Erschienen in:
Tagungsband des Deutschen
IT-Sicherheitskongress 2005

Web Application Security auf sechs Ebenen – Ein Klassifizierungsschema zur Sicherheit von Webanwendungen

Thomas Schreiber[1]

Kurzfassung
Web Application Security, die Sicherheit von Webanwendungen, ist eine noch sehr junge Disziplin innerhalb der IT-Security. Eine systematische Darstellung existiert bisher ebenso wenig wie eine methodische Aufbereitung für den praktischen Umgang. Im vorliegenden Papier nennen wir sechs Themenbereiche, aus denen sich die Web Application Security zusammen setzt, grenzen diese gegen verwandte Felder der IT-Security ab und machen einen Vorschlag für eine Darstellung in einem Ebenenmodell.

Stichworte: Web Application Security, Sicherheit von Webanwendungen, Anwendungssicherheit, Klassifizierung

1 Einleitung

Die systematische Aufbereitung und Weiterentwicklung der Web Application Security findet gegenwärtig überwiegend innerhalb einer enthusiastischen Community statt, die in Mailing-Listen und verschiedenen Open Source Projekten zusammen gefunden hat. Das Wissen ist auf eine Vielzahl von Veröffentlichungen – zumeist im Internet und in Open Source Guides, zunehmend auch in Buchform – verteilt. Wer in das Thema einsteigen möchte, findet in den genannten Quellen zwar reichhaltiges Material, die Orientierung fällt jedoch schwer und eine sinnvolle Umsetzung und Anwendung im eigenen Unternehmen ist auf diese Weise nur schwer zu erreichen.

Mit dem vorliegenden Papier möchten wir einen Beitrag zur Strukturierung dieses Gebietes der IT-Security leisten. Im Folgenden benennen wir die Themen, die wir als der Web Application Security zugehörig ansehen. Wir untergliedern diese in sechs Rubriken, die wir in einem Ebenenmodell darstellen. Zusätzlich stellen wir eine Abgrenzung zu anderen Bereichen der IT-Security her, zu denen ein enger Bezug besteht. Der Übersicht halber stellen wir das Ergebnis, das Ebenenmodell, an den Anfang des Papiers.

Diese Klassifizierung ist Grundlage einer soeben erschienenen Studie des Bundesamtes für Sicherheit in der Informationstechnik zur Web Application Security [1]. Teil der Studie ist ein Leitfaden und Maßnahmenkatalog, der Industrie und Verwaltung praxisnahe Richtlinien, Vorgehensweisen und Best Practices zur Herstellung der Sicherheit von Webanwendungen an die Hand gibt.

2 Das Ebenenmodell

Web Application Security wird häufig ausschließlich unter dem Gesichtspunkt der fehlerhaften (bzw. richtigen) Programmierung und Konfiguration gesehen. Die Implementierungsebene ist zwar ein wichtiger Aspekt, aber sie ist nur einer von mehreren. Eine Webanwendung kann erst dann als hinreichend sicher eingestuft werden, wenn

[1] Thomas Schreiber, SecureNet GmbH, München

alle Ebenen betrachtet worden sind und auch das – möglicherweise ungünstige – Zusammenwirken der einzelnen Ebenen untersucht worden ist.

Wir gliedern die Web Application Security in sechs Bereiche. Auch wenn eine Zuordnung der einzelnen Schwachstelle, Bedrohung, Angriffstechnik oder Schutzmaßnahme zu genau einer Ebene nicht immer möglich ist, erweist sich dieses Ebenenmodell doch als sehr hilfreich für das Verständnis der Thematik und die Organisation des Umgangs mit der Web Application Security.

	Ebene	Inhalt
6	**Vorschriften und Bestimmungen**	Einhaltung gesetzlicher Regelungen und unternehmensspezifischer Vorgaben
5	**Semantik**	Schutz vor Täuschung und Betrug
4	**Logik**	Absicherung von Prozessen und Workflows als Ganzes
3	**Implementierung**	Vermeiden von Programmierfehlern, die zu Schwachstellen führen
2	**Technologie**	Richtige Wahl und sicherer Einsatz von Technologie
1	**System**	Absicherung der auf der Systemplattform eingesetzten Software
0	**Netzwerk & Host**	Absicherung von Host und Netzwerk

Ebene 0 – Netzwerk und Host

Wenngleich die Web Application Security auch in Abhängigkeit zur Sicherheit von Netzwerk, Hardware und Host steht, rechnen wir ihr diese Ebene 0 nicht zu. Dieser Bereich ist heutzutage gut verstanden und in den Sicherheitsprozessen der Unternehmen in der Regel verankert.

