IoT Smartphone Application Analysis of GoatDroid

Preston Chapman

NTW233 - IoT Architecture and Security

University of Advancing Technology

Briant Becote

Sunday, July 13, 2025

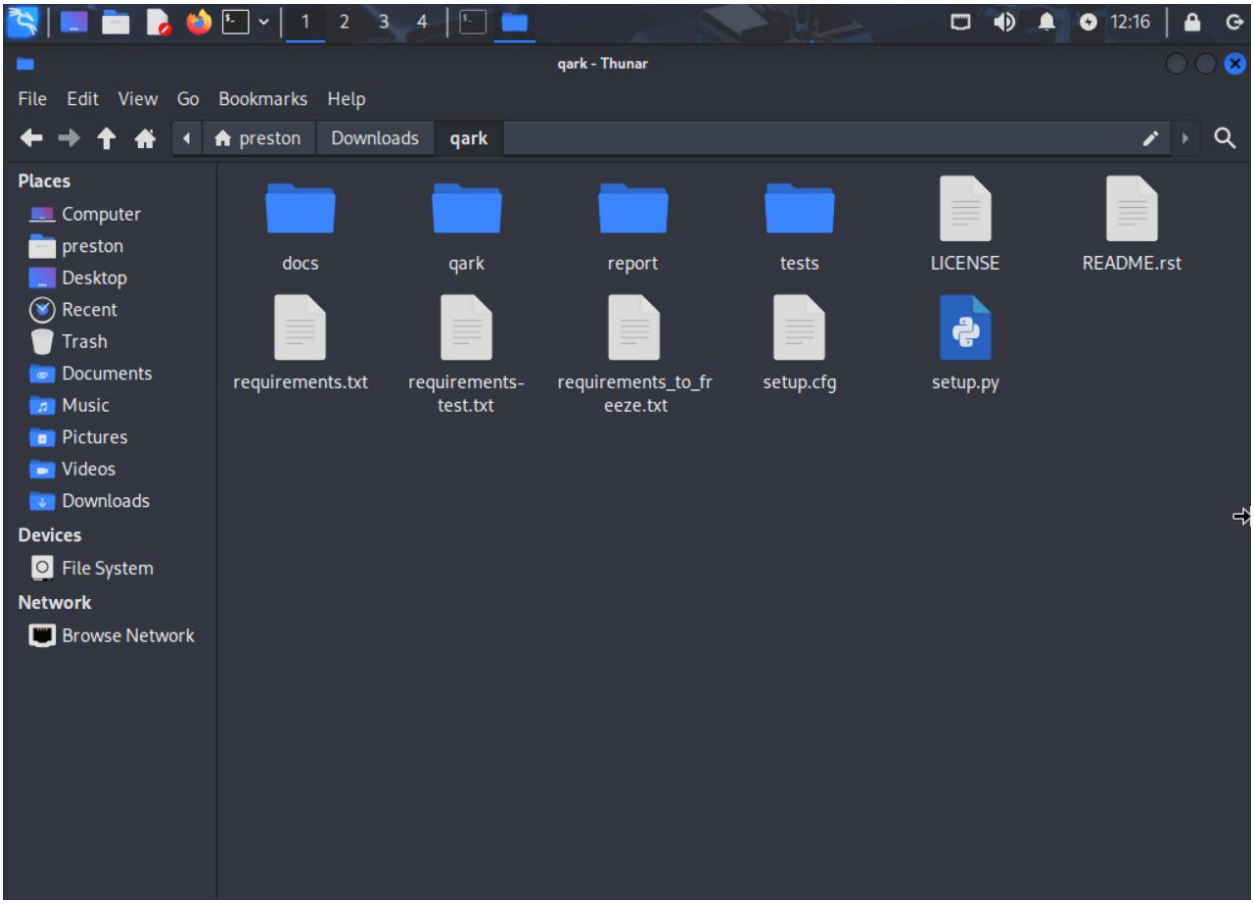
Summary

In this assignment we were tasked with reverse engineering an IoT application, instead of working with our existing device we were tasked with using Goat Droid. Please note that everything in this document is for educational purposes only and should not be replicated.

