Similar to client/server applications and web applications, mobile applications and HTML sites need to be tested from a security penetration perspective. Penetration testing is a blend of art and science, where each tester brings their unique skills and experience for the art, and manual and automated testing and tools for the science. To help with the latter part, this appendix provides a list of a few free penetrating testing tools helpful with auditing mobile applications, mobile networking, and mobile HTML sites.

This list is not exhaustive, but just a start in providing resources for the new emerging area. An updated list can always be found at www.isecpartners.com/mobile_application_tools.html.

Mobile Platform Attack Tools and Utilities

True mobile-specific security tools are still somewhat rare; however, as of 2009, security research into mobile applications and mobile platforms has increased significantly, resulting in a number of tools dedicated entirely to this area.

Manifest Explorer

- Author: Jesse Burns
- Location: www.isecpartners.com/mobile_application_tools.html

Manifest Explorer is a tool that can be used on any device using the Google Android operating system. On Android, every application must have an AndroidManifest.xml file in its root directory. The AndroidManifest.xml file does a few things, which are all explained at http://developer.android.com/guide/topics/manifest/manifest-intro.html. From a security perspective, the file is most interesting because it defines the permissions the application must have to other applications or protected parts of the API. The Manifest Explorer tool can be used to review the AndroidManifest.xml file, specifically the security permissions of the application, and to give the pen-tester a view of the basic attack surface of the application. The attack surface is a critical starting point to understand the security of the application and how it affects the mobile device itself.

The tool is quite simple to use. As shown in Figure B-1, the tool lists all the system's applications, allows the user to select one, and then displays the contents of the AndroidManifest.xml file that pertains to the selected application. A menu option enables saving the extracted manifest, so the testers can read it more comfortably on a PC for manual inspection.

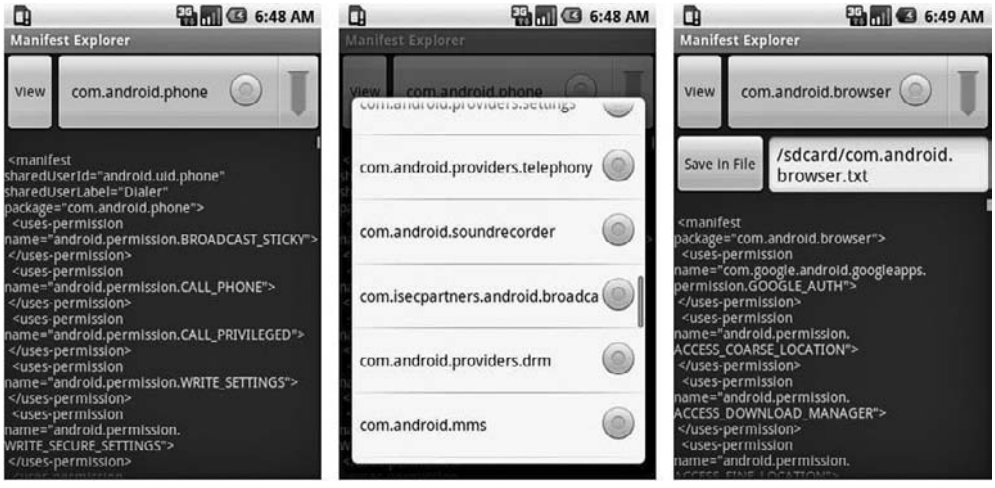

Figure B-1 *Manifest Explorer main screen, displaying the com.android.phone manifest*

Package Play

- Author: Jesse Burns
- Location: www.isecpartners.com/mobile_application_tools.html

Package Play is a tool that can be used on any device using the Google Android operating system. Package Play shows the user all installed packages on the mobile device. This helps the user in the following ways:

- Provides an easy way to start exported Activities
- Shows defined and used permissions
- Shows Activities, Services, Receivers, Providers, and instrumentation, as well as their export and permission status
- Switches to Manifest Explorer or the Settings application's view of the application

Figure B-2 shows a screenshot of Package Play. The first step with Package Play is to select the package to examine. By reviewing the list, the user may see software they did not originally install, such as software preloaded by the hardware manufacturer but not included in the open-source Android OS.

