



Spis treści

O autorach	15
O recenzencie technicznym	17
Wstępniak	19
Wprowadzenie	21
Ogólne omówienie książki	22
W jaki sposób zorganizowana jest ta książka?	22
Kto powinien przeczytać tę książkę?	27
Niezbędne narzędzia	27
Co znajdziesz w witrynie internetowej?	27
Rozdział 1. (Nie)bezpieczeństwo aplikacji mobilnych	29
Ewolucja aplikacji mobilnych	30
Najczęstsze kategorie aplikacji mobilnych	31
Zalety aplikacji mobilnych	32
Bezpieczeństwo aplikacji mobilnych	32
Kluczowe czynniki problemu	35
Projekt OWASP Mobile Security	36
Narzędzia rekomendowane przez OWASP Mobile Security	40
Przyszłość dziedziny zapewniania bezpieczeństwa aplikacjom mobilnym	42
Podsumowanie	43

Rozdział 2. Analiza aplikacji iOS	45
Poznajemy model bezpieczeństwa	45
Inicjalizacja systemu iOS za pomocą łańcucha procedur bezpiecznego rozruchu	46
Wprowadzenie Secure Enclave	47
Ograniczenie procesów aplikacji dzięki podpisywaniu kodu	47
Izolacja aplikacji za pomocą piaskownicy na poziomie procesu	48
Ochrona informacji za pomocą szyfrowania przechowywanych danych	48
Ochrona przed atakami dzięki funkcjom ograniczającym możliwość wykorzystania luk w zabezpieczeniach	49
Poznajemy aplikacje iOS	50
Dystrybucja aplikacji iOS	51
Struktura aplikacji	52
Instalacja aplikacji	53
Poznajemy uprawnienia aplikacji	54
Omówienie jailbreakingu	56
Powody przeprowadzania jailbreakingu urządzenia	57
Rodzaje jailbreakingu	58
JailbreakMe v3 Saffron	58
Jailbreak evasi0n	60
Jailbreak evasi0n7	60
Przygotowanie środowiska testowego	61
Przygotowanie podstawowego zestawu narzędzi	62
Przeglądanie systemu plików	67
Pliki typu property list	69
Binarne pliki cookie	69
Bazy danych SQLite	70
Poznajemy API Data Protection	70
Poznajemy pęk kluczy w systemie iOS	73
Kontrola dostępu i polityka uwierzytelniania w iOS 8	75
Uzyskanie dostępu do pęku kluczy w iOS	76
Poznajemy Touch ID	77
Inżynieria odwrotna plików binarnych w iOS	79
Analiza plików binarnych w systemie iOS	79
Identyfikacja funkcji związanych z zapewnianiem bezpieczeństwa	82
Deszyfrowanie plików binarnych pochodzących ze sklepu App Store	85
Analiza odszyfrowanych plików binarnych	88
Deasemblacja i dekompilacja aplikacji iOS	93
Podsumowanie	93