Setup

For this assignment I decided to use Hyper-V to manage my virtual machine, the machine I am using is a Kali Linux machine, with the default settings and programs. I specifically needed Python and PIP, as well as Git, these are needed to run qark, and to install it, respectively. Once the Kali Linux machine was installed, I ran the following command in the terminal: "git clone https://github.com/linkedin/qark.git", this creates a copy of the qark repository onto the Linux machine.

```
┌──(preston㉿kali)-[~/Downloads]
└─$ git clone https://github.com/linkedin/qark.git
Cloning into 'qark' ...
remote: Enumerating objects: 9314, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9314 (delta 4), reused 1 (delta 1), pack-reused 9306 (from 2)
Receiving objects: 100% (9314/9314), 52.16 MiB | 4.41 MiB/s, done.
Resolving deltas: 100% (2809/2809), done.
```



Once finished I navigated to the root of the project and then ran the following command: "sudo python3 setup.py install" doing so will install all the libraries needed for this project.

```
┌──(preston㉿kali)-[~/Downloads/qark]
└─$ sudo python3 setup.py install
[sudo] password for preston:
/usr/lib/python3/dist-packages/setuptools/dist.py:759: SetuptoolsDeprecationW
arning: License classifiers are deprecated.
!!

        ********************************************************************
**********
        Please consider removing the following classifiers in favor of a SPDX
 license expression:

        License :: OSI Approved :: Apache Software License

        See https://packaging.python.org/en/latest/guides/writing-pyproject-t
oml/#license for details.
        ********************************************************************
**********
```

```
Adding certifi 2025.1.31 to easy-install.pth file

Using /usr/lib/python3/dist-packages
Searching for urllib3==2.3.0
Best match: urllib3 2.3.0
Adding urllib3 2.3.0 to easy-install.pth file

Using /usr/lib/python3/dist-packages
Searching for idna==3.10
Best match: idna 3.10
Adding idna 3.10 to easy-install.pth file

Using /usr/lib/python3/dist-packages
Searching for charset-normalizer==3.4.2
Best match: charset-normalizer 3.4.2
Adding charset-normalizer 3.4.2 to easy-install.pth file
Installing normalizer script to /usr/local/bin

Using /usr/lib/python3/dist-packages
Searching for MarkupSafe==2.1.5
Best match: MarkupSafe 2.1.5
Adding MarkupSafe 2.1.5 to easy-install.pth file

Using /usr/lib/python3/dist-packages
Finished processing dependencies for qark==4.0.0
```