Ebene 1 - Systemebene

Ebene 1 beinhaltet all jene Software, die eine Webanwendung benötigt, um überhaupt ablaufen zu können. Dazu gehören der Webserver und der Application Server, aber auch die Datenbank und beteiligte Backend-Systeme. Alle diese Komponenten müssen bei der Betrachtung der Sicherheit einer Webanwendung mit einbezogen werden. Eine Webanwendung, die frei von Sicherheitsmängeln programmiert ist, ist trotzdem unsicher, wenn die von ihr verwendete Datenbank über einen anderen Kanal, etwa den nicht ausreichend abgesicherten und für 'Innentäter' zugänglichen Direktzugriff per Database-Client manipulierbar ist und diese Manipulation im Web von einem Angreifer ausgenutzt werden kann.

Ebene 2 - Technologie

In diesem Bereich geht es um die Verwendung der für den jeweiligen Zweck und Schutzbedarf richtigen Technologie – und um deren richtigen Einsatz. Eine Webanwendung, die sensible Daten unverschlüsselt über das Internet transferiert, setzt nicht die richtige Technologie ein. Und eine Webanwendung, die Passworte zwar verschlüsselt, dafür aber einen nicht ausreichend langen Schlüssel verwendet, setzt die richtige Technologie falsch ein. Die erste Anwendung ist gegen Ausspähen auf dem Übertragungswege nicht geschützt, die andere nicht ausreichend gegen Passwort-Cracking. Beide Anwendungen sind also unsicher, auch wenn sie im Programmcode keine Sicherheitslücken enthalten.

Ebene 3 - Implementierung

Die Implementierungsebene ist die offensichtliche Ebene der Web Application Security. Es ist der Bereich, in dem unbeabsichtigte Programmierfehler (Bugs) auftreten und zu Sicherheitsproblemen führen, oder aber fehlerhafte Programmierung, wie nicht vorhandene oder ungenügende Data Validation, stattfindet. Es ist aber auch die Ebene von ungenügendem Testen, dem Testen mit einseitigem Fokus (nur die Funktionalität und Laststabilität, aber nicht die Sicherheit wird getestet) und die Vernachlässigung der Qualitätssicherung zugunsten des Inbetriebnahmetermins oder aus Kostengründen.

Ebene 4 - Logik und Ebene 5 - Semantik

Diese beiden Ebenen sind diejenigen, deren Bedeutung für die Sicherheit von Webanwendungen gegenwärtig noch kaum erkannt wird. Dabei sind sie von großer Wichtigkeit – und werden es im Zeitalter von Phishing und Identitätsdiebstahl immer stärker werden – wenn man die Sicherheit von Webanwendungen nicht allein als den Schutz des Servers vor Eindringversuchen versteht, sondern sie umfassend begreift: Eine Webanwendung ist sicher, wenn sie selbst mitsamt dem sie beherbergenden System sicher ist und wenn sie darüber hinaus auch die Benutzer und deren vertrauenswürdige Daten vor Schaden bewahrt. Der Anbieter einer Webanwendung trägt nicht nur die Verantwortung für sein eigenes System, sondern auch für alle an der Nutzung Beteiligten. Auf der Ebene der Logik und der Semantik kommt dieser Aspekt ganz besonders zum Tragen:

Ebene 4 - Logik

Diese Ebene betrifft die Logik der Abläufe innerhalb einer Anwendung – die Anwendungs- und Business-Logik – und in der Interaktion mit dem Benutzer. Ist diese zu 'zweckorientiert' implementiert, d.h. ist die Möglichkeit, dass sie anders als beabsichtigt genutzt wird, zu wenig berücksichtigt, ist häufig eine Angreifbarkeit gegeben. Wird etwa, um Passwort-Enumeration zu verhindern, der Benutzer nach dem fünften fehlerhaften Loginversuch gesperrt, so kann ein Angreifer den Benutzer gezielt aussperren. Das wird weiter erleichtert, wenn die Benutzerkennung leicht zu erraten ist, z.B. wenn dazu die Emailadresse verwendet wird. Eine Denial-of-Service Attacke auf die gesamte Anwendung bzw. das gesamte System ist dann möglich, wenn die Benutzerkennungen zusätzlich leicht durchprobiert werden können, was z.B. bei aufeinander folgenden Zahlen als Benutzerkennungen der Fall ist. Die massenweise Provokation

von fehlerhaften Logins könnte eine Überlastung des Helpdesks durch Hilfe suchende Benutzer zur Folge haben.

