Einführung

Dieses Buch führt in die große Welt des *ethischen Hackings* ein, also dem Prozess, Schwachstellen verantwortungsvoll aufzudecken und diese dem Eigner der Anwendung zu melden. Als ich mit dem Hacken begann, wollte ich nicht nur wissen, *welche* Schwachstellen Hacker gefunden hatten, sondern auch *wie* sie diese Lücken aufgedeckt hatten.

Ich suchte nach Informationen, doch es blieben immer die gleichen Fragen:

- Welche Schwachstellen finden Hacker in Anwendungen?
- Wie finden sie diese Schwachstellen?
- Wie beginnen sie mit der Infiltration einer Site?
- Wie sieht das Hacking aus: Läuft alles automatisiert, oder ist es Handarbeit?
- Wie kann ich selbst mit dem Hacking beginnen und Schwachstellen aufspüren?

Letztlich landete ich bei HackerOne, einer sogenannten Bug-Bounty-Plattform. Ihr Ziel ist es, ethische Hacker mit Unternehmen zusammenzubringen, die nach Hackern suchen, um ihre Anwendungen zu testen. HackerOne umfasst Funk-

tionen, die es Hackern und Unternehmen erlauben, aufgedeckte und behobene Bugs zu veröffentlichen.

Während ich die veröffentlichten HackerOne-Reports las, kämpfte ich damit zu verstehen, welche Lücken die Hacker gefunden hatten und wie man sie ausnutzen konnte. Häufig musste ich den gleichen Report zwei- oder dreimal lesen, um ihn zu verstehen. Mir wurde schnell klar, dass ich (und andere Einsteiger) von leicht verständlichen Erläuterungen realer Schwachstellen sehr profitieren würde. Und so kam es schließlich zu diesem Buch.

Hacking und Bug Hunting ist eine Referenz, die Ihnen dabei hilft, die unterschiedlichen Arten von Sicherheitslücken im Web zu verstehen. Sie werden lernen, wie man solche Schwachstellen findet, wie man sie meldet, wie man dafür bezahlt wird und (gelegentlich) auch, wie man defensiven Code entwickelt. Doch das Buch enthält nicht nur erfolgreiche Beispiele. Es zeigt Ihnen auch Fehler und wichtige Erkenntnisse aus der praktischen Arbeit; viele davon sind meine eigenen.

Wenn Sie mit dem Buch durch sind, haben Sie die ersten Schritte unternommen, um das Web zu einem sichereren Ort zu machen, und sollten dabei auch noch etwas Geld verdienen können.

Wer dieses Buch lesen sollte

Dieses Buch richtet sich an Hacker-Neulinge. Es spielt keine Rolle, ob Sie Webentwickler, Webdesigner, in Elternzeit, ein 10-jähriges Kind oder ein 75-jähriger Rentner sind.

Zwar ist es keine Voraussetzung für das Hacking, doch etwas Programmiererfahrung und die Vertrautheit mit Webtechnologien sind hilfreich. Sie müssen beispielsweise kein Webentwickler sein, um ein Hacker zu werden, doch das Verständnis der HTML-Struktur einer Webseite oder Kentnisse darüber, wie CSS (Cascading Style Sheets) ihr Aussehen definiert und wie JavaScript dynamisch mit Webseiten interagiert, hilft Ihnen dabei, Lücken aufzuspüren und die Auswirkung des entdeckten Bugs zu beurteilen.

Programmieren zu können ist hilfreich, wenn man Schwachstellen sucht, die die Logik einer Anwendung betreffen, und wenn man sich Gedanken darüber macht, welche Fehler ein Programmierer gemacht haben könnte. Wenn Sie sich in den Programmierer hineinversetzen und absehen können, wie er etwas implementiert hat, oder (falls verfügbar) seinen Code lesen können, erhöhen sich Ihre Erfolgsaussichten.

Wenn Sie etwas über Programmierung lernen wollen, finden Sie, u. a. beim dpunkt.verlag, eine Vielzahl hilfreicher Bücher. Sie können sich auch die kostenlosen Kurse auf Udacity und Coursera ansehen. Anhang B führt weitere Ressourcen auf.

