MWR LABS

Public

EXTERNAL

drozer

Black Hat Tools Arsenal

1st August 2013

What is drozer?

drozer is the leading security testing framework for Android

(previously known as Mercury)

What is drozer?

[Local assessment of apps/devices]

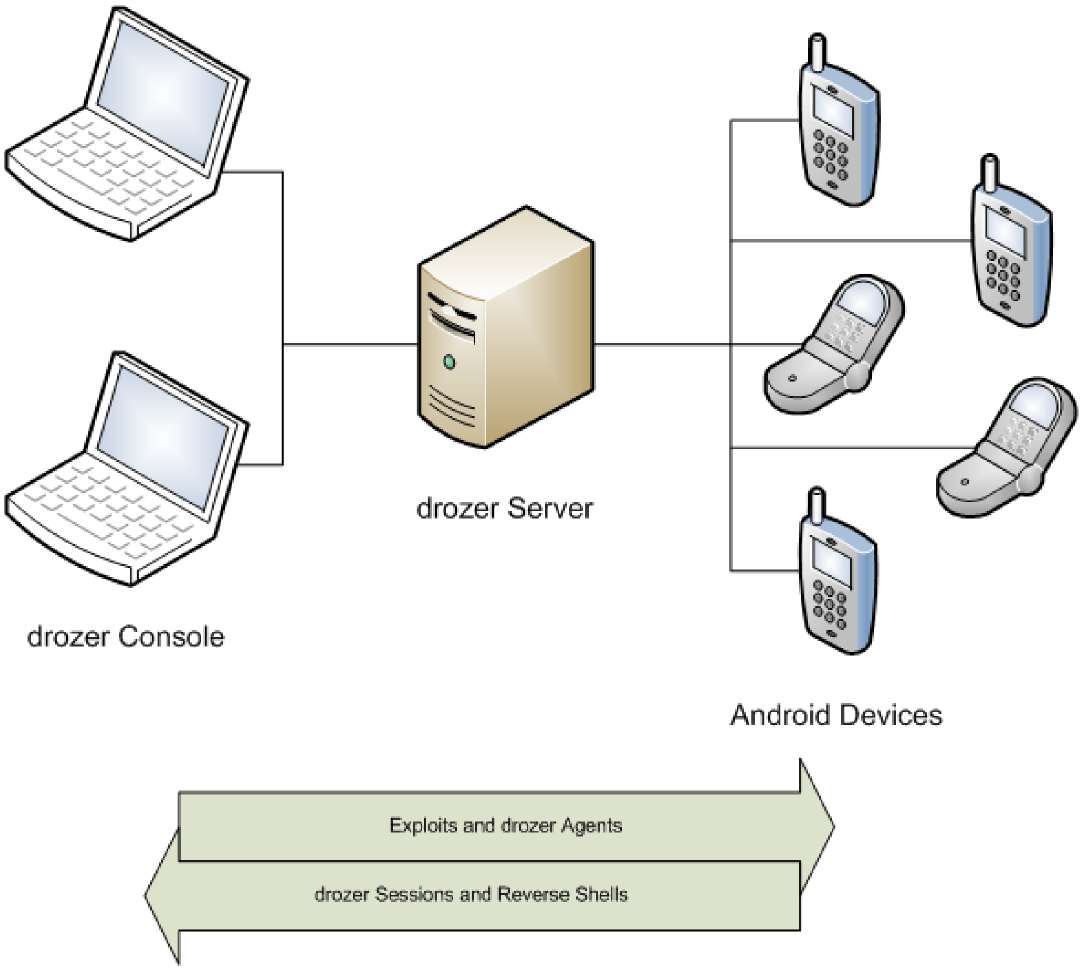
drozer enables you to search for security vulnerabilities in apps and devices by assuming the role of an app and interacting with the Dalvik VM, other apps' IPC endpoints and the underlying OS.

What is drozer?

[Remote exploitation of devices]

drozer provides tools to help you use and share public Android exploits. It helps you to deploy a drozer agent by using weasel – MWR's advanced exploitation payload.

