Graf Arved von Stackelberg

(E-Mail: arved.stackelberg@hp.com)
ist Country Manager von Fortify, einem Tochterunternehmen von HP im Bereich Security, das auf Lösungen für eine sichere Softwareentwicklung spezialisiert ist. Seit mehreren Jahren ist Graf von Stackelberg für die Leitung des Betriebs der Fortify Aktivitäten im deutschsprachigen Raum verantwortlich und war während dieser Zeit in eine Vielzahl von großen Sicherheitsprojekten involviert. Zuvor leitete er über 10 Jahre das Zentraleuropageschäft von mehreren Softwareunternehmen, die sich auf Qualität in der Softwareentwicklung spezialisiert haben.

Applikationssicherheit in der Cloud – sicherer Code für wolkige Zeiten

Applikationen bilden heute das Hauptziel von Hackern: Gartner-Analyst John Pescatore warnte schon vor Jahren, dass bis zu 75 Prozent der Angriffe auf Webserver auf der Applikationsebene stattfinden [mzd]. Der Grund dafür: Anwendungen sind zunehmend komplex geworden und müssen heute deutlich mehr als früher untereinander sowie nach außen kommunizieren. Die Methoden der klassischen Netzwerksicherheit sind damit hoffnungslos überfordert, auch Application Firewalls bringen aufgrund schwieriger Handhabung nicht den gewünschten Erfolg. Der effektivste Weg für den sicheren Anwendungsbetrieb ist, die Applikationen von vornherein sicher zu entwickeln.

Die Hacker von heute arbeiten viel professioneller und ausdauernder als ihre Vorgänger aus den „Kindertagen des Internets“ – und wesentlich zielgerichteter. Wie sieht ein Hackerangriff heute aus? Hier ein Beispiel aus dem Bankenumfeld:

Ein Angreifer (oder eine organisierte Bande) erfährt aus einer einschlägigen Internetseite von der XSS-Anfälligkeit (Cross-Site Scripting) einer Online-Banking-Website. An Tausende Empfänger – E-Mail-Adressen stehen massenhaft preiswert zum Verkauf – sendet er eine Phishing-E-Mail, es liege auf der Banken-Site eine Nachricht vor. Bankkunden loggen sich auf der Site der Bank ein und per XSS greift der Hacker ihre Logins und Passwörter ab, bevor die Besucher sich verwundert unverrichteter Dinge wieder ausloggen. Damit hat der Hacker mittels mehrerer Logins Zugriff auf das Self-Service-System der Bank.

Durch ausdauerndes Ausprobieren unter verschiedenen Benutzeridentitäten findet er einen Fehler im Message-System, mit dem Kunden technischen Support bei der Bank anfordern: Er entdeckt, dass die Eingabe einer „0“ in das Textfeld eine Privileg-Eskalation bewirkt, und erhält damit Admin-Rechte auf dem Server. Diesen Admin-Zugang nutzt der Hacker nicht selbst, sondern verkauft ihn an Dritte – vermutlich aus Kreisen des organisierten Verbrechens.

Diese Personen wiederum gehen trickreicher vor, als nun einfach die Konten „abzuräumen“; vielmehr schicken sie an alle Bankkunden eine Kaufempfehlung für billige Wertpapiere („Penny Stocks“), die sie im Vorfeld erworben haben. Viele Kunden folgen dieser Kaufempfehlung, der Kurs der Penny Stocks steigt massiv an, und die Bankbetrüger verkaufen ihre Wertpapiere eine Stunde später mit Millionengewinn. Die betroffene Bank hat den Trick übrigens drei Jahre lang nicht bemerkt. Ähnliches ist bereits mehrfach passiert.

Professionalisierung der Angreiferszene

Auffällig ist der hohe Grad der Arbeitsteilung auf Angreiferseite: Die einen handeln mit E-Mail-Adresskontingenten für Spam und Phishing, andere hacken – oft in monate-, sogar jahrelanger Kleinarbeit – das System und wieder andere, meist organisierte Betrügerbanden, kassieren umso schneller ab.

