Application Security

Team Hotel, testing team Juliette

Security tests results

1. **Automated-security-helpe**r has not found anything. In the case of Android Applications written in Java/Kotlin this tool is not useful, because it doesn't verify any code written in Java/Kotlin. It was used only for searching for git secrets. It's the same job as the **gitleaks** tool performed.

   Although report suggests there is a problem with git ownership (this problem does not appear locally on my machine so must be related to Docker):
   ##
   Start of /home/pitu/studia/as-mob/testing/XtremeDnotes//git_report_result.txt
   ##
   fatal: detected dubious ownership in repository at '/app'
   To add an exception for this directory, call:

   git config --global --add safe.directory /app
   ##
   End of /home/pitu/studia/as-mob/testing/XtremeDnotes//git_report_result.txt
   ##
   It doesn't affect test results.

2. **Betterscan-ce** has found only warnings. Report is attached as a file in git repository as it's too long to be put here - see *betterscan.txt*

3. **SpotBugs** - by itself, the GUI doesn't work inside Android Studio, but checking the Gradle build resulted in a detailed security overview. We will not elaborate further about results, as **SonarQube** (point 7.) gave a similar report. It doesn't work because the GUI originates from last year's October (so it's pretty outdated). Full report available in *lint-results-debug.html.*

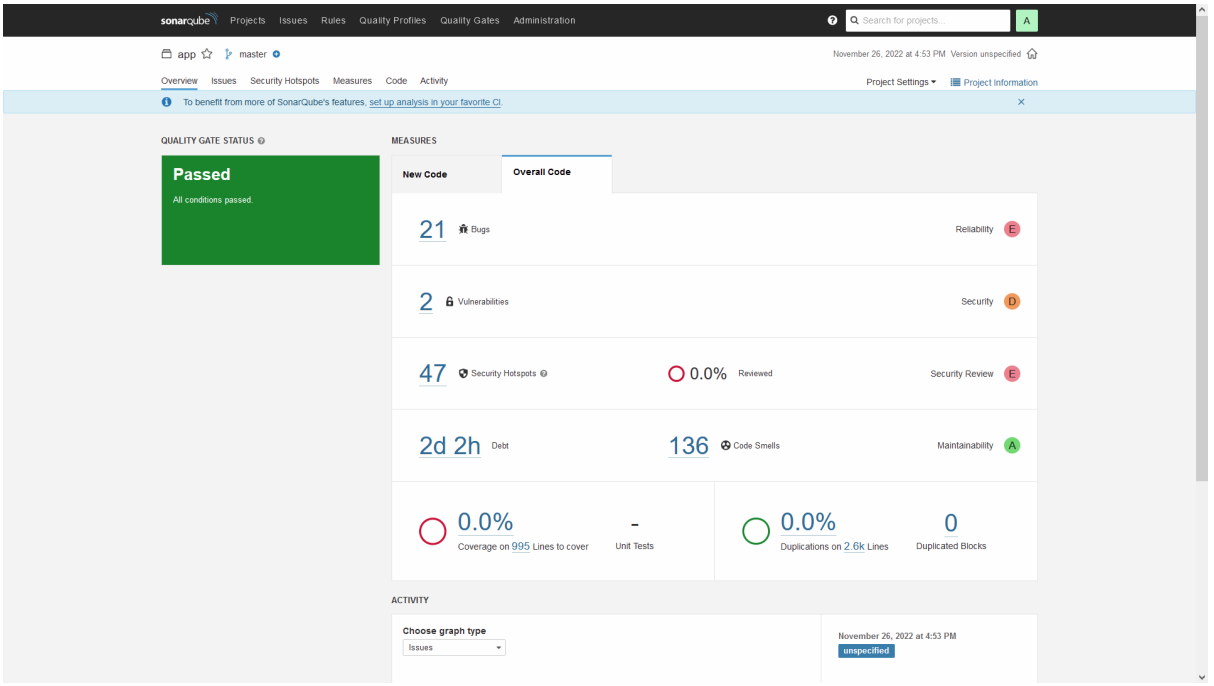
4. **Fluid Attack's Scanner** - this tool was unable to check for repository vulnerabilities due to unknown problems with repository structure. More details will be included in the attached file under name *fluids.txt.* We don't know how to use this tool and its documentation is very unclear and obfuscated (wink wink).

