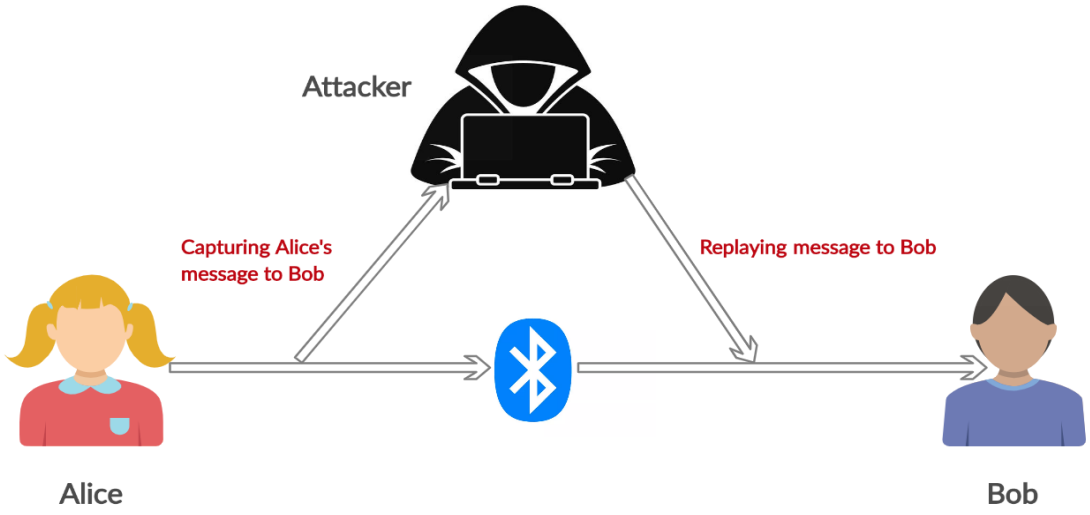
Weekly Blog 2

What have I done this week?

This week, I focussed mainly upon the security risks, cryptographic solutions of Exposure Notification API, prevention of relay attacks, and learning more about MobSF for testing, static analysis, code analysis.

Some of the most common security attacks which are found in contact tracing apps are-

1. Relay/Replay attack
2. Enumeration attack
3. Denial of service
4. Deanonymization attacks



Replay/relay attacks are the most common and difficult to tackle. It is possible to prevent these attacks by a mitigation strategy that involves storing the timestamp of a received proximity identifier and later matching the timestamp to the time interval number found for the proximity identifier but there is a fundamental flaw in this approach, there is a lot of identifiable information stored which is going to compromise the privacy.

There is another effective solution, modifying the data argument of RPI to include current location cell number can prevent relay attacks.

Normal RPI key is defined as-

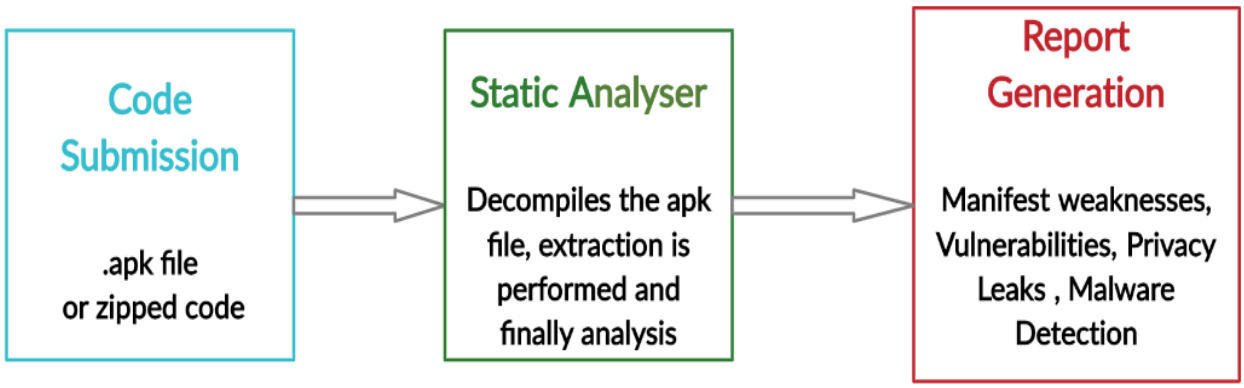
$RPI_{i,j} \leftarrow Truncate(AES(dtk_i, (UTF8("CT\text{-}RPI")||TIN_j)), 16)$