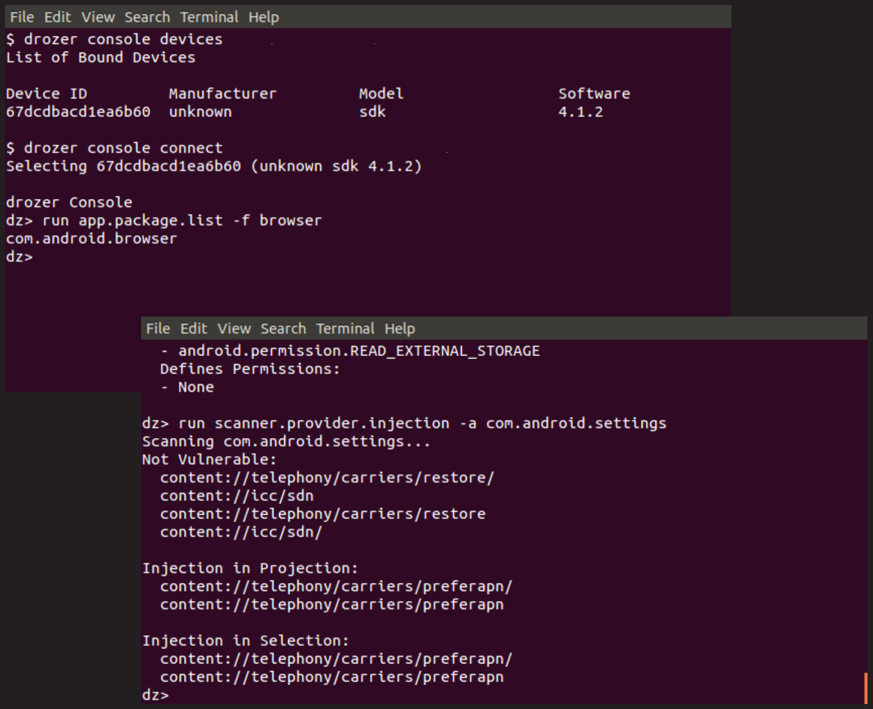
Under the Hood


drozer Console
drozer Server
Android Devices
Exploits and drozer Agents
drozer Sessions and Reverse Shells

The drozer Console

The console is your key interface to the power of drozer.

It provides modules that are run on the connected device to help you find and exploit vulnerabilities.



File Edit View Search Terminal Help

```
$ drozer console devices
List of Bound Devices

Device ID         Manufacturer         Model                 Software
67dcdbacd1ea6b60  unknown              sdk                   4.1.2

$ drozer console connect
Selecting 67dcdbacd1ea6b60 (unknown sdk 4.1.2)

drozer Console
dz> run app.package.list -f browser
com.android.browser
dz>
```

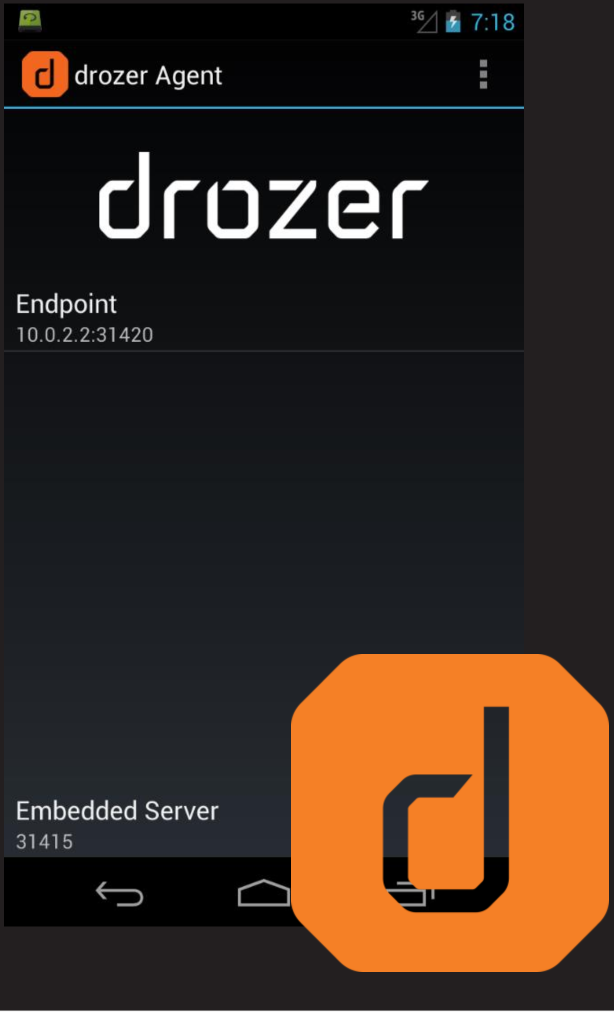
File Edit View Search Terminal Help

```
  - android.permission.READ_EXTERNAL_STORAGE
  Defines Permissions:
  - None

dz> run scanner.provider.injection -a com.android.settings
Scanning com.android.settings...
Not Vulnerable:
  content://telephony/carriers/restore/
  content://icc/sdn
  content://telephony/carriers/restore
  content://icc/sdn/

Injection in Projection:
  content://telephony/carriers/preferapn/
  content://telephony/carriers/preferapn

Injection in Selection:
  content://telephony/carriers/preferapn/
  content://telephony/carriers/preferapn
dz>
```

