Mobile Application Threat Modeling Transcript

Chimene Tchokoko Diboma

School or Institution Name (University at Place or Town, State)

**Mobile Application Threat Modeling Transcript**

**Introduction:**

Advancements made in communication and information technologies have changed the way people do business. Mobile phones are turned into powerful computing devices. A modern smartphone is capable of handling intensive computing tasks. People rely on mobile devices for their digital needs. Depending on the exponential penetration of mobile devices in our lives, businesses are going mobile friendly as well. Most of the businesses have developed mobile applications to provide their customers with customized services. However, extensive growth in the mobile application development market has raised concerns as well. Mobile applications of major business outlets process personal information of their users making them a potential target of cyber-criminals (Roy et al., 2019). The cyber threat landscape is shifting from server or mainframe computer markets to mobile applications. Secure mobile applications cannot be developed without appropriate modeling of threats posed to mobile applications. The report provides an insight into threats available to mobile applications and secure application development techniques.

**Mobile Application Architecture:**

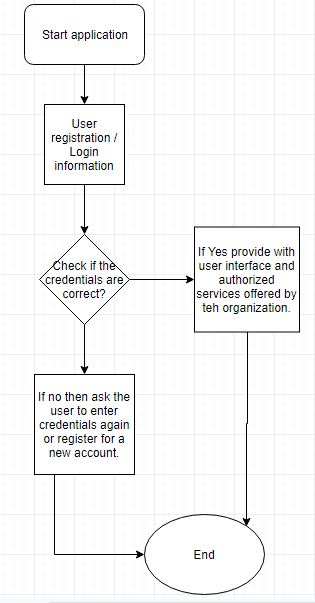
In mobile application development, appropriate choice of application architecture plays the central role. As a cyber-threat analyst, I have evaluated the architecture for mobile application for a business that provides retail items to end users at their doorsteps. The application is used by the customers to place an order. The details of the order are collected by the application and processed by the backend corporate information technology infrastructure. Before, designing a mobile application, the developers must be aware of the fact that who will be using that application (Schliep & Hopper, 2018). If the application is intended for business customers such as online retail store customers, then the application must provide limited functionality to the end user. It is important to hide extra features from client-side applications as novice users can be confused with technical features. The application must be simple and provide core functionality on mobile devices without bugs.

The application must be compatible with the mobile platform such as it should be compatible with the host device. Mobile applications are platform dependent. For example, applications developed for iOS (Proprietary mobile operating system of Apple) cannot be used on Android-powered devices. It is more useful to develop separate applications for different operating systems. Each of the operating systems will have different threat landscapes. Android is the most popular mobile operating system and due to the open source nature of operating system cybercriminals have all access to core operating system functions (Kaur & Kaur, 2019). On the other hand, iOS developed by Apple is a closed source operating system and threat surface for this operating system is very small as compared to that of the Android operating system. Apple uses built-in device encryption and strong privacy controls to make the iOS ecosystem more secure as compared to rivals. Therefore, operating system dependent mobile applications must have different characteristics for both of the platforms while keeping the core functionality the same.

Android devices do not offer built-in encryption mechanisms as compared to iOS devices. Android application version of the application must not store critical information on the user device. To create platform independence up to some extent, client-server application architecture can be used for mobile application development. In this architecture, the mobile application will use internet connection wither cellular networks such as 3G or 4G connection or available WiFi connection to communicate with the backend server of the company. The application may interact with other applications installed on the device such as Camera. Inter-app connectivity must be refined by testing the application in real-world environments. Camera access may be required to scan a bar code or QR code (Coles, Faily, & Ki-Aries, 2018). The application may use direct access to camera hardware or may be assisted with the third-party camera application. Accessing the camera directly can reduce the risk of third party spying on application operations. In case of using a third party application for camera access that third party may collect information about the use of application potentially compromising the privacy of the user and corporate network as well.

Mobile applications transferring authentication information such as usernames and passwords must use encryption algorithms to secure the transmission of sensitive information. Developers mostly deal with the application layer while developing mobile applications however, corporate applications dealing with sensitive information must support transport layer security such as SSL certificates. Third party application API’s must be integrated using verified software development kits. Open source code available for API integration can reduce development overhead and provides more efficient integration of application operations. Data stored by the application or accessed by the application will be as secure as the host device itself. If the host device is compromised with malicious code then the logical measures to secure local data can fail miserably (Atwater & Goldberg, 2018). Depending on the popularity of mobile applications, they are the most attractive target of cybercriminals. Cybercriminals can access sensitive information by attacking mobile application rather than compromising large-scale server architecture. For user authentication, the application must use multiple authentication factors and must not transfer sensitive information over insecure wireless channels.

