        finally {
            d.close();
        }
    }
```

FindSecurityBugs is a extension for **FindBugs** which include security rules for Java applications.

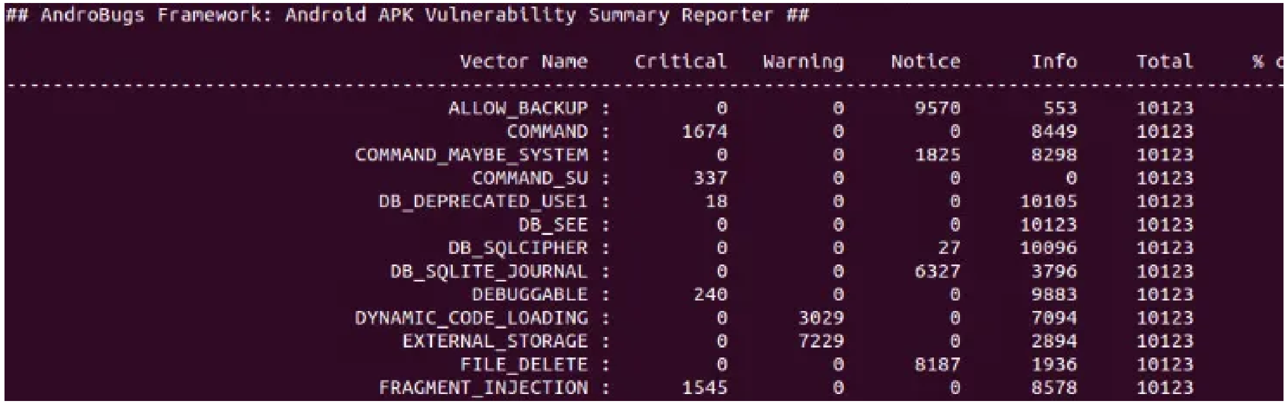
Quick Android Review Kit (Qark)

A tool designed to look for several security related Android application vulnerabilities, either in source code or packaged APKs.

Secure, Unified, Powerful and Extensible Rust Android Analyzer

SUPER is a command-line application, developed in **Rust**, that can be used in Windows, MacOS X and Linux, that analyzes .apk files in search for vulnerabilities.

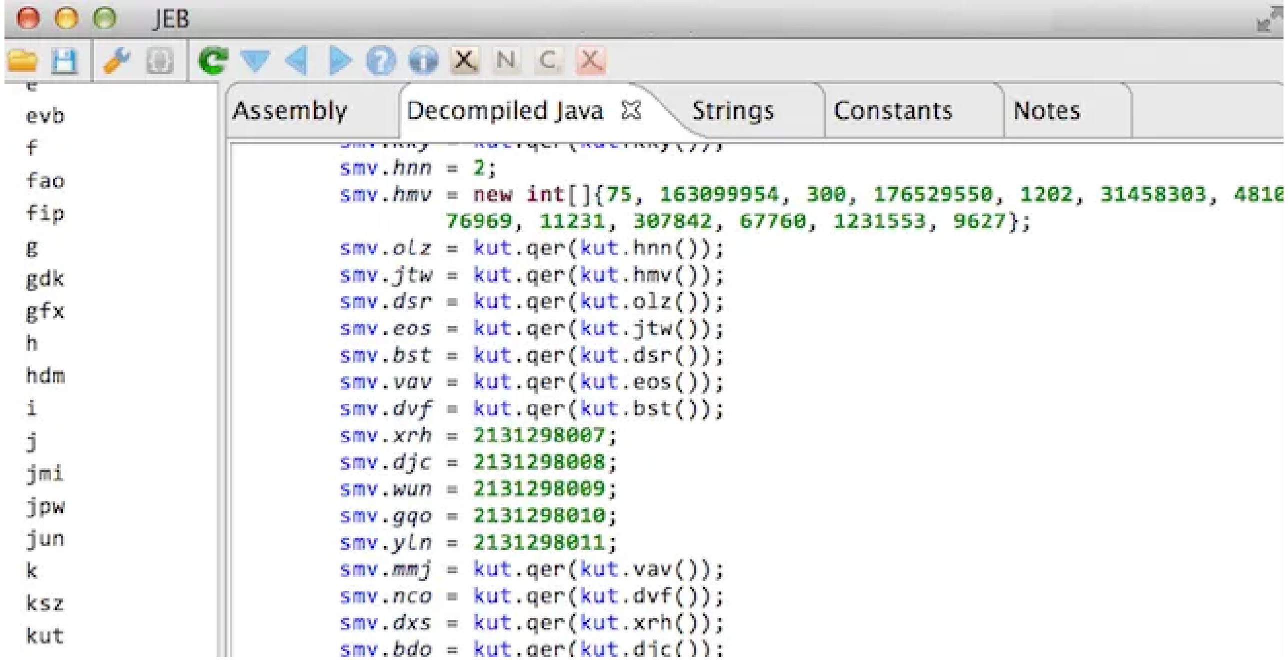
AndroBugs Framework



