Threat Modeling Android Apps : A Case Study

I. Introduction

Introduction will go here

II. Related Work

Forgoing early software security methodologies in the software development process exacerbates the cost and time of software projects in the long term. Research shows thinking about security early greatly reduces the time and cost required to maintain and update applications. In mobile apps, Android applications occupy 55.1 % of the mobile application market in 2019. (However,) [?] searched for several known vulnerabilities and found that 28 per cent of the studied apps had at least one of these vulnerabilities. In a related study, Bhoraskar et al. [?] mined 1010 apps from Google Play and used static analysis. They found that 13 of 200 apps using the Facebook SDK were vulnerable to known attacks. The security risks exist everywhere from the libraries used to the way Android developers structure and program their code. Wei et al [?] suggested that unlike the mature HTTPS standard and solutions we have in traditional web programming, there are gaps between theoretical HTTPS usage and in-the-wild implementation of the protocol in Android applications which exposes sensitive user data to man-in-the-middle attacks. (In addition,) an empirical study in collaborative question answering sites (e.g. Stack Overflow) [?] pointed out that security misconceptions and lack of 'usable' security libraries are key deterrent factors for developers to implement security features properly. Security vulnerabilities occur not only in the most well-known apps but also in financial apps which are essential to society and are supposed to be safe in high-security standards. Analysis of Branchless Banking Applications [?] uncovers the vulnerabilities that can manifest in current financial applications and services, where security is of utmost importance.

Security issues and risks result from inherent difficulties in Android app development. An empirical case study on StackOverflow [?] tried to explore security challenges which developers concluded were confusing and hard to implement [?]. Security tools are not widely adopted by developers [?] while the fact that programmers without the help of security tools tend to provide code with vulnerabilities suggested by Xie et al [?].

Static analysis tools are not a panacea for developers [?]. In reality, static analysis tools that are widely used can fail to provide reliable insights for developers to identify and make corrections to the potential vulnerabilities in their code [?], [?], which is also the main reason that these tools are not widely adopted in Android programming [?], [?].

Threat modelling is a common approach in the field of security which involves risk-ranking or attempts to estimate resources, capabilities or motivations and includes techniques such as DFD, STRIDE and countermeasures [?]. Threat modelling can be used as foundations for the specification of security requirements [?]. There are several studies focus on applying threat modelling in different areas [?]. Paper [?] proposes how developers can use a threat modelling game during Agile software development, and Tymchuk et al [?] examines an IDE feature implemented to provide static analysis during the coding process. Shostack et al [?] utilized the lessons learned during the process at the Microsoft software company to present the acceptance of threat modelling in the industry. Kang et al [?] published an example where the concept is applied to smart bands. In the network area, Akash et al [?] viewed the security framework built from applying threat modelling to the web application layer, and Myagmar et al [?] published an approach outlined for identifying threats of networked systems. By enumerating the threats to a system, threat modelling helps programmers to develop realistic security requirements [?] and contributes to developers in following a "Security by Design" approach [?].

However, prior work does not provide a practical methodology targeting on the programming process to help improve the security in Android app development. Our work specifically focuses on this open area of research and provides Android security analysis as well as an open-source tool [?] to help developers do threat modelling [?] of their source code. In doing so, programmers gain the ability to review their code from a security perspective, even if they don't have the necessary background.

III. Background

```
- Threat modeling in general
- Developer centric threat modeling
- Limitation of static analysis in
 Android App Analysis
```

IV. Discussion and Conclusion

References