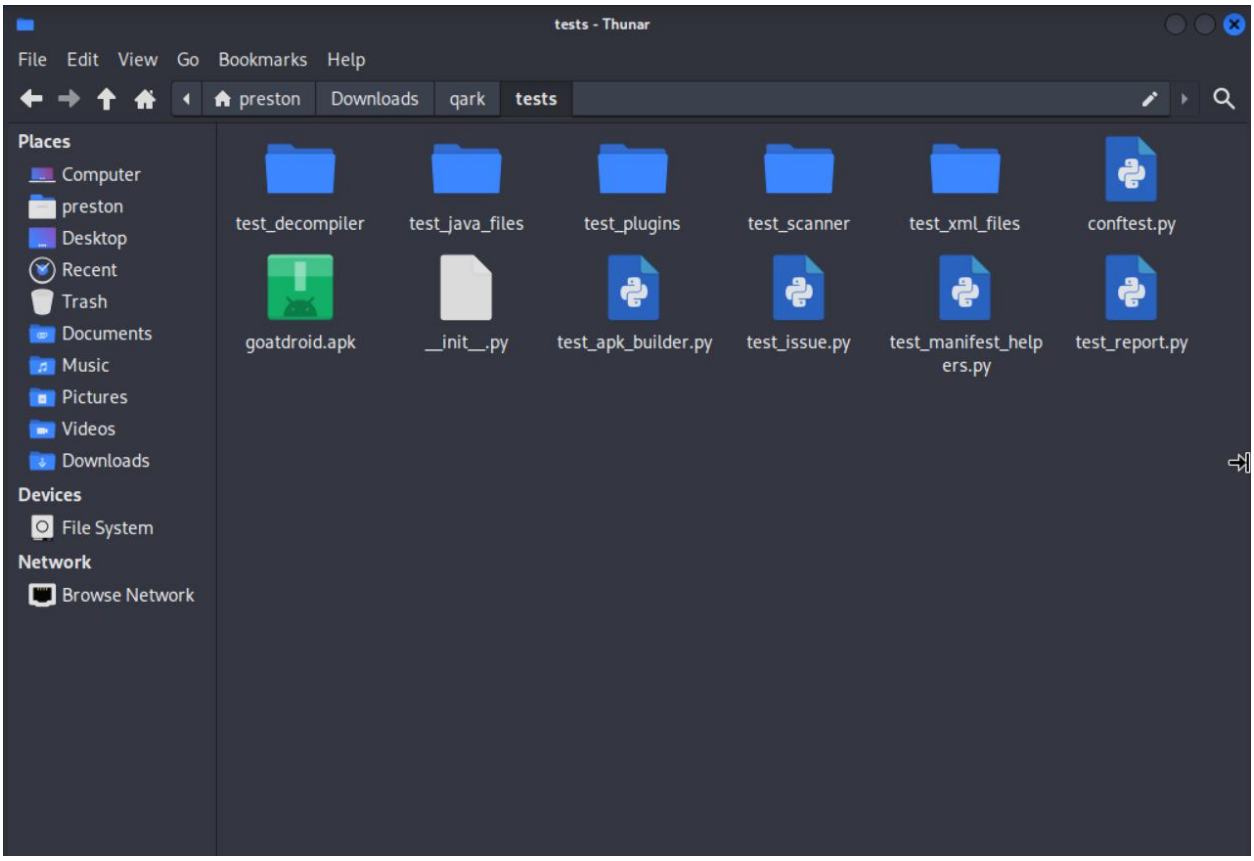
Now that it is installed, we can run the new “qark” command, which will show that the command is installed.

```
┌──(preston㉿kali)-[~]
└─$ qark
Please pass a source for scanning through either --java or --apk
Usage: qark [OPTIONS]

Options:
  --sdk-path DIRECTORY            Path to the downloaded SDK directory if
                                  already downloaded. Only necessary if
                                  --exploit-apk is passed. If --exploit-apk
                                  is passed and this flag is not
                                  passed,QARK will attempt to use the
                                  ANDROID_SDK_HOME, ANDROID_HOME,
                                  ANDROID_SDK_ROOT environment variables
                                  (in that order) for a path.
  --build-path DIRECTORY          Path to place decompiled files and
                                  exploit APK.  [default: build]
  --debug / --no-debug            Show debugging statements (helpful for
                                  issues).  [default: no-debug]
```

Usage

Now that the Qark is installed, we can navigate to the tests folder, where we will see the "goatdroid.apk" file.

