Tobias Kutzer

(E-Mail: tobias.kutzer@de.ibm.com)
IBM Rational Technical Sales, Certified IT Specialist – Quality Management ist seit 2007 im IBM Rational Technical Sales Team in Deutschland.
Sein Schwerpunkt innerhalb des Rational Portfolios ist das Quality Management, welches neben dem Testmanagement auch die Aspekte Functional, Performance und Security Testing umfasst.

Mit AppScan PCI Compliance erreichen

Für viele Unternehmen ist die Website, inklusive Kundenportal, einer der wichtigsten – wenn nicht der zentrale – Vertriebsauftritt. Wenn Unternehmen mit Kreditkartendaten von Kunden arbeiten, müssen Maßnahmen zur Sicherung dieser Daten ergriffen werden, und zwar obligatorisch. Diese Maßnahmen sind im aktuellen PCI DSS v2.0 (Payment Card Industry Data Security Standard) festgelegt. PCI DSS ist damit eine der wenigen Compliance-Vorgaben, in denen von Unternehmen konkrete Sicherungsaktionen für Kundeninformationen verlangt werden. Doch was bedeutet dies für die betroffenen Unternehmen?

Web-Anwendungen müssen geschützt werden

PCI DSS enthält diverse Abschnitte, die sich mit dem Sichern von Systemen befassen. So wird zum Beispiel im Abschnitt 6 verlangt, dass eine der folgenden Methoden von den Unternehmen zum Schutz von Web-Anwendungen angewendet werden muss: Review und Überprüfung des Anwendungscodes durch eine Security-Organisation oder die Nutzung einer Web Application Firewall. Beim Review der Anwendungen wird zudem aufgeführt, dass sowohl ein automatisierter Source Code Review oder die Nutzung eines Web Application Scanning Tools (Black-Box-Test-Werkzeug), sowie dessen richtiges Aufsetzen und Anwenden, den Vorgaben genügt. Wie in Abbildung 1 dargestellt, kann allein die Problematik der mangelnden Überprüfung von Nutzereingaben Angriffe wie Cross-Site Scripting oder SQL Injection ermöglichen, welche dann dazu führen, dass PCI nicht eingehalten wird.

Welche Maßnahme sind am besten einzusetzen?

Wenn man auf den Einsatz einer Web Application Firewall (WAF) setzt, dann mag dies Unternehmen kurzfristig und vermeintlich schnell in die Lage versetzen, dass man der regulatorischen Maßnahme genügt. Doch ist in diesem Fall die Erfüllung der Vorgabe oft nur Selbstzweck und bietet nur eingeschränkten Schutz für den Webauftritt: Die Nutzung einer Web Application Firewall hat entscheidende Nachteile. Zum einen ist so gut wie allen Unternehmen die Anwendungsentwicklungsabteilung eine andere als diejenige, die für den Betrieb der Infrastruktur und damit auch der WAF zuständig ist. Dabei muss die kleinste Änderung im Webauftritt, sei es die Umbenennung eines Parameters oder die Erweiterung des Webauftritts, sich auch sofort in der Konfiguration der WAF widerspiegeln, was einen extrem hohen Aufwand und Overhead bedeutet. Weiterhin ist eine WAF als solches immer reaktiv. Das heißt es wird versucht, mögliche Lücken in der Anwendung zu stopfen. Dies hat erneut zwei Nachteile: zum einen könnte es die Entwickler dazu veranlassen, noch unnachsichtiger mit dem Thema Sicherheit umzugehen, frei nach dem Motto „die WAF wird es schon richten“. Zudem ist das reaktive Vorgehen nicht wirklich effizient: Denn warum soll man einen hohen Aufwand in ein zusätzliches System investieren, dessen Aufgaben es lediglich ist, die Lücken der Web-Anwendungen zu stopfen? Warum investiert man diesen Aufwand nicht direkt dort, wo er sinnvoller ist, nämlich in der Web-Anwendung selbst?

Web-Anwendungslücken identifizieren und schließen

Sinnvoll ist es, die Energie zum Schutz dort zu investieren, wo sie am effektivsten genutzt werden kann. Im Fall einer Web-Anwendung ist nichts naheliegender als die Web-Anwendung selbst. Denn wenn man die Schwachstellen im Anwendungscode selbst findet und schließt, bedarf es keines weiteren Systems, wie einer WAF, die wiederum Personal, Kosten und Zeit bindet. Wie in PCI DSS beschrieben, gibt es bereits zwei bewährte Methoden, um Sicherheitslücken direkt in Web-Anwendungen zu identifizieren: die statische Analyse des Quellcodes der Anwendung (White Box Testing) und die dynamische Analyse der installierten, laufenden Anwendung (Black Box Testing). Sowohl Black-Box-Tests als auch White-Box-Tests lassen sich auf manuelle und auf automatisierte Art und Weise durchführen. Einer der Vorteile bei der Nutzung automatisierter Tests durch Werkzeuge ist das breite Spektrum für unterschiedliche Technologien und Frameworks, sowie die schnelle und präzise Testausführung. Ein vergleichbares umfassendes Vorgehen würde bei einem rein manuellen Ansatz schnell in tage-

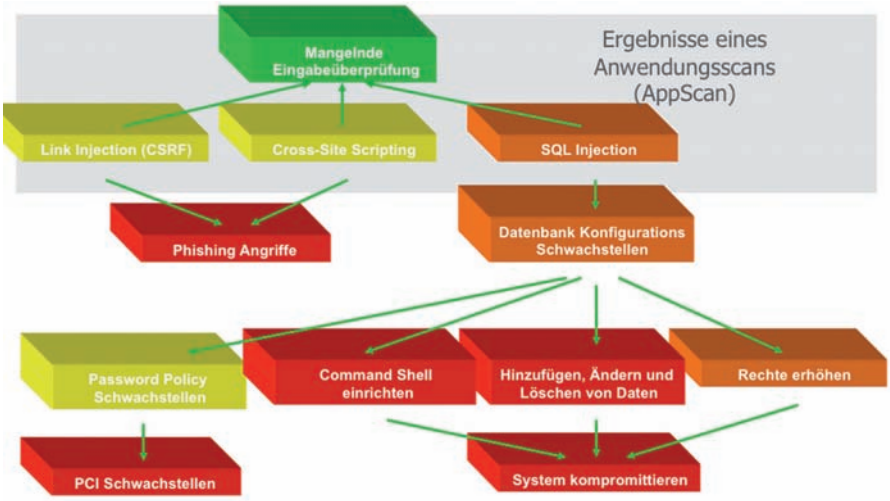

Abb. 1: Ausnutzen von Schwachstellen führt zu PCI-Problemen

bzw. wochenlange Routinearbeit ausufern, um eine gleichwertige Abdeckung an Sicherheitstests zu erreichen. Das automatische Generieren von Berichten über die gefundenen Schwachstellen erleichtert zudem beim werkzeugbasierten Ansatz die wohl mühevollste Arbeit eines manuellen Testers.

Dynamische und statische Analysen mit AppScan

Um den Vorgaben von PCI DSS gerecht zu werden, bietet IBM Werkzeuge, die beide Welten – dynamische und statische Analyse – abdecken.

Das Aufspüren von Sicherheitslücken mittels Black-Box-Tests, wie mit AppScan Standard, lässt sich am besten als cleveres Erraten über manipulierte HTTP-Anfragen beschreiben. Dieses Erraten bzw. Herantasten an eine Schwachstelle macht unter anderem auch den Reiz für Hacker aus, die auf ähnliche Art und Weise vorgehen. Der große Vorteil einer solchen Untersuchung liegt darin, dass man nahezu unabhängig von der verwendeten Technologie jede auf HTTP(S) basierende Anwendung mit den gleichen Tests prüfen kann. Diese Black-Box-Tests betrachten üblicherweise das gesamte System, d. h. Server (Applikationsserver, Webserver, Datenbankserver, etc.), externe Schnittstellen, Netzwerk, Firewalls und Drittanbieter-Komponenten. Ein Vorteil bei AppScan Standard liegt unter anderem in der hybriden Analyse von Web-Anwendungen. Das bedeutet, dass neben der klassischen dynamischen Analyse parallel auch eine statische Analyse des client-seitigen Javascript-Codes durchgeführt wird, um so z. B die Schwachstelle DOM-basiertes Cross-Site-Scripting ausfindig zu machen.

Das Aufspüren von Sicherheitslücken mittels White-Box-Test, wie mit der AppScan Source Edition, basiert auf der Untersuchung des Quellcodes einer Anwendung. Dies gleicht einem Code Audit, z. B auf unternehmensspezifische Programmiervorgaben oder Security Coding Guidelines. Um eine solche Untersuchung durchzuführen, ist eine sehr gute Kenntnis der verwendeten Programmiersprachen und der eingesetzten Frameworks notwendig. Mit dieser Form der Untersuchung lassen sich Schwachstellen auch unabhängig von der Systemkonfiguration finden. Zwei große Vorteile dieser Vorgehensweise: der Quellcode steht zur Verfügung und so kann theoretisch jedes mögliche Verhalten der Anwendung in jedem Winkel des Quellcodes nachvollzogen werden. Zudem kann man die statische Analyse wesentlich früher als die dynamische Analyse im Entwicklungsprozess einsetzen.

Fazit und Ziel: Reports für PCI DSS auf Knopfdruck erzeugen

Der effektivste Weg, um Angriffe auf Web-Anwendungen zu verhindern, ist es, die Schwachstellen direkt in den Web-Anwendungen zu erkennen und zu schließen. Sowohl AppScan Standard als auch die AppScan Source Edition sind spezielle Werkzeuge für diesen Zweck. Zudem können beide maßgeschneiderte Berichte passend auf PCI DSS erstellen, und das auf Knopfdruck. Dies erlaubt es, innerhalb kürzester Zeit auf Bedrohungen und interne oder externe Compliance-Anfragen zum Thema PCI DSS schnell und aktuell reagieren zu können.