```
## AndroBugs Framework: Android APK Vulnerability Summary Reporter ##

                                       Vector Name    Critical    Warning    Notice       Info      Total     % c
-----------------------------------------------------------------------------------------------------------------
                                      ALLOW_BACKUP :         0          0      9570        553      10123
                                           COMMAND :      1674          0         0       8449      10123
                              COMMAND_MAYBE_SYSTEM :         0          0      1825       8298      10123
                                        COMMAND_SU :       337          0         0          0      10123
                                 DB_DEPRECATED_USE1 :        18          0         0      10105      10123
                                            DB_SEE :         0          0         0      10123      10123
                                       DB_SQLCIPHER :         0          0        27      10096      10123
                                  DB_SQLITE_JOURNAL :         0          0      6327       3796      10123
                                         DEBUGGABLE :       240          0         0       9883      10123
                               DYNAMIC_CODE_LOADING :         0       3029         0       7094      10123
                                   EXTERNAL_STORAGE :         0       7229         0       2894      10123
                                        FILE_DELETE :         0          0      8187       1936      10123
                                 FRAGMENT_INJECTION :      1545          0         0       8578      10123
```

Android vulnerability scanner that helps pentesters to find potential security vulnerabilities in Android applications.

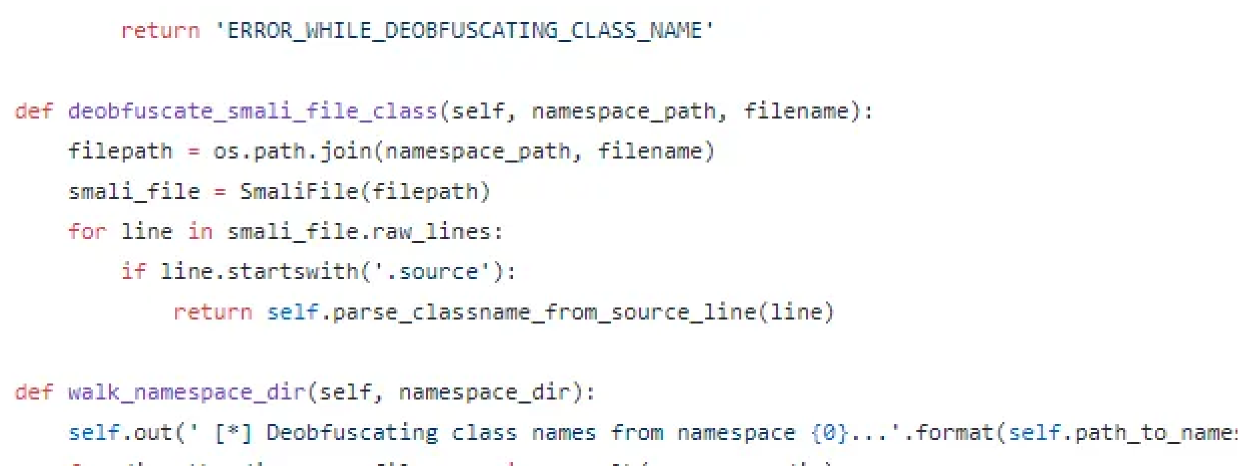
Simplify



```
        smv.hnn = 2;
        smv.hmv = new int[]{75, 163099954, 300, 176529550, 1202, 31458303, 4810
                76969, 11231, 307842, 67760, 1231553, 9627};
        smv.olz = kut.qer(kut.hnn());
        smv.jtw = kut.qer(kut.hmv());
        smv.dsr = kut.qer(kut.olz());
        smv.eos = kut.qer(kut.jtw());
        smv.bst = kut.qer(kut.dsr());
        smv.vav = kut.qer(kut.eos());
        smv.dvf = kut.qer(kut.bst());
        smv.xrh = 2131298007;
        smv.djc = 2131298008;
        smv.wun = 2131298009;
        smv.gqo = 2131298010;
        smv.yln = 2131298011;
        smv.mmj = kut.qer(kut.vav());
        smv.nco = kut.qer(kut.dvf());
        smv.dxs = kut.qer(kut.xrh());
        smv.bdo = kut.qer(kut.djc());
```

Tool for de-obfuscating android package into Classes.dex which can be use Dex2jar and JD-GUI to extract contents of dex file.

ClassNameDeobfuscator



```
            return 'ERROR_WHILE_DEOBFUSCATING_CLASS_NAME'

    def deobfuscate_smali_file_class(self, namespace_path, filename):
        filepath = os.path.join(namespace_path, filename)
        smali_file = SmaliFile(filepath)
        for line in smali_file.raw_lines:
            if line.startswith('.source'):
                return self.parse_classname_from_source_line(line)

