

OWASP Application Security Verification Standard 2009

Web Application Standard





Vorwort

Dieses Dokument definiert vier Stufen ("Level") für die Verifikation der anwendungsseitigen Sicherheit von Webanwendungen. Anwendungsseitige Sicherheit zielt dabei auf die Analyse all jener Komponenten ab, welche innerhalb des OSI-Modells (Open Systems Interconnection Reference Model) der Anwendungsschicht zugeordnet werden können. Aspekte darunterliegender Schichten, wie etwa des Netzwerks oder des Betriebssystems, werden hier dagegen weniger betrachtet. Jedes Verifikationslevel, welches in diesem Dokument beschrieben wird, beinhaltet eine Reihe von Anforderungen an die Verifikation der Effektivität der von einer Webanwendung verwendeten Sicherheitsmechanismen.

Die Anforderungen wurden mit den folgenden Zielen entwickelt:

- Verwendung als Metrik Bietet Entwicklern und Eignern von Webanwendungen ein Messinstrument, mit dem sie den Vertrauensgrad in ihre Anwendung feststellen lassen,
- Verwendung als Entwicklungsleitlinie (Coding Guideline) Bietet Entwicklern von Sicherheitsmechanismen eine Leitlinie auf deren Basis sie Sicherheitsanforderungen umsetzen können³ sowie die
- Verwendung als Beschaffungsrichtlinie Bietet eine Grundlage mit der sich anwendungsseitige Sicherheitsanforderungen in Verträgen (z.B. Pflichtenheften) oder Ausschreibungen spezifizieren lassen.⁴

Die in diesem Standard beschriebenen Anforderungen wurden im Hinblick auf diese Ziele entwickelt, indem die Anforderungen festlegen wie Sicherheitsmechanismen zu entwerfen, zu implementieren und einzusetzen sind. Es wird weiterhin festgelegt, dass die von einer Anwendung eingesetzten Sicherheitsmechanismen mittels Deny-by-Default-Strategie arbeiten und dass sie zentralisiert, serverseitig und überall dort zum Einsatz kommen wo dies erforderlich ist.

Copyright und Lizenz

Copyright © 2008 - 2009 The OWASP Foundation.



Dieses Dokument ist unter der Creative Commons Attribution ShareAlike 3.0 Lizenz veröffentlicht. Bei jeglicher Form von Wiederverwendung oder Verbreitung ist auf die Lizenzbestimmung dieser Arbeit hinzuweisen.

³ Für mehr Informationen zur Erstellung und Verwendung von Sicherheitsmechanismen die konform zu ASVS-Anforderungen sind, siehe *Enterprise Security API (ESAPI)* (OWASP, 2009).

⁴ Für mehr Informationen zur Verwendung von ASVS in Verträgen, siehe *Contract Annex* (OWASP, 2009).



Inhaltsverzeichnis	
Einführung	1
Vorgehensweise	2
Danksagungen	4
Anmerkungen zur Deutschen Übersetzung	5
Verifikationslevel für Anwendungssicherheit	6
Level 1 - Automatische Verifikation. Level 1A - Dynamischer Scan (partiell automatische Verifikation) Level 1B - Sourcecodeanalyse (partiell automatische Verifikation) Level 2 - Manuelle Verifikation Level 2A - Security Test (partiell manuelle Verifikation) Level 2B - Code Review (partiell manuelle Verifikation) Level 3 - Design Verifizierung Level 4 - Interne Verifikation Interpretation und Praxisbeispiele von Anforderungen	9 9 13 13
Anforderungen an detaillierte Verifikation	20
V1 - Verifikationsanforderungen zur Dokumentation der Sicherheitsachitektur V2 - Verifikationsanforderungen zur Authentisierung V3 - Verifikationsanforderungen zum Session Management V4 - Verifikationsanforderungen an Zugriffskontrollen V5 - Verifikationsanforderungen zur Eingabevalidierung V6 - Verifikationsanforderungen zur Ausgabeenkodierung /-Escaping V7 - Verifikationsanforderungen zur Kryptographie V8 - Verifikationsanforderungen zur Fehlerbehandlung und Logging V9 - Verifikationsanforderungen zur Datensicherheit V10 - Verifikationsanforderungen zur Komunikationsicherheit V11 - Verifikationsanforderungen zur HTTP-Sicherheit V12 - Verifikationsanforderungen zur Sicherheitskonfiguration V13 - Verifikationsanforderungen zur Schadcodesuche V14 - Verifikationsanforderungen zur Internen Sicherheit	22 23 25 27 30 31 34 35 36 37
Anforderungen an den Verifikationsbericht	
R1 - Berichtseinführung	40
Glossar	43
Weiterführende Informationen	46



