OWASP Risk Analysis Driven Security Requirements Specification for Secure Android Mobile Software Development

Kai Qian, Reza M. Parizi, Dan Lo
College of Computing and Software Engineering, Kennesaw State University, Marietta, GA, USA
{kqian, rparizi1, dlo2}@kennesaw.edu

***Abstract* — The security threats to mobile applications are growing explosively. Mobile apps flaws and security defects open doors for hackers to break in and access sensitive information. Defensive requirements analysis should be an integral part of secure mobile SDLC. Developers need to consider the information confidentiality and data integrity, to verify the security early in the development lifecycle rather than fixing the security holes after attacking and data leaks take place. Early eliminating known security vulnerabilities will help developers increase the security of apps and reduce the likelihood of exploitation. However, many software developers lack the necessary security knowledge and skills at the development stage, and that's why Secure Mobile Software Development education is very necessary for mobile software engineers. In this paper, we propose a guided security requirement analysis based on OWASP Mobile Top ten security risk recommendations for Android mobile software development and its traceability of the developmental controls in SDLC. Building secure apps immune to the OWASP Mobile Top ten risks would be an effective approach to provide very useful mobile security guidelines.**

Keywords– Security requirements, OWASP, SDLC, Mobile apps, Android development

I. INTRODUCTION

Smartphones have become a mine of personal information, holding bank account data, credit card information, emails and addresses, making them the preferred target for cybercriminals, experts warn. "Cybercriminals go where there is value, and they have understood that the smartphone has become the preferred terminal for online shopping and payment" [1]. Cybercriminals have progressed from smartphone ransomware attacks to using Trojan Horse malware that can steal the login credentials of mobile banking users. There was a 17 percent increase in attacks targeting banking apps last year around the world [1]. The mobile apps must protect the confidentiality and integrity of data at rest and data in motion. As the time-to-market pressures for mobile app development are increasing, its development cycle is getting shorter. As a result, many mobile app developers overlook the security quality of the software in the development cycle. Mobile app flaws and security defects could open doors for hackers to lunch sophisticated attacks. Early elimination of possible security vulnerabilities will help to secure the software, and mitigate the security risk threats from potential malicious attacking. However, many software engineers lack the necessary security knowledge and skills at the development stage. For example, protection for mobile data at rest and on the move is extremely important for the software confidentiality and integrity, but it is often observed in the breach. Any mobile software must prevent sensitive data loss via cache buffers, in data storage, and during data transfer. Therefore, mobile software needs security requirements of the strategic development process to prevent potential risks and improve the security quality.

Security requirements specify traceable security goals that a mobile app must provide, and the project team including designers, developers, and QAs must meet these goals in the design, implementation, testing, and deployment of the product. To build security requirements, we must specify security needs and knowledge of the vulnerabilities that could lead to security attacks or confidential data leaks. This work aims to derive a set of unified security requirements and its defensive developmental controls based on the OWASP [2] most top critical mobile threats and risks.

II. RELATED WORK

Security threats are growing unabatedly, particularly in m-commerce domains. To combat the rising of security risks, many software engineers acknowledge the risks and analyze security with abuse case or misuse case for security requirements at the start of system development [3-4], [6]. Security threat modeling [5] is also applied in security requirements specification process. Security requirements process has become a vital part of SDLC. Inspired by the success of the security threat modeling, we propose our OWASP [2] guided security requirements analysis specific to Android Mobile Software Development and its related traceable development activities including secure design principle, good code practice, and testing approaches.

III. OWASP GUIDED SECURITY REQUIREMENTS ANALYSIS FOR ANDROID APPS DEVELOPMENT

There are majorly 3 types of Android Mobile Apps:

1. Native apps - Java APIs access all Android device features, such as the mobile SD, camera, and geolocation.
2. Web apps - purely run with the phone's browser such as mobile booking and banking.

978-1-5386-5790-4/18/$31.00 ©2018 IEEE

3. Hybrid apps - a combination of native app and Web app such as BYOD.

Almost all types of mobile apps directly access to personal data, where the safety includes security for data at rest, data in in motion and device platform security. Data protection is one of the most security-preserving requirements for any mobile app especially for the financial apps, healthcare apps, and business apps. When we decide what a software will do, we also need to know what it should not do. When we specify security requirements, we must understand threat models and security vulnerabilities because that aid secure system design, secure development and testing. Knowing how to think like an attacker is an asset, and it should be the mindset of requirements analysts.

Table I shows the threat risk analysis driven security requirements and its defensive major development's activities in secure SDLC including design principles, code implementation, and testing.

