M9: Insecure Data Storage

Threat Agents

Application Specific

Insecure data storage in a mobile application can attract various threat agents who aim to exploit the vulnerabilities and gain unauthorised access to sensitive information. These threat agents include skilled adversaries who target mobile apps to extract valuable data, malicious insiders within the organisation or app development team who misuse their privileges, state-sponsored actors conducting cyber espionage, cybercriminals seeking financial gain through data theft or ransom, script kiddies utilising pre-built tools for simple attacks, data brokers looking to exploit insecure storage for selling personal information, competitors and industrial spies aiming to gain a competitive advantage, and activists or hacktivists with ideological motives.

These threat agents exploit vulnerabilities like weak encryption, insufficient data protection, insecure data storage mechanisms, and improper handling of user credentials. It is crucial for mobile app developers and organisations to implement strong security measures, such as robust encryption, secure data storage practices, and adherence to best

practices for mobile application security, to mitigate the risks associated with insecure data storage.

Attack vectors

Exploitability EASY

Insecure data storage in a mobile application exposes vulnerabilities to various attack vectors that threat actors can exploit. Attack vectors include unauthorised access to the device's file system through physical or remote means, exploiting weak encryption or lack thereof, intercepting data transmissions, and leveraging malware or malicious apps installed on the device. Additionally, rooted or jailbroken devices provide an opportunity for attackers to bypass security measures and gain direct access to sensitive data. Other attack vectors include social engineering techniques to deceive users into providing access to their data or manipulating the application's behaviour.

Overall, insecure data storage on a mobile application opens avenues for attacks ranging from direct data extraction to interception of sensitive information, emphasising the critical need for robust encryption, secure transmission protocols, and thorough security measures in mobile app development.

Security weakness

Prevalence COMMON

Detectability AVERAGE

Insecure data storage in a mobile application encompasses various security weaknesses that can jeopardise the confidentiality and integrity of stored information. These weaknesses include the use of

weak or nonexistent encryption, allowing attackers to easily access and decipher sensitive data. Additionally, storing data in easily accessible locations within the device's filesystem, such as plain text files or unprotected databases, exposes it to unauthorised extraction or manipulation. Insufficient access controls and user authentication mechanisms further compound the problem, enabling unauthorised individuals to gain access to sensitive data.

Furthermore, the absence of secure data transmission protocols leaves data vulnerable to interception during communication between the mobile app and external servers. Collectively, these security weaknesses in mobile application data storage create opportunities for data breaches, unauthorised access, and data tampering, emphasising the critical need for robust encryption, secure storage practices, and stringent access controls to mitigate these risks.

Technical Impact

Impact SEVERE

Insecure data storage on a mobile application can have significant technical impacts that undermine the overall security and functionality of the app. These impacts include:

Data breaches: Insecure data storage makes sensitive information susceptible to unauthorised access and data breaches. Attackers can exploit vulnerabilities to extract or manipulate sensitive data, leading to potential privacy violations and loss of confidential information.

Compromised user accounts: Inadequate data storage practices can result in the compromise of user accounts. Attackers may gain access to login credentials or personal information stored insecurely, leading to unauthorised account access, identity theft, or unauthorised activities on behalf of the user.

Data tampering and integrity issues: Without proper data protection measures, attackers can modify or tamper with the stored data. This can lead to data integrity issues, inaccurate information, or the injection of malicious content into the app's data stores.

Unauthorised access to application resources: Insecure data storage can provide attackers with the ability to gain unauthorised access to critical application resources. This includes sensitive files, configuration files, or cryptographic keys stored within the app, which can be leveraged to compromise the app's functionality or exploit its underlying systems.

Reputation and trust damage: If an app is found to have insecure data storage, it can severely damage the reputation and trust of the app developer or organisation. Users may lose confidence in the app's security, resulting in decreased user adoption and potential legal and regulatory consequences.

Compliance violations: Insecure data storage can lead to non-compliance with industry regulations and data protection standards. App developers may be subject to penalties or legal actions if they fail to adequately protect user data and maintain secure data storage practices.

Business Impacts

Impact SEVERE

The business impact of insecure data storage on a mobile application can be significant and wide-ranging. Here are some key business impacts:

Reputational damage: Insecure data storage can lead to data breaches and compromised user accounts, which can severely damage the reputation and trust of the organisation. News of data breaches can spread quickly, resulting in negative publicity, customer dissatisfaction, and potential loss of business.

Loss of customer trust: When sensitive customer data is compromised due to insecure data storage, customers may lose trust in the organisation's ability to protect their information. This loss of trust can lead to a decrease in customer loyalty, increased customer churn, and a negative impact on overall customer satisfaction.

Legal and regulatory consequences: Inadequate data storage practices may result in non-compliance with industry regulations and data protection laws. Organisations may face legal repercussions, including fines, penalties, or lawsuits for failing to protect user data adequately. Compliance violations can also damage the organisation's reputation and trustworthiness in the eyes of customers and business partners.

Financial implications: Data breaches and the resulting fallout can have significant financial implications for organisations. This includes the costs associated with investigating the breach, notifying affected customers, providing identity theft

protection services, potential legal settlements, and loss of business opportunities.

Competitive disadvantage: In today's highly competitive landscape, organisations that experience data breaches or have a reputation for insecure data storage can face a competitive disadvantage. Customers are increasingly concerned about the security of their data, and they may choose competitors who have a better track record of safeguarding sensitive information.

Am I Vulnerable To 'Insecure Data Storage'?