Solch ein Vorgehen ist beileibe kein Einzelfall: Die Studie „Die IT-Sicherheitsbranche in Deutschland“ von Dr. Rainer Bernnat et al. [Ber10], erstellt im Auftrag des Bundesministeriums für Wirtschaft und Technologie, konstatiert eine „zunehmende Professionalisierung bzw. Industrialisierung der Angriffe“. Angriffe würden heute, so die Studie, „vermehrt gezielt geplant und zentral gesteuert“, via Zero Day Exploits nutze man entdeckte Sicherheitslücken sehr schnell aus und Hacker verfolgten konkrete kommerzielle Ziele: „Mittlerweile werden derartige Angriffe

und die Nutzung der entsprechenden Werkzeuge als Dienstleistung gegen Entgelt angeboten“, heißt es in der Studie. „Hierbei erfolgt die Preisbildung nach Art eines funktionierenden Marktes durch Angebot und Nachfrage.“

Steigende Abhängigkeit von der IT

Diese Industrialisierung der Angriffe auf IT-Systeme findet vor dem Hintergrund rasant steigender Abhängigkeit der Unternehmen und Privatpersonen von der IT statt: Immer mehr Geschäftsprozesse laufen applikationsgestützt ab, immer mehr Unternehmen und Haushalte sind breitbandig mit dem Internet verbunden und immer mehr Dienste – von Voice over IP über Video bis hin zu sozialen Netzen – arbeiten internetbasiert. Da Netzwerke und Server heute häufig mit ausgefeilten Mechanismen abgesichert sind, richtet sich das Augenmerk der Angreifer auf die Anwendungen (siehe Abbildung 1).

Verstärkt wird diese Entwicklung durch das Cloud Computing: „Der gegenwärtige Trend, die Bereitstellung von IT-Anwendungen zu virtualisieren und den Speicherort von Daten zu flexibilisieren (z. B. in Form von Cloud Computing), führt zu einer Aufweichung der Außengrenzen (engl. perimeter) von Unternehmensnetzen und Anwendungen“, so Bernnat et al. [Ber10]. Derart flexible virtualisierte Anwendungen seien nur schwer mit vertrauten Mechanismen wie Firewalls, Intrusion-Detection-Systemen (IDS) oder Content Filtering zu schützen, warnen die Autoren der Studie.

Zwar bietet die Security-Branche mit WAFs (Web Application Firewalls) längst Equipment zum Schutz der Webanwendungen. Da die Applikationsentwicklung allerdings heute ständig im Fluss ist, gestaltet sich das andauernde Nachziehen von Änderungen auf den WAFs so aufwendig, dass viele Administratoren die Geräte nur noch zum Monitoring der Anwendungen verwenden und nicht wirklich als Firewall. Sprich: Viele Unternehmen setzen unter verschärften Bedingungen weiterhin auf alte, aber nicht unbedingt bewährte Schutzmechanismen.

Cloud: viel Neues – und viele alte Probleme

Cloud Computing wird dieser Tage gerne als „Revolution“ gehandelt. Denn aktuelle Virtualisierungs- und Automatisierungs-technik ermöglicht eine derart flexible Bereitstellung von IT-Ressourcen (Infrastruktur, Entwicklungsplattformen und Software per Internet („as a Service“, „aus der Cloud“), durch die ganz neue Wege der Kapazitätsauslastung und der Servicebereitstellung möglich sind. Komplette Geschäftsfunktionen werden in die Internet-Wolke verlagert – etwa mit der Cloudbasierten CRM-Lösung von Salesforce.com – und manche Anbieter sprechen gar schon vom „Business Process as a Service“.

Aus Developer-Sicht bedeutet dieser Trend schlicht: Anwendungen werden immer öfter über ein Web-Frontend bereitgestellt, sind immer engmaschiger über Webservice-Protokolle wie SOAP miteinander verknüpft und erlauben Benutzern wie eben auch Angreifern damit immer öfter direkt oder indirekt den Durchgriff auf Backend-Systeme. Auch Vertrauensbeziehungen zwischen den IT-Systemen werden immer komplexer, über Protokolle wie OAuth authentifizieren sich Anwendungen untereinander gegenseitig.

Dennoch sind die häufigsten Risiken, Schwachstellen und Angriffsmethoden alte Bekannte. So veröffentlicht das auf Anwendungssicherheit fokussierte Entwicklerprojekt OWASP (Open Web Application Security Project) jedes Jahr einen Report mit den zehn größten Risiken und auch der Report für 2010 (siehe Kasten 1) zeigte wieder: Entwickler missachten altbekannte Schwachstellen.