The drozer Agent

The drozer agent allows the console to interact with a device.

It provides a simple interface for execution of code pushed by the console, so all of the heavy lifting is done by the console and modules.



Endpoint
10.0.2.2:31420

Embedded Server
31415

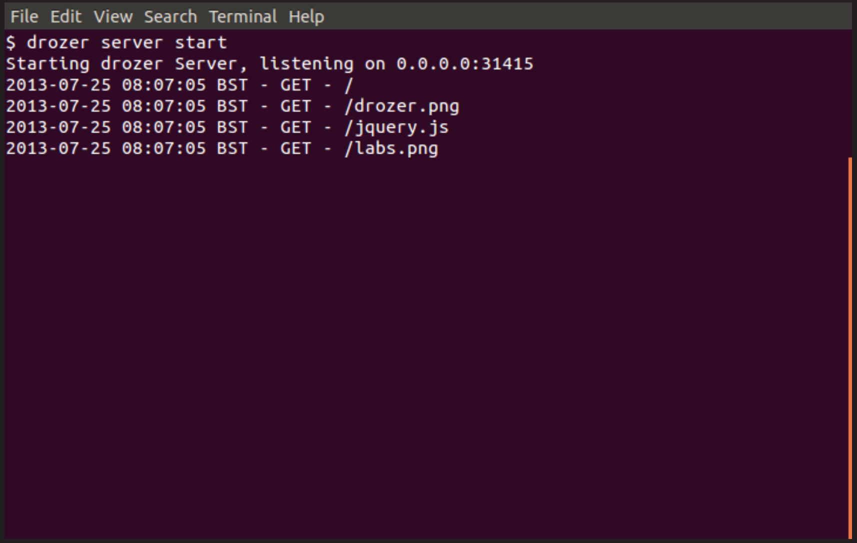
The drozer Server

Catches drozer sessions and reverse shells sent by exploit payloads.

Serves the WWW exploit page for browser exploits.

Speaks 3 protocols:

- HTTP
- drozerp
- Shell



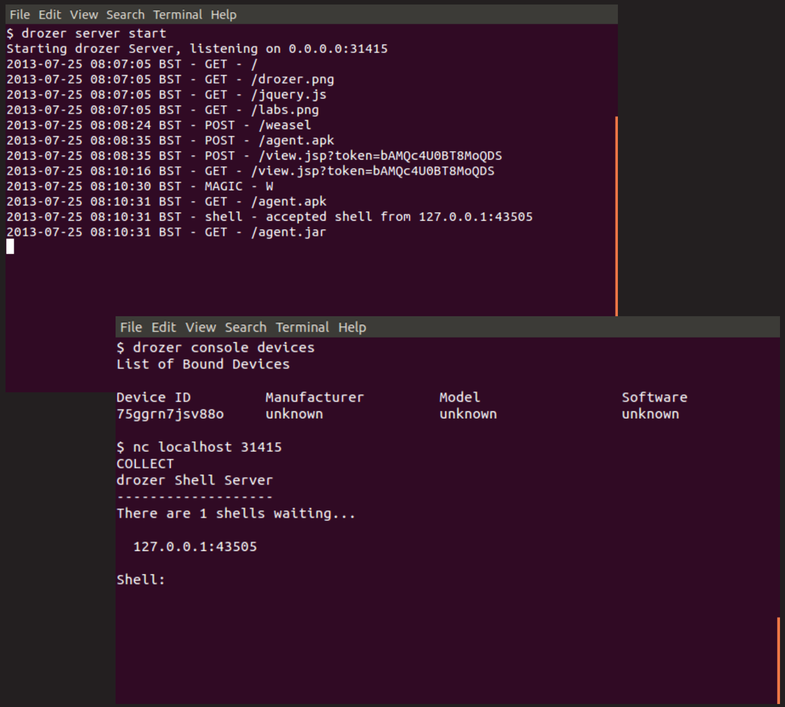
```
$ drozer server start
Starting drozer Server, listening on 0.0.0.0:31415
2013-07-25 08:07:05 BST - GET - /
2013-07-25 08:07:05 BST - GET - /drozer.png
2013-07-25 08:07:05 BST - GET - /jquery.js
2013-07-25 08:07:05 BST - GET - /labs.png
```

The drozer Server

Exploit modules dynamically push new resources to the “web server”.

Advanced options include:

- User-Agent Checking
- ‘Magic’ Byte
- Forced MIME Type



```
$ drozer server start
Starting drozer Server, listening on 0.0.0.0:31415
2013-07-25 08:07:05 BST - GET - /
2013-07-25 08:07:05 BST - GET - /drozer.png
2013-07-25 08:07:05 BST - GET - /jquery.js
2013-07-25 08:07:05 BST - GET - /labs.png
2013-07-25 08:08:24 BST - POST - /weasel
2013-07-25 08:08:35 BST - POST - /agent.apk
2013-07-25 08:08:35 BST - POST - /view.jsp?token=bAMQc4U0BT8MoQDS
2013-07-25 08:10:16 BST - GET - /view.jsp?token=bAMQc4U0BT8MoQDS
2013-07-25 08:10:30 BST - MAGIC - W
2013-07-25 08:10:31 BST - GET - /agent.apk
2013-07-25 08:10:31 BST - shell - accepted shell from 127.0.0.1:43505
2013-07-25 08:10:31 BST - GET - /agent.jar
```

```
$ drozer console devices
List of Bound Devices

Device ID        Manufacturer         Model                Software
75ggrn7jsv88o    unknown              unknown              unknown

$ nc localhost 31415
COLLECT
drozer Shell Server
-------------------
There are 1 shells waiting...

  127.0.0.1:43505

Shell:
```

drozer Exploits

- Publicly known vulnerabilities
- Public exploits with improved reliability
- Some of MWR’s own exploits

Browser exploits
File format exploits
Social engineering
DoS

weasel – MWR's new Android payload

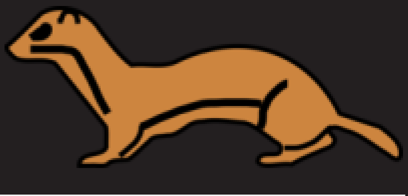
So much more than silly old shells…

A binary created in Android NDK in C

Provides 3 'weasels':

- privileged_weasel() – INSTALL_PACKAGES?
- sneaky_weasel() – app_process JAR loading
- defeated_weasel() – send a shell

Maximum leverage on device from the start



weasel – MWR's new Android payload

Stager for weasel does the following:

- Connects to the drozer Server
- Sends "W"
- dup2()'s stdin/stdout/stderr to network socket
- EXECVE('/system/bin/sh')

Server responds with:

- Echo weasel binary into /data/data/…
- Run weasel

Get drozer

http://mwr.to/drozer

@mwrdrozer +drozer