M6: Inadequate Privacy Controls

Threat Agents

Application Specific

Privacy controls are concerned with protecting Personally Identifiable Information (PII), e.g., names and addresses, credit card information, e-mail and IP addresses, information about health, religion, sexuality and political opinions.

This information is valuable to attackers for several reasons. For example, an attacker could

- Impersonate the victim to commit a fraud,
- Misuse the victim's payment data,
- Blackmail the victim with sensitive information or
- Harm the victim by destroying or manipulating the victim's critical data.

In general, PII could either be leaked (i.e., a violation of confidentiality), manipulated (violation of integrity) or destroyed/blocked (violation of availability).

Attack Vectors

Exploitability AVERAGE

Typical sources for PII are well protected, e.g., the sandbox of the app, the network communication with the server, the app's logs and backups. Some have less protection but are still hard to access, like URL query parameters and clipboard content.

Obtaining PII thus requires the attacker to first breach security on another level. Attackers could eavesdrop on the network communication, access file system, clipboard, or logs with a trojan or get their hands on the mobile device and create a backup to analyze. Since PII is just data that can be stored, processed, and transmitted by all means available on mobile devices, the possibilities to extract or manipulate it are manifold.

Security Weakness

Prevalence COMMON

Detectability EASY

Almost all apps process some kind of PII. Many even collect and process more than they need to fulfill their purpose, which makes them more attractive as a target without business needs.

Risks of privacy violations increase due to careless handling of PII by developers. PII should always be processed with the possibility in mind that an attacker could access communication and storage media.

Hence, an app is vulnerable to privacy infringements if some personal data it collects motivates an attacker to manipulate or abuse that data through a storage or transmission medium that is insufficiently secured.

Technical Impacts

Impact LOW

Privacy violations usually have little technical impact on the system as a whole. Only if the PII includes information like authentication data, it can affect certain global security properties, e.g., traceability.

If user data is manipulated it might render the system unusable for that user. Through ill-formed data, also the backend may be disturbed if it is missing proper sanitization and exception handling.

Business Impacts

Impact SEVERE

The extent and severity of the business impact, which a privacy violation has, strongly depends on the number of affected users, the criticality of the affected data, and the data protection regulations that apply where the violation happened. The business impact of privacy violations will typically result in the following at a minimum:

Violation of legal regulations: Regulations are the biggest issue regarding privacy controls. GDPR (Europe), CCPA (California, US), PDPA (Singapore), PIPEDA (Canada), LGPD (Brazil), Data Protection Act 2018 (UK), POPIA (South Africa), PDPL (China) are examples of relevant regulations with known sanctions against companies for not protecting their users' data.

Financial damage due to victims' lawsuits: Whoever is personally affected by a privacy violation might sue the app provider that let the violation happen. These lawsuits might be successful, depending on the legal regulations that apply and the ability of the provider to show that they hat adequate and up to date protection mechanisms in place.

Reputational damage: If a privacy violation affects users on a large scale, it is likely published in media, thus, generating negative publicity for the provider of the app. As a consequence, sales and usage for the app and even other, unrelated products of the same provider might drop.

Loss or theft of PII: Actual information stolen might be misused, even for attacks on the provider of the app. For example, specific user data could be used to employ a social engineering attack on the provider by impersonating a victim.

Am I Vulnerable To 'Inadequate Privacy Controls'?

An app can only be vulnerable to Inadequate Privacy Controls if it processes some form of personally identifiable information. This is almost always the case: Client apps' IP addresses visible to a server, logs of the apps' usage, and metadata sent with crash reports or analytics are PII that apply to most apps. Usually, an app will collect and process additional, more sensitive PII from its users, like accounts, payment data, locations and more.

Given an app that uses PII, it might expose it like any other sensitive data. This most notably happens through

- Insecure data storage and communication (cf. M5, M9),
- Data access with insecure authentification and authorization (cf. M3, M1), and
- Insider attacks on the app's sandbox (cf. M2, M4, M8).

The other OWASP Mobile Top 10 risks provide deeper insights on how an app might be vulnerable to the different attack vectors.