Wie man dieses Buch liest

Jedes Kapitel, das einen bestimmten Schwachstellen-Typ beschreibt, hat die folgende Struktur:

1. Eine Beschreibung des Schwachstellen-Typs
2. Beispiele für diese Art von Schwachstelle
3. Eine Zusammenfassung mit Schlussfolgerungen

Jedes Beispiel einer Schwachstelle umfasst:

- meine Einschätzung des Schwierigkeitsgrads, die Schwachstelle aufzuspüren und zu belegen
- den URL mit dem Fundort der Schwachstelle
- einen Link auf den Original-Report oder die Rezension
- das Datum, an dem die Sicherheitslücke gemeldet wurde
- den Betrag, der für die Schwachstelle gezahlt wurde
- eine klare Beschreibung der Schwachstelle
- die Kernpunkte, die man für sein eigenes Hacking nutzen kann

Sie müssen dieses Buch nicht von vorne bis hinten durchlesen. Wenn Sie ein bestimmtes Kapitel besonders interessiert, dann lesen Sie es zuerst. In manchen Fällen spreche ich Konzepte an, die in früheren Kapiteln behandelt wurden. Ich gebe dann aber auch an, wo ein Begriff definiert wurde, damit Sie den entsprechenden Abschnitt schnell finden. Sie sollten das Buch neben sich haben, während Sie hacken.

Was Sie in diesem Buch finden

Hier eine Übersicht dessen, was Sie in den einzelnen Kapiteln finden:

Kapitel 1: Bug-Bounty-Grundlagen erklärt, was Schwachstellen und Bug-Bounties sind, sowie den Unterschied zwischen Clients und Servern. Es erläutert auch, wie das Internet funktioniert, was HTTP-Requests, -Responses und -Methoden sind und was »HTTP ist zustandslos« bedeutet.

Kapitel 2: Offene Redirects behandelt Angriffe, die das in eine gegebene Domain gesetzte Vertrauen ausnutzen, um Nutzer auf eine andere Domain umzuleiten.

Kapitel 3: HTTP-Parameter-Pollution zeigt, wie Angreifer HTTP-Requests manipulieren, zusätzliche Parameter einschleusen (denen die verwundbare Website vertraut) und wie dies zu unerwartetem Verhalten führt.

Kapitel 4: Cross-Site-Request-Forgery zeigt, wie ein Angreifer eine bösartige Website nutzen kann, um einen angegriffenen Browser dazu zu bringen, einen

HTTP-Request an eine andere Website zu senden. Die andere Website agiert dann so, als wäre der Request legitim und gewollt gesendet worden.

Kapitel 5: HTML-Injection und Content-Spoofing erläutert, wie böswillige Nutzer eigene HTML-Elemente in die Webseiten einer angegriffenen Site einschleusen.

Kapitel 6: Carriage Return/Line Feed-Injection zeigt, wie Angreifer codierte Zeichen in HTTP-Nachrichten einfügen, um deren Interpretation durch Server, Proxies und Browser zu verändern.

Kapitel 7: Cross-Site-Scripting erläutert, wie Angreifer Sites ausnutzen, die Benutzereingaben nicht ausreichend prüfen, um eigenen JavaScript-Code auf der Site auszuführen.

Kapitel 8: Template-Injection erklärt, wie Angreifer Template-Engines ausnutzen, wenn Sites die Benutzereingaben nicht ausreichend prüfen, die in den Templates genutzt werden. Das Kapitel enthält client- und serverseitige Beispiele.

Kapitel 9: SQL-Injection beschreibt, wie es Schwachstellen in einer datenbankgestützten Anwendung einem Angreifer ermöglichen, die Datenbank der Site abzufragen oder anzugreifen.

Kapitel 10: Server-Side-Request-Forgery erläutert, wie ein Angreifer einen Server dazu bringt, unbeabsichtigte Netzwerk-Requests durchzuführen.

Kapitel 11: Externe Entitäten bei XML zeigt, wie Angreifer die Art und Weise ausnutzen, in der eine Anwendung XML-Eingaben verarbeitet und externer Entitäten in die Eingabe einbindet.

Kapitel 12: Remote-Code-Execution diskutiert, wie Angreifer einen Server oder eine Anwendung missbrauchen, um eigenen Code auszuführen.