**Requirements for Mobile Application:**

Mobile applications developed for different businesses will have different requirements as per the type of business. Mobile application developed for a retail business that provides end users with an online ordering system will have the requirements to collect information from the user and transmit it to the back end infrastructure for order processing. In a typical retail business application, the user will be provided with the application interface asking for registration. In case of an already registered customer, the application will ask for the login credentials. Login details will then be transmitted to the backend server of the company. After successful authentication of the user, authorized services will be provided to the customer. The communication between the mobile application and the backend server of the company will use the available network connection of the device. Following flow diagram presents a rough outline of the application login procedure.

The application will not store any user information on host device except the user identification files including session cookies. Cookies are small files used by web applications to identify users and to provide a personalized experience. Regarding authentication of the customers with the network of organization, username and password based approach are not secure enough to ensure confidentiality, integrity, and availability of the information. Passwords and authenticating credentials can be sniffed from wireless networks using packet sniffing tools (Verdecchia, 2018). Criminals can then use compromised credentials to impersonate the original user. Similar actions can be performed by hijacking the session cookies that contain the authentication functions. If a session cookie is compromised then the personalized content can be accessed by the criminal without requiring the password. Attacks to compromise cookies stored in the temporary storage of the device are known as session hijacking attacks. A hijacked session can provide unlimited access to protected content.

The authentication process in mobile applications can be improved without investing in additional hardware. Most of the modern mobile devices support biometric identification methods to secure the mobile device. Biometric identification sensors and devices embedded in mobile devices must be used by third party applications as well to perform user authentication. Multifactor authentication can pose a lower probability risk on the other side. If the application is designed to access and use the biometric hardware of the device then it may store and transfer information about the geolocation of the device and unique device identifiers (Yoran & Amoroso, 2018). Users may not want to transfer or reveal their location information but the application will not function appropriately without having access to metadata of device location. In order to support payment processing, there will be API's of payment Processor as well. These API's may serve as an attack vector for the application compromising the security and privacy of data. Most of the API integrations of payment processing mechanisms store transaction histories in the local storage of the application. Poor encryption mechanisms in Android platforms can compromise this information with relative ease as compared to the iOS platform.

**Threats and Threat Agents:**

The mobile application will provide an extreme level of convenience to customers as they will be able to place orders online for their desired items. Increased level of convenience will bring more security and privacy challenges as sensitive information such as credit card details, social security numbers are being transmitted between the devices and databases. Inaccurate storage of credentials such as authentication details including passwords, session cookies can put users at risk of data loss. If the authentication tokens, location data, usernames, and unique device identification numbers are not stored in encrypted containers will be compromised if the device is lost or stolen (Yu & Hou, 2018). Loss of these sensitive information records including application data such as debug information, and transaction histories can occur due to the poor implementation of network communication protocols. Servers of the organizations used to host application data can also act like the threat agents for mobile applications. If the server hosting the application data or the server providing services is compromised by the attackers then end-user devices will also be compromised by the same attacker.

As the application will use available network man in the middle attacks can be performed as well. A monitored wifi hotspot can provide criminals with full access of the device to server communication. Man in the middle attacks can be mitigated by using secure coding techniques and transport layer security implementation built into the application core. Another considerable risk is present for the application by the compromised host. If the device of a user is already infected with credential-stealing malware then the application will be compromised as soon as installed on the host device regardless of the platform (Venkatasen & Mani, 2018). As with the Android platform the applications are downloaded and run from the devices, attackers can use client-side injection attacks. They can inject scripts into the local interpreter process that will compromise all of the newly installed applications regardless of the vendor. Injection attacks designed for SQL databases and techniques can be fatal if the application support multiple users accounts on the same device (Navas & Beltrán, 2019). Allowing multiple user accounts in a single application is a useful feature in some cases but it can compromise the security and confidentiality of data as well.

**Methods of Attack:**

Cyber-criminals are always devising new and improved attack methods to avoid detection by the defense mechanisms of the applications and operating system of the target devices. Most popular methods of attacks used by the criminals include man in the middle attacks, phishing, XSS attacks, password sniffing, eavesdropping, malware, and monitored network attacks. All of the attacks methods used by the criminals focus on compromising the defense mechanism of applications and operating systems. Man in the middle attacks can be used to extract critical information from communication between the application and the server. Any function call that is not structured appropriately by the programmer can allow code injection into the application processes (Rodríguez-Mota, Escamilla-Ambrosio, Aguirre-Anaya, & Happa, 2018). If the device is connected to an insecure public wifi hotspot then the probability of man in the middle attacks is much greater than on a private network connection. With the availability of integrated development environments to the general public and end users, it is very easy for an attacker to design a lookalike piece of application to steal credentials known as a phishing attack. It will not be possible for the end user to differentiate between a forged version of the application and an official version of the application.