    def walk_namespace_dir(self, namespace_dir):
        self.out(' [*] Deobfuscating class names from namespace {0}...'.format(self.path_to_name
```

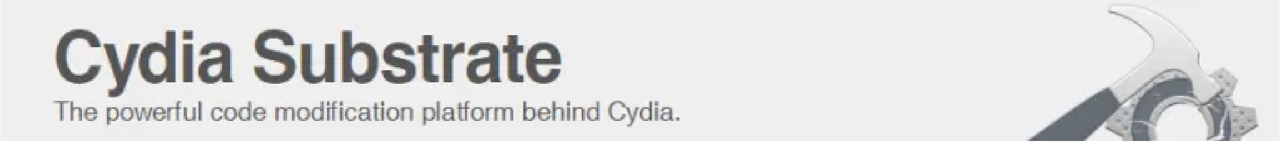
Python script to parse through the .smali files produced by **apktool** and extract the .source annotation lines.

Android backup extractor

Utility to extract and repack Android backups created with **adb backup** (ICS+). More info about adb backup **here**.

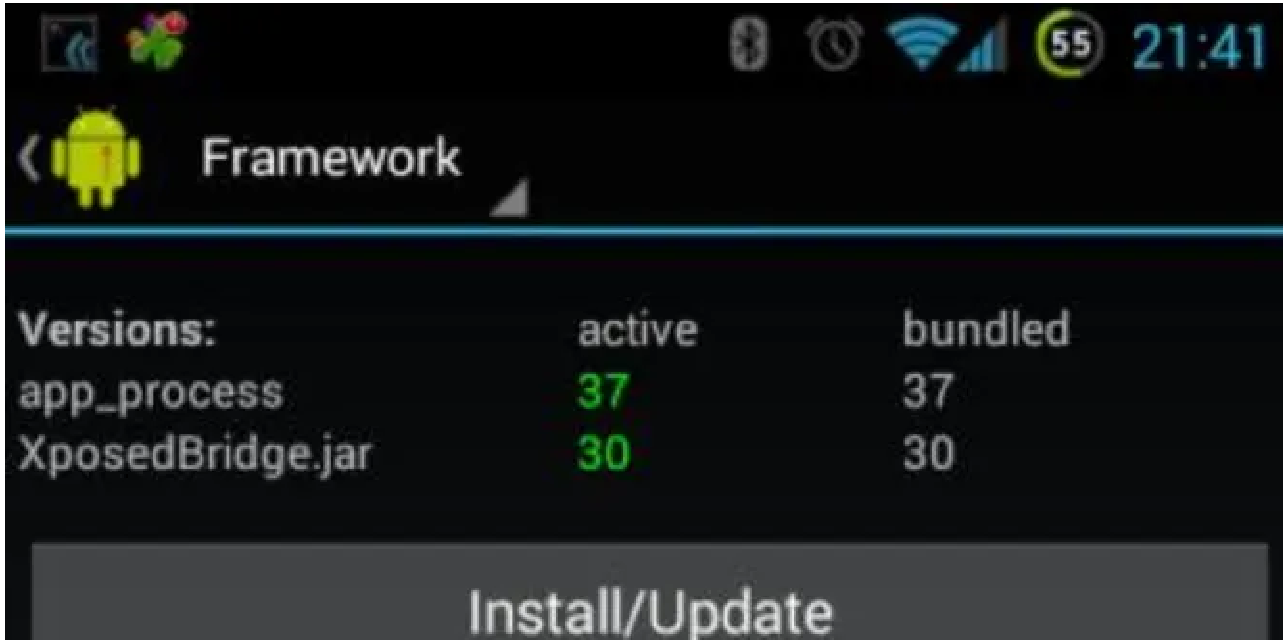
Dynamic Analysis

Cydia Substrate



Android version of well-known **iOS's Cydia Substrate**: it enables developers to make changes to existing software with extensions that are injected in to the target process's memory.