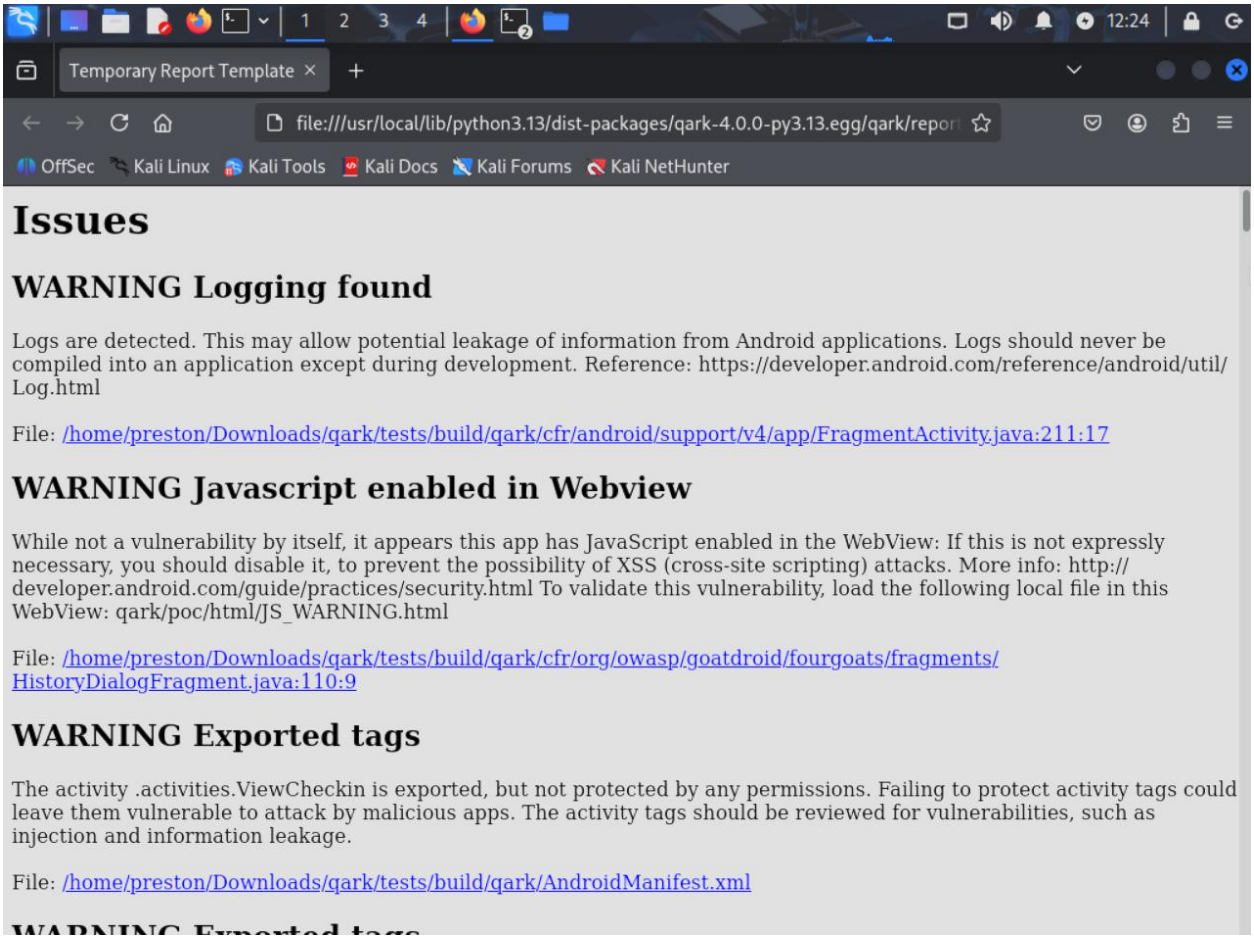


We can now open a new terminal in this directory, or copy the path to use elsewhere, and run the following command: “sudo qark –apk goatdroid.apk,” this will decompile the APK file for this app. Afterwords generating a report in this location: “/usr/local/lib/python3.11/dist-packages/qark-4.0.0-py3.11.egg/qark/report/report.html”

```
┌──(preston㉿kali)-[~/Downloads/qark/tests]
└─$ sudo qark --apk goatdroid.apk
[sudo] password for preston:
Decompiling ...
dex2jar /home/preston/Downloads/qark/tests/build/qark/classes.dex → /home/pr
eston/Downloads/qark/tests/build/qark/goatdroid.jar
```

```
Running scans ...
Finish scans ...
Writing report ...
Finish writing report to /usr/local/lib/python3.13/dist-packages/qark-4.0.0-p
y3.13.egg/qark/report/report.html ...
```

We can now open this HTML report file in Firefox to see the results.



Analysis

Qark has now provided us with an in-depth analysis about basic vulnerabilities when it comes to this application, most of these are not too important, such as logging being enabled, and JavaScript allowed in WebView's, these often can be beneficial but can result in vulnerabilities if they were not handled properly. XSS, also known as Cross Site Scripting, is a vulnerability where the user can load their own scripts, often from other websites, and input them into the application or website. Having JavaScript enabled in this way can be good, especially for debugging and development, but most of the time when it comes to a released product this is not

good and can allow for a multitude of attacks. These attacks can range from modifying user information, elevating existing permissions, and more. Although this is not a direct example of the JavaScript size of XSS, these methods will get equivalent results that you will find in the Try Hack Me room “Authentication Bypass” (2021).

References

Ashishb. (n.d.). GitHub - ashishb/android-security-awesome: A collection of android security related resources. GitHub. https://github.com/ashishb/android-security-awesome

Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution. (2025, June 13). Kali Linux. Retrieved July 13, 2025, from https://www.kali.org/

TryHackMe | Cyber Security Training. (2021, August 17). TryHackMe. Retrieved July 13, 2025, from https://tryhackme.com/room/authenticationbypass

Types of XSS | OWASP Foundation. (2022). https://owasp.org/www-community/Types_of_Cross-Site_Scripting

Types of XSS (Cross-site Scripting). (2025, February 19). Acunetix. https://www.acunetix.com/websitesecurity/xss/