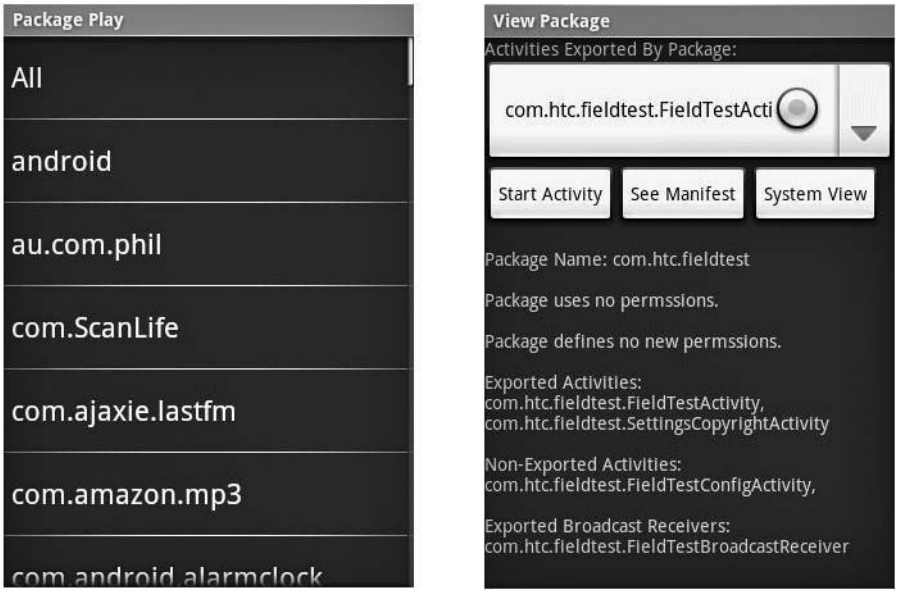

Figure B-2 *Package Play listing packages and exploring package activities*

Intent Sniffer

- Author: Jesse Burns
- Location: www.isecpartners.com/mobile_application_tools.html

Intent Sniffer is a tool that can be used on any device using the Google Android operating system. On the Android OS, an Intent is a description of an action to be performed, such as startService to start a service. The Intent Sniffer tool performs monitoring of runtime routed broadcasts Intents. It does not see explicit broadcast Intents, but defaults to (mostly) unprivileged broadcasts. There is an option to see recent tasks' Intents (GET_TASKS). Activities' Intents are visible when started. The tool can also dynamically update Actions and Categories. Figure B-3 shows a screenshot of Intent Sniffer.

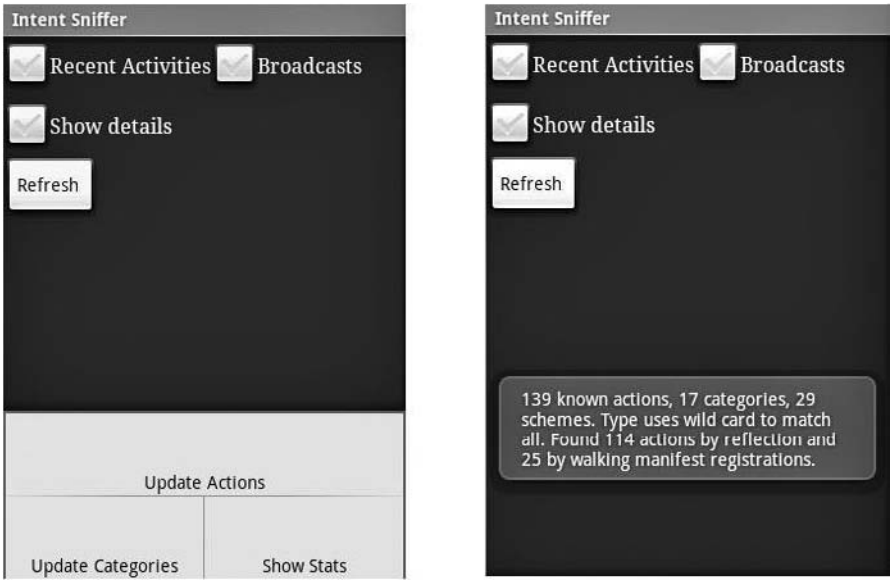

Figure B-3 *Intent Sniffer output*

Intent Fuzzer

- Author: Jesse Burns
- Location: www.isecpartners.com/mobile_application_tools.html

Intent Fuzzer is a tool that can be used on any device using the Google Android operating system. Intent Fuzzer is exactly what is seems—it is a fuzzer. It often finds bugs that cause the system to crash as well as performance issues on the device. The tool can either fuzz a single component or all components. It works well on Broadcast receivers, and works average on Services. As for Activities, only single Activities can be fuzzed, not all of them. Instrumentations can also be started using this interface, and Content Providers are listed, but are not an Intent-based interprocess communication (IPC) mechanism. Figure B-4 shows a screenshot of Intent Fuzzer.