Modified RPI key

$RPI_{i,j,k} \leftarrow Truncate(AES(dtk_i, (UTF8("CT\text{-}RPI")||TIN_j||LCN_k)), 16)$

The modification limits the scope of the attack within the location cell. Moreover, it has no bearing on privacy, since an attacker who receives such a proximity identifier already knows his or her current location (to some granularity), just like they know the current time. The device associates private encryption of the location cell with the received proximity identifier its stores. This prevents the relay attack

Testing – For testing, MobSF can be used, as it is an automated, open-source, all-in-one mobile application (Android/iOS/Windows) pen-testing framework capable of performing static, dynamic, and malware analysis. It is suggested by OWASP MSTG for static analysis of security in mobile applications.



Code analysis :

- Analysis result of java code by a static analyzer.
- Identifies potential vulnerabilities, determines their severity & the files in which this type of vulnerability was found.

CVSS :

- Common Vulnerability Scoring System

- Vulnerability is assigned a CVSS base score between 0.0 & 10.0.

0.0 → No risk

0.1–3.9 → Low risk

4.0–6.9 → Medium risk

7.0–8.9 → High risk

9.0–10.0 → Critical risk score

For added security, some other measures can be used;-

1) Regression Testing
2) Usability Testing
3) Performance Testing

Paper Review 1

Yaron Gvili, "Security Analysis of the COVID-19 Contact Tracing Specifications by Apple Inc. and Google Inc. “Cryptology ePrint Archive, Report 2020/428

https://eprint.iacr.org/2020/428.pdf

Paper Summary

https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/wiki/LOKESH-PATHAK---Security-Analysis-of-the-COVID-19-Contact-Tracing-Specifications-by-Apple-Inc.-and-Google-Inc.

What research questions does the paper address?

1) The primary research question addressed in the research paper is to find out the information and system security attacks that can be performed on the contact tracing ability developed by Google Inc. and Apple Inc.

2) Also, the author wants to evaluate the consequences of these security vulnerabilities on the efficiency of the system and also suggest measures to mitigate them as far as possible.

3) Another important research question is to demonstrate the techniques that can be employed to enable efficient location cell determination

What was the conclusion of the research?

The main conclusions of the research paper were to show that the contact tracing technology developed by Google and Apple, has been designed to keep data security and data privacy on priority, yet there are several system security and information security risks that can be found in the system. Some of the common system security risks are power and storage drain attacks, relay, and replay attacks. Similarly, some of the commonly encountered information security attacks are tracking and deanonymization attacks, linking attacks, etc. The author also concluded that most of the attacks can be mitigated effectively by adopting measures that do not require major changes to the specification, for example introducing a public key or adopting bidirectional communication instead of unidirectional.

How can you apply this knowledge to your project? How can you apply this knowledge to your project?

As our solution is based on a decentralized approach using the exposure notification API, this research paper gives many suggestions to adopt, so that security and privacy can be offered as default by design. Also, it shows that the exposure notification API has a lot of scope for improvement, and to develop an effective and trustworthy application, all the system security attacks and information security have to be taken into consideration. Also, most of the possible attacks can be reduced if the system is executed at the operating system level. Usually, more stress is given on information security but in the research paper, the research showed effectively how relay/replay attacks, power, and storage drain attacks can leave the whole system useless by making it impossible for the application to work normally. This has given us the motivation to be extra careful while designing the architecture and specifications of the system so that the exposure notification app can work as intended.

