

Spis treści

PRZEDMOWA	15
WSTĘP	17
CZĘŚĆ I. BRANŻA	21
1	
WYBÓR PROGRAMU BUG BOUNTY	23
Stan branży	23
Typy zasobów	24
Serwisy i aplikacje społecznościowe	25
Ogólne aplikacje internetowe	26
Aplikacje mobilne (Android, iOS i Windows)	26
Interfejsy API	27
Kod źródłowy i programy wykonywalne	27
Sprzęt i internet rzeczy	28
Platformy bug bounty	28
Zalety	29
Wady	29
Zakresy, wypłaty i czasy reakcji	30
Zakres programu	30
Kwoty wypłat	31
Czas reakcji	32
Programy prywatne	32
Wybór właściwego programu	33
Szybkie porównanie popularnych programów	34
2	
JAK ZAPRACOWAĆ NA SUKCES?	36
Pisanie dobrych raportów	36
Krok 1. Wymyśl opisowy tytuł	37
Krok 2. Napisz przejrzyste podsumowanie	37
Krok 3. Uwzględnij ocenę dotkliwości	37

Krok 4. Podaj szczegółowe kroki do odtworzenia ataku	39
Krok 5. Przedstaw dowód koncepcji	40
Krok 6. Opisz wpływ i scenariusze ataku	40
Krok 7. Zarekomenduj możliwe środki zaradcze	41
Krok 8. Sprawdź raport	41
Dodatkowe wskazówki dotyczące pisania lepszych raportów	41
Budowanie relacji z zespołem programistycznym	43
Stany raportu	43
Radzenie sobie z konfliktami	45
Budowanie partnerstwa	45
Dlaczego nie odnosisz sukcesu?	46
Dlaczego nie znajdujesz błędów?	46
Dlaczego Twoje raporty są odrzucane?	48
Co robić, gdy utkniesz w miejscu?	50
Krok 1. Zrób sobie przerwę!	50
Krok 2. Zbuduj zestaw umiejętności	51
Krok 3. Zyskaj nową perspektywę	51
Na koniec kilka słów z doświadczenia	52

CZĘŚĆ II. ZACZYNAMY! 53

3	
JAK DZIAŁA INTERNET?	55
Model klient-serwer	55
System nazw domenowych	56
Porty internetowe	57
Żądania i odpowiedzi HTTP	58
Kontrola bezpieczeństwa w internecie	60
Kodowanie zawartości	60
Zarządzanie sesją i pliki cookie HTTP	61
Uwierzytelnianie oparte na tokenach	62
Tokeny sieciowe JSON	63
Reguła tego samego pochodzenia	66
Ucz się programować	67

4	
KONFIGURACJA ŚRODOWISKA I PRZECHWYTYWANIE RUCHU	68
Wybór systemu operacyjnego	68
Konfigurowanie narzędzi podstawowych — przeglądarka i serwer proxy	69
Otwieranie osadzonej przeglądarki	70
Konfigurowanie przeglądarki Firefox	70
Konfigurowanie Burpa	73
Korzystanie z Burpa	75
Proxy	75
Intruder	78

Repeater	80
Decoder	81
Comparer	81
Zapisywanie żądań Burpa	82
Ostatnie uwagi na temat robienia notatek	82
5	
REKONESANS PRZED HAKOWANIEM APLIKACJI INTERNETOWYCH	84
Ręczna trawersacja celu	85
Google dorking	85
Wykrywanie zakresu	89
WHOIS i odwrotne WHOIS	89
Adresy IP	89
Parsowanie certyfikatu	91
Enumeracja poddomen	92
Enumeracja usług	93
Brute forcing katalogów	94
Indeksowanie witryn	96
Hostowanie zewnętrzne	98
Rekonesans z wykorzystaniem GitHuba	101
Inne podstawowe techniki białego wywiadu	102
Fingerprinting stosu technologicznego	103
Pisanie własnych skryptów rekonesansowych	106
Podstawy pisania skryptów bashowych	106
Zapisywanie w pliku danych wyjściowych narzędzia	109
Dodanie do danych wyjściowych daty skanowania	111
Dodawanie opcji wyboru uruchamianych narzędzi	111
Uruchamianie dodatkowych narzędzi	112
Parsowanie wyników	116
Tworzenie raportu głównego	118
Skanowanie wielu domen	120
Pisanie biblioteki funkcji	125
Budowanie interaktywnych programów	126
Używanie specjalnych zmiennych i znaków	129
Planowanie automatycznego skanowania	132
Kilka słów na temat interfejsów API rekonesansu	134
Zaczynj hakować!	135
Narzędzia wymienione w tym rozdziale	135
Wykrywanie zakresu	135
Biały wywiad	137
Fingerprinting stosu technologicznego	137
Automatyzacja	138

