

# Die Top 10 der Webapplikations-sicherheit

Spätestens seit dem Auftauchen des Begriffes Web 2.0 möchten Unternehmen vermehrt Applikationen ins Web portieren oder diese gleich dort aufbauen. Entwickler für webbasierte Anwendungen sollten deshalb mindestens ein Grundwissen der Webapplikationssicherheit mitbringen. *Sven Vetsch*

Das Open Web Application Security Project (OWASP – [www.owasp.org](http://www.owasp.org)) betreut neben vielen anderen Projekten auch die so genannten OWASP Top 10, die die zehn akutesten Probleme in Bezug auf Webapplikationen wiedergeben. Die OWASP Top 10 sind international anerkannt und werden von Unternehmen wie Sun Microsystems, IBM, Swiss Federal Institute of Technology, British Telecom und vielen mehr bei der Applikationsentwicklung sowie beim Testen eben solcher eingesetzt. Auch haben zum Beispiel die Michigan State University (MSU) und die University of California at San Diego (UCSD) die OWASP Top 10 in den Lehrplan aufgenommen. Für den Payment Card Industry (PCI) Data Security Standard gelten diese ab 2008 gar als offizielles Requirement für jegliche Security Code Reviews.

## A1 – Cross Site Scripting (XSS)

Cross Site Scripting, besser bekannt unter der Abkürzung XSS, ist die momentan am weitesten verbreitete Verwundbarkeit bei Webapplikationen. XSS sind dann vorhanden, wenn Daten, die von einem User eingegeben werden, in ungeprüfter/ungefilterter Form von Seiten des Servers zurückgesendet und im Webbrowser ausgegeben werden.

XSS erlaubt einem Angreifer das Ausführen von Scriptcode im Webbrowser des Opfers, wodurch das Übernehmen der User Session, Defacen einer Website, Einfügen eigener Contents, Durchführen von Phishing-Attacken und das Missbrauchen des Webbrowsers zur Ausführung von scriptbasierter Malware möglich wird. Dieser schadhafte Scriptcode wird meist mit Javascript geschrieben, jedoch ist dies mit jeder Scriptsprache möglich, die vom Webbrowser des Opfers unterstützt wird.

## A2 – Injection Flaws

Injection Flaws, hauptsächlich SQL Injections, sind eine weit verbreitete Verwundbarkeit in Webapplika-

The screenshot shows the OWASP website's main page. At the top, there's a navigation bar with links for 'article', 'discussion', 'view source', and 'history'. Below that is the 'Main Page' header. A large banner for 'get training' is visible. The main content area includes a 'Welcome to OWASP' section with a brief description of the project. To the right, there's a 'OWASP Conferences' section with news about the 6th OWASP AppSec Conference in Italy and the 7th OWASP AppSec Conference in San Jose, CA. A 'Featured Story' section highlights that OWASP is funding over 25 new application security projects. At the bottom, there's a 'OWASP Moderated AppSec News Feed' with a list of recent articles.

Das Open Web Application Security Project (OWASP – [www.owasp.org](http://www.owasp.org)) betreut neben vielen anderen Projekten auch die so genannten OWASP Top 10, die die zehn akutesten Probleme in Bezug auf Webapplikationen wiedergeben.

tionen. Solche Injections erfolgen dann, wenn die an eine Applikation gesendeten Daten von einem Interpreter auf Seiten des Servers als Command, Query oder als Teil eines solchen verstanden und ausgeführt werden. Injection Flaws können es einem Angreifer erlauben, beliebige Daten und/oder Dateien zu lesen, zu schreiben, zu erweitern oder sogar auszuführen. Im schlimmsten Fall kann es einem Angreifer durch diese Verwundbarkeit möglich sein, die ganze Applikation oder gar den ganzen Server zu kompromittieren.

## A3 – Insecure Remote File Include

Auch Insecure Remote File Includes können in vielen Webapplikationen gefunden werden, dies weil Entwickler dem Input eines Users vertrauen und die von diesem gesendeten Daten in einer File-, Include-, oder Stream-Funktion ungeprüft und/oder ungefiltert verwenden. Auf vielen Plattformen erlauben Frameworks das Einbinden von externen Objekten, wie zum Beispiel Websites oder auch lokale Dateien. Wenn solche Daten ungeprüft eingelesen wer-

den, kann dies dazu führen, dass die inkludierten Daten Code enthalten, der beim Verarbeiten durch den Interpreter zur Ausführung kommt.

## A4 – Insecure Direct Object Reference

Als Insecure Direct Object Reference bezeichnet man den Verweis auf ein Objekt, wie zum Beispiel Files, Ordner, Datenbankeinträge oder Keys, durch den diese intern eingebunden werden. Sofern kein Access Control Check implementiert wurde, kann ein Angreifer durch die Manipulation der Parameter, die auf solche Files verweisen, auf andere Ressourcen zugreifen ohne sich authentifizieren zu müssen.