Zu den größten Risiken zählen nach wie vor Code Injection (SQL-, LDAP- oder OS-Injections), Cross-Site Scripting sowie Lücken in Authentifizierung und Session-Management. Lediglich eines der „Top-Ten-Risiken“ – Cross-Site Request Forgery (CSRF) – ist vergleichsweise jüngeren Datums. Der Rest – etwa mangelnde Restriktion der URL-Zugriffsrechte – sind Lücken, die leider immer wieder auftreten.

Denn bei den meisten heutigen Anwendungen handelt es sich eben nicht um speziell für das Hochrisikoumfeld Cloud entwickelte Applikationen, sondern um teils Jahrezehnte alte, ursprünglich interne Anwendungen, die man nachträglich „webifiziert“ hat. Selten wurden sie gemäß den rigorosen Qualitätssicherungskriterien einer sicherheitsorientierten Applikationsentwicklung programmiert.

Hinzu kommt Software, die Unternehmen von externen Entwicklungspartnern übernommen, aber nie ausreichend „auf Herz und Nieren“ abgeklopft haben, oder Open-Source-Software, die man – weniger aus Gutgläubigkeit als vielmehr aufgrund stets hohen Zeitdrucks – mehr oder weniger ungeprüft mit in die Applikationslandschaft eingebaut hat.

Entwickler machen es den Hackern häufig zudem durch Nachlässigkeit oder Bequemlichkeit allzu leicht. Auch im eingangs genannten Beispiel nutzte der Hacker einen Entwicklerfehler aus: Ein Developer hatte eine Testroutine („0“ für Admin-Zugriff) programmiert und diese wurde – versehentlich (?), aus Leichtsinn (?) – im Code belassen. Mitunter hinterlegen Entwickler sogar Passwörter im Code, um sie sich nicht merken zu müssen. Solche kompromittierenden Restbestände von Testläufen zu entdecken, kann Monate dauern, aber die heutige Angriffsindustrie ist angesichts hoher Gewinne hartnäckig. Hinzu kommen Backdoors, die bösartige Insider im Unternehmen oder bei Outsourcern vorsätzlich einbauen, um sie dann weiterzuverkaufen.

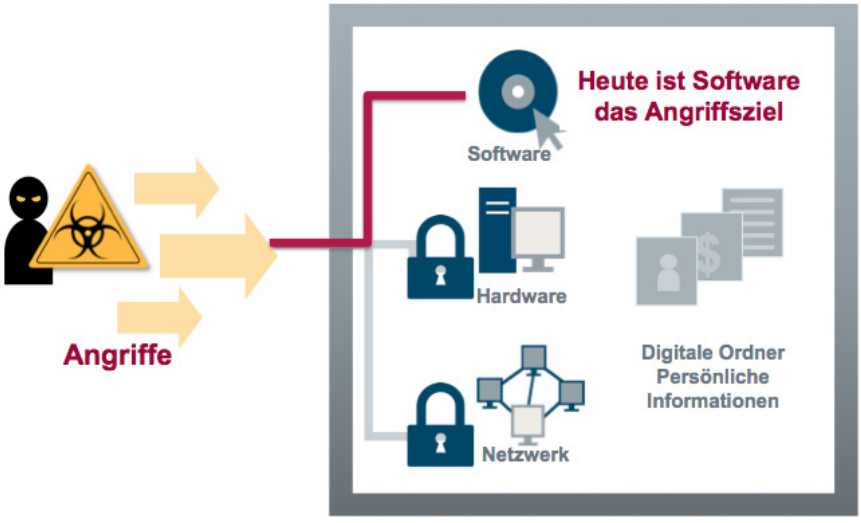

Abb. 1: Hacker-Angriffe setzen verstärkt auf Applikationsebene an.

Mehr Risikobewusstsein gefragt

Zwei Folgerungen liegen nahe: Erstens müssen Softwarehäuser und Unternehmen mit hausinternem Software-Development unter ihren Entwicklern durch Fortbildungen und Awareness-Trainings ein größeres Risikobewusstsein schaffen. Zweitens sind Entwicklungsprozesse erforderlich, in denen eine Security-Validierung von Software bereits frühzeitig, also während der Entwicklung ansetzt. Auch in puncto Security gilt eine Maxime der Qualitätssicherung: „Test early, test often."