Rozdział 3.	Atakowanie aplikacji iOS	95
	Wprowadzenie do bezpieczeństwa w warstwie transportu	95
	Identyfikacja niebezpieczeństw czyhających w warstwie transportu	96
	CVE-2014-1266: SSL/TLS „Goto Fail”	101
	Przechwytywanie szyfrowanej komunikacji	103
	Niebezpieczeństwa związane z instalacją profili	106
	Ominięcie mechanizmu przypinania certyfikatu	106
	Niebezpieczeństwa związane z instalacją narzędzi pozwalających obejść kwestie zaufania	107
	Identyfikacja niebezpiecznego magazynu danych	107
	Modyfikacja aplikacji za pomocą deasemblera Hopper	111
	Atak na środowisko uruchomieniowe iOS	117
	Poznajemy języki Objective-C i Swift	118
	Instrumentacja środowiska uruchomieniowego iOS	120
	Poznajemy komunikację międzyprocesową	142
	Atak na procedurę obsługi protokołu	142
	Niebezpieczna procedura obsługi w aplikacji Skype	145
	Rozszerzenia aplikacji	145
	Ataki typu injection	147
	Atak injection na komponent UIWebView	147
	Aplikacja Skype na platformie iOS i ataki XSS	150
	Atak typu injection na magazyn danych po stronie klienta	150
	Atak typu injection na analizator składni XML	151
	Atak typu injection na procedurę obsługi pliku	153
	Podsumowanie	154
Rozdział 4.	Identyfikowanie problemów w implementacji aplikacji iOS	155
	Ujawnienie danych osobowych	155
	Obsługa identyfikatora urządzenia	156
	Przetwarzanie książki adresowej	157
	Obsługa danych geolokalizacji	157
	Wykrywanie wycieku danych	158
	Wyciek danych w dziennikach zdarzeń aplikacji	159
	Wykrywanie wycieku danych poprzez schowek systemowy	159
	Obsługa zmiany stanu aplikacji	161
	Buforowanie klawiatury	162
	Buforowanie odpowiedzi HTTP	163
	Uszkodzenie pamięci w aplikacjach iOS	164
	Luki w zabezpieczeniach związane z ciągiem tekstowym formatowania	164
	Próba użycia obiektu po jego usunięciu	167
	Inne problemy związane z implementacjami kodu natywnego	168
	Podsumowanie	168

Rozdział 5.	Tworzenie bezpiecznych aplikacji iOS	169
	Ochrona danych w aplikacji	169
	Ogólne reguły projektowe	170
	Implementacja szyfrowania	171
	Ochrona danych w trakcie transportu	174
	Unikanie luk w zabezpieczeniach pozwalających na przeprowadzenie ataku typu injection	176
	Unikanie ataków typu SQL injection	176
	Unikanie ataków typu XSS	177
	Ochrona aplikacji z użyciem zabezpieczeń binarnych	178
	Wykrycie przeprowadzenia jailbreakingu urządzenia	179
	Zabezpieczenie środowiska uruchomieniowego aplikacji	183
	Zabezpieczenie aplikacji przed modyfikacjami	187
	Implementacja zabezpieczeń antydebugowania	188
	Zaciemnienie aplikacji	189
	Podsumowanie	190
Rozdział 6.	Analiza aplikacji Android	191
	Przygotowanie pierwszego środowiska Android	192
	Poznajemy aplikacje Android	197
	Podstawy systemu Android	197
	Poznajemy pakiety Androida	199
	Użycie narzędzi do przeglądania zasobów Androida	203
	Wprowadzenie do komponentów aplikacji	213
	Spojrzenie pod maskę	218
	ART, czyli zamiennie środowisko uruchomieniowe dla oprogramowania Dalvik	222
	Poznajemy model zapewniania bezpieczeństwa	223
	Podpisanie kodu	223
	Poznajemy uprawnienia	229
	Słowo ostrzeżenia dotyczące taktyki najczęściej stosowanej przez oprogramowanie typu malware	235
	Piaskownica aplikacji	235
	Szyfrowanie systemu plików	237
	Ogólne mechanizmy obronne przed wykorzystaniem luk w zabezpieczeniach	239
	Dodatkowe mechanizmy obronne jądra przed eskalacją uprawnień	241
	Poznajemy kwestię uzyskania uprawnień użytkownika root	241
	Gingerbreak — wykorzystanie luk w zabezpieczeniach kodu jądra AOSP	245
	Exynos — wykorzystanie luk w zabezpieczeniach niestandardowych sterowników	245
	Samsung Admire — nadużycie uprawnień pliku za pomocą dowiązań symbolicznych	246

