Gefährliche Lücken

Schwachstellen in Web-Applikationen

Die Betriebssysteme werden immer sicherer. Für Hacker wird es schwerer, über das Betriebssystem Zugriff auf einen fremden Rechner zu erlangen. Daher versuchen sie es über Schwachstellen in Webseiten. Dieser Artikel geht auf drei dieser Angriffe detaillierter ein: Scareware, SQL-Injection und XSS.

Autor: Andreas Wisler

Jede Woche werden über 60 (!) neue Schwachstellen in Web-Applikationen entdeckt. Mit diesen Lücken können Hacker Manipulationen am Inhalt der Webseite vornehmen oder weitere Angriffe ausführen.

Scareware

Eine beliebte Art ist der Faktor Angst und Verunsicherung. Beim Besuch einer scheinbar seriösen Webseite erscheint ein zusätzliches Fenster, das anzeigt, dass der eigene Rechner von einem Virus befallen ist. Dieses Fenster (Pop-Up) ist haargenau einem bekannten Antivirenprogramm nachgebildet. Die Google Forscher haben rund 240 Millionen Webseiten genauer untersucht und sind dabei auf über 11'000 Seiten gestossen, die solche Scareware genannten Falschprogramme vertreiben. Wird nun das angebotene Programm zum Entfernen des Virus heruntergeladen, hat man gleich doppelten Schaden. Das Säuberungsprogramm ist in der Regel nicht kostenlos und die nutzlose Installation öffnet für den Angreifer weitere Hintertürchen.

SQL-Injection

Praktisch keine Webseite kommt mehr ohne eine Datenbank aus. Darin werden Informationen zur Firma, beispielsweise für einen Shop oder Benutzerlogindaten hinterlegt. Ein Eingabefenster verlangt in der Regel nach Benutzernamen und Passwort zur Identifizierung des Benutzers. Stimmen beide Angaben mit den in der Datenbank hinterlegten Daten überein, wird der Zugriff gewährt. Das Ziel des Hackers ist es nun, nicht den Benutzernamen sondern SQL Code in das Eingabefenster einzugeben. SQL ist die Sprache der Datenbank. Sucht jemand im Shop nach einem bestimmten Artikel, «baut» die Webseite die Anfrage in SQL um und schickt diese an die Datenbank. Die Antwort wird wiederum in eine schön formatierte Darstellung umgewandelt. Bei der bereits erwähnten Abfrage nach dem richtigen Benutzernamen und Passwort sieht der Befehl etwa so aus: SELECT * FROM Benutzer WHERE benutzername='<Benutzername>' AND passwort='<passwort>'. Wichtig für unser Beispiel ist der Teil ab WHERE (auf Deutsch Wo). Nur wenn beide Bedingungen, <Benutzername> und <Passwort>, korrekt sind, werden die Daten aus der Datenbank ausgelesen. Falls nicht, kommt es zu einem Fehler. Doch anstelle der geforderten Angaben gibt ein Hacker nun ' or ''=" ein. Dies ergibt für den WHERE-Teil nun benutzername='' or ''='' AND passwort='' or ''=''. Der Teil ''='' ist immer korrekt (leer ist identisch zu leer) und damit auch die Anfrage. Die Datenbank liefert nun in der Regel den ersten Eintrag der Datenbank aus und dies ist in den häufigsten Fällen der Administrator. Mit einer kleinen «Manipulation» der Anfrage hat ein Hacker gute Chancen, sehr hohe Rechte zu erlangen. Dieser triviale Angriff ist zwar sehr einfach abzufangen, trotzdem funktioniert dies auf vielen Webseiten noch immer.

Cross-Site-Scripting

Beim dritten Angriff wird versucht, an vertrauliche Informationen des Benutzers zu gelangen. Im Gegensatz zu Phishing, wo eine Webseite nachgebildet und unter einem

Zum Autor

Andreas Wisler, (Tel.: 052 320 91 20), Dipl. Ing. FH, CISSP, ISO 27001 Lead Auditor, ist Geschäftsführer der GO OUT Production GmbH, welche sich mit ganzheitlichen und produktneutralen IT-Sicherheitsüberprüfungen und -beratungen auseinandersetzt. System Hardening rundet das Profil ab. Regelmässig veröffentlicht er einen informativen Newsletter zu aktuellen Sicherheitsthemen, der kostenlos und unverbindlich auf www.gosecurity.ch (INFONEWS) heruntergeladen werden kann. Für Blickpunkt:KMU beleuchtet er in jeder Ausgabe einen neuen Aspekt der IT-Sicherheit.

ähnlichen Namen ins Internet gestellt wird, manipuliert der Hacker die Originalwebseite und versucht das Opfer auf die veränderte Originalwebseite zu locken. Diese Angriffsmethode nennt sich Cross-Site-Scripting oder kurz XSS. Analog dem vorherigen Beispiel mit den manipulierten SQL-Befehlen wird bei XSS ein Javascript-Befehl in das Formularfeld eingegeben. Javascript ermöglicht es vielen Webseitenbetreibern dynamische Inhalte auszuführen. Das Problem ist, dass die Webserver nicht nur den in der Webseite hinterlegten Javascriptcode ausführen, sondern auch denjenigen, der in das Formularfeld eingegeben wurde. Ein einfacher Test mit <script>alert("Testing XSS vulnerability");</script> zeigt das folgende Fenster (siehe unten)

Natürlich wird dem Opfer nicht eine Fehlermeldung angezeigt, sondern der Javascript-Code wird dazu benutzt, die Originalwebseite dynamisch umzubauen. Anstelle des Original-Anmeldefensters wird ein anderes eingefügt. Gibt der Besucher nun seinen Benutzernamen und sein Passwort ein, werden diese Angaben an den Hacker geschickt. Nun muss natürlich das Opfer noch dazu gebracht werden, die Webseite mit dem veränderten Code aufzurufen. Dies kann beispielsweise in einem Forumseintrag, einem E-Mail oder einem Word-Dokument der Fall sein. Für das Opfer ist es nur sehr schwer herauszufinden, dass noch zusätzlicher Code nachgeladen wurde, denn das Opfer befindet sich wirklich auf der gewünschten Webseite. Aus diesem Grund hat Microsoft einen ersten Schritt gemacht und den Internet Explorer 8 mit einer Schutzfunktion ausgestattet.

Fazit

Die Hacker verlagern ihre Angriffe immer mehr ins Internet. Das Ziel ist es, die Gutgläubigkeit der Opfer auszunutzen. Für die Opfer wird es immer schwerer zu unterscheiden, ob Manipulationen vorgenommen wurden oder nicht. Daher gilt, jeden Link genau anzusehen, bevor man auf diesen klickt. Mit wachsamen Augen können die meisten Angriffe erfolgreich abgewehrt werden. ◆