WSTĘP

Ta książka wprowadzi Cię do obszernego świata *hackingu etycznego* lub inaczej procesu odpowiedzialnego odkrywania podatności bezpieczeństwa i zgłaszania ich do właścicieli aplikacji. Kiedy sam zaczynałem naukę hakowania, nie tylko chciałem wiedzieć, *jakie* podatności hakerzy znajdują, ale również, *jak* to robią.

Szukałem wszędzie informacji, lecz zawsze pozostawałem z tymi samymi pytaniami:

- Jakie podatności hakerzy znajdują w aplikacjach?
- Jak hakerzy uczyli się o tych podatnościach?
- Jak hakerzy zaczynają infiltrację strony?
- Jak wygląda samo hakowanie? Wszystko jest zautomatyzowane, czy może robione ręcznie?
- Jak mogę zacząć hakować i znajdować podatności?

W końcu znalazłem się na HackerOne, platformie bug bounty stworzonej do łączenia hakerów etycznych z firmami, które szukają osób do przetestowania ich aplikacji. HackerOne pozwala hakerom i firmom upubliczniać błędy, które zostały znalezione i naprawione.

Czytając te ujawnione zgłoszenia, starałem się zrozumieć, w jaki sposób ludzie znajdują podatności oraz jak je wykorzystują. Często musiałem przeczytać to samo zgłoszenie dwa lub trzy razy, zanim je zrozumiałem. Tak samo jak pozostali początkujący, zdałem sobie sprawę z tego, że mógłbym wynieść o wiele więcej z objaśnień znalezionych podatności, gdyby były napisane prostym językiem.

Na tropie błędów. Przewodnik hakerski jest na to autorską odpowiedzią, która pomoże Ci zrozumieć wiele rodzajów podatności w aplikacjach internetowych. Nauczysz się szukać podatności, zgłaszać je, zarabiać na tym oraz, przy okazji, pisać bezpieczny kod. Książka ta nie omawia jedynie pomyślnych przykładów: znajdziesz w niej również błędy i wyciągnięte wnioski, z których wiele należy do mnie.

Do czasu, gdy skończysz czytanie, postawisz swój pierwszy krok w kierunku zwiększania bezpieczeństwa oraz powinieneś być w stanie zarabiać na tym pieniądze.

Dla kogo jest ta książka?

Ta książka została napisana z myślą o początkujących hakerach. Nie ma znaczenia, czy tworzysz strony internetowe, czy projektujesz je, czy jesteś rodzicem przebywającym w domu, 10-letnim dzieckiem, czy 75-letnim emerytem.

Powiedziawszy to, choć nie jest to warunek konieczny do hakowania, pewne doświadczenie w programowaniu i znajomość z technologiami internetowymi mogą okazać się pomocne. Na przykład nie musisz być programistą internetowym, aby być hakerem, jednak znajomość podstawowego hipertekstowego języka znaczników (HTML), struktury strony internetowej, wyglądu kaskadowych arkuszy stylów (CSS) oraz tego, w jaki sposób JavaScript dynamicznie wchodzi w interakcję z witrynami, pomogą Ci odkrywać podatności i oceniać znaczenie znalezionych błędów.

Umiejętność programowania może pomóc, gdy szukasz podatności dotyczących logiki aplikacji oraz przy domysłach, gdzie programista mógł popełnić błędy. Jeśli potrafisz patrzeć z punktu widzenia programistów, zgadywać, w jaki sposób coś zostało zaimplementowane, bądź czytać ich kod (o ile jest dostępny), będziesz miał większą szansę na powodzenie.

Jeśli chcesz nauczyć się programowania, polecam Ci sprawdzić darmowe kursy na platformach Udacity oraz Coursera. Dodatkowe materiały znajdziesz w załączniku B.

Jak czytać tę książkę?