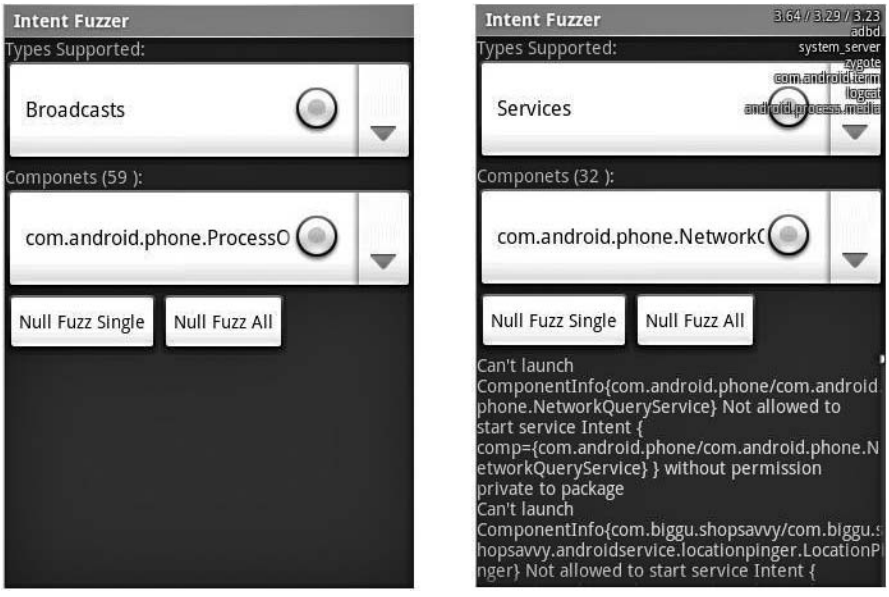

Figure B-4 *Intent Fuzzer*

pySimReader

- Authors: Zane Lackey and Luis Miras
- Location: www.isecpartners.com/mobile_application_tools.html

pySimReader is a tool to write out arbitrary raw SMS PDUs to a SIM card. It is a modified version of Todd Whiteman's PySimReader code. Additionally, debugging output has been added to allow the user to view all Application Protocol Data Units (APDUs) that are sent between the SIM card and pySimReader. The requirements for this tool are Windows XP with Python 2.5 and the ACS ACR 38t SIM reader. Here's a sample usage:

```
# Start the app
python pySimReader.py

# To run with debugging mode enabled
# (This will print out all APDUs sent between the SIM and
pySimReader)

python pySimReader.py  d
```

Browser Extensions

Several add-ons to the Firefox browser are available that are useful for security testing, web development, and mobile device simulation. Often, we find it easier to do the testing of mobile HTML sites from a desktop browser than from a mobile device or simulator. Here are a few of our favorites.

WMLBrowser

- Author: Matthew Wilson
- Location: https://addons.mozilla.org/en-US/firefox/addon/62

The WMLBrowser Firefox add-on simulates WAP browsing by parsing and rendering pages written in the Wireless Markup Language. This is useful for performing testing of mobile sites with a WAP component, because you can leverage all of your existing Firefox tools and network proxies—and of course you use an actual keyboard. Note that some sites will not deliver WML content without detecting the correct User-Agent.

User Agent Switcher

- Author: Chris Pederick
- Location: https://addons.mozilla.org/en-US/firefox/addon/59

Because some sites make decisions on what content to show you by examining your browser's "User-Agent" header, you may sometimes want to trick the server into thinking you're a different browser (such as a WebKit-based mobile browser like on the iPhone, Android, and Symbian). The User Agent Switcher Firefox add-on allows you to change your User-Agent header at will, from a list of User-Agent strings that you define (see Figures B-5 and B-6). This can allow you to interact with mobile sites from the comfort of your own desktop browser.

FoxyProxy

- Author: Eric H. Jung
- Location: https://addons.mozilla.org/en-US/firefox/addon/2464