Ab	bil	ldı	un	g	en

Abbildung 1 - OWASP ASVS Level	2
Abbildung 2 - Mögliches Vorgehen zur Integration der Verifikation als SDLC-Aktivität	
Abbildung 3 - OWASP ASVS Level 1, 1A und 1B	
Abbildung 4 - Beispiel einer Sicherheitsarchitektur bei OWASP ASVS Level 1	
Abbildung 5 - OWASP ASVS Level 2, 2A und 2B	
Abbildung 6 - OWASP ASVS Level 2 Beispiel einer Sicherheitsarchitektur	
Abbildung 7 - OWASP ASVS Level 3	14
Abbildung 8 - OWASP ASVS Level 3 Beispiel einer Sicherheitsarchitektur	16
Abbildung 9 - OWASP ASVS Level 4	17
Abbildung 10 - OWASP ASVS Level 4 Beispiel für nichtbetrachteten Code	
Abbildung 11 - Berichtsanforderungen	40

Tabellen

Tabelle 1 - OWASP ASVS Verifikationsanforderungen Dokumentation und Sicherheitsarchitektur (V	/1)
	21
Tabelle 2 - OWASP ASVS Verifikationsanforderungen Authentisierung (V2)	22
Tabelle 3 - OWASP ASVS Verifikationsanforderungen zum Session Management (V3)	24
Tabelle 4 - OWASP ASVS Verifikationsanforderungen an Zugriffskontrollen (V4)	26
Tabelle 5 - OWASP ASVS Verifikationsanforderungen zur Eingabevalidierung (V5)	27
Tabelle 6 - OWASP ASVS Verifikationsanforderungen zur Ausgabeenkodierung /-Escaping (V6)	28
Tabelle 7 - OWASP ASVS Verifikationsanforderungen zur Kryptographie (V7)	30
Tabelle 8 - OWASP ASVS Verifikationsanforderungen zur Fehlerbehandlung und Logging (V8)	31
Tabelle 9 - OWASP ASVS Verifikationsanforderungen zur Datensicherheit (V9)	33
Tabelle 10 - OWASP ASVS Verifikationsanforderungen zur Kommunikationssicherheit (V10)	34
Tabelle 11 - OWASP ASVS Anforderungen zur HTTP-Sicherheit (V11)	36
Tabelle 12 - OWASP ASVS Verifikationsanforderungen zur Sicherheitskonfiguration (V12)	37
Tabelle 13 - OWASP ASVS Anforderungen zur Schadcodesuche (V13)	37
Tabelle 14 - OWASP ASVS Verifikationsanforderungen zur internen Verifikation (V14)	
Tabelle 15 - OWASP ASVS Inhalte der Verifikationsergebnisse im Bericht	41



Einführung

Das Open Web Application Security Project (OWASP) ist eine offene Community die das Ziel verfolgt, Unternehmen bei der Entwicklung, Anschaffung und dem Betrieb von vertrauenswürdigen Anwendungen zu unterstützten. Alle Tools, Dokumente, Foren und Chapter der OWASP sind frei verfügbar und offen für jeden der an Anwendungssicherheit interessiert ist. Wir verstehen Anwendungssicherheit als ein durch Menschen, Prozesse und Technologien bedingtes Problem, da der effektivste Umgang mit Anwendungssicherheit immer die Verbesserung in allen dieser drei Bereiche erfordert. Finden können Sie uns unter www.owasp.org.

Die OWASP ist eine neue Form von Organisation. Unsere Ungebundenheit von kommerziellem Druck erlaubt uns unparteiische, praktikable und kosteneffiziente Informationen zur Anwendungssicherheit bereitzustellen. Die OWASP ist an keinerlei Technologiefirma angegliedert. Dennoch befürworten wir den bedarfsgerechten Einsatz von kommerzieller Sicherheitstechnologie. Ähnlich wie dies bei vielen Projekten mit Open Source Software der Fall ist, produziert die OWASP zahlreiche Arten von Materialien in einer kooperativen, offenen Form. Die OWASP Foundation ist eine gemeinnützige Einrichtung welche einen langfristigen Erfolg dieses Projekts sicherstellt.

Das primäre Ziel des OWASP ASVS-Projektes ist es, die Bandbreite der Abdeckung und Strenge bei Sicherheitstests von Webanwendungen mittels eines kommerziell einsetzbaren und offenen Standards zu harmonisieren. Der Standard liefert hierfür eine Basis zum Testen von technischen Sicherheitsmechanismen von Anwendung für andere Umgebungen, die Angriffe wie etwa Cross-Site Scripting (XSS) oder SQL Injection⁵ abwehren müssen. Dieser Standard ermöglicht es, ein bestimmtes Vertrauen in die Sicherheit von Webanwendungen zu setzen.

⁵ Für weiteren Informationen zu geläufigen Sicherheitslücken von Webanwendungen, siehe die *OWASP Top Ten* (OWASP, 2007).



Vorgehensweise

OWASP ASVS definiert Anforderungen an Verifikation und Dokumentation. Der Standard beschreibt dabei insgesamt vier hierarchische Verifikationslevel (kurz: "Level"). Jedes erfordert ein unterschiedlichen Grad an Abdeckung (engl. "Coverage") und Strenge (engl. "Rigor") der Verifikation. Dadurch wird etwa bei Level 2 eine Strenge und Abdeckung an die Verifikation gefordert als bei Level 1.

