



OWASP

Mariusz Burdach
Prevenity
www.prevenity.com
mariusz.burdach@prevenity.com

OWASP

09 10 2012

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Bankowość mobilna – analiza ryzyka na przykładzie telefonów iPhone

- Rodzaje aplikacji mobilnych
- iOS
- Zagrożenia
- Weryfikacja
- Podsumowanie

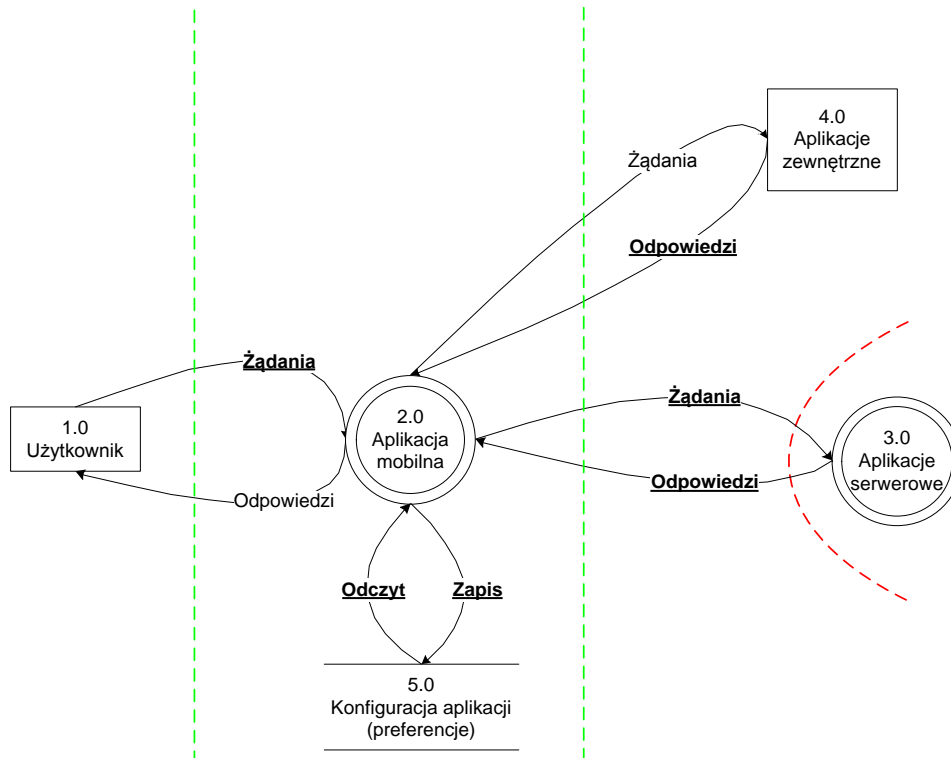
Rodzaje aplikacji mobilnych

■ Przeglądarka

■ Aplikacja natywna

■ Funkcje

- ▶ Token
- ▶ Autoryzacja
- ▶ IPC
- ▶ Apple Push
- ▶ Facebook
- ▶ Photo Faktura
- ▶ GPS
- ▶ ...



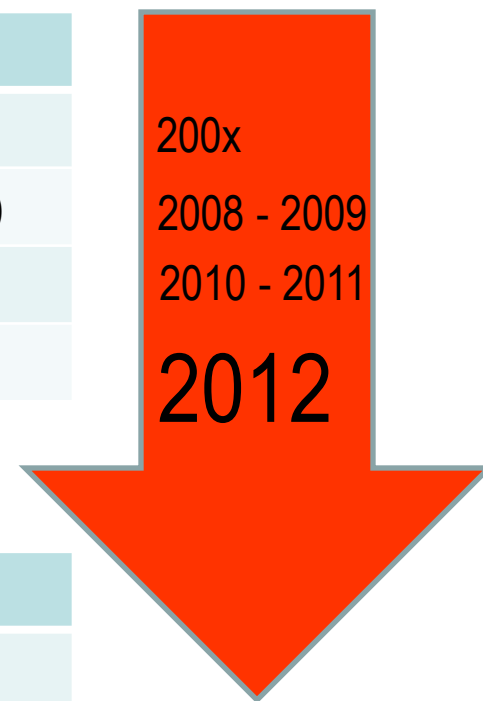
Historia

■ Bankowość internetowa

Zagrożenia	Ochrona
Keylogger/MitM	Token OTP/Karta z OTP
MitB	OoB (SMS/Token z HMAC)
MitMo + Socjotechnika	Token z HMAC (OoB)
MitB + Socjotechnika	-/FDS