Acer Iconia — wykorzystanie luk w zabezpieczeniach plików binarnych SUID	247
Błędy klucza głównego — wykorzystanie luk w zabezpieczeniach kodu systemowego AOSP	247
Towelroot — wykorzystanie luk w zabezpieczeniach jądra w systemie Android	248
Modyfikacja w urządzeniu Nexus własnego obrazu przeznaczonego do odzyskiwania systemu	249
Inżynieria odwrotna aplikacji	250
Pobieranie plików APK	251
Wyświetlenie pliku manifestu	252
Wyświetlanie plików XML	253
Wygenerowanie danych wyjściowych do pliku	254
Deasemblacja kodu bajtowego DEX	254
Dekompilacja kodu bajtowego DEX	256
Dekompilacja zoptymalizowanego kodu DEX	259
Deasemblacja kodu natywnego	261
Narzędzia dodatkowe	261
Praca ze środowiskiem ART	262
Podsumowanie	264
Rozdział 7. Atakowanie aplikacji Android	265
Dziwactwa modelu zapewniania bezpieczeństwa	266
Współpraca z komponentami aplikacji	266
Atak na komponenty aplikacji	272
Poznajemy intencje	273
Poznaj Sieve, czyli pierwszą atakowaną aplikację	275
Przeprowadzanie ataku na czynności	279
Kilka słów na temat aliasów czynności	281
Rzeczywisty przykład: CVE-2013-6271 — usunięcie blokady urządzenia z wydania Android 4.3 lub starszego	283
Rzeczywisty przykład — zmiana kodu PIN w urządzeniu bez podania dotychczasowego	287
Wykorzystanie luk w niezabezpieczonych dostawcach treści	289
Użycie istniejących narzędzi do wykrywania SQL injection	295
Rzeczywisty przykład — luki w zabezpieczeniach wielu aplikacji Androida zainstalowanych standardowo w urządzeniach Samsunga	295
Rzeczywisty przykład — aplikacja Shazam	300
Przeprowadzanie ataku na niezabezpieczone usługi	301
Rzeczywisty przykład — usługa schowka w urządzeniach firmy Samsung	302
Błędy podczas kompilacji własnych klas Javy	310
Przeprowadzanie ataku na odbiorcę komunikatów	310
Systemowe komunikaty rozgłoszeniowe	311

Rzeczywisty przykład: CVE-2013-6272 — zainicjowanie lub zerwanie połączenia bez odpowiednich uprawnień w systemie Android 4.4.2 lub starszym	312
Rzeczywisty przykład — zdalne usunięcie zawartości urządzenia Samsung Galaxy	318
Uzyskanie dostępu do pamięci masowej i dzienników zdarzeń	319
Uprawnienia plików i katalogów	319
Rzeczywisty przykład — możliwy do zapisu przez wszystkich użytkowników skrypt DroidWall wykonany przez użytkownika root	322
Praktyki dotyczące szyfrowania pliku	324
Pamięć masowa na karcie SD	325
Rzeczywisty przykład — pamięć masowa aplikacji WhatsApp	326
Rejestracja danych	326
Błędne zastosowanie niezabezpieczonej komunikacji	327
Przegląd ruchu sieciowego	327
CVE-2012-6636 — wykonanie dowolnego kodu zdefiniowanego w metodzie addJavaScriptInterface()	335
Inne mechanizmy komunikacji	336
Inne płaszczyzny ataku	340
Przeprowadzanie ataku na kod natywny	340
Wykorzystanie luk w zabezpieczeniach błędnie skonfigurowanych atrybutów pakietu	347
Przeprowadzenie ataku na aplikację z włączoną opcją debuggable z poziomu innej aplikacji bez szczególnych uprawnień	354
Dodatkowe techniki testowania	355
Stosowanie poprawek w aplikacji	356
Błędy podczas podpisywania pakietu	358
Manipulacja środowiskiem uruchomieniowym	359
Podsumowanie	364
Rozdział 8. Identyfikowanie problemów w implementacji aplikacji Android	365
Przegląd standardowo zainstalowanych aplikacji	365
Wyszukanie aplikacji o szerokich uprawnieniach	366
Wyszukiwanie płaszczyzn dla zdalnego ataku	369
Wyszukiwanie lokalnych luk w zabezpieczeniach	376
Wykorzystanie luk w zabezpieczeniach urządzeń	377
Narzędzia ataku	377
Poznajemy poziomy uprawnień	385
Praktyczne ataki fizyczne	387
Praktyczne zdalne ataki	398
Infiltracja danych użytkownika	426
Użycie istniejących modułów narzędzia drozer	426
Inne techniki do zastosowania w uprawnionych scenariuszach	431
Podsumowanie	436

Rozdział 9.	Tworzenie bezpiecznych aplikacji Android	437
	Reguła najmniejszego odkrycia się	437
	Komponenty aplikacji	438
	Magazyn danych	438
	Współpraca z niezaufanymi źródłami	438
	Żądanie minimalnych uprawnień	438
	Sprawdzenie plików znajdujących się w pakiecie APK	439
	Podstawowe mechanizmy obronne	439
	Przegląd punktów wejścia w komponentach aplikacji	439
	Bezpieczne przechowywanie plików	445
	Bezpieczne udostępnianie plików innym aplikacjom	449
	Prowadzenie bezpiecznej komunikacji	450
	Zabezpieczenie komponentu WebView	453
	Konfiguracja pliku manifestu	455
	Rejestracja zdarzeń	457
	Zmniejszenie ryzyka związanego z kodem natywnym	457
	Skrypt checksec nie działa	458
	Zaawansowane mechanizmy zabezpieczeń	458
	Wykrycie obniżenia poziomu ochrony	459
	Ochrona niewyeksportowanych komponentów	460
	Spowolnienie procesu inżynierii odwrotnej	460
	Zaciemnianie kodu źródłowego	460
	Wykrywanie użycia konta użytkownika root	461
	Wykrycie debugowania	463
	Wykrycie modyfikacji aplikacji	463
	Podsumowanie	464
Rozdział 10.	Analiza aplikacji Windows Phone	467
	Poznajemy model zapewniania bezpieczeństwa	468
	Podpisywanie kodu i DRM	468
	Piaskownica aplikacji	469
	Szyfrowanie przechowywanych danych	471
	Proces zgłaszania aplikacji do sklepu Microsoft Store	472
	Poznajemy funkcje mechanizmów obronnych	474
	Poznajemy aplikacje na platformie Windows Phone 8.x	482
	Pakiety aplikacji	482
	Języki programowania i typy aplikacji	482
	Manifest aplikacji	484
	Katalogi aplikacji	489
	Rozpowszechnianie aplikacji Windows Phone	489
	Przygotowanie środowiska testowego	493
	Narzędzia SDK	494
	Odblokowanie możliwości urządzenia	499

Wykorzystanie dostępu do systemu plików	512
Wykorzystanie dostępu do rejestru	514
Użyteczne narzędzia hakera	514
Analiza plików binarnych aplikacji	515
Proces inżynierii odwrotnej	515
Analiza funkcji mechanizmów obronnych	516
Podsumowanie	517
Rozdział 11. Atakowanie aplikacji Windows Phone	519
Analiza pod kątem punktów wejścia danych	519
Kontrolki WebBrowser i WebView	520
Bluetooth	522
Sesje HTTP	524
Gniazda sieciowe	525
NFC	525
Kod kreskowy	527
Karta SD	528
Interfejsy komunikacji międzyprocesowej	530
Powiadomienia typu toast	532
Atak na warstwę transportową	533
Identyfikacja i przechwytywanie komunikacji w postaci zwykłego tekstu	533
Identyfikacja i przechwytywanie komunikacji HTTPS	537
Przechwytywanie ruchu sieciowego innego niż HTTP i HTTPS	539
Błędy związane z weryfikacją certyfikatu SSL	539
Ataki na kontrolki WebBrowser i WebView	541
Ataki typu XSS	541
Ataki skryptów lokalnych	543
Komunikacja między językami JavaScript i C#	548
Identyfikacja luk w zabezpieczeniach implementacji IPC	549
Procedura obsługi protokołu	549
Procedura obsługi pliku	553
Powiadomienia typu toast	557
Atak na analizator składni XML	566
Wprowadzenie do API XDocument	566
Atak DoS podczas rozwinięcia encji	569
Atak typu XXE	571
Atak na bazę danych	574
API LINQ to SQL	574
SQLite i SQLCipher	575
Atak na procedurę obsługi pliku	579
Wprowadzenie do obsługi pliku	579
Ataki polegające na poruszaniu się po katalogach	581
Modyfikacje podzespołu .NET	584
Podsumowanie	591