## A5 – Cross Site Request Forgery (CSRF)

Cross Site Request Forgery ist keine neue Verwundbarkeitsart in Bezug auf Webapplikationen, dennoch ist sie simpel und gleichzeitig kann dadurch grosser Schaden entstehen. Ein CSRF-Angriff richtet sich gegen den im Webbrowser eingeloggten User einer Applikation. Es wird ver-

sucht, aus dem Browser einen Request an die verwundbare Webapplikation zu senden, wo die gewünschte Aktion dann mit den Berechtigungen des Benutzers ausgeführt wird. Diese Verwundbarkeit besteht in fast jeder Webapplikation im Netz, da jede Applikation, die die Legitimität eines Requests nur anhand von automatisch übertragenen Credentials durch den Browser beurteilt, potenziell verwundbar ist.

Diese Verwundbarkeit ist auch unter verschiedenen anderen Namen bekannt, wie etwa Session Riding, One-Click Attacks, Cross Site Reference Forgery, Hostile Linking und Automation Attack. Teilweise wird auch die Abkürzung XSRF verwendet. OWASP sowie MITRE haben sich auf die Bezeichnung Cross Site Request Forgery mit der zugehörigen Abkürzung CSRF festgelegt.

## A6 – Information Leakage and Improper Error Handling

Applikationen können in unbeabsichtigter Weise durch verschiedenste Fehler Informationen über Konfigurationen, interne Abläufe oder Personen preisgeben. Applikationen können auch Informationen preisgeben über die Dauer der Verarbeitung von gewissen Operationen oder auch über die verschiedenen Antworten auf verschiedene Anfragen, so zum Beispiel denselben Error-Text, jedoch mit anderer Error-Nummer. Webapplikationen geben oft sensible Informationen in Form von teilweise sehr detaillierten (Debug) Error Messages preis.



Autor

**Sven Vetsch**  
ist Leader  
des OWASP  
Switzerland  
Local Chapters,  
angehender

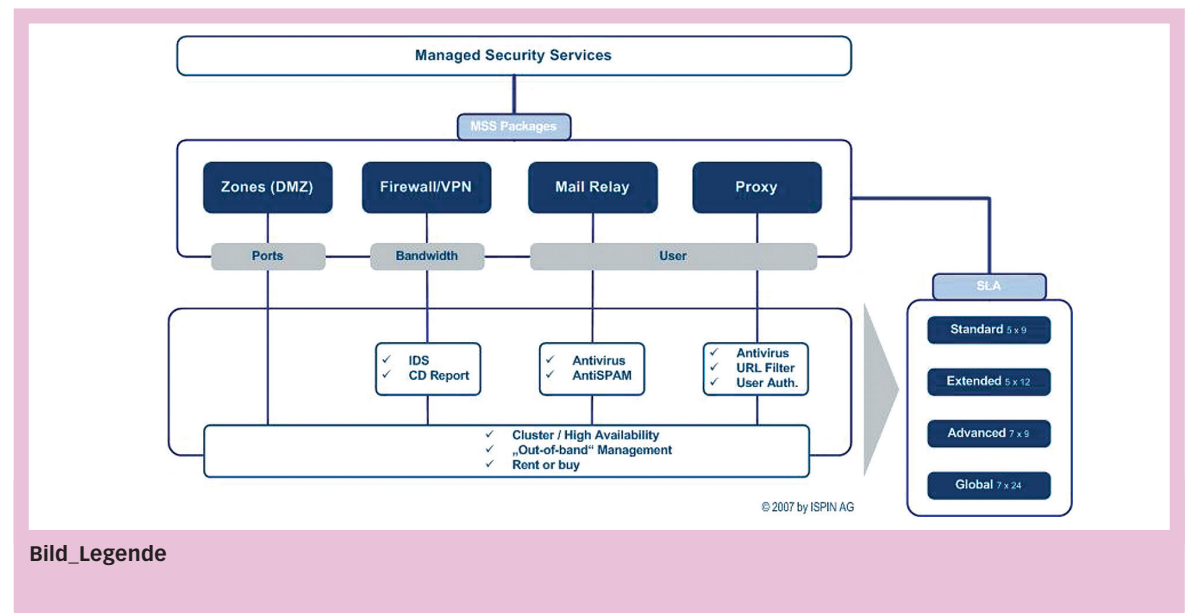
Student der Informatik sowie Security-Tester, Analyst und Engineer mit Spezialisierung auf Web Technologies bei der Dreamlab Technologies Ltd. in Bern.