Eine Hürde stellt der Umstand dar, dass viele Unternehmen mit Reifegradmodellen (Maturity Models) für sichere Entwicklungszyklen nicht vertraut sind. Für die Optimierung der Entwicklungsprozesse verwendet HP das hausintern erstellte Prozessmodell SSA (Software Security Assurance), Know-how rund um SSA hat HP in das OWASP-Projekt eingebracht. Inzwischen liegt mit SAMM (Software Assurance Maturity Model) Version 1.0 sogar ein Framework für sichere Softwareentwicklung vor. SAMM ist auf www.opensamm.org frei verfügbar [SAM].

Prozesse und Tools für sichere Entwicklung

Mit Lösungen für die Source-Code-Analyse wie HP Fortify SCA aus HPs Application-Security-Portfolio (siehe Kasten 2) erhalten Entwicklungsteams Werkzeuge, um Best Practices für die sichere Softwareentwicklung in die Tat umzusetzen und den Erfolg laufend zu kontrollieren. HP Fortify SCA ist als Standalone-Lösung erhältlich, aber auch als Plug-in für IDEs wie Visual Studio oder Eclipse.

In der Praxis funktioniert dies wie folgt: Der Entwickler lädt neu erstellten Code auf den Build-Server hoch, dort läuft Fortify im Hintergrund und prüft den Code auf Schwachstellen – natürlich nicht nur auf die OWASP Top Ten, verfügt HP doch über die wohl größte Datenbank für die Source-Code-Fehlerfindung am Markt. Die Ergebnisse dieser Prüfung werden automatisch wieder in die IDE zurückgespielt.

Entwickler bekommen somit zeitnah Feedback darüber, wo Lücken bestehen, und zugleich erhalten sie auf der Basis der Fortify-Knowledge-Base Hilfestellung für die Verbesserung des Codes. So können die Developer mit jedem Einschecken von Code ihr Know-how über eine sichere Entwicklung stetig ausbauen, was nicht nur der Qualität der Software, sondern letztlich auch ihrer Karriere zugutekommt (siehe Abbildung 2).

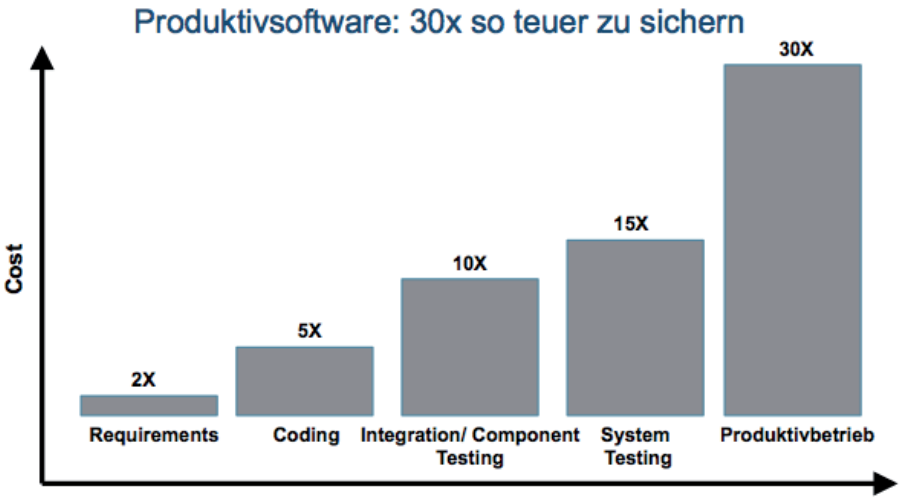

Abb. 2: Sicherheitslücken bereits während der Anwendungsentwicklung zu beheben, ist deutlich preisgünstiger, als sie nachträglich aus einem Produktivsystem zu entfernen.