CZĘŚĆ III. LUKI W ZABEZPIECZENIACH SIECIOWYCH 139

6

CROSS-SITE SCRIPTING 141

Mechanizmy	142
Rodzaje ataków XSS	146
Zapisywany XSS	146
Ślepy XSS	147
Odbijany XSS	148
XSS oparty na modelu DOM	148
XSS własny	150
Zapobieganie	150
Tropienie błędów XSS	152
Krok 1. Szukaj możliwości wprowadzania danych	152
Krok 2. Wstawiaj ładunki	154
Krok 3. Potwierdź działanie ładunku	158
Omijanie ochrony przed atakami XSS	159
Alternatywna składnia JavaScriptu	159
Wielkość liter i kodowanie	159
Błędy logiki filtra	161
Eskalacja ataku	161
Automatyzacja tropienia błędów XSS	162
Szukanie pierwszego błędu XSS!	163

7

OTWARTE PRZEKIEROWANIA 164

Mechanizmy	164
Zapobieganie	166
Tropienie błędów open redirect	166
Krok 1. Poszukaj parametrów przekierowania	166
Krok 2. Użyj Google dorkingu, aby znaleźć dodatkowe parametry przekierowania	167
Krok 3. Przetestuj otwarte przekierowania oparte na parametrach	169
Krok 4. Przetestuj otwarte przekierowania oparte na odsyłaczach	169
Omijanie ochrony przed atakami open redirect	169
Używanie autokorekty przeglądarki	170
Eksploracja logiki wadliwego walidatora	171
Korzystanie z adresów URL danych	172
Eksploracja dekodowania URL	173
Łączenie technik eksploatacji	175
Eskalacja ataku	175
Szukanie pierwszego otwartego przekierowania!	177

8	
PORYWANIE KLIKNIĘĆ	178
Mechanizmy	178
Zapobieganie	184
Tropienie błędów clickjackingu	186
Krok 1. Poszukaj działań zmieniających stan	186
Krok 2. Sprawdź nagłówki odpowiedzi	187
Krok 3. Potwierdź lukę w zabezpieczeniach	187
Obchodzenie zabezpieczeń	188
Eskalacja ataku	189
Uwagi na temat dostarczania ładunku clickjackingu	190
Szukanie pierwszej luki clickjackingu!	191
9	
FAŁSZOWANIE ŻĄDAŃ PRZESYŁANYCH MIĘDZY WITRYNAMI	192
Mechanizmy	192
Zapobieganie	196
Tropienie błędów CSRF	198
Krok 1. Namierz działania zmieniające stan	199
Krok 2. Poszukaj brakujących zabezpieczeń przed CSRF	199
Krok 3. Potwierdź lukę w zabezpieczeniach	200
Omijanie ochrony przed CSRF	201
Wykorzystanie clickjackingu	202
Zmiana metody żądania	202
Obejście tokenów CSRF przechowywanych na serwerze	203
Omijanie podwójnie przesyłanych tokenów CSRF	205
Omijanie kontroli nagłówka referencyjnego CSRF	207
Omijanie ochrony przed CSRF przy użyciu ataku XSS	209
Eskalacja ataku	209
Wykorzystanie luki CSRF do spowodowania wycieku informacji o użytkownikach	209
Wykorzystanie luki CSRF do utworzenia zapisywanego XSS-a własnego	210
Wykorzystanie luki CSRF do przejmowania kont użytkowników	211
Dostarczanie ładunku CSRF-a	212
Szukanie pierwszej luki CSRF!	214
10	
NIEZABEZPIECZONE BEZPOŚREDNIE ODWOŁANIA DO OBIEKTÓW	215
Mechanizmy	215
Zapobieganie	217
Tropienie błędów IDOR	218
Krok 1. Utwórz dwa konta	218
Krok 2. Odkryj oferowane funkcjonalności	219
Krok 3. Przechwytyj żądania	220
Krok 4. Zmieniaj identyfikatory	220