Rozdział 12. Identyfikowanie problemów w implementacji aplikacji Windows Phone	593
Identyfikacja niebezpiecznego magazynu danych ustawień aplikacji	594
Wykrywanie wycieku danych	597
Magazyn danych plików cookie HTTP(S)	598
Buforowanie HTTP(S)	599
Rejestracja danych w aplikacji	599
Identyfikacja niebezpiecznego magazynu danych	600
Niezaszyfrowany magazyn danych plików	600
Niezaabezpieczony magazyn bazy danych	603
Niebezpieczne generowanie liczby losowej	607
Przewidywalność System.Random	607
Wiele egzemplarzy System.Random	610
Bezpieczeństwo wątku System.Random	610
Niebezpieczna kryptografia i użycie hasła	611
Klucze kryptograficzne na stałe umieszczone w kodzie	612
Niebezpieczny magazyn kluczy kryptograficznych	612
Przechowywanie klucza lub hasła w niemodyfikowalnym obiekcie ciągu tekstowego	613
Nieudane usunięcie z pamięci klucza kryptograficznego lub hasła	614
Niebezpieczne generowanie klucza	615
Użycie słabych algorytmów kryptograficznych i trybów oraz kluczy o niewystarczającej długości	617
Użycie statycznych wektorów inicjalizacji	620
Błędne użycie API Data Protection na platformie Windows Phone	621
Wykrywanie luk w zabezpieczeniach kodu natywnego	623
Przepełnienie bufora stosu	624
Przepełnienie bufora sterty	626
Inne błędy związane z obsługą liczb całkowitych	628
Błędy ciągu tekstowego formatowania	630
Błędy związane z indeksowaniem tablicy	631
Błędy związane z odmową usług	632
Niebezpieczny kod C#	632
Podsumowanie	633
Rozdział 13. Tworzenie bezpiecznych aplikacji Windows Phone	635
Ogólne rozważania dotyczące bezpiecznego projektu aplikacji	635
Bezpieczne szyfrowanie i przechowywanie danych	636
Bezpieczne algorytmy i tryby szyfrowania	636
Generowanie klucza i zarządzanie nim	636
Szyfrowanie pliku	637
Szyfrowanie bazy danych	639

Bezpieczne generowanie liczby losowej	640
Zapewnianie bezpieczeństwa danych w pamięci i usuwanie zawartości pamięci	640
Uniknięcie ataku typu SQL injection	642
Implementacja bezpiecznej komunikacji	643
Użycie SSL i TLS	643
Weryfikacja certyfikatu SSL i TLS	644
Uniknięcie ataków typu XSS w komponentach WebBrowser i WebView	645
Użycie SSL i TLS w komunikacji sieciowej	645
Wyłączenie obsługi JavaScript	646
Bezpieczne tworzenie dynamicznego kodu HTML i JavaScript	646
Unikanie ataków przeprowadzanych przez skrypty lokalne	647
Bezpieczne przetwarzanie danych XML	647
Usunięcie bufora internetowego i plików cookie	648
Usunięcie plików cookie	648
Usunięcie bufora internetowego	649
Unikanie błędów kodu natywnego	649
Użycie funkcji mechanizmów obronnych	650
Podsumowanie	651
Rozdział 14. Tworzenie aplikacji mobilnych niezależnych od platformy	653
Wprowadzenie do aplikacji mobilnych niezależnych od platformy	653
Zastosowanie funkcjonalności natywnej	655
Udostępnienie funkcjonalności natywnej w systemie Android	656
Udostępnienie funkcjonalności natywnej w systemie iOS	657
Udostępnienie funkcjonalności natywnej w systemie Windows Phone	658
Poznajemy frameworki PhoneGap i Apache Cordova	659
Funkcje standardowe PhoneGap	659
Zapewnienie bezpieczeństwa aplikacji utworzonych za pomocą frameworków PhoneGap i Cordova	660
Wiele luk w zabezpieczeniach frameworka Cordova	661
Ominięcie białej listy we frameworku Cordova dla protokołu innego niż HTTP	663
Podsumowanie	664
Skorowidz	665