Ebene 5 - Semantik

Die semantische Ebene der Web Application Security umfasst inhaltliche und die Kommunikation betreffende Aspekte. Sie stellt den Vertrauenskontext für die Interaktion mit dem Benutzer her. Wird in diesem Bereich nicht sehr viel Sorgfalt aufgewandt, so kann eine Webanwendung oder Website von Dritten leicht dazu missbraucht werden, den Benutzer zu täuschen und zu betrügen. Das betrifft die Art der Informationen, die dem Benutzer gegeben werden, wie ihm Inhalte präsentiert werden und wie mit ihm umgegangen wird. Dieser Bereich kann in der Betrachtung selten auf eine einzelne Anwendung beschränkt bleiben. Er ist in der Regel vielmehr website- oder unternehmensübergreifend zu definieren. Angriffe, die sich Fehler auf der semantischen Ebene zunutze machen, sind Social Engineering, Phishing, Identitätsdiebstahl, Täuschung, Fälschung, Betrug, Aufbrechen des Datenschutzes und des Schutzes der Privatsphäre.

Ebene 6 – Vorschriften und Bestimmungen

In diesen Bereich fallen Regelungen aus dem eigenen Haus und von Seiten Dritter sowie gesetzliche Anforderungen. Die Einhaltung von Datenschutzrichlinien, die Berücksichtigung der Vorgaben des KontraG oder das Nachkommen der Verpflichtungen von Haftungsfragen sind Erfordernisse dieser Ebene. Auch wenn es sich hier nicht mehr um IT-Security im eigentlichen Sinne handelt, kann durch eine Missachtung der Vorschriften auf dieser Ebene dem Anbieter Schaden entstehen.

Ebenen unterstützen die Intuition

Als Darstellungsform für das Klassifizierungsschema haben wir das Ebenenmodell gewählt. Obwohl es sich bei den Ebenen nicht, wie z. B. beim ISO/OSI-Schichtenmodell, um unmittelbar aufeinander aufbauende Schichten handelt, ist diese Form der Darstellung doch gut geeignet, da sie die intuitiv erkennbare Hierarchie zum Ausdruck bringt. Dabei liegt den verschiedenen Ebenen nicht unbedingt dasselbe Prinzip zugrunde, nach dem sie mit der darüber- bzw. darunter liegenden Ebene in Beziehung stehen: Die Systemebene (Ebene 1) ist die unterste Ebene, auf der die Anwendung mit ihrer Implementierung (wir lassen die Technologieebene zunächst außer Acht) aufsetzt. Die Implementierungsebene (Ebene 3) braucht und nutzt die bereitgestellte Systemumgebung. Die Anwendungslogik (Ebene 4) steht über der Ebene der Implementierung, da hier Bausteine und Abläufe einer größeren Granularität zu betrachten sind. Auf der semantischen Ebene (Ebene 5) wird die Betrachtung weiter ausgedehnt: Inhalte und die Interaktion mit dem Benutzer kommen hinzu. Die Ebene 6 der Vorschriften und Bestimmungen schließlich bringt noch weiter übergeordnete Aspekte hinein.

Die Technologieebene fügt sich hier nicht so ohne weiteres ein. Der Einfachheit halber haben wir sie trotzdem als Ebene aufgenommen und sie zwischen die Systemebene – da sie auf Systemvoraussetzungen wie etwa SSL fußt – und die Implementierungsebene – da die Implementierung sich wiederum auf die Technologie abstützt – platziert.

Die Aufeinanderfolge der Ebenen von unten nach oben spiegelt darüber hinaus die Intensität wieder, mit der Sicherheit gegenwärtig jeweils betrieben wird. Der Systemebene kommt dabei am meisten Aufmerksamkeit zu. Diese schwächt sich nach oben hin deutlich ab bis hin zur Semantischen Ebene, die größtenteils noch nicht einmal als Gegenstand der Web Application Security erkannt worden ist. Ebene 6 wird bisher in der Regel nicht als Thema der IT-Sicherheit angesehen und im Rahmen der Seitenredaktion mit abgehandelt.