How Do I Prevent 'Inadequate Privacy Controls'?

Something that does not exist cannot be attacked, so the safest approach to prevent privacy violations is to minimize the amount and variety of PII that is processed. This requires full awareness of all PII assets in a given app. With that awareness, the following questions should be assessed:

- Is all PII processed really necessary, e.g., name and address, gender, age?
- Can some of the PII be replaced by less critical information, e.g., fine-grained location by coarse-grained location?
- Can some of the PII be reduced, e.g., location updates every hour instead of every minute?
- Can some of the PII be anonymized or blurred, e.g., by hashing, bucketing, or adding noise?
- Can some of the PII be deleted after some expiration period, e.g., only keep health data of the last week?
- Can users consent to optional PII usage, e.g., to receive a better service but also be aware of the additional risk?

The remaining PII should not be stored or transferred unless absolutely necessary. If it must be stored or transferred, access must be protected with proper authentication and possibly authorization. Also defense in depth should be considered for particularly critical data. For example, health data may be encrypted with a key sealed in the device's

TPM in addition to its storage in the app's sandbox. So, if an attacker manages to circumvent the sandbox restrictions, the data is still not readable. The other OWASP Mobile Top 10 risks suggest measures to securely store, transfer, access and otherwise handle sensitive data.

Threat modeling can be used to determine the most likely ways that privacy violations may occur in a given app. The effort of securing PII could then be focused on these.

Static and dynamic security checking tools might reveal common pitfalls, like logging of sensitive data or leakage to clipboard or URL query parameters.

Example Attack Scenarios

The following scenarios showcase inadequate privacy controls in mobile apps:

Scenario #1: Inadequate sanitization of logs and error messages.

Reporting of logs and exceptions is essential for quality assurance of a productive app. Crash reports and other usage data helps developers to fix bugs and learn about how their app is used. However, logs and error messages might contain PII if the developers chose to include this data in log or error messages. Also, third party libraries might include PII in their error messages and logs as well. An example of a frequent issue are database exceptions that reveal part of the query or result. This will most likely be visible to any platform provider used for collecting and evaluating crash reports. It might also become visible to the user if the error is displayed on screen or to attackers who can read device logs. Developers should be especially careful in what they

log and ensure that exception messages are sanitized before displaying them to the user or reporting them to a server.

Scenario #2: Using PII in URL query parameters.

URL query parameters are often used to transmit request arguments to a server. However, URL query parameters are visible at least in the server logs, but often also in website analytics and possibly in the local browser history. So sensitive information should never be transmitted as query parameters. Instead, they should be sent as a header or part of the body.

Scenario #3: Exclusion of personal data in backups/not setting hasFragileUserData.

Most PII processed by an app is stored in its sandbox. The app should explicitly configure what data to include in device backups. An attacker might obtain a device and create a backup or get a backup from another source, from which the sandbox content could be extracted.

Alternatively, by setting hasFragileUserData to 'true' in Android, an app may preserve its data upon uninstallation. An attacker who manages to install a malicious app with the same package id later can access this data.

Hence, both settings should be explicitly set for apps to make the developers' intent transparent and to control the information flow through backups or between subsequent installations of an app.

References

- OWASP
  - User Privacy Protection Cheat Sheet

    - Testing User Privacy Protection (MASTG)
    - OWASP Top 10 Privacy Risks
    - OWASP Top 10 for Large Language Models: LLM06: Sensitive Information Disclosure
- External
    - EU General Data Protection Regulation

Edit on GitHub

Spotlight: Approov

Approov is the world leader in mobile security. As cybercriminals adopt AI tools and a mobile-first attack strategy, Approov helps organizations stay ahead of threats with runtime, zero-trust protection of both the mobile apps that run your business and all the APIs they use. Approov secures the world's most targeted mobile apps and APIs with patented app attestation and runtime API protection. Our cloud-first approach ensures only your genuine mobile app, running on a safe device, can access your backends—stopping fake and repackaged apps, emulators, bots, and man-in-the-middle attacks.

Corporate Supporters