Gerade bei umfangreichen Softwareprojekten oder der Erweiterung von Legacy-Code kann anfangs eine Flut von Sicherheitswarnungen auftreten. Man sollte deshalb dafür im Entwicklungsprozess ausreichend Zeit einplanen. Dies muss aber kein Problem darstellen: Frühzeitig erkannte und ausgebesserte Lücken im Code ersparen später in den Testphasen ein Vielfaches der anfangs investierten

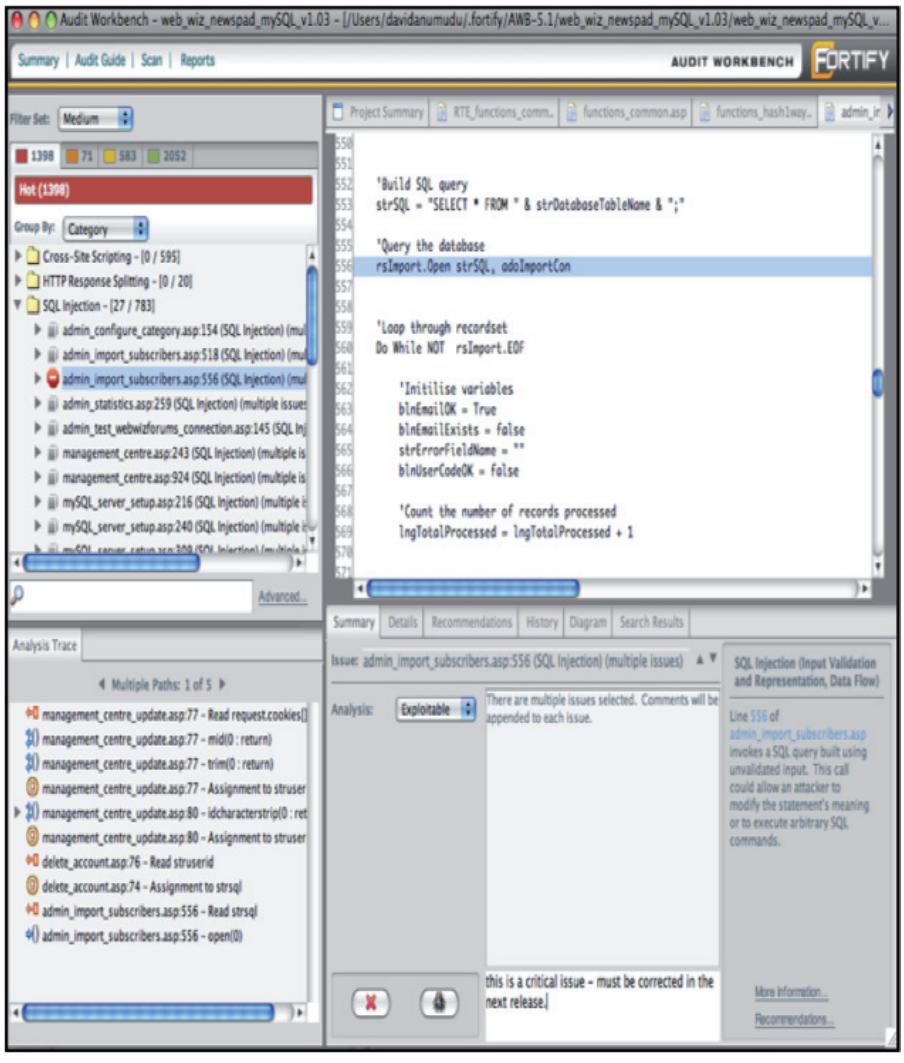

Abb. 3: HP Fortify SCA gibt Feedback über den Sicherheitszustand programmierten Codes.

Zeit, da deutlich weniger Issues nachträglich zu beheben sind.

Auf abschließende Penetration-Tests kann aber nicht verzichtet werden: Die Source-Code-Analyse zeigt Mängel auf Code-Ebene auf, nicht in der Logik der Software. Verweist ein Code etwa auf eine Datenbanktabelle, die es gar nicht gibt, so ermittelt dies erst der Penetration-Test. Die wirkungsvolle Kombination beider Testverfahren, wie sie auch HP empfiehlt und unterstützt, nennt sich „Hybrid Analysis". Sie sorgt für maximale Sicherheit. Das Verhältnis gefundener Issues liegt dabei im Schnitt ungefähr bei 90:10 (Source-Code-Analyse zu Pen-Test) (siehe Abbildung 3).

