Q

THE SH3LLCOD3R'S BLOG

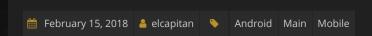
HOME CONTACT CTF WALKTHROUGHS EXPLOIT DEVELOPMENT MOBILE SECURITY NETWORK

SECURITYTUBE - LINUX ASSEMBLY EXPERT 32-BIT SECURITYTUBE - OFFENSIVE IOT EXPLOITATION SECURITYTUBE EXAMS

CISCO EMBEDDED

Home / Android / Main / Mobile / Androguard usage

Androguard usage



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

Install Androquard

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

This blog is dedicated to my research and experimentation on ethical hacking. The methods and techniques published on this site should not be used to do illegal things. I do not take responsibility for acts of other people.

	environment is a container and has its own installation directories for python modules.	RECENT POSTS
	which python2	Androguard usage
	On my computer this is <code>/usr/bin/python2</code> , 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 (<code>'pip install virtualenv'</code>). <code>virtualenv -p /usr/bin/python2 .vepy27</code> .vepy27 is the name of the folder, that will contain the installed python modules. Now we should activate the environment.	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
	source .vepy27/bin/activate	
		CATEGORIES
	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.	Android (5)
		Fusion (2)
	pip install androguard	IoT (13)
	pip install ipython	Main (3)
		Mobile (6)
		Protostar (24)
	When you finished your work, simply deactivate the virtual environment with:	SLAE32 (8)
	deactivate	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 small code of a method of a class: d.CLASS_xxx.METHOD_yyy.show() Instead of small, the Java source code can be listed with: d.CLASS_xxx.METHOD_yyy.source()