Xposed Framework



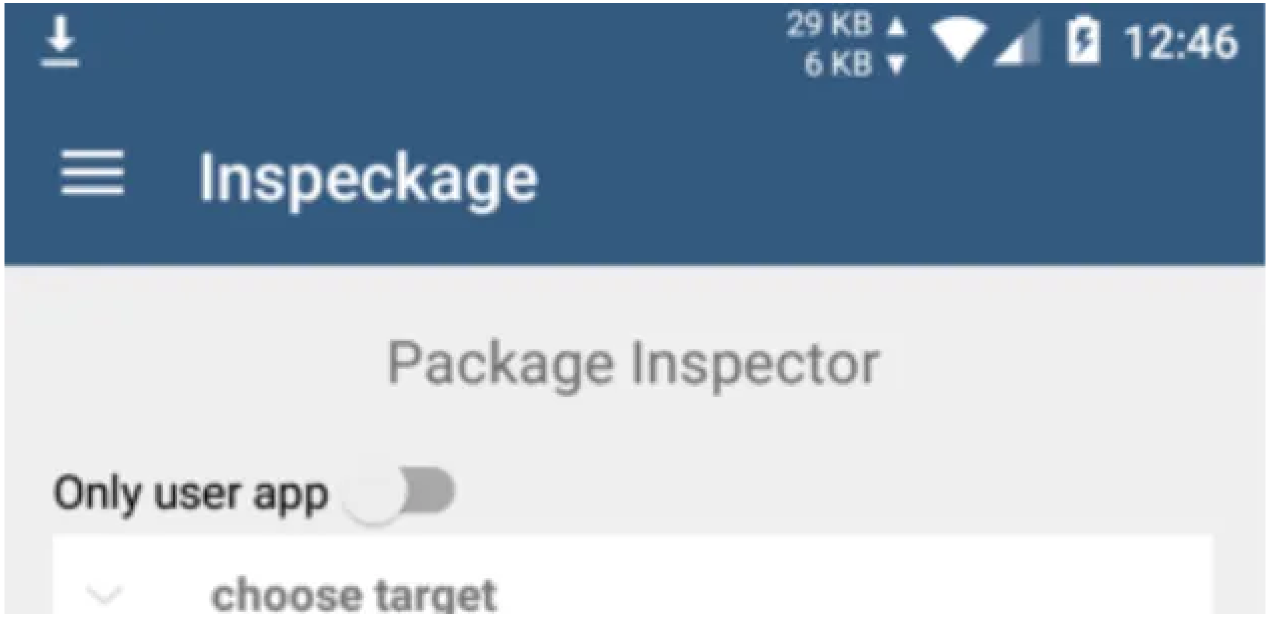
Versions:	active	bundled
app_process	37	37
XposedBridge.jar	30	30

Xposed framework enables analysts to modify the system or application behaviour at runtime, without modifying any package or re-flashing.

logcat-color

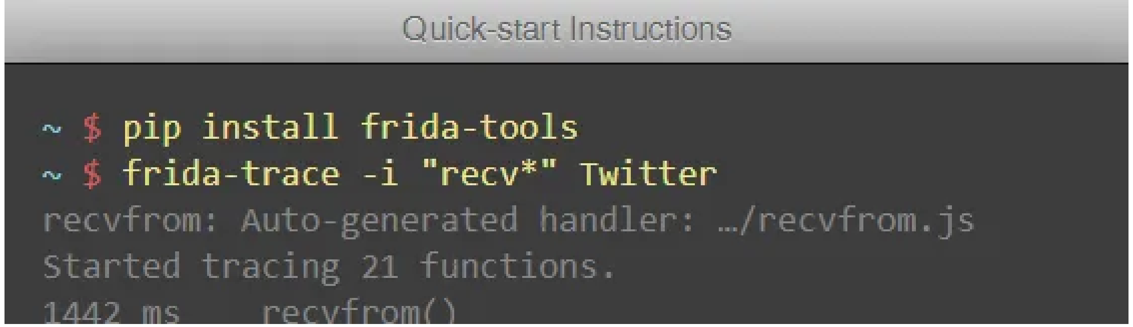
A colorful and highly configurable alternative to the **adb logcat** command from the Android SDK.

Inspeckage



Tool developed for dynamic analysis of Android applications.
By applying hooks to functions of the Android API, **Inspeckage** will help analysts to track what an Android application is doing at runtime.

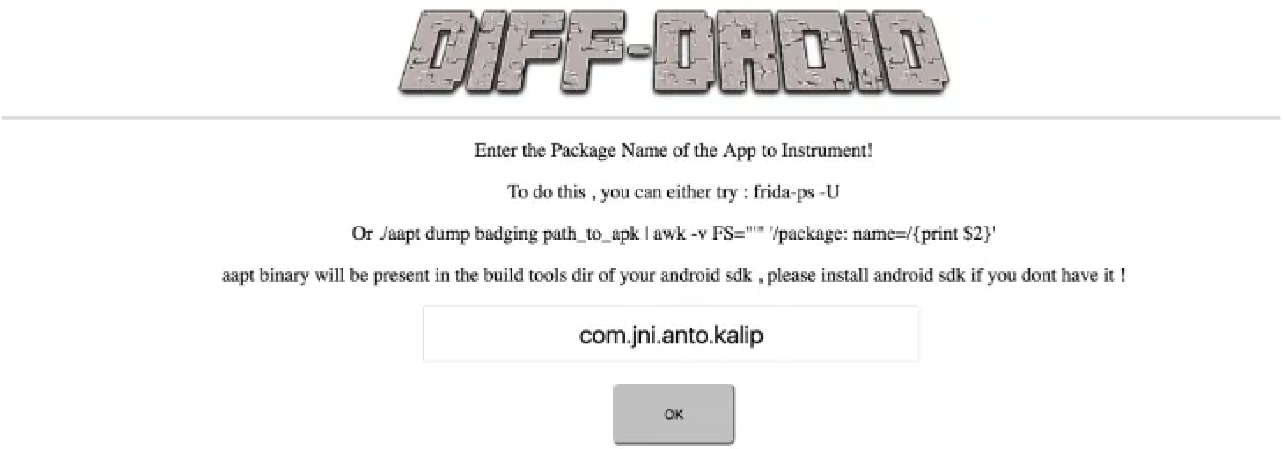
Frida