■ Bankowość mobilna

Zagrożenia	Ochrona
Keylogger	
MitM	
Socjotechnika	



Największe ryzyko dla banku związane jest z kradzieżą środków z konta

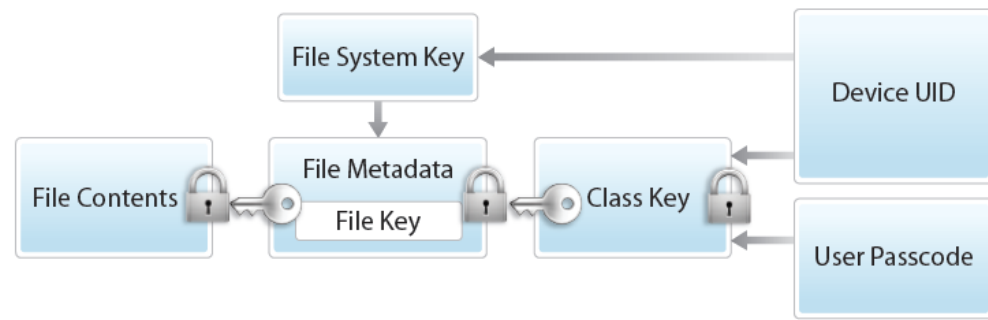
Zmniejszenie ryzyka nie musi się wiązać z ograniczeniem funkcjonalności

(np. wyłącznie operacje zdefiniowane lub limity dla transakcji dowolnych)

iOS

■ Mechanizmy bezpieczeństwa iOS

- ▶ Secure Boot Chain
- ▶ Application Code Signing + Vetting Process
- ▶ Runtime Process Security
 - ASLR/KASLR
 - Code Signing Enforcement
 - Sandbox
 - Entitlements
- ▶ Encryption & Data Protection
 - AES engine
 - Class keys
 - Keychain
- ▶ Device Access
 - Passcode



XCode

- -fstack-protector-all
- Automatic Reference Counting
- Background
- NSURLRequest/CFNetwork
- Obsługa IPC
- NSData/NSFileManager
- SecItemAdd/SecItemUpdate
- Implementacja protokołów
- NSLog
- NSAssert
- UITextField -> UITextAutocorrectionNo
- ...

Analiza

Zagrożenia	Wektor ataku	Prawdopodobieństwo
Keylogger lub inny malware Dostęp do pliku z kluczem	0-day exploit Socjotechnika Fizyczny dostęp „Ominięcie” procesu weryfikacji (np. jailbreak)	średnie
MitM	Socjotechnika	średnie
Kradzież telefonu	Fizyczny dostęp	małe
Dostęp do kopii bezpieczeństwa	Kompromitacja stacji roboczej	średnie

Analiza aplikacji

■ Analiza

- ▶ statyczna
- ▶ dynamiczna
- ▶ kodu źródłowego

■ Budowa aplikacji

- ▶ Mach O
- ▶ Objective C

■ Architektura ARM

Weryfikacja aplikacji bankowości mobilnej




- Komunikacja
- Lokalne dane
 - ▶ Klasy bezpieczeństwa
 - ▶ Keychain/SQLite/.plist
- Kryptografia
 - ▶ Funkcje (np. PBKDF2/HMAC)
 - ▶ Parametry
- Zarządzanie obiektami
 - ▶ User-after-free, double-free
 - ▶ Poufne dane w pamięci RAM
- Walidacja danych
 - ▶ Overflows
 - ▶ Format string
- Inne elementy
 - ▶ personalizacja/aktywacja/obsługa błędów i logowania/snapshot/keyboardcache/itd

Demonstracja wybranych elementów

- Weryfikacja ASLR
- Odtworzenie klas oraz zmiennych
- Identyfikacja istotnych funkcji w aplikacji
 - ▶ Bezpieczeństwo plików
 - ▶ Bezpieczeństwo elementów w keychain
 - ▶ Obsługa zdarzeń
 - ▶ Funkcje kryptograficzne
- Automatyczna walidacja danych z wykorzystaniem dedykowanego rozszerzenia do Burp Suite

Podsumowanie

- Funkcjonalność aplikacji jest krytyczna
- Bezpieczeństwo aplikacji jest związane z bezpieczeństwem urządzenia mobilnego
- „Zależności” pomiędzy bankowością mobilną a bankowością internetową (np. te same dane uwierzytelniające)
- Rekomendacje dla użytkowników

Aktualizacja systemu operacyjnego	
Ustawienie PIN/hasło (nie tylko cyfry) do urządzenia	
Bez jailbreak	
Transakcje zdefiniowane	



Źródła

- „iOS Hacker’s Handbook”, Ch. Miller, D. Blazakis, D. Dai Zovi, S. Esser, V. Iozzo, R-P Weinmann
- „Secure Mobile Development Best Practices”, viaForensics.com
- „Secure Development on iOS”, David Thiel
- „iOS Application (In)Security”, Dominic Chell
- „iOS Security”, Apple 2012
- „iOS Hardening Configuration Guide”, Australian Government DoD
- „The Dark Art of iOS Application Hacking”, Jonathan Zdziarski
- „Automation in iOS Application Assessment”, J. Engler, S. Law, J. Dubik, D. Vo.
- OWASP iGoat