TABLE I. THREAT RISK DRIVEN MOBILE SECURITY REQUIREMENTS AND ITS DEFENSIVE DEVELOPMENT'S ACTIVITIES

OWASP threats	Security Req. (R)	Design (D)	Implementation (I)	Testing (T)
1.Improper Platform Usage **Issues**: Compromise device system security and IPC communication	Ensure platform security controls Secure IPC communication	Secure Android component design patterns for intent communication Protection of Rooting to prevent from viruses and malwares infection	Secure IPC intra-app with protection of explicit intent IPC and proper setting of exported attribute Secure inter-app programming with permission Avoid coding with root privilege	Static Code Analysis (SCA) with Android FindSecurityBugs detectors for direct root access, intent spoofing, intent eavesdropping (in Android Studio), FlowDroid, DroidSafe Or Dynamic Code Analysis(DCA) such as TaintDroid
2. Insecure Data Storage and Unintended Data Leakage **Issues**: Loss of Confidentiality and Privacy for data at rest	Identify and protect sensitive data on the mobile device with data encryption Protect caching memory data	Apply the principle of minimal data disclosure: File encryption Check API calls to sensitive data Store sensitive data on server instead of device Protect possible leakage via channel caches, temp storage, contacts, and multimedia (Temp sensitive data is limited to a max. retention period) Remote wipe data on device for theft or loss	Conform the secure design rules for coding such as avoid direct access data on local storage unless the data is encrypted and protected, Avoid sensitive data in temp storage too long.	Android Debug Bridge (adb) SCA with Android FindSecurityBugs , FlowDroid, DroidSafe detectors for direct root access Or DCA such as TaintDroid, Inspeckage
3. Insecure Communication **Issues**: loss of Confidentiality and integrity by data tampering	Protect data in motion	Avoid clear text communication of sensitive assets	Programming with SSL/TLS	SCA with Android ComDroid, AmaDroid, FindSecurityBugs , FlowDroid, DroidSafe detectors for inline string or binary data or DCA such as TaintDroid, Inspeckage
4. Insecure Authentication, Authorization **Issues**: Unauthorized access and privilege escalation	Protect access controls	Avoid using device ID or subscriber ID as sole immutable authenticator Executing an API endpoint - Rest service	Implement access token or signature/system permission whenever necessary Implement API endpoint with Service interface whenever necessary	SCA with Android FindSecurityBugs , FlowDroid, DroidSafe detectors for explicit ID exposure Or DCA such as TaintDroid, Inspeckage
5. Insecure Mobile Input Injection **Issues**: SQL injection and XSS client-side flaws, lead to unauthorized access to sensitive data	Prevent malicious code Input injection with input validation Protect output reddening via XSS attacks	White list and Black list Input validation Output filtering	Proper input validation checking Proper output escaping/encoding against XSS attacks	SCA with Android FindSecurityBugs , FlowDroid, DroidSafedetectors for input injection and output Or DCA such as TaintDroid, Inspeckage
6. Reverse Engineering and Code Tampering **Issues**: Credentials disclosed and Intellectual property exposed	Prevent reverse analysis on source code and assets from its binary core Prevent binary patching, local resource, memory modification, method hooking	Keep proprietary and sensitive business logic on the server No hardcoding password	Implement anti reverse engineering attacks with Anti-reverse engineering controls approaches of AntiDebug, Checksum, Renaming, Obfuscation and others for sensitive code and data whenever necessary	Use Java decompiler tools for Apk and Dex Android files such as Jadx to check the possible disclosure of credentials and Intellectual property, Radare

IV. CONCLUSION

Security requirements specification is an important process which must be incorporated into mobile apps SDLC. We proposed an OWASP-based security requirements analysis specific for Android mobile software development and its defensive development's activities.

ACKNOWLEDGMENT

The work is partially supported by the National Science Foundation (NSF) under awards: 1723578, 1623724.

REFERENCES

[1] T. de Coatpont, Cybercriminals are increasing their attacks on smartphones, easy data-rich targets, http://www.firstpost.com/ tech/news-analysis/cybercriminals-are-increasing-their-attacks-on-smartphones-easy-data-rich-targets-3698697. html, 2017.

[2] Mobile Top 10 Security Threat Risk, www.owasp.org/index.php/ Mobile_Top_10_2016-Top_10, 2017.

[3] J. Boote, Are you making software security a requirement? https://www.synopsys.com/blogs/software-security/software-security-requirement/, 2016.

[4] Get Started with Software Security Requirements, https://www.sdelements.com/media/pdf/software-security-requirements-5-steps.pdf, 2013.

[5] Threat Risk Modeling, https://www.owasp.org/index.php/Threat_Risk_Modeling, 2017.

[6] R. M. Parizi, K. Qian, H. Shahriar, F. Wu, and L. Tao, Benchmark Requirements for Assessing Software Security Vulnerability Testing Tools, *In IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 825–826, 2018.