Bild: fotolia

tigte Personen die App nutzen können? Werden die Benutzerdaten verschlüsselt übertragen? Wird dafür ein sicherer Mechanismus verwendet? Wo werden Benutzernamen und Passwort abgelegt? Wie und wo erfolgt die Datenhaltung? Solche und weitere Fragen können anhand eines Source Code Reviews beantwortet werden. Steht der Source Code den Testern nicht zur Verfügung, kann mittels Reverse Engineering dasselbe Testergebnis erreicht werden. Dabei wird die App Schritt für Schritt mit dafür geeigneten Tools in ihre einzelnen Funktionsteile zerlegt. So kann der Auditor wichtige Erkenntnisse darüber erhalten, was mit möglichem sensiblen Inhalt geschieht, wie die Kommunikation zwischen mobilem Gerät und dem zugehörigen Server vonstatten geht und wie die Daten geschützt sind.

Application Security Audit der Mobile App: Dieser Schritt überprüft, ob die typischen Sicherheitsziele Verfügbarkeit, Vertraulichkeit und Integrität gewährleistet sind. Bei den Tests sollte darauf geachtet werden, dass mindestens auf die bereits erwähnten OWASP-Mobile-Top-Ten-Risiken geprüft wird. Aus der Erfahrungsbasis von über 50 überprüften Mobile Apps ist unter anderem ein besonderes Augenmerk auf die folgenden Punkte zu legen:

- Weak Server Side Controls: In diese Kategorie gehören Risiken, die auf dem Server lokalisiert sind, der mit der Mobile App kommuniziert. Oft sind dies dieselben Risiken, die typischerweise auch bei der Sicherheitsüberprüfung einer normalen Website auftauchen. Häufige Mängel sind fehlende oder nur lückenhaft bereitgestellte Sicherheitsfunktionen oder Schnittstellen zu Sicherheitsfunktionen des Betriebssystems, oder dass darauf vertraut wird, dass das mobile OS die volle Sicherheit gewährleistet. Insbesondere wenn eine App für verschiedene mobile Betriebssysteme entwickelt werden soll, nutzt mancher Hersteller den grössten gemeinsamen Sicherheitsnenner der zu unterstützenden Betriebssysteme, um den Aufwand zu optimieren. So bleibt oft die Sicherheit auf der Strecke. Dies wird nur noch durch die Tatsache übertroffen, dass bei gewissen Entwicklungsvorhaben keinerlei Sicherheitsvorgaben vom Auftraggeber definiert werden und sich die Entwicklungsfirma nicht genötigt fühlt, auf diesen Umstand hinzuweisen.
- Insecure Data Storage: Bei vielen Audits ist aufgefallen, dass die Benutzerdaten nicht an sicheren Orten gespeichert werden, sondern dort, wo es sich gerade anbietet. Dies hat zur Folge, dass ein Angreifer diese anschliessend einfach auslesen und missbrauchen kann. Populäre, aber nicht empfohlene Ablageorte für solche Daten sind unter anderem eine verwendete SQLite-Datenbank, die SD-Karte, Cookies sowie XML- oder Binary Data Stores. Empfohlen ist, die Daten, wenn möglich, nicht auf dem Gerät, sondern im Fall von Benutzerdaten mittels Standard- oder API-Log-in-Verfahren über eine verschlüsselte Verbindung auf dem Server zu speichern. Sollte die Speicherung auf dem Gerät zwingend notwendig sein, sollten bewährte Encryption Libraries wie beispielsweise «CommonCrypto» oder «javax.crypto» verwendet werden. Das iOS bietet auch eine lokale Keychain an, die aber, sollte das Telefon «gejailbraked» oder mittels Exploit angegriffen werden, relativ einfach ausgelesen werden kann.
- Insufficient Transport Layer Protection: Im Gegensatz zu einer Intranetanwendung ist es bei mobilen Apps besonders wichtig, jeglichen Verkehr zu schützen, da diese Applikationen oft via WLAN oder ein sonstigs ungeschütztes Netzwerk verwendet werden und so die Gefahr besteht, dass die Verbindung abgehört wird. Auch kann der Benutzer nicht – wie bei einem herkömmlichen Browser – anhand eines Symbols oder grünen Balkens verifizieren, ob die Verbindung verschlüsselt ist. Wichtig für den Schutz der Übertragung ist, dass alle Verbindungen, zu den eigenen Servern wie zu externen Stellen, gemäss aktuellen Security Best Practices verschlüsselt sind. Wichtig ist auch, dass die Zertifikate nicht abgelaufen und nicht selbstsigniert sind sowie starke Verschlüsselungsalgorithmen mit einer Schlüssellänge von mindestens 256 Bit für symmetrische und 2048 Bit für asymmetrische Algorithmen verwendet werden. Sensitive Inhalte wie Benutzernamen oder Passwörter dürfen nie über einen alternativen Kanal wie SMS oder MMS versendet werden – die einzige Ausnahme bilden sogenannte mTAN-Verfahren, bei denen dem Benutzer zwecks zusätzlicher Authentisierung ein Ein-Mal-Passwort per SMS gesendet wird.
- Ein weiteres Augenmerk ist auf die Autorisierung und Authentisierung zu legen. Wobei hier oft bemängelt werden muss, dass als einziger Faktor nur feste Merkmale eines Telefons wie IMEI- oder Telefonnummer als Authentisierungsmerkmal verwendet werden. Mithilfe eines einfachen Proxys kann die Anfrage abgefangen und manipuliert werden, sofern auf dem Backend nicht die notwendigen Massnahmen getroffen wurden.

Penetration Test des Backend: Zwingend ist auch der Backend-Server zu prüfen, der die Kommunikation mit der mobilen Applikation ermöglicht (bei nativen Applikationen) respektive die App zur Verfügung stellt (Webapps). Oft geht vergessen, dass dieser Service vom Internet her genauso via Browser oder ähnlichen Technologien für jedermann zugänglich ist. Die Realität zeigt leider oft, dass davon ausgegangen wird, dass nur berechtigte Benutzer auf den Service zugreifen und so wichtige Schutzmechanismen nicht oder nur unzureichend implementiert werden.

Um die Backend-Systeme überprüfen zu können, eignet sich ein Penetration Test, der den unprivilegierten Zugang sowie die zur Verfügung gestellten Services detailliert aus der Perspektive eines Angreifers überprüft.

Audit als vorbeugende Massnahme

Mobile Applikationen unterliegen teilweise denselben Sicherheitsproblemen wie herkömmliche Webapplikationen. Obwohl Empfehlungen, Standards und Richtlinien für eine sichere Umsetzung vorhanden sind, werden diese oft aufgrund von Zeitdruck oder aus wirtschaftlichen Überlegungen vernachlässigt. Dabei gehört die Durchführung eines Security Audits, der nur einen Bruchteil des Budgets ausmacht, mittlerweile zu Best Practices und schützt vor den unerfreulichen Folgen einer erfolgreichen Hackerattacke.