Creating phishing applications and tricking users into installing them will allow the attackers to form botnets. Botnets are networked of compromised devices that are used to carry out distributed denial of service attacks. In a distributed denial of service attacks, attackers send floods of useless traffic to a target server that cause congestion on the links. Access of legitimate users is blocked to the service in such type of attacks. Distributed denial of service attacks using mobile applications is not that much popular as compared to other types of attacks. Hijacking sessions by using compromised cookies is a popular attack method being actively used by cybercriminals (Tyagi, Sharma, Malhotra, & Khosla, 2018). Cybercriminals hijack session cookies using cross-site scripting attacks. If the application connects to a third party website to offer specific service then attackers can inject malicious java scripts in that third-party website. The injected code will be delivered to the application as part of the HTML body of the website that will then be executed by the application on the host device. The malicious javascript can be programmed to send the session cookies of the user to the attacker without generating an alert to the user. Therefore, compromised third-party resources pose serious risks to mobile application security.

**Controls:**

Threats for mobile applications can be mitigated and rendered useless by utilizing efficient security controls. The very first step in securing mobile applications is to use platform integration appropriately in mobile applications. For example, Apple devices provide a key chain storage area for application data as compared to local device storage space provided by other platforms. Applications designed for the iOS platform must use keychain data storage efficiently. Applications designed for Android and other mobile platforms such as ARM architectures must use trusted platform module chips to store encryption keys as these storage spaces are known to be tamper proof (Ngalo, Xiao, Christianson, & Zhang, 2018). Mobile applications have to communicate with the parent server for various operations. The initial communication required for authentication and authorization operation is protected using transport layer security such as SSL and TLS certificates. However, complete communication security can be provided using secure communication protocols for every outside request made on the network. Security at the transport layer level also rules out the possibility of eavesdropping attacks on sensitive user information.

Mobile applications must go through rigorous testing as per the OWASP project standards. Rushing towards the publication of application without fixing the bugs can create a disaster. An infected host platform can compromise even well secure applications as well. Therefore, platform dependent security features must also be integrated with the application framework by the developer (Kang, Kim, & Kim, 2018). There are dedicated security businesses that recognize and remove unauthorized versions of mobile applications from the internet. However, mobile application developers must sign their application using valid digital signatures. Signed applications will not only remove the risk of supply chain attacks but will also make the reverse engineering of code extremely difficult for cyber-criminals.

References

Atwater, E., & Goldberg, I. (2018). Shatter Secrets: Using Secret Sharing to Cross Borders with Encrypted Devices (Transcript of Discussion). *Cambridge International Workshop on Security Protocols*, 295–303. Springer.

Coles, J., Faily, S., & Ki-Aries, D. (2018). Tool-supporting Data Protection Impact Assessments with CAIRIS. *2018 IEEE 5th International Workshop on Evolving Security & Privacy Requirements Engineering (ESPRE)*, 21–27. IEEE.

Kang, S., Kim, H. M., & Kim, H. K. (2018). Trustworthy Smart Band: Security Requirement Analysis with Threat Modeling. *ArXiv Preprint ArXiv:1812.02361*.

Kaur, A., & Kaur, K. (2019). A COSMIC Function Points based Test Effort Estimation Model for Mobile Applications. *Journal of King Saud University-Computer and Information Sciences*.

Navas, J., & Beltrán, M. (2019). Understanding and mitigating OpenID Connect threats. *Computers & Security*.

Ngalo, T., Xiao, H., Christianson, B., & Zhang, Y. (2018). Threat Analysis of Software Agents in Online Banking and Payments. *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, 716–723. IEEE.

Rodríguez-Mota, A., Escamilla-Ambrosio, P. J., Aguirre-Anaya, E., & Happa, J. (2018). Reassessing Android malware analysis: From apps to IoT system modeling. *EAI Endorsed Transactions*, *18*(10).

Roy, S., Das, A. K., Chatterjee, S., Kumar, N., Chattopadhyay, S., & Rodrigues, J. J. (2019). Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. *IEEE Transactions on Industrial Informatics*, *15*(1), 457–468.

Schliep, M., & Hopper, N. (2018). End-to-End Secure Mobile Group Messaging with Conversation Integrity and Minimal Metadata Leakage. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2282–2284. ACM.

Tyagi, R., Sharma, N. K., Malhotra, K., & Khosla, A. (2018). Comprehensive Methodology for Threat Identification and Vulnerability Assessment in Ad hoc Networks. *Cyber Security: Proceedings of CSI 2015*, 335–347. Springer.

Venkatasen, M., & Mani, P. (2018). A risk-centric defensive architecture for threat modelling in e-government application. *Electronic Government, an International Journal*, *14*(1), 16–31.

Verdecchia, R. (2018). Identifying architectural technical debt in Android applications through automated compliance checking. *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*, 35–36. ACM.

Yoran, E., & Amoroso, E. G. (2018). The Role of Commercial End-to-End Secure Mobile Voice in Cyberspace. *The Cyber Defense Review*, *3*(1), 57–66.

Yu, J., & Hou, B. (2018). Survey on IMD and Wearable Devices Security Threats and Protection Methods. *International Conference on Cloud Computing and Security*, 90–101. Springer.