Androguard usage

February 15, 2018 elcapitan Android Main Mobile

Androguard is a python tool for analyzing Android applications. It can decompile and analyze APK files.

Install Androguard

Androguard is written in python 2.7. The first step in installing Androguard is determining the path to python 2.7 and creating a virtual environment. The virtual

environment is a container and has its own installation directories for python modules.

which python2

On my computer this is */usr/bin/python2*, which is a symlink to python2.7. The next command creates a virtual environment for python. If virtualenv is not installed, then it should be installed first ('*pip install virtualenv*').

virtualenv -p /usr/bin/python2 .vepy27

.vepy27 is the name of the folder, that will contain the installed python modules. Now we should activate the environment.

source .vepy27/bin/activate

The '(.vepy27)' is visible at the beginning of the command prompt. This is a sign, that we have a virtual environment. Now install the androguard.

pip install androguard

pip install ipython

When you finished your work, simply deactivate the virtual environment with:

deactivate

RECENT POSTS

Androguard usage

How to debug an iOS application with Appmon and LLDB

OWASP Uncrackable – Android Level3

OWASP Uncrackable – Android Level2

How to install Appmon and Frida on a Mac

CATEGORIES

Android (5)

Fusion (2)

IoT (13)

Main (3)

Mobile (6)

Protostar (24)

SLAE32 (8)

VulnServer (6)

Windows Reverse Shell (2)

Analyze an APK with Androguard

androlyze.py -s

The command prompt changes. Now analyze an APK file.

In [1]: a, d, dx = AnalyzeAPK("path_to_apk", decompiler="dad")

Get the activities:

a.get_activities()

Get the permissions:

a.get_permissions()

Show which classes and methods use each permission:

show_Permissions(dx)

Get the AndroidManifestXml:

a.get_android_manifest_xml().toxml()

View the smali code of a method of a class:

d.CLASS_xxx.METHOD_yyy.show()

Instead of smali, the Java source code can be listed with:

d.CLASS_xxx.METHOD_yyy.source()

Get method information:

```
for meth in d.CLASS_xxx.get_methods():
    meth.show_info()
```

Get files in the APK:

```
a.get_files()
```

Get resources:

```
aobj = a.get_android_resources()
aobj.values
```

Get a string resource value:

```
pkg = aobj.packages.keys()[0]
aobj.get_string(pkg, 'resource_key')
```

Decompile with Androguard

```
androdd.py -i <APK_FILE> -o <OUTPUT_DIRECTORY>
```

« PREVIOUS POST