Omijanie ochrony przed IDOR-ami	221
Kodowane i mieszane identyfikatory	222
Wyciekające identyfikatory	223
Przekaż aplikacji identyfikator, nawet jeśli o niego nie prosi	223
Szukaj IDOR-ów ślepych	224
Zmień metodę żądania	225
Zmień typ żądanego pliku	225
Eskalacja ataku	226
Automatyzacja ataku	226
Szukanie pierwszego IDOR-a!	227
11	
WSTRZYKNIĘCIE SQL	228
Mechanizmy	229
Wstrzykiwanie kodu do zapytań SQL	230
Korzystanie ze wstrzyknięcia SQL drugiego rzędu	233
Zapobieganie	234
Tropienie błędów wstrzyknięcia SQL	237
Krok 1. Szukaj klasycznych wstrzyknięć SQL	238
Krok 2. Szukaj ślepych wstrzyknięć SQL	239
Krok 3. Eksfiltruj informacje za pomocą wstrzyknięć SQL	241
Krok 4. Szukaj wstrzyknięć NoSQL	242
Eskalacja ataku	244
Zdobądź wiedzę o bazie danych	244
Uzyskaj dostęp do powłoki internetowej	245
Automatyzacja wstrzyknięć SQL	246
Szukanie pierwszego wstrzyknięcia SQL!	246
12	
SYTUACJE WYŚCIGU	248
Mechanizmy	248
Kiedy sytuacja wyścigu staje się luką w zabezpieczeniach?	250
Zapobieganie	253
Tropienie sytuacji wyścigu	253
Krok 1. Znajdź funkcjonalności podatne na sytuacje wyścigu	254
Krok 2. Wysyłaj żądania jednocześnie	254
Krok 3. Sprawdź wyniki	254
Krok 4. Przygotuj dowód słuszności koncepcji	255
Eskalacja sytuacji wyścigu	255
Szukanie pierwszej sytuacji wyścigu!	256

13	
FAŁSZOWANIE ŻĄDAŃ WYKONYWANYCH PO STRONIE SERWERA	257
Mechanizmy	257
Zapobieganie	259
Tropienie błędów SSRF	260
Krok 1. Znajdź funkcjonalności podatne na ataki SSRF	260
Krok 2. Przekaż potencjalnie podatnym punktom końcowym wewnętrzne adresy URL	262
Krok 3. Sprawdź wyniki	263
Omijanie zabezpieczeń przed atakami SSRF	265
Omijanie list elementów dozwolonych	265
Omijanie list elementów blokowanych	267
Eskalacja ataku	270
Wykonaj skanowanie sieci	270
Pozyskaj metadane instancji	272
Eksploituuj ślepe SSRF	274
Zaatakuj sieć	275
Szukanie pierwszego SSRFI!	276
14	
NIEZABEZPIECZONA DESERIALIZACJA	277
Mechanizmy	277
PHP	278
Java	288
Zapobieganie	291
Tropienie błędów niezabezpieczonej deserializacji	292
Eskalacja ataku	293
Szukanie pierwszej niezabezpieczonej deserializacji!	293
15	
ENCJA ZEWNĘTRZNA XML-A	294
Mechanizmy	294
Zapobieganie	296
Tropienie błędów XXE	297
Krok 1. Znajdź punkty wejścia danych XML-a	297
Krok 2. Przeprowadź testy pod kątem klasycznych XXE	298
Krok 3. Przeprowadź testy pod kątem ślepych XXE	299
Krok 4. Osadź ładunki XXE w różnych typach plików	300
Krok 5. Przeprowadź testy pod kątem ataków XInclude	301
Eskalacja ataku	302
Odczytywanie plików	302
Inicjowanie ataków SSRF	303
Używanie ślepych XXE	303
Przeprowadzanie ataków DoS	306
Kilka słów o eksfiltracji danych przy użyciu XXE	307
Szukanie pierwszego błędu XXE!	309