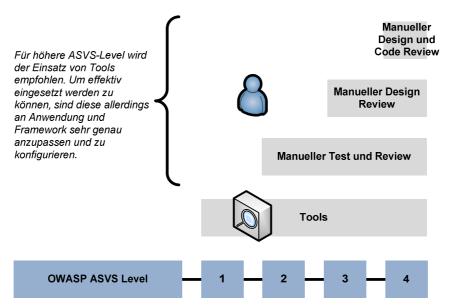


Abbildung 1 - OWASP ASVS Level

Die Verifikation der Sicherheit einer Webanwendung orientiert sich an deren Datenflusslogik. Dazu werden alle Datenflüsse zwischen sämtlichen Ein- und Austrittspunkten der untersuchten Anwendung (Verifikationsgegenstand, Taget of Verification oder TOV) analysiert. Bei komplexeren Anwendungen ist dies natürlich üblicherweise mit einem größeren Aufwand verbunden, was zu längeren und damit auch kostenintensiveren Verifikationen führt. Zur Bemessung der Komplexität stellt dabei die Anzahl an Codezeilen (Lines of Code) nur eine der möglichen Messgrößen dar. Auch der Einsatz vieler unterschiedlicher Technologien schlägt sich häufig in einem deutlich höheren Analyseaufwand nieder. Verwenden einfach aufgebaute Anwendungen lediglich nur einzelne Bibliotheken oder Frameworks, binden mittelgroße Anwendungen mitunter ganze Web 1.0-Anwendungen, komplexere sogar mitunter Web 2.0-Anwendungen oder auch eigene Webtechnologien ein.

Der ASVS-Standard unterteilt die Level 1 und 2 in jeweils zwei Verifikationsstufen. Als Beispiel benötigt eine Verifikation nach Level 1 dabei sowohl die Berücksichtigung der Anforderungen aus Level 1A als auch aus die aus Level 1B. Dadurch ist eine Anwendung, die konform zu Level 1A und 1B ist, automatisch auch konform zu Level 1.

Der Standard definiert für die Bereiche Verifikation und Dokumentation drei Arten von Prüfanforderungen: Allgemeinen Anforderungen (High-Level Requirements), Detailanforderungen und Berichtsanforderungen. Erstere legen übergeordnete Anforderungen für die Implementierung und Verifikation einer Anwendung fest. Die Detailanforderungen definieren spezifische Maßgaben an Implementierung und Verifikation (z.B. spezifische Elemente die zu verifizieren sind). Die Berichtsanforderungen schließlich legen fest, in welcher Form Ergebnisse einer ASVS-konformen Verifikation zu dokumentieren sind.



Zahlreiche Ressourcen werden durch die OWASP bereitgestellt, die Organisationen dabei zu unterstützen, sichere Anwendungen zu entwickeln und zu betreuen. Der OWASP ASVS⁶, OWASP Anhang Contract und OWASP ESAPI⁷ können dazu genutzt werden einen Softwareentwicklungsprozess (Software Development Life Cycle, SDLC) mit Sicherheitsaspekten zu erweitern, wie dies in folgender Abbildung dargestellt ist:

An dieser Stelle kann An dieser Stelle finden An dieser Stelle finden An dieser Stelle können die ESAPI eingesetzt Sie heraus, ob Ihre Sie heraus, ob ihre Sie den Contract Annex Anwendungen enthält werden um die Anwendung Schwachnutzen, um ein ASVS-Schwachstellen zu weiterhin Schwachstellen wie Cross-Site Level zu spezifizieren stellen wie Cross-Site korrigieren. Scripting (XSS), SQL Injection, CSRF, etc., Scripting (XSS), SQL Injection, CSRF, etc. besitzt. Entwickle und Führe initiale Lehre und Korrigiere und Verifiziere Verifikation Erfasse Plane innerhalb des durch **SDLC** Hier kann der Hier kann der **ASVS zum ASVS zum** Einsatz Einsatz kommen kommen

Abbildung 2 - Mögliches Vorgehen zur Integration der Verifikation als SDLC-Aktivität⁸

⁶ Für mehr Informationen zur Verwendung von ASVS in Verträgen, siehe *Contract Annex* (OWASP, 2009).

⁷ Für mehr Informationen dazu, wie die ESAPI in Anwendungen integriert werden kann, diese "ESAPI-Enabled" (oder "ES-Enabled") gemacht werden können, sie das OWASP ESAPI Projekt (OWASP 2009). ⁸ Für mehr Informationen zur Einführung von sicherheitsrelevanten Aktivitäten in Ihren existierenden SDLC, siehe hierzu das *OWASP CLASP* (OWASP 2008) oder das *OWASP SAMM* Projekt (OWASP 2009).



Danksