Bezpieczeństwo aplikacji mobilnych

Wykład 7

Bezpieczeństwo aplikacji mobilnych – analiza i ochrona

Mateusz Pawełkiewicz

1. Wprowadzenie do analizy bezpieczeństwa aplikacji mobilnych

Bezpieczeństwo aplikacji mobilnych wymaga stałej analizy i implementacji najlepszych praktyk, aby chronić dane użytkowników przed potencjalnymi zagrożeniami. Deweloperzy muszą rozumieć procesy związane z dystrybucją aplikacji, ich analizą oraz zabezpieczeniami, które pomagają w ochronie przed złośliwym oprogramowaniem i nieautoryzowanymi modyfikacjami. W tym wykładzie omówimy sposoby dystrybucji aplikacji, narzędzia do analizy bezpieczeństwa oraz metody zabezpieczania aplikacji przed reverse engineeringiem.

2. Sposoby dystrybucji aplikacji i związane z nimi ryzyka

Aplikacje mobilne są dystrybuowane głównie za pośrednictwem oficjalnych sklepów, takich jak Google Play dla Androida i App Store dla iOS. Jednak istnieją również alternatywne metody dystrybucji, które niosą ze sobą pewne ryzyko:

- **Sklepy trzecich stron**: Instalowanie aplikacji z nieoficjalnych źródeł zwiększa ryzyko pobrania oprogramowania zawierającego złośliwy kod.
- **Pliki APK i sideloading**: Użytkownicy Androida mogą ręcznie instalować pliki APK, co otwiera drogę do ataków, jeśli plik pochodzi z niezaufanego źródła.
- **Certyfikaty i podpisy cyfrowe**: Aplikacje powinny być podpisywane cyfrowo przez dewelopera, co potwierdza ich autentyczność. Brak podpisu cyfrowego lub użycie fałszywego certyfikatu może oznaczać, że aplikacja została zmodyfikowana przez osoby trzecie.

Deweloperzy muszą mieć świadomość tych ryzyk i implementować środki ochrony, takie jak weryfikacja integralności aplikacji i używanie bezpiecznych metod dystrybucji.

3. Analiza form binarnych aplikacji

Zrozumienie form binarnych aplikacji i ich dystrybucji, takich jak APK dla Androida oraz IPA dla iOS, jest kluczowe dla deweloperów zajmujących się bezpieczeństwem. Pliki te mogą być analizowane przez osoby trzecie w celu znalezienia luk w zabezpieczeniach lub modyfikacji ich działania. Główne aspekty analizy form binarnych to:

- **Reverse engineering**: Proces polegający na dekompilacji aplikacji w celu analizy jej kodu źródłowego. Deweloperzy mogą stosować narzędzia takie jak JADX czy Apktool na Androidzie oraz narzędzia typu Cycript i Frida na iOS.
- **Formaty plików**: Na Androidzie pliki APK składają się z kodu, zasobów, manifestu oraz plików DEX (Dalvik Executable). Na iOS pliki IPA zawierają kod w formacie Mach-O i zasoby aplikacji.

Aby zabezpieczyć aplikacje przed dekompilacją, deweloperzy powinni stosować techniki obfuskacji kodu oraz szyfrowania zasobów.

4. Reverse engineering i jego konsekwencje

Reverse engineering, czyli inżynieria wsteczna, polega na analizie i modyfikacji kodu aplikacji w celu zrozumienia jej działania lub wprowadzenia zmian. Może to prowadzić do poważnych zagrożeń:

- **Naruszenie własności intelektualnej**: Kod źródłowy aplikacji może zostać skopiowany lub zmodyfikowany przez osoby trzecie.
- **Modyfikacje złośliwe**: Aplikacje mogą być zmieniane w taki sposób, aby działały na szkodę użytkownika, np. wykradały dane lub wprowadzały złośliwe funkcje.

Deweloperzy mogą stosować narzędzia obfuskacji, takie jak ProGuard czy R8 dla Androida, aby utrudnić analizę kodu oraz blokować użycie narzędzi debugujących.

5. Metody utrudniania analizy i modyfikacji aplikacji

W celu ochrony aplikacji przed reverse engineeringiem deweloperzy powinni wdrażać następujące techniki:

- **Obfuskacja kodu**: Utrudnia czytanie kodu aplikacji przez zamianę nazw zmiennych, funkcji i klas na nieczytelne dla człowieka. Obfuskacja zmniejsza ryzyko skopiowania i modyfikacji kodu.
- **ASLR (Address Space Layout Randomization)**: Mechanizm polegający na losowym rozmieszczeniu elementów pamięci aplikacji, co utrudnia ataki typu buffer overflow.
- **Wykrywanie debuggerów**: Implementacja technik, które pozwalają aplikacji wykrywać, czy jest debugowana przez osoby trzecie, co zapobiega analizie jej działania w czasie rzeczywistym.

6. Wykrywanie środowisk z podwyższonymi uprawnieniami

Aplikacje powinny być w stanie rozpoznać, czy działają na urządzeniach, które zostały zmodyfikowane poprzez rootowanie (Android) lub jailbreaking (iOS). Wdrożenie tego rodzaju zabezpieczeń może:

- **Zwiększyć poziom ochrony danych**: Aplikacje mogą ograniczać dostęp do wrażliwych funkcji, jeśli wykryją, że działają na urządzeniu z podwyższonymi uprawnieniami.
- **Zminimalizować ryzyko naruszenia bezpieczeństwa**: Deweloperzy mogą implementować funkcje, które automatycznie kończą działanie aplikacji na urządzeniach z rootem lub jailbreakingiem.

7. Narzędzia wspomagające analizę bezpieczeństwa aplikacji

Istnieją liczne narzędzia wspierające deweloperów w analizie bezpieczeństwa aplikacji, takie jak:

- **OWASP Mobile Security Framework (MobSF)**: Narzędzie do analizy aplikacji mobilnych pod kątem bezpieczeństwa, które umożliwia wykrywanie luk i ocenę zabezpieczeń.
- **Burp Suite**: Narzędzie używane do testowania bezpieczeństwa aplikacji webowych i mobilnych, które pozwala na przeprowadzanie testów penetracyjnych.
- **Frida i Cycript**: Narzędzia do dynamicznej analizy aplikacji na iOS, które umożliwiają inżynierię wsteczną i analizę w czasie rzeczywistym.

8. Podsumowanie i wnioski

Bezpieczeństwo aplikacji mobilnych wymaga stałego monitorowania i wdrażania zaawansowanych mechanizmów ochrony przed analizą i modyfikacją. Deweloperzy powinni stosować techniki obfuskacji, wykrywania środowisk z podwyższonymi uprawnieniami oraz wdrażać narzędzia do analizy i testowania zabezpieczeń. Dzięki temu mogą zwiększyć poziom ochrony swoich aplikacji, chroniąc dane użytkowników i zapewniając bezpieczeństwo funkcjonalności aplikacji.

Literatura:

Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse – *Bezpieczeństwo aplikacji mobilnych. Podręcznik hakera*, Helion.