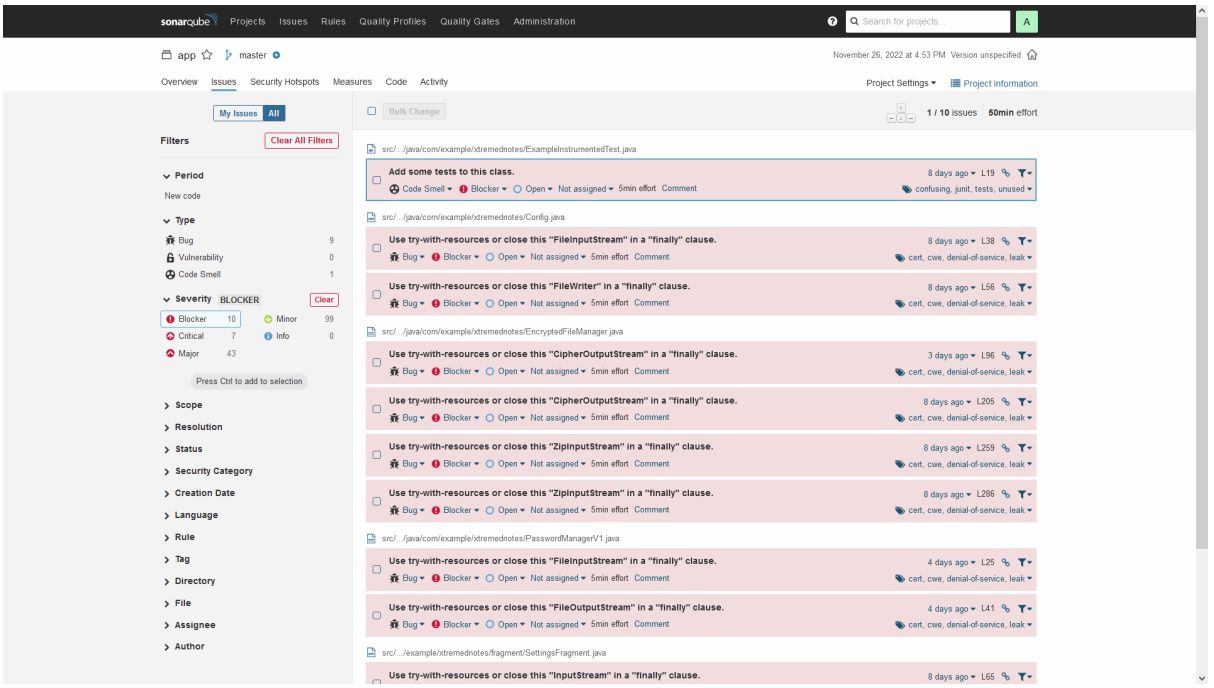
5. **Horusec** has found 33 possible vulnerabilities. Short version of report below (in git repository there's a file *horusec.txt* with full report). The most important vulnerabilities are related to potentially hard-coded passwords and weak block mode for Cryptographic Hash Function. Other 20 potential HIGH level severity vulnerabilities are about usage of *javax.crypto* library which is necessary and can't be replaced. Usage of Streams, Base64 and IV vector is also potentially vulnerable.

```
548 ================================================================================
549
550 In this analysis, a total of 33 possible vulnerabilities were found and we classified them into:
551 Total of Vulnerability CRITICAL is: 1
552 Total of Vulnerability HIGH is: 22
553 Total of Vulnerability MEDIUM is: 6
554 Total of Vulnerability LOW is: 4
555
556 ================================================================================
557
```

6. **Mobile Security Framework** - we used an online version available at https://mobsf.live/static_analyzer/?name=XtremeDnotes-master.zip&checksum=5d0a6b0123731bdfdd504b31d7b869e7&type=zip#activities. Overall, the application's result is 54/100, that means the level of risk is medium. According to the report, the biggest risk is permission **WRITE_EXTERNAL_STORAGE**. However this is a key feature of the application so it cannot be avoided. On the other hand, the report indicates a possibility for hard coded secrets, passwords, etc. but after code inspection it turns out that it's not true. Another suggestion of the report is to disable the possibility of creating backups in the Manifest file.

7. **SonarQube SCA** - this tool found 21 bugs, 2 vulnerabilities, 47 security hotspots and 136 code smells. The biggest issue in the code is improper handling of streams. Streams are surrounded by a try-catch clause however they are not closed in the final block (which a must have in case of any failure). Other major bugs are related to NPE. Security hotspots inform about the need to make sure that given function are implemented correctly and securely. Two vulnerabilities are about the same issue. Authors implement a static IV, which should be dynamically generated. Also authors left plenty of unused imports which are marked as a Code Smell and they should be removed. Other found issues are mainly skipping default case in switches and in two cases lack of static mark for properties.





8. **Gitleaks** has not found any hard coded passwords, API keys or tokens in the project's repository.

```
pitu@pitu-os:~/studia/as-mob/testing$ docker run -v /home/pitu/studia/as-mob/testing/:/src zricethezav/gitleaks:latest detect --source="/src" -r /src/result.json

    gitleaks

11:59AM INF 0 commits scanned.
11:59AM INF scan completed in 75.3ms
11:59AM INF no leaks found
```