16

WSTRZYKNIĘCIE SZABLONU	310
Mechanizmy	310
Silniki szablonów	311
Kod wstrzyknięcia szablonu	312
Zapobieganie	315
Tropienie błędów wstrzyknięcia szablonu	315
Krok 1. Szukaj lokalizacji wprowadzania danych przez użytkownika	315
Krok 2. Wykrywaj wstrzykiwanie szablonu dzięki przesyłaniu ładunków testowych	316
Krok 3. Określ stosowany silnik szablonów	317
Escalacja ataku	318
Szukanie dostępu do systemu za pomocą kodu Pythona	319
Ucieczka z piaskownicy przy użyciu wbudowanych funkcji Pythona	320
Przesyłanie ładunków do testowania	323
Automatyzacja wstrzyknięcia szablonu	324
Szukanie pierwszego błędu wstrzyknięcia szablonu!	324

17

BŁĘDY LOGIKI APLIKACJI I USZKODZONA KONTROLA DOSTĘPU	326
Błędy logiki aplikacji	327
Uszkodzona kontrola dostępu	329
Udostępnione panele administracyjne	329
Luki trawersacji katalogów	330
Zapobieganie	331
Tropienie błędów logiki aplikacji i uszkodzonej kontroli dostępu	331
Krok 1. Poznaj swój cel	332
Krok 2. Przechwytyj żądania podczas przeglądania	332
Krok 3. Myśl niesablonowo	332
Escalacja ataku	332
Szukanie pierwszego błędu logiki aplikacji lub uszkodzonej kontroli dostępu!	333

18

ZDALNE WYKONYWANIE KODU	334
Mechanizmy	334
Wstrzyknięcie kodu	335
Załączenie pliku	337
Zapobieganie	339
Tropienie błędów RCE	340
Krok 1. Zbierz informacje o celu	341
Krok 2. Zidentyfikuj podejrzone lokalizacje wprowadzania danych przez użytkownika	341
Krok 3. Prześlij ładunki testowe	341
Krok 4. Potwierdź lukę w zabezpieczeniach	343
Escalacja ataku	343
Omijanie ochrony przed RCE	344
Szukanie pierwszego błędu RCE!	346