```
~ $ pip install frida-tools
~ $ frida-trace -i "recv*" Twitter
recvfrom: Auto-generated handler: …/recvfrom.js
Started tracing 21 functions.
1442 ms     recvfrom()
```

The toolkit works using a client-server model and lets you inject in to running processes not just on Android, but also on iOS, Windows and Mac.

Diff-GUI



Enter the Package Name of the App to Instrument!

To do this , you can either try : frida-ps -U

Or ./aapt dump badging path_to_apk | awk -v FS="'" '/package: name=/{print $2}'

aapt binary will be present in the build tools dir of your android sdk , please install android sdk if you dont have it !

com.jni.anto.kalip

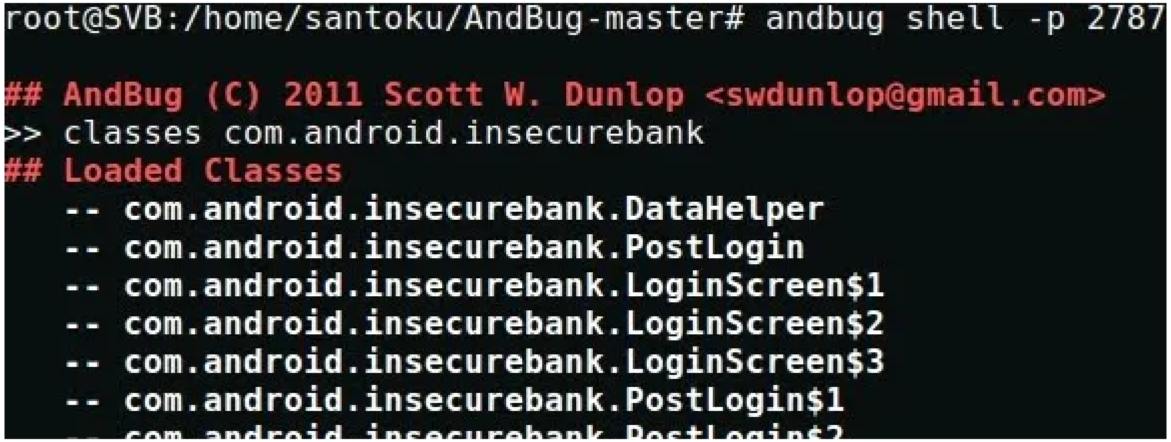
A Web framework to start instrumenting with the available modules, hooking on native, inject JavaScript using Frida.

House

