

Security Check 1. Teil: Denn sicher ist sicher

Eine Betrachtung von Michael Lausch, Security Experte bei ANECON

Sicherheit ist in den letzten Jahren zu einem der Hauptthemen in der IT geworden. Waren früher hauptsächlich die Netzwerkspezialisten mit Sicherheitsfragen beschäftigt, verlagert sich heute der Fokus auf die Softwarespezialisten.

Die Herausforderungen für Netzwerkspezialisten liegen heute im Durchsatz der aktiven Netzwerkkomponenten und in der Verarbeitung von Informationen, die zur Bekämpfung von Distributed Deny Of Service-Attacken dienen.

Fast alle anderen Angriffe die heute stattfinden, sind nicht auf die Netzwerk Ebene beschränkt, sondern zielen auf Applikationen und immer mehr auch auf Security Software, wie zum Beispiel Virens Scanner, ab.

Bevor sich Ihre "Bugs" vermehren

Diese Entwicklung erhöht die Anforderungen an Applikationssoftware, unabhängig davon, ob diese eine Web Applikation oder eine konventionelle Client Server Applikation ist.

Um diese erhöhten Anforderungen gerecht zu werden, müssen die Software Architekten, Designer und Entwickler aber andere Prioritäten setzen, als vom Projektplan vorgegeben.

Das Zitat „Security is a Process, not a Product“ von Bruce Schneier, beschreibt das Vorgehen bei sicherheitsrelevanten Themen. Techniken, die vor einigen Jahren noch als völlig korrekt angesehen wurden, sind heute ein massives Problem.

Als Beispiel kann der kürzlich entdeckte und im Anschluss genau analysierte Bug in der Implementierung des Windows Meta File Formats (WMF) dienen.

Ein konkretes Beispiel

Die verwendete Programmierertechnik war, als WMF entwickelt wurde, völlig ausreichend. Heute wird von Spezialisten der Verdacht geäußert, dass die Implementierung von WMF absichtlich so gewählt wurde, um ein Backdoor zu schaffen, das es Microsoft ermöglicht, Zugriff auf Rechner zu erhalten, die die Microsoft Website besuchen.

Diese Interpretation des Bugs wird von Microsoft aufs heftigste dementiert. Trotzdem zeigt dieses Beispiel, dass Image im Bereich Security sehr viel ausmacht.

Hätte Microsoft in diesem Bereich nicht einen so schlechten Ruf über die Jahre erworben, wären diese Dementi entweder gar nicht notwendig gewesen, da niemand auf diese Idee gekommen wäre, oder es hätte weit weniger (Internet-) mediale Anstrengung gebraucht, um diese Theorie aus der Welt zu schaffen.

Dieses Beispiel zeigt deutlich, **dass sich die Anforderung an Software bezüglich der Sicherheit ändern.**

Techniken und Prozesse, die auch von Fachleuten als ausreichend angesehen wurden, sind immer wieder aufs Neue in Frage zu stellen und ihre Gültigkeit immer wieder zu überprüfen.

Einige Testmethoden im Security Bereich, wie zum Beispiel OSSTMM, beinhalten diese Tatsache in der Form einer Berechnungsformel, die angibt für welchen Zeitraum eine Sicherheitsüberprüfung Gültigkeit hat und wann die Überprüfung wiederholt werden muss.

Allerdings kann auch dieser Zeitraum zu lange sein, wenn neue Angriffstechniken entwickelt werden, die aufgrund der Verbreitung von neuen Technologien aussichtsreich werden. AJAX, also die Technologie auf der „quasi interaktive“ Websites wie <http://maps.google.com> basieren, hat sicher das Potential zu einer Bedrohung zu werden. Dynamische Änderungen von Inhalten einer Webpage durch Javascript werden, wenn es als Angriffstool eingesetzt wird, als Cross-Site-Scripting bezeichnet. Im Fall von AJAX ist die Infrastruktur, nämlich Javascripts zum Ausführen von HTTP Requests an andere Webserver, und die Verarbeitung von HTML Inhalten in diesem Request, schon im Browser geladen. Natürlich nicht zum Ausführen von Cross-Site-Scripting Attacken, sondern von AJAX Funktionalitäten. Je mehr Funktionen eine Applikation oder deren Framework anbietet, desto einfacher wird es diese Website anzugreifen.

Vorsorge ist besser als „Nachsorge“

Sicherheit kostet Geld. Leider nicht nur einmalig, da Tests immer wieder durchgeführt werden müssen.

Unter diesem Aspekt ist Sicherheit durchaus mit Backups zu vergleichen.

Keiner will im Grunde dafür zahlen oder die Funktionsfähigkeit von Software prüfen. Solange sie funktioniert oder ein potentieller Fehler von niemandem bemerkt wird, bietet es keinerlei Zusatznutzen für das Produkt. Erst wenn einmal etwas nicht mehr klappt und man es dringend braucht, wird die Wichtigkeit erkannt. Dann ist es oft zu spät, so wie man eine kaputte Tape-Cartridge nicht mehr für einen Restore verwenden kann.

Viel schlimmer noch, wenn man merkt, dass wichtige Daten nicht vom Backup-Konzept erfasst wurden, ähnliche Analogien gibt es im Sicherheitsbereich.

Das Korrigieren von Berechtigungen ist ein überschaubarer Aufwand, das Korrigieren von Design Fehlern hingegen, die es einem Angreifer möglich machen in eine Website einzubrechen, sind ungleich aufwendiger. Daher sollte schon beim Entwurf von Software darauf geachtet werden, nur Technologien einzusetzen, die von den Entwicklern wirklich gründlich verstanden werden. Alle Vorgänge, die von den Frameworks „Behind the Scene“ durchgeführt werden, ersparen dem Entwickler zwar Programmierarbeit, das Verständnis für sie darf aber auf keinen Fall fehlen.

Anhand des WMF Bugs wird eine „Best Practice“ von Security deutlich herausgehoben. Man darf keinesfalls davon ausgehen, dass eine Applikation, die nach allen Regeln der Kunst und vorerst frei von Sicherheitsproblem entwickelt wurde, diesen Status für die Ewigkeit behalten wird.

Die Entwicklung von Programmiersprachen, Frameworks und Entwicklungsmethoden schreitet fort. Dies trifft leider auch auf die dunkle Seite der IT zu. Hacker und Industriespione sind interessierter und schneller als je zuvor. Entwickler, die noch mit den Bugs des letzten Release beschäftigt sind, legen durch den Druck von Time-To-Market Prioritäten den Fokus auf maximale Funktionalität in kürzester Entwicklungszeit.