Każdy rozdział, który omawia określony rodzaj podatności, zbudowany jest w następujący sposób:

1. Opis rodzaju podatności.
2. Przykłady podatności.
3. Podsumowanie wraz z wnioskami.

Każdy przykład podatności zawiera następujące części:

1. Moją ocenę trudności w szukaniu i udowadnianiu tej podatności.
2. Adres URL powiązany z miejscem, w którym dana podatność została znaleziona.
3. Link do oryginalnego zgłoszenia bądź artykułu.
4. Data zgłoszenia podatności.
5. Kwota otrzymana za przesłanie informacji.
6. Przejrzysty opis podatności.
7. Wnioski, które warto zapamiętać.

Nie ma potrzeby czytania tej książki od deski do deski. Jeśli znajdziesz konkretny rozdział, którym jesteś zainteresowany, przeczytaj go najpierw. W niektórych przypadkach odnoszę się do pojęć omawianych w poprzednich rozdziałach, jednak robiąc to, staram się odnotować miejsce, w którym zdefiniowałem ten termin, dzięki czemu łatwiej Ci będzie znaleźć odpowiednią sekcję. Podczas hakowania trzymaj tę książkę otwartą.

Co znajdziesz w tej książce?

Oto przegląd tego, co znajdziesz w każdym z rozdziałów:

Rozdział 1: Podstawy Bug Bounty wyjaśnia, czym są podatności oraz programy bug bounty, oraz tłumaczy różnice między klientem a serwerem. Omawia również, w jaki sposób działa internet, a w tym żądania HTTP, odpowiedzi i metody, oraz co to znaczy, że HTTP jest bezstanowy.

Rozdział 2: Otwarte przekierowanie omawia ataki, które wykorzystują zaufanie użytkowników do określonej domeny w celu przekierowywania ich do innej.

Rozdział 3: HTTP Parameter Pollution omawia sposób, w jaki hakerzy manipulują żądaniami HTTP, wstrzykując dodatkowe parametry, którym dana witryna ufa, prowadząc w ten sposób do nieoczekiwanych rezultatów.

Rozdział 4: Cross-Site Request Forgery wyjaśnia sposób, w jaki atakujący może wykorzystać złośliwą stronę, aby przeglądarka ofiary wykonała żądanie HTTP do innej strony. Strona, która otrzymuje żądanie, postępuje tak, jakby żądanie zostało wysłane umyślnie przez ofiarę.

Rozdział 5: HTML Injection i fałszowanie treści tłumaczy, w jaki sposób złośliwy użytkownik może wstrzykiwać własne elementy HTML na docelowe strony internetowe.

Rozdział 6: Carriage Return Line Feed Injection pokazuje, jak atakujący wstrzykują zakodowane znaki do wiadomości HTTP, aby zmienić to, w jaki sposób interpretują je serwery, proxy i przeglądarki.

Rozdział 7: Cross-Site Scripting wyjaśnia, w jaki sposób atakujący wykorzystują strony, które nie filtrują danych wejściowych użytkownika, w celu wykonywania na nich własnego kodu JavaScript.

Rozdział 8: Template Injection pokazuje, jak atakujący wykorzystują template engine'y w przypadku, gdy strona nie filtruje odpowiednio wejścia użytkownika i używa ich w swoich szablonach. Rozdział pokrywa zarówno przypadki po stronie klienta, jak i serwera.

Rozdział 9: SQL Injection opisuje, jak podatność po stronie aplikacji (serwera) może pozwolić atakującemu na zaatakowanie bazy danych.

Rozdział 10: Server-Side Request Forgery wyjaśnia, w jaki sposób atakujący sprawiają, że serwer wykonuje niezamierzone żądania.

Rozdział 11: XML External Entity pokazuje, jak atakujący wykorzystują sposób, w jaki aplikacja parsuje dane wejściowe XML i przetwarza w nich zewnętrzne encje.

Rozdział 12: Zdalne wykonanie kodu dotyczy wykorzystywania serwera bądź aplikacji do uruchomienia na nich własnego kodu.