3 Abgrenzung

Die Web Application Security befindet sich in – teilweise enger – Abhängigkeit und Wechselwirkung zu anderen Bereichen der IT-Security. Eine klare Abgrenzung ist jedoch wichtig, um die richtigen Maßnahmen an den richtigen Stellen ergreifen zu können. Insbesondere die im Folgenden genannten Bereiche spielen dabei eine Rolle.

Desktop-Security

Bei der Desktop-Security geht es um den Schutz des PCs und des Benutzers. Im Hinblick auf die Webanwendung also um die *Client*-Rolle in der Kommunikation. Bei der Web Application Security steht demgegenüber der Schutz der *Server*-Seite im Vordergrund. Doch die Web Application Security muss sich auch mit dem Schutz des Benutzers befassen: Indirekt dadurch, dass sie Sorge dafür trägt, dass z.B. ein Eindringen in das Benutzerkonto nicht möglich ist. Etwas direkter dadurch, dass die Webanwendung einem Betrüger z.B. einen Phishing-Angriff durch eine XSS-Schwachstelle nicht noch erleichtert.

Virenschutz ist Teil der Desktop-Security. Eine unsichere Webanwendung kann dazu missbraucht werden, Viren auf einem Weg einzuschleusen, den der Benutzer für sicher hält. Etwa dann, wenn die Webanwendung über eine Schwachstelle (Ebene 3) verfügt, über die Dateien auf dem Server abgelegt werden können. Auf der Ebene 5 (semantische Ebene) gibt jede XSS-Schwachstelle einem Angreifer die Möglichkeit, über einen präparierten Link einen Virus vermeintlich über diese Webanwendung zu schleusen, so dass der Benutzer keinen Verdacht schöpft [6]. Auch hier gilt also, dass die Webanwendung Verantwortung für die Sicherheit des Benutzers übernimmt und nicht dazu beiträgt, dass dieser unter Zuhilfenahme der Webanwendung von einem Dritten getäuscht oder geschädigt werden kann.

Produkte wie Content-Filter und Popup-Blocker sind Gegenstand der Desktop-Security. Sie können den Benutzer aber teilweise auch vor Gefahren schützen, die durch Fehler in der von ihm benutzten Webanwendung entstehen. Die vom BSI propagierte Schutzmaßnahme, Aktive Inhalte inklusive JavaScript abzuschalten, ist Gegenstand der Desktop-Security. Sie hat die Begleiterscheinung, dass sie einige Angriffstechniken aus der Web Application Security wirksam verhindert.

Netzwerk- und Host-Security

Dieser Bereich schließt sich nach unten an die Ebene 1 an (wir haben sie als Ebene 0 in das Schema mit aufgenommen). Ein sicheres Netzwerk und ein sicherer Host sind die wichtigste Voraussetzung für eine sichere Webanwendung. Trotzdem ist sie klar von der Web Application Security zu trennen: Die Skills, die benötigt werden, um

Netzwerk und Host abzusichern, sind völlig anderer Natur als diejenigen für die Web Application Security. Letzteres erfordert die Erfahrungen von Softwareentwicklern, Betriebssystem- und Netzwerkkenntnisse sind da nicht ausreichend. Auch die organisatorische Einbindung unterscheidet sich voneinander: Die Zuständigkeit für die Netzwerksicherheit ist im Unternehmen an zentraler Stelle verankert. Die Zuständigkeit für die Sicherheit der Anwendung muss wegen der untrennbaren Verzahnung der Software mit den von ihr abgebildeten Geschäftsprozessen hingegen beim Fachbereich, von dem sie betrieben wird, angesiedelt werden.

Zu den Sicherheitsanforderungen auf der Ebene 0 gehört seitens der Webanwendung, dass hier eine *Second-Line-of-Defense* etabliert werden muss. Besitzt die Webanwendung auf der Ebene 3 einen Fehler, der es einem Angreifer erlaubt, in den Host einzudringen, so darf es diesem trotzdem nicht gelingen, dort Schaden anzurichten, der über den Bereich der betroffenen Webanwendung hinausgeht. Durch entsprechende Konfiguration des Hosts sind Schutzzonen zu schaffen, die dies verhindern.