Insecure data storage and unintended data leakage in a mobile application can manifest in several ways, leading to potential privacy breaches and unauthorised access to sensitive information. Here are common manifestations of these issues:

Lack of Access Controls: Insufficient access controls within the application may allow unauthorised users or attackers to gain access to sensitive data stored on the device or in the app's databases.

Inadequate Encryption: Failure to properly encrypt sensitive data can result in unintended data leakage if an attacker gains access to the storage location. Without encryption, the data is easily readable and can be exploited.

Unintentional Data Exposure: Mobile applications may inadvertently expose sensitive data through application logs, error messages, or debug features, allowing unauthorised individuals to view or capture sensitive information.

Poor Session Management: Weak session management can lead to unintended data leakage. If session tokens or user authentication information are not adequately protected or managed, they can be intercepted or manipulated, allowing unauthorised access to sensitive data.

Insufficient Input Validation: Inadequate input validation and data sanitization can lead to unintended data leakage. Attackers may exploit this weakness to inject malicious scripts or retrieve sensitive data by manipulating input fields.

Cloud Storage Misconfigurations: If the mobile application uses cloud storage services for data storage and the configurations are mismanaged or misconfigured, it can result in unintended exposure or unauthorised access to stored data.

Third-Party Library Vulnerabilities: Insecure third-party libraries used in the mobile application may have vulnerabilities that could lead to unintended data leakage. Attackers can exploit these vulnerabilities to gain unauthorised access to sensitive information.

Unintended Data Sharing: Improper handling of data sharing features within the application can result in unintended data leakage. If sensitive data is shared with unintended recipients or if the sharing process is not adequately secured, it can lead to privacy breaches.

How Do I Prevent 'Insecure Data Storage'?

To prevent insecure data storage in a mobile application and ensure the protection of sensitive

data, the following security measures should be implemented:

Use Strong Encryption: Implement robust encryption algorithms and practices to protect sensitive data both at rest and in transit. Utilise industry-standard encryption algorithms and ensure that encryption keys are securely stored and managed.

Secure Data Transmission: Utilise secure communication protocols (e.g., HTTPS, SSL/TLS) to protect data during transmission between the mobile application and backend servers. Avoid sending sensitive data over unsecured channels.

Implement Secure Storage Mechanisms: Store sensitive data in secure storage locations that are inaccessible to unauthorised users. Use platform-specific secure storage mechanisms provided by the mobile operating system, such as Keychain (iOS) or Keystore (Android).

Employ Proper Access Controls: Implement strong access controls to restrict unauthorised access to sensitive data. Authenticate users securely, enforce role-based access controls, and validate user permissions before granting access to sensitive information.

Validate Input and Sanitize Data: Implement input validation and data sanitization techniques to prevent injection attacks and ensure that only valid and expected data is stored. Validate user inputs to mitigate the risk of malicious code injection or unintended data leakage.

Apply Secure Session Management: Implement secure session management techniques, such as using randomly generated session tokens, setting

proper session timeouts, and securely storing session data on the client and server sides.

Regularly Update and Patch Dependencies: Keep all libraries, frameworks, and third-party dependencies up to date, as they may contain security vulnerabilities that could lead to insecure data storage. Regularly apply security patches and updates provided by the respective vendors.

Stay Informed: Stay up to date with the latest security threats and vulnerabilities in the mobile application landscape. Monitor security forums, security advisories, and mobile platform updates to ensure timely mitigation of emerging risks.

Example Attack Scenarios

Few example scenarios that illustrate potential instances of insecure data storage in a mobile application:

Storing Passwords in Plain Text: The mobile application stores user passwords in plain text format within a local database or file, making it easy for an attacker to retrieve and abuse these credentials if they gain unauthorized access to the device.

Unsecured Local Storage: The mobile application stores sensitive user data, such as personally identifiable information (PII), locally on the device without utilizing proper access controls or encryption. This allows anyone with physical access to the device to extract and view the data.

Insecure Data Caching: The mobile application caches sensitive data, such as user authentication tokens or session information, without implementing appropriate security measures. If an attacker gains

access to the device's cache, they can obtain these credentials and impersonate the user.

Unprotected Logging: The mobile application logs sensitive data, including user actions, API responses, or error messages, without proper security controls. This can lead to unintentional exposure of sensitive information if an attacker gains access to the device or intercepts the log files.

Insecure Cloud Storage Configuration: The mobile application utilizes cloud storage services to store user data but misconfigures the storage permissions, allowing unauthorized access to the stored information. This can result in data leakage or unauthorized exposure of sensitive data.

Improper Handling of Temporary Files: The mobile application creates temporary files to process or store sensitive data, but fails to properly handle and delete these files afterward. This leaves sensitive information exposed and vulnerable to unauthorized access.

References

- OWASP
  - OWASP
- External
  - External References

Edit on GitHub

Spotlight: Guardsquare

Guardsquare offers the most complete approach to mobile application security on the market. Guardsquare's software integrates seamlessly across the development cycle: from app security testing to code hardening to real-time visibility into the threat landscape. Guardsquare products provide enhanced mobile application security from early in the development process through publication. More than 900 customers worldwide across all major industries rely on Guardsquare to help them identify security risks and protect their mobile applications against reverse engineering and tampering. Learn more at www.guardsquare.com.

Corporate Supporters

Become a corporate supporter

products or services, allowing our community to remain vendor neutral with the collective wisdom of the best minds in software security worldwide.