# Wenn die Bänder stillstehen, ist der Schaden gross

Die ständige Verfügbarkeit von Firewalls, Serversystemen, Routern und anderen, dem Internet zugewandten Systemen und Netzwerkkomponenten, ist zu einem geschäftskritischen Faktor geworden. Der Ausfall nur eines dieser Sicherheitssysteme kann die Produktivität des ganzen Unternehmens lähmen. *Patrick Hulliger*

Dem hohen Anspruch an Business Continuity und der damit laufenden Überwachungsarbeit stehen die oft knappen Ressourcen in den IT-Security-Abteilungen gegenüber. Neben Anspruch auf Kompetenz wird der Kostendruck immer stärker. Gleichzeitig steigen die Anforderungen an Systemadministratoren und Security-Verantwortliche, immer komplexere globale Netzwerke und Infrastrukturen vor externen und internen Angriffen zu schützen und sichere IT-Plattformen für neue Businessprozesse zur Verfügung zu stellen. Eine funktionsfähige IT ist essenziell für den Erfolg eines Unternehmens, teilweise sogar überlebenswichtig.

Einen Ausweg hierfür stellen Managed Security Services (MSS) oder das vollständige Outsourcing der IT Security dar. Das heisst, ausgelagertes Überwachen und Verwalten von Security Devices, Systemen oder Prozessen durch Spezialisten. Dies



Bild\_Legende

spart Kosten, entlastet Kapazitäten und steigert durch Synergieeffekte die Sicherheitslage im Unternehmen nachhaltig und langfristig.

## Auf die Begeisterung folgt der Betrieb

Der Fokus der Unternehmen lag in den letzten Jahren auf der Siche-

rung des Perimeters und hatte eine verstärkte Nachfrage nach dem Management von Firewalls, Intrusion-Detection-Diensten, Gateway-Anti- ▶

## ► A7 – Broken Authentication and Session Management

Ein korrekt funktionierendes Session Management und eine saubere Authentifizierung sind zwei kritische Faktoren im Bereich der Webapplikationssicherheit. Oft bestehen Fehler in diesem Bereich darin, dass Credentials und/oder Session Tokens während deren Gültigkeit nicht ausreichend geschützt werden. Durch diese Schwächen kann es einem Angreifer möglich sein, Userdaten zu stehlen oder die aktuelle Session eines Users zu übernehmen. Weiter können auch Autorisationsprozesse oder Accountability Controls umgangen werden sowie Datenschutzverletzungen auftreten.

## A8 – Insecure Cryptographic Storage

Webapplikationen nutzen sehr oft Kryptographie zum Schutz sensibler Daten, häufig werden die dazu nötigen Funktionen jedoch falsch implementiert oder genutzt. Sensible Daten einfach unverschlüsselt

zu lassen, ist die wohl grösste Gefahr in diesem Bereich, jedoch ist es nicht selten der Fall, dass entweder schwache kryptographische Algorithmen verwendet werden oder aber, dass starke Verschlüsselungen falsch angewendet werden, wodurch diese ihre Wirkung verfehlen. Insecure Cryptographic Storage kann dazu führen, dass sensitive Daten einsehbar sind oder auch, dass gegen gewisse Compliance-Vorgaben verstossen wird.

## A9 – Insecure Communications

Oftmals wird vergessen, Netzwerk Traffic durch die Applikation zu verschlüsseln, wo dies zum Schutz von sensiblen Daten nötig wäre. Verschlüsselung (meist SSL) muss für jede Verbindung verwendet werden, die nicht sowieso jedem zugänglich ist. Besonders zu beachten ist dies bei Webseiten, die vom Internet aus erreichbar sind. Doch sollten auch Backend-Verbindungen geschützt werden, denn sonst können darüber

sensible Daten wie Authentication und/oder Session Tokens preisgegeben werden. Weiter sollte Verschlüsselung auch immer dann verwendet werden, wenn sensible Daten wie Kreditkarten-Informationen oder medizinische Daten übertragen werden. Es ist auch zu beachten, dass die Applikation nicht dazu gebracht werden kann, die Verschlüsselung auszuschalten oder auf einen veralteten und unsicheren Algorithmus auszuweichen, da dies durch einen Angreifer leicht ausgenutzt werden kann.

## A10 – Failure to Restrict URL

Immer wieder kommt es vor, dass bestimmte Bereiche innerhalb einer Webapplikation nicht durch einen Login-Mechanismus oder Ähnliches ge-

schützt sind, sondern lediglich durch das Aufrufen der korrekten URL Daten eingesehen werden können oder gar administrative Funktionen verwendet werden können. Ein motivierter und kompetenter Angreifer, der auch noch ein wenig Glück hat, kann zum Beispiel durch blosses Ausprobieren Zugang zu diesen versteckten Seiten erhalten. In der Vergangenheit hat es sich immer wieder gezeigt, dass «Security by Obscurity» keine geeignete Schutzmassnahme ist und somit davon abgesehen werden sollte. Es sollten immer Access Control Checks implementiert werden, wo Funktionen angeboten werden, die nicht öffentlich zugänglich sein sollen. Damit wird nicht autorisierten Benutzern dieser Zugang verweigert.

## Security Zone

Der Autor ist einer unter vielen Fachreferenten an der diesjährigen Security Zone. Der Fachkongress und die Ausstellung finden am 19. und 20. September in Zürich-Oerlikon statt. Informationen und Anmeldung: [www.security-zone.info](http://www.security-zone.info)