Rozdział 13: Podatności w manualnej obsłudze pamięci omawia sposoby, w jakie hakerzy wykorzystują zarządzanie pamięcią aplikacji do powodowania niezamierzonych działań, włączając w to możliwość wykonywania własnych poleceń.

Rozdział 14: Przejęcie subdomeny pokazuje, w jaki sposób atakujący może przejąć kontrolę nad subdomeną w imieniu uprawnionej domeny.

Rozdział 15: Race Condition wyjaśnia, jak atakujący wykorzystują sytuacje, w których zmiana warunku początkowego danego procesu ma wpływ na jego końcowy wynik.

Rozdział 16: Insecure Direct Object Reference omawia wszelkie podatności pojawiające się, gdy atakujący ma dostęp do odniesienia do obiektu, do którego nie powinien mieć dostępu, na przykład pliku, rekordu w bazie danych bądź konta.

Rozdział 17: Podatności OAuth pokazuje błędy w implementacji protokołu stworzonego do uproszczenia i ustandaryzowania bezpiecznego uwierzytelniania w aplikacjach internetowych, mobilnych i desktopowych.

Rozdział 18: Podatności w logice i konfiguracji aplikacji wyjaśnia, w jaki sposób atakujący mogą wykorzystać wadliwą logikę w kodzie lub błąd w konfiguracji do wykonania niezamierzonych działań.

Rozdział 19: Poszukiwanie podatności na własną rękę dostarcza wskazówki, gdzie i jak szukać podatności, bazując na moich doświadczeniach i metodologii. Ten rozdział nie jest jednak poradnikiem krok po kroku do włamywania się na stronę.

Rozdział 20: Zgłaszanie podatności omawia, w jaki sposób pisać wiarygodne i informacyjne zgłoszenia podatności, dzięki czemu programy nie odrzucą twoich raportów.

Załącznik A: Narzędzia omawia popularne narzędzia stworzone do hakowania, włączając w to przechwytywanie ruchu sieciowego, enumerację subdomen, wykonywanie zrzutów ekranu i wiele więcej.

Załącznik B: Źródła stanowi listę dodatkowych materiałów do dalszego rozwoju w dziedzinie hackingu. Znajdziesz tu między innymi ćwiczenia online, popularne platformy bug bounty i polecane blogi.

Zastrzeżenie dotyczące hakowania

Patrząc na upublicznione zgłoszenia podatności, a dokładniej na kwoty pieniędzy, które hakerzy za nie otrzymują, naturalnie może się wydawać, że hacking to łatwa i szybka droga do wzbogacenia się. Tak jednak nie jest. Poszukiwanie błędów potrafi być satysfakcjonujące, jednak mało prawdopodobne jest, że napotkasz wiele historii o porażkach, które zdarzają się po drodze (z wyjątkiem tej książki, gdzie dzielę się kilkoma naprawę kompromitującymi opowieściami). Ponieważ będziesz głównie słyszał o sukcesach hakerów, możesz rozwinąć nierealistyczne oczekiwania wobec swojej hakerskiej przygody.

Sukces może przyjść do Ciebie bardzo szybko. Jednak jeśli masz problemy ze znajdowaniem błędów, nie przestawaj się zagłębiać. Programiści zawsze będą pisać nowy kod, a błędy stanowią nieuniknioną część tego procesu. Im bardziej się starasz, tym więcej błędów będziesz znajdował.

W tej sprawie możesz śmiało do mnie napisać na Twitterze @yaworsk i dać znać, jak sobie radzisz. Nawet jeśli nie odnosisz sukcesów, chciałbym to od Ciebie usłyszeć. Bug hunting potrafi być samotną pracą, szczególnie jeśli sprawia Ci trudności. Jednak wspaniale jest też wspólnie świętować, a może nawet znajdziesz coś, co będę mógł zamieścić w następnym wydaniu tej książki.

Powodzenia i szczęśliwego hakowania.