Kapitel 13: Speicher-Schwachstellen erklärt, wie Angreifer das Speichermanagement einer Anwendung ausnutzen, um unerwartetes Verhalten herbeizuführen, einschließlich der möglichen Ausführung eingeschleuster Befehle.

Kapitel 14: Übernahme von Subdomains zeigt, wie es zur Übernahme von Subdomains kommt, das heißt, wie ein Angreifer eine Subdomain einer gültigen Domain kontrollieren kann.

Kapitel 15: Race Conditions erklärt, wie Angreifer Situationen ausnutzen, in denen die Prozesse einer Site ihre Arbeit basierend auf Ausgangsbedingungen abschließen wollen, die während der Ausführung der Prozesse nicht mehr gelten.

Kapitel 16: Insecure Direct Object References beschreibt Sicherheitslücken, die auftreten, wenn ein Angreifer die Referenz auf ein Objekt (eine Datei, einen Datensatz aus einer Datenbank, einen Account) nutzen oder modifizieren kann, auf die er eigentlich keinen Zugriff haben sollte.

Kapitel 17: OAuth-Schwachstellen behandelt Bugs in der Implementierung des Protokolls, das die sichere Autorisierung für Web-, Desktop- und mobile Anwendungen vereinfachen und standardisieren soll.

Kapitel 18: Schwachstellen in Anwendungslogik und -konfiguration erläutert, wie ein Angreifer Fehler in der Programmlogik oder der Konfiguration einer Anwendung ausnutzen kann, um die Site einige unbeabsichtigte Aktionen ausführen zu lassen, die zu einer Schwachstelle führen.

Kapitel 19: Eigene Bug-Bounties gibt Tipps, wo und wie man (basierend auf meiner Erfahrung und Methode) nach Sicherheitslücken suchen kann. Dieses Kapitel ist keine Schritt-für-Schritt-Anleitung zum Hacken einer Site.

Kapitel 20: Bug-Reports diskutiert, wie man glaubwürdige und informative Reports zu Schwachstellen verfasst, damit die entsprechenden Bug-Bounty-Programme ihre Meldungen nicht ablehnen.

Anhang A: Tools stellt beliebte Werkzeuge für Hacker vor, darunter Web-Traffic-Proxies, Subdomain-Auflistung, Screenshots und vieles mehr.

Anhang B: Ressourcen führt zusätzliche Ressourcen auf, die Ihr Hacking-Wissen erweitern. Dazu gehören Online-Trainings, beliebte Bounty-Plattformen, empfohlene Blogs und so weiter.

Ein Disclaimer zum Hacking

Wenn Sie in den Medien über Schwachstellen lesen und sehen, wie viel Geld einige Hacker verdienen, ist es nur natürlich zu glauben, dass das Hacking eine einfache und schnelle Möglichkeit ist, reich zu werden. Doch das ist es nicht. Hacking kann lohnenswert sein, doch Geschichten über das Scheitern auf diesem Weg werden Sie kaum finden (außer in diesem Buch, wo ich einige sehr peinliche Geschichten mit Ihnen teilen werde). Da Sie hauptsächlich von den Erfolgen einiger Hacker hören werden, könnten Sie unrealistische Erwartungen in Bezug auf ihre eigene Hacker-Karriere entwickeln.

Sie können schnell Erfolg haben, doch oft wird es so sein, dass Sie Bugs nicht auf Anhieb finden und sich die Suche dann zeitaufwendig gestaltet. Geben Sie aber nicht auf. Entwickler werden immer neuen Code schreiben, und Bugs finden immer ihren Weg in den Produktiv-Code. Je öfter Sie Bugs suchen, desto routinierter werden Sie sein.

In diesem Sinne ermuntere ich Sie, mir eine Nachricht auf Twitter @yaworsk zu senden und mir zu schreiben, wie es Ihnen mit dem Bug Hunting geht. Selbst wenn Sie keinen Erfolg haben, würde ich gerne von Ihnen hören. Die Jagd nach Bugs kann eine einsame Arbeit sein. Doch es ist auch großartig, miteinander zu feiern, und vielleicht finden Sie ja etwas, das ich in die nächste Auflage dieses Buchs aufnehmen kann.

Viel Glück und viel Spaß beim Hacken!