```
        House

Dynamic Mobile Analysis Tool
Contact: hao.ke@nccgroup.com
```

A runtime mobile application analysis toolkit with a Web GUI, powered by Frida, is designed for helping assess mobile applications by implementing dynamic function hooking and intercepting and intended to make Frida script writing as simple as possible.

AndBug



```
root@SVB:/home/santoku/AndBug-master# andbug shell -p 2787

## AndBug (C) 2011 Scott W. Dunlop <swdunlop@gmail.com>
>> classes com.android.insecurebank
## Loaded Classes
   -- com.android.insecurebank.DataHelper
   -- com.android.insecurebank.PostLogin
   -- com.android.insecurebank.LoginScreen$1
   -- com.android.insecurebank.LoginScreen$2
   -- com.android.insecurebank.LoginScreen$3
   -- com.android.insecurebank.PostLogin$1
   -- com.android.insecurebank.PostLogin$2
```

AndBug is a debugger targeting the Android platform's Dalvik virtual machine intended for reverse engineers and developers.

Introspy-Android

Blackbox tool to help understand what an Android application is doing at runtime and assist in the identification of potential security issues.

Drozer

Drozer allows you to search for security vulnerabilities in apps and devices by assuming the role of an app and interacting with the Dalvik VM, other apps' IPC endpoints and the underlying OS.

Bypassing Root Detection and SSL Pinning

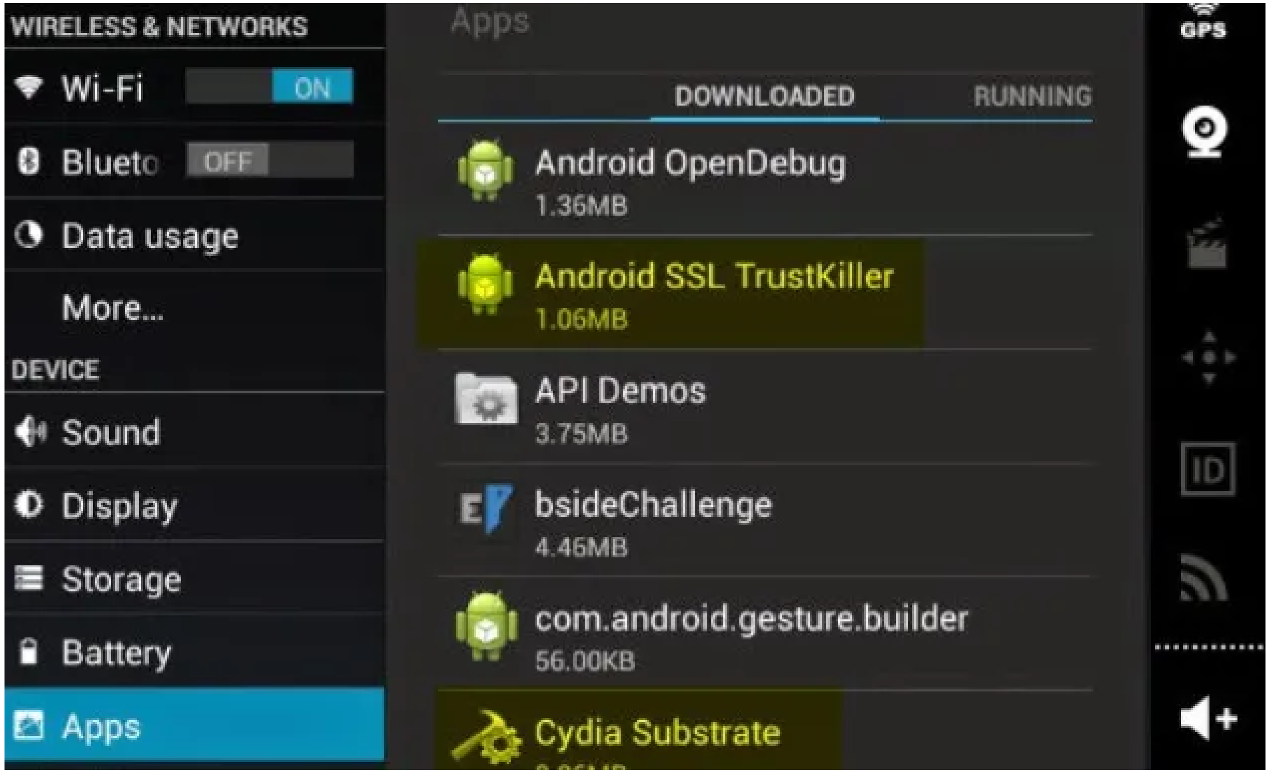
Xposed: Just Trust Me

Xposed Module to bypass SSL certificate pinning.

Xposed: SSLUnpinning

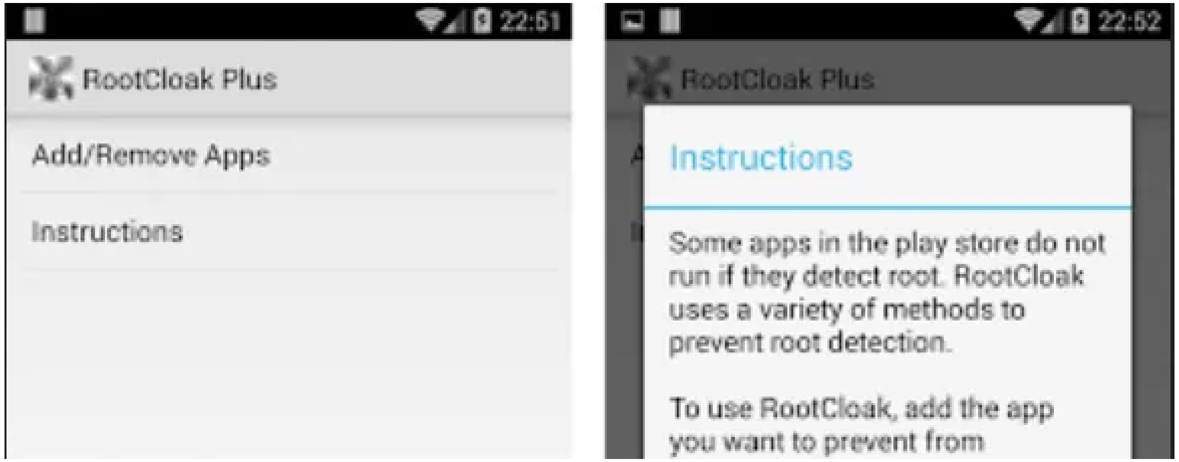
Android Xposed Module to bypass SSL certificate validation (Certificate Pinning).

Cydia Substrate: Android SSL Trust Killer



Blackbox tool to bypass SSL certificate pinning for most applications running on a device.

Cydia Substrate: RootCoak Plus



Some apps in the play store do not run if they detect root. RootCloak uses a variety of methods to prevent root detection.

To use RootCloak, add the app you want to prevent from

Patch root checking for commonly known indications of root.

Android Pinning

```
// Define an array of pins.  One of these must be present
// in the certificate chain you receive.  A pin is a hex-encoded
// hash of a X.509 certificate's SubjectPublicKeyInfo. A pin can
// be generated using the provided pin.py script:
// python ./tools/pin.py certificate_file.pem
String[] pins                 = new String[] {"f30012bbc18c231ac1a44b788e410ce754182513"};
URL url                       = new URL("https://www.google.com");
HttpsURLConnection connection = PinningHelper.getPinnedHttpsURLConnection(context, pins, url);

return connection.getInputStream();
```

A standalone library project for certificate pinning on Android.

Android-ssl-bypass

Android debugging tool that can be used for bypassing SSL, even when certificate pinning is implemented, as well as other debugging tasks.

Related posts

Share this:

Like this:

Loading...

TAGS

ANDROID FEATURE PENETRATION TESTING TOOLS

PRINT

Andrea Fortuna

Elon Musk unveils Neuralink: tiny wires in the brain to read electrical pulses and let humans 'merge with computers'

COMMENTS

Enter your comment here...

This site uses Akismet to reduce spam. Learn how your comment data is processed.