19		
BŁĘDY REGUŁY TEGO SAMEGO POCHODZENIA		347
Mechanizmy		348
Eksploatacja mechanizmu CORS		349
Eksploatacja metody postMessage()		351
Eksploatacja JSONP		353
Omijanie SOP-u przy użyciu luki XSS		355
Tropienie błędów obejścia SOP-u		355
Krok 1. Sprawdź stosowanie technik rozluźniania SOP-u		355
Krok 2. Znajdź błędną konfigurację CORS-u		356
Krok 3. Znajdź błędy postMessage		357
Krok 4. Znajdź błędy JSONP		358
Krok 5. Rozważ środki zaradcze		358
Eskalacja ataku		358
Szukanie pierwszego błędu obejścia SOP-u!		359
20		
KWESTIE BEZPIECZEŃSTWA ZWIĄZANE Z POJEDYNCZYM LOGOWANIEM		360
Mechanizmy		361
Współdzielenie plików cookie		361
SAML		363
OAuth		366
Tropienie błędów przejęcia poddomeny		370
Krok 1. Zrób listę poddomen celu		370
Krok 2. Znajdź niezarejestrowane strony		371
Krok 3. Zarejestruj stronę		371
Monitorowanie pod kątem przejęcia poddomeny		372
Tropienie błędów SAML-a		373
Krok 1. Znajdź odpowiedź SAML-a		373
Krok 2. Analizuj pola odpowiedzi		374
Krok 3. Omiń podpis		374
Krok 4. Ponownie zakoduj komunikat		375
Tropienie kradzieży tokenów OAuth		375
Eskalacja ataku		375
Szukanie pierwszego błędu obejścia SSO!		376
21		
UJAWNIEŃ INFORMACJI		377
Mechanizmy		377
Zapobieganie		378
Tropienie błędów ujawnienia informacji		379
Krok 1. Spróbuj ataku trawersacji ścieżek		379
Krok 2. Przeszukaj Wayback Machine		380
Krok 3. Przeszukaj strony wklejania i udostępniania tekstu		382

Krok 4. Zrekonstruuuj kod źródłowy z udostępnionego katalogu .git	382
Krok 5. Znajdź informacje w publicznych plikach	386
Escalacja ataku	387
Szukanie pierwszego błędu ujawnienia informacji	387

CZĘŚĆ IV. TECHNIKI ZAAWANSOWANE 389

22

PRZEPROWADZANIE INSPEKCJI KODU 391

Porównanie testów białej i czarnej skrzynki	392
Podejście szybkie: grep jest Twoim najlepszym przyjacielem	392
Niebezpieczne wzorce	392
Wyciekające sekrety i słabe szyfrowanie	395
Nowe poprawki i nieaktualne zależności	396
Komentarze programistów	397
Funkcjonalności debugowania, pliki konfiguracyjne i punkty końcowe	397
Podejście szczegółowe	398
Ważne funkcje	398
Dane wprowadzone przez użytkownika	398
Ćwiczenie — znajdź luki w zabezpieczeniach	401

23

HAKOWANIE APLIKACJI SYSTEMU ANDROID 404

Konfigurowanie serwera proxy dla urządzeń mobilnych	405
Omijanie przypinania certyfikatu	407
Anatomia APK	407
Narzędzia	409
Android Debug Bridge	409
Android Studio	410
Apktool	410
Frida	410
Mobile Security Framework	411
Tropienie luk w zabezpieczeniach	411

24

HAKOWANIE INTERFEJSÓW API 413

Czym są interfejsy API?	413
Interfejsy RESTful API	415
Interfejsy API SOAP	416
Interfejsy GraphQL API	417
Aplikacje skoncentrowane na API	420
Tropienie błędów API	420
Przeprowadzanie rekonesansu	421
Testowanie pod kątem uszkodzonej kontroli dostępu i wycieków informacji	423
Testowanie pod kątem problemów z ograniczaniem przepustowości	425
Testowanie pod kątem błędów technicznych	425

AUTOMATYCZNE WYKRYWANIE LUK W ZABEZPIECZENIACH ZA POMOCĄ FUZZERÓW	427
Czym jest fuzzing?	428
Jak działa fuzzer internetowy?	428
Proces fuzzingu	429
Krok 1. Określ punkty wstrzyknięcia danych	429
Krok 2. Wybierz listę ładunków	430
Krok 3. Rozpocznij fuzzing	431
Krok 4. Monitoruj wyniki	432
Fuzzing za pomocą Wfuzza	433
Enumeracja ścieżek	433
Brute forcing uwierzytelniania	434
Testowanie pod kątem typowych luk w zabezpieczeniach sieciowych	436
Więcej informacji o Wfuzzie	437
Porównanie fuzzingu i analizy statycznej	437
Pułapki fuzzingu	438
Poszerzanie zestawu narzędzi do zautomatyzowanego testowania	438
SKOROWIDZ	440