4 Nutzen und Anwendungsbeispiele

Die folgenden Beispiele zeigen Anwendungsmöglichkeiten des Ebenenmodells.

Festlegung von Verantwortlichkeiten

Das Klassifizierungsschema eignet sich gut, um Verantwortlichkeiten innerhalb einer Organisation abzugrenzen. Als Ausgangspunkt kann das nachfolgend dargestellte Gerüst herangezogen werden.

<table>
<tr><th></th><th>Ebene</th><th>Stelle</th><th>Funktion</th></tr>
<tr><td>6</td><td>Vorschriften und Bestimmungen</td><td>Zentrale</td><td rowspan="3">Plan</td></tr>
<tr><td>5</td><td>Semantik</td><td rowspan="2">Fachstelle (Anforderer)</td></tr>
<tr><td>4</td><td>Logik</td></tr>
<tr><td>3</td><td>Implementierung</td><td>Entwickler (Umsetzer)</td><td rowspan="2">Build</td></tr>
<tr><td>2</td><td>Technologie</td><td>Fachstelle, Entwickler, Betrieb</td></tr>
<tr><td>1</td><td>System</td><td rowspan="2">Betrieb</td><td rowspan="2">Run</td></tr>
<tr><td>0</td><td>Netzwerk & Host</td></tr>
</table>

Vorschriften und Bestimmungen (Ebene 6) werden von der Unternehmensführung unternehmensweit festgelegt. Die semantische Ebene (Ebene 5) ist in Zusammenarbeit der Unternehmensführung mit den Fachstellen zu behandeln. Die Ebene der Anwendungslogik (Ebene 4) ist Sache der anfordernden Fachstelle. Diese verlässt sich darauf, dass die Entwicklungsstelle die Anwendung sicher implementiert (Ebene 3) hat und dass der Betrieb die Anwendung sicher hostet (Ebene 1). Auf der Technologieebene

(Ebene 2) schließlich tragen Fachstelle, Entwickler und Betrieb für ihre jeweiligen Zuständigkeiten Verantwortung.

Im Rahmen einer Plan/Build/Run-Organisation ergibt sich darüber hinaus grob die in der dritten Spalte dargestellte Zuordnung zu den drei Funktionen.

Sicherheitstests

Eine Webanwendung ist vor der Inbetriebnahme genauso, wie sie auf Laststabilität und funktionale Fehlerfreiheit geprüft wird, einem Test der Sicherheit zu unterziehen. Das folgende Bild zeigt für die einzelnen Bereiche, welche Kenntnisse jeweils erforderlich sind. In der zweiten Spalte ist größenordnungsmäßig dargestellt, in welchem Umfang dabei auf Tooluntstützung gesetzt werden kann.

	Ebene	Skills	Tool-unterstützung
6	**Vorschriften und Bestimmungen**	Juristischer Background	■. . . . .
5	**Semantik**	Corporate Identity und Unternehmens-kommunikation	. . . . .
4	**Logik**	Kenntnisse der Geschäftsprozesse	■. . . . .
3	**Implementierung**	Softwareentwicklungs-skills	■■■■. . . . .
2	**Technologie**	Allg. IT-Security	■■. . . . .
1	**System**	Netzwerk- und Systemadministration	■■■■■■■■. .
0	**Netzwerk & Host**		

Schwachstellen und Schutzmaßnahmen

Das Klassifizierungsschema hat sich als sehr hilfreich für das Verständnis von Schwachstellen in Webanwendungen erwiesen – bei der Darstellung gegenüber Fachfremden ebenso wie bei der Schulung von Softwareentwicklern und Sicherheitsverantwortlichen. So liegt es auch der BSI-Studie [1] zugrunde. Darin sind Schwachstellen und zugehörige Schutzmaßnahmen den Ebenen zugeordnet. Einige Beispiele:

	Ebene	Schwachstellen
6	**Vorschriften und Bestimmungen**	Fehlende Belehrungen zum Datenschutz Nichteinhalten von Bestimmungen des KontraG Fehlerhafte Kennzeichnung des Contents gemäß ICRA Preisgabe vertraulicher Informationen
5	**Semantik**	Informationen ermöglichen Social Engineering-Angriffe Gebrauch von Popups u.ä. erleichtern Phishing-Angriffe Keine Absicherung für den Fall der Fälschung der Website
4	**Logik**	Verwendung unsicherer Email in einem ansonsten gesicherten Workflow Angreifbarkeit des Passworts durch nachlässig gestaltete "Passwort vergessen"-Funktion Die Verwendung sicherer Passworte wird nicht erzwungen
3	**Implementierung**	Programmierfehler Cross-Site Scripting SQL-Injection Session Riding Information Disclosure
2	**Technologie**	Unverschlüsselte Übertragung sensitiver Daten Authentisierungsverfahren, die nicht dem Schutzbedarf angemessen sind Ungenügende Randomness von Token
1	**System**	Fehler in der Konfiguration des Webservers "Known Vulnerabilities" in den eingesetzten Softwareprodukten Mangelnder Zugriffsschutz in der Datenbank

5 Zielkonflikte

Vorgaben auf der einen Ebene stehen manchmal im Widerspruch zu Erfordernissen auf einer anderen Ebene. Beispiel:

Auf der Logischen Ebene lautet eine Schutzmaßnahme: Bevor dem Benutzer Zugang zu sensiblen Daten oder Funktionen, wie beispielsweise zum Ändern der Bankverbindung, gewährt wird, ist das Passwort erneut abzufragen, selbst wenn der Benutzer bereits eingeloggt ist. Damit wird sichergestellt, dass ein Angreifer, der z.B. dadurch in das Konto eingedrungen ist, dass er die Session-ID ausgespäht hat, trotzdem nicht in der Lage ist, diese Funktion auszuführen. Demgegenüber lautet eine Regel der Semantischen Ebene: Mit Aufforderungen, das Passwort einzugeben, sollten Webanwendungen sehr sparsam umgehen, um den Benutzer nicht daran zu gewöhnen, Passwortabfragen als etwas Selbstverständlich zu akzeptieren. Angreifer würden dann mit Phishing-Angriffen leichter zum Ziel kommen, da der Benutzer nicht so schnell Verdacht schöpft.

Die Lösung dieses Zielkonflikts kann nur durch Abwägung der Vor- und Nachteile im jeweiligen Anwendungsfall erfolgen.

6 Weiterführende Arbeiten

Abschließend möchten wir auf Arbeiten verweisen, denen eine Klassifizierung zugrunde liegt bzw. die eine solche einführen:

1. Das *Web Application Security Consortium (WASC)* berücksichtigt in seiner *Threat Classification* [2] nur Aspekte der Implementierungsebene und der logischen Ebene und unterteilt diese in folgende Klassen: Authentication, Authorization, Client-side Attacks, Command Execution, Information Disclosure und Logical Attacks.

2. Einen ähnlichen Ansatz verfolgt das *Open Web Application Security Consortium (OWASP)*. Es verwendet in der *Web Application Penetration Checklist* die *OASIS WAS Vulnerability Types*. Wir verweisen hierzu auf [3].

3. Im *OASIS Consortium* befassen sich zwei Technical Committees mit Themen, bei denen es auch um die Standardisierung der Web Application Security geht: *OASIS Application Vulnerability Description Language (AVDL) Technical Committee* [4] und das *OASIS Web Application Security (WAS) Technical Committee* [5]. Letzteres hat sich die Entwicklung eines offenen Datenformats zur Beschreibung von Schwachstellen in Webanwendungen zum Ziel gesetzt.

Quellen

[1] *Studie zur Sicherheit von Webanwendungen*, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2005

[2] *Web Security Threat Classification*, Web Application Security Consortium (WASC) http://www.webappsec.org/tc/WASC-TC-v1_0.pdf

[3] *Web Application Penetration Checklist,* Open Web Application Security Consortium (OWASP), http://prdownloads.sourceforge.net/owasp/OWASPWebAppPenTestList1.1.pdf

[4] *OASIS Application Vulnerability Description Language (AVDL) Technical Committee*, OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=avdl

[5] *OASIS Web Application Security (WAS) Technical Committee*, OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=was

[6] *Die semantische Ebene der Sicherheit von Webanwendungen*, Thomas Schreiber, Securenet GmbH, 2003, http://www.securenet.de/papers/WebApplicationSecurity_Semantik.pdf