Paper Review 2

Baumgärtner, Lars & Dmitrienko, Alexandra & Freisleben, Bernd & Gruler, Alexander & Höchst, Jonas & Kühlberg, Joshua & Mezini, Mira & Miettinen, Markus & Muhamedagic, Anel & Nguyen, Thien & Penning, Alvar & Pustelnik, Dermot & Roos, Filipp & Sadeghi, Ahmad-Reza & Schwarz, Peter Michael & Uhl, Christian. (2020). Mind the GAP: Security & Privacy Risks of Contact Tracing Apps.

https://arxiv.org/pdf/2006.05914.pdf

Paper Summary

https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/wiki/LOKESH-PATHAK-Mind-the-GAP:-Security-&-Privacy-Risks-of-Contact-Tracing-Apps.

What research questions does the paper address?

Some of the research questions tackled in this research paper are:-

1) The researchers intend to provide empirical evidence in real-world scenarios for two crucial risks found in GAP (Google Apple Protocol): one concerning privacy i.e. Profiling infected persons, and the other one concerning security i.e. Wormhole Attack, which is a kind of relay attack.
2) To show how there is a trust deficit in users regarding the contact tracing applications, the researcher also intends to find out the benefits

of having a full software stack (app code, GAP API code, and backend-server code) as open source.

3) To evaluate the effects of deploying countermeasures to address the aforementioned risks on the privacy and security properties of the application.

What was the conclusion of the research?

With the help of real-world experiments the author of the paper, showed us that the current GAP design is vulnerable to (i) profiling and possibly de-anonymizing infected persons, and (ii) relay-based wormhole attacks that can generate fake contacts that can severely hamper the accuracy of the system. For example, the attacker can use a large number of collected GAP Rolling Proximity Identifiers to make a victim appear to have had a high number of contacts, ultimately resulting in a high probability of an infected contact and a restriction of the user's freedom, since he or she must most likely follow the instructions of self-quarantine. Besides, an (unnecessary) medical test for infection is likely to be performed, which puts a strain on the available test capacities. Finally, the author has proved, why a revision in the GAP protocol is important to increase its wide-scale acceptability. There is a need to deploy countermeasures without compromising on security and privacy and using unacceptably high resource demands, such as battery consumption.

How can you apply this knowledge to your project? How can you apply this knowledge to your project?

This research paper discusses in detail about the damaging effects and user rights violations that can be done by hackers who are always ready to exploit the vulnerabilities of systems like this. This has made me more aware of the responsibilities of making an app like this and how important it is to maintain high privacy and safety standards.

Also, the project has suggested ideas to employ to prevent profiling and wormhole attacks. These include stricter data encryption standards, delayed authentication, and rigorous testing before deploying the application on the market.

Paper Review 3

Shahriar, Hossain & Talukder, Md Arabin Islam & Islam, Md Saiful. (2019). An Exploratory Analysis of Mobile Security Tools.

https://digitalcommons.kennesaw.edu/cgi/viewcontent.cgi?article=1095&context=ccerp

Paper Summary

https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/wiki/LOKESH-PATHAK--An-Exploratory-Analysis-of-Mobile-Security-Tools.

What research questions does the paper address?

Some of the research questions tackled in this research paper are:-

1) Explore the efficiency of vulnerability detection for static and dynamic analysis tools
2) Investigate how the majority of malicious mobile attacks take advantage of vulnerabilities in mobile applications, such as sensitive data leakage, faulty encryption, unsecured data storage, etc.
3) Create a detailed analysis report on the performance of the security tools on various parameters.

What was the conclusion of the research?

- After detailed analysis, it was found that flowdroid and mobsf are the most efficient for malware detection and code analysis.
- Android Studio which is the most used, android development platform does not have a usable security analysis module
- The development of a plugin or a native module for Android Studio can greatly reduce the risk of privacy invasion, and data leakage and stealth with the use of malwares and spywares.

How can you apply this knowledge to your project? How can you apply this knowledge to your project?

The research paper provides a detailed overview of various tools that can be used to perform static and dynamic analysis, malware detection on our application. Particularly MobSF, which is an automated, all-in-one mobile

application (Android/iOS/Windows) pen-testing, malware analysis, and security assessment framework. The author of the research has explained it's working by running many test cases and explained each module with great detail. The research paper was one of the most prominent factors to consider MobSF for the project.