Mobile Apps – aber sicher

Applikationen für mobile Geräte wie Smartphones und Tablets werden immer beliebter und populärer. Dies führte dazu, dass die Anzahl bekannter Malware für mobile Geräte massiv gestiegen ist. Auf welche Punkte sollte angesichts dieser Tatsache beim Entwickeln einer mobile App geachtet werden, und wie prüft man deren Sicherheit? Tobias Ellenberger

Die hohe Akzeptanz und Verbreitung von Smartphones im geschäftlichen und privaten Umfeld hat viele Firmen, die Produkte und Dienstleistungen für eine breite Käuferschicht anbieten, faktisch dazu gezwungen, mobile Apps für einzelne oder mehrere Plattformen anzubieten. Dabei muss unter anderem beachtet werden, dass der Grossteil der Smartphones und die darauf laufenden Apps primär auf maximale Benutzerfreundlichkeit und für den privaten Gebrauch entwickelt wurden. Dass mit steigender Akzeptanz und Verbreitung Smartphones auch für Firmen interessant wurden, ist die logische Konsequenz. Somit gewinnt auch der Aspekt der Sicherheit zunehmend an Bedeutung, insbesondere dann, wenn sensibler Inhalt wie zum Beispiel Kreditkartendaten, Kontoinformationen oder Zahlungen über das Gerät abgewickelt werden. Kaum ein Unternehmen möchte als Opfer einer erfolgreichen Hackerattacke genannt werden. Doch was ist bei der Entwicklung zu beachten?

Entwicklung und Architektur

Soll eine mobile Applikation programmiert werden, stellt sich die Frage, für welchen Designtyp man sich entscheiden soll. Grundsätzlich wird zwischen nativen und für mobile Geräte optimierte Webapplikationen unterschieden. Eine native Applikation wird ausschliesslich auf dem Endgerät des Benutzers betrieben, kann auf praktisch alle Daten des mobilen Geräts zugreifen und diese auch für eigene Zwecke verwenden. Der gesamte Programmablauf und die Datenhaltung finden somit auf dem mobilen Endgerät statt. Diese Architektur zwingt sich bei allen Stand-alone-Apps wie beispielsweise Passwort-Vaults oder Games auf.

Bei einer sogenannten Webapplikation oder Webapp wird die Applikation auf dem Endgerät im Browser aufgerufen und dargestellt. Die ganze Intelligenz verbleibt auf einem Webserver, der den Inhalt analog einer Website für den Benutzer bereitstellt. Typische Vertreter dieses Designtyps sind mobile Clients von Onlineshops oder Mobile-Banking-Apps. Der Vorteil dieses Ansatzes ist, dass die Funktionalität und Datenhaltung auf den zugehörigen Servern abläuft, also auf Systemen, die von den Betreibern besser abgesichert werden können als die mobilen Clients. Der Nachteil ist, dass die Rechenleistung zentral bereitgestellt werden muss und nicht dezentral von den mobilen Clients gestellt wird. Dieser Faktor ist insbesondere bei Applikationen mit sehr vielen parallelen User Sessions nicht zu vernachlässigen – denn kaum jemand akzeptiert in der heutigen Zeit noch lange Reaktionszeiten oder Verbindungsabbrüche. Da beide, native und Webapps, Vor- und Nachteile haben, sind sogenannte Hybrid-Apps entstanden, bei denen versucht wird, die Vorteile der beiden Ansätze zu vereinen. Doch hierbei gilt es, die Funktionen detailliert zu hinterfragen und zu klären, was lokal und was zentral ablaufen, beziehungsweise wo die Datenhaltung erfolgen soll.

Tobias Ellenberger ist COO und Teilhaber der One-Consult GmbH. www.oneconsult.com

Bei der Umsetzung empfiehlt es sich, auf die vom Hersteller präferierten Programmiersprachen und Frameworks zu setzen, die sich allgemein als Standard für das jeweilige Betriebssystem etabliert haben. Matchentscheidend ist auch, für welche Programmiersprachen und Frameworks entsprechendes Know-how bei der internen Entwicklungsabteilung oder dem externen Entwicklungspartner vorliegt. Anschliessend sollte die Applikation von einer darauf spezialisierten Firma auf ihre korrekte Funktionalität und Belastungsresistenz überprüft werden, wobei unterschiedliche Endgeräte berücksichtigt werden müssen, da alle ihre Eigenheiten haben. Nachdem die App gemäss Pflichtenheft läuft, sollte sie vor dem «Go Live» auf Sicherheitslücken untersucht werden, da diese Anforderungen leider allzu oft bei der Formulierung des Pflichtenhefts gar nicht oder zu oberflächlich berücksichtigt werden.

Implementierung

Es ist insbesondere von herkömmlichen (Web-) Applikationen bekannt, dass es sich lohnt, bei der Entwicklung auf etablierte Standards zu setzen und offene sowie frei verfügbare Projekte, die sich unter anderem mit der Sicherheit von Applikationen beschäftigen, wie das Open Web Application Security Project (OWASP), bereits im Entwicklungsprozess zu berücksichtigen. Daran hat sich auch für die Entwicklung von Mobile Apps nichts geändert. Neben den bekannten Entwicklungsstandards und Richtlinien der Hersteller, hat auch OWASP ein Framework speziell für Mobile Apps entwickelt, das den Programmierern hilft, eine möglichst hohe Sicherheit bei der App-Programmierung zu erreichen. Hier lohnt es sich, einen Blick auf das OWASP Mobile Security Project zu werfen. Wichtig hierbei ist, dass den Entwicklern daraus die «Top Ten Mobile Risks» sowie die «Top Ten Mobile Controls» bekannt sind und die Erkenntnisse daraus in die Entwicklung miteinfliessen.

Jetzt kann die App veröffentlicht werden. Oder nicht? In der Praxis ist es oft so, dass zu diesem Zeitpunkt das Budget aufgebraucht oder bereits überschritten wurde und für eine neutrale Beurteilung der Funktionalität und Sicherheit weder Zeit noch Budget übrig sind. Doch sollte die App insbesondere bei sensiblen Inhalten im Sinne einer Qualitätskontrolle und zum Schutz der Daten des Kunden auf ihre Sicherheit überprüft werden. Im Optimalfall wurde dies bereits bei der Projektierung eingeplant. Doch wie wird eine solche App getestet, für welchen Testtyp soll man sich entscheiden und was soll überhaupt überprüft werden?

Typen von technischen Security Audits

Um eine mobile App auf ihre Sicherheit zu überprüfen, bieten sich folgende Testtypen an, die sich je nach der Art der Architektur (native, Web- oder Hybrid-App) ergänzen oder unterscheiden können. Grundsätzlich wird die Entscheidung, welche Testtypen zum Einsatz kommen, anhand der Architektur, des Einsatzzweckes und den Sicherheitsanforderungen an die App getroffen.

Source Code Review/Reverse Engineering: Hierbei wird der Source Code der App einer detaillierten Analyse unterzogen. Dabei wird das Augenmerk nicht auf die Einhaltung von programmiertypischen Standards gelegt, sondern auf die integrierten Sicherheitsfunktionen. Wie wird sichergestellt, dass nur berech-

▶

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.