HP Fortify bietet zahlreiche Funktionen, um einer Überlastung und Demotivation der Entwickler entgegenzuwirken. So erhält der Entwickler die Aufstellung gefundener Fehler im übersichtlichen Baumdiagramm, in Kategorien unterteilt und gewichtet nach Kritikalität. Damit die Entwickler trotz einer Vielzahl von Issues stets wissen, welche Schritte am dringlichsten sind, kann der CISO, Entwicklungsleiter oder Security Team Lead mit HP Fortify eigene Regeln der Gewichtung erstellen. So steuert er in jeder Projektphase genau, wie sein Team den Sicherheitsmängeln begegnet.

Im Cloud-Zeitalter müssen Fragen der Sicherheit bei der Anwendungsentwicklung viel stärker in den Vordergrund rücken. Wünschenswert wäre ein Kulturwandel in den Entwicklungsabteilungen hin zu mehr Risikobewusstsein. Mit Prozessmodellen wie SAMM und Werkzeugen für die sichere Entwicklung wie HP Fortify stehen Hilfsmittel bereit, die den Weg zur Entwicklung sicherer Applikationen ebnen.

Top-10-Risiken der Applikationssicherheit laut OWASP (2010)

1. Injection (SQL-, LDAP-, OS-Injection etc.)
2. Cross-Site Scripting (XSS)
3. Lücken in Authentifizierung und Session-Management
4. Unsichere direkte Verweise auf Objekte
5. Cross-Site Request Forgery (CSRF)
6. Security-Fehlkonfiguration
7. Unsicherer kryptografischer Speicher
8. Mangelnde Restriktionen beim URL-Zugang
9. Unzureichender Schutz der Transportschicht
10. Nicht-validierte Redirects und Forwards

Das HP Application Security Portfolio

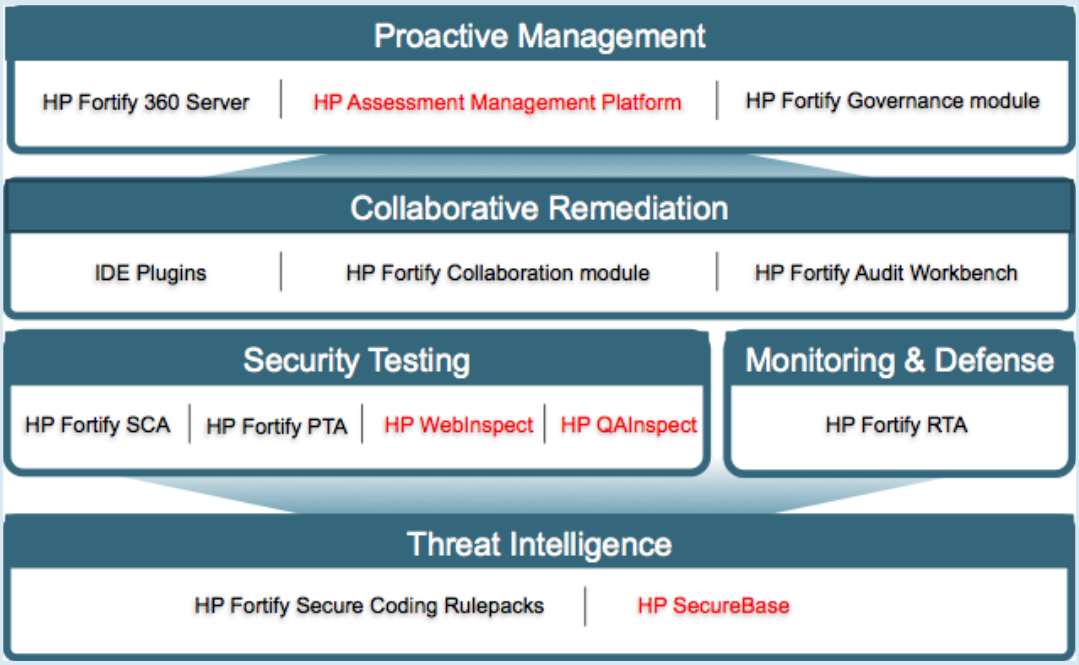

Abb. 4: HP bietet ein umfassendes Portfolio für Applikationssicherheit.

HP hat jüngst sein Portfolio für die Anwendungsmodernisierung um sieben neue Produkte und Services erweitert. Die neuen Lösungen sollen Softwarehäusern und Unternehmen helfen, ihre Applikationslandschaften flexibel zu gestalten, sodass sie Innovationen schnell auf den Markt bringen können.

Zum Portfolio gehört auch eine neue Generation von Tools für die Applikationssicherheit: HP Application Security Center 9.0 (inklusive HP Application Management Platform, HP Web-Inspect und HP QAInspect) sowie HP Fortify 360 3.0 (mit HP Fortify SCA, PTA, RTA, Governance und on Demand).

Die Lösungen helfen Softwarehäusern, Unternehmen und Behörden, sich vor Schadsoftware zu schützen und Risiken zu reduzieren. HP Fortify Real-Time Hybrid Analysis bietet ein Analysewerkzeug, das Softwareschwachstellen in Echtzeit erkennt. Mit Fortify on Demand steht außerdem eine gehostete Security-Testing-Lösung für Applikationen zur Verfügung.

Applikationen in der Cloud: Was zu beachten ist

Bei der Entwicklung von Cloud-Applikationen gibt es im Vergleich zu „herkömmlichen“ Webanwendungen keinen nennenswerten Unterschied: Was zur Absicherung von Web-Applikationen ratsam ist, gilt ebenso für die Cloud. Einige beachtenswerte Unterschiede gibt es allerdings im Hinblick auf den Applikationsbetrieb:

1. *Vertragsgrundlage:* Da Cloud-Applikationen in der Regel von einem externen Cloud-Provider gehostet werden (außer beim internen Betrieb einer Private Cloud), ist eine vernünftige vertragliche Grundlage sehr wichtig. Der Vertrag sollte die Sicherheitsanforderungen explizit aufführen (Absicherung, Monitoring, Benachrichtigungen, Eskalation, Recht zur Auditierung beim Anbieter etc.).

2. *Datenübertragung:* Die Verbindung zwischen Kunde und Provider muss gesichert und verschlüsselt sein (zum Beispiel über SSL). Dies gilt nicht nur für die HTTP-Verbindung, sondern auch für andere Protokolle wie FTP.

3. *Datenschutz:* Bei der Verarbeitung personenbezogener Daten fordert das Bundesdatenschutzgesetz seit 2009 einen dedizierten Vertrag mit dem Auftragsdatenverarbeiter, der das Zusammenspiel der Vertragsparteien regelt.

4. *Updates/Patches:* Die Gestaltung und Durchführung von Updates und Patches muss im Vertrag mit dem Cloud-Provider klar beschrieben sein.

5. *Administrationszugriff:* Die Sicherung des Administrationszugriffs für den Kunden sollte über HTTPS und/oder einen Remote-Access-Server mit Token oder Einmalpasswort erfolgen.

6. *Anwendungsinteraktion:* Die Kommunikation zwischen Anwendungen muss grundsätzlich authentifiziert und verschlüsselt erfolgen, die Transaktionssicherheit muss gegeben sein.

7. *Mandantentrennung:* Die Trennung der einzelnen Mandanten muss beim Cloud-Provider auf Softwareseite nicht nur vorhanden, sondern auch vertraglich festgeschrieben und auditierbar sein.

8. *Virtualisierungsplattform:* Cloud-Services nutzen Virtualisierungsplattformen wie VMware, Xen oder Parallels. Diese Plattformen gilt es ebenfalls abzusichern, hier bestehen die gleichen Risiken wie beim lokalen Betrieb virtualisierter Server.

Neben diesen Risiken und Fallstricken bietet die Cloud allerdings viele Vorteile gegenüber dem lokalen Applikationsbetrieb: Insbesondere die größeren Cloud-Provider können sich aufgrund der Skalierungsvorteile Spezialisten mit großem fachlichen (und eben auch Security-) Know-how leisten. Cloud-Provider können damit in puncto Dienstgüte, Monitoring und Sicherheit mehr bieten als das Gros unternehmensinterner IT-Abteilungen, vor allem ein schnelleres Incident-Management bei Störfällen, einen 24/7-Service-Desk sowie einen preisgünstigen sicheren Applikationsbetrieb. ■

Literatur & Links

[mzd] http://m.zdnet.com.au/e-tailers-watch-your-price-tags-120217173.htm

[Ber10] Dr. Rainer Bernnat et al., „Die IT-Sicherheitsbranche in Deutschland: Aktuelle Lage und ordnungspolitische Handlungsempfehlungen“ (Booz & Co. im Auftrag des Bundesministeriums für Wirtschaft und Technologie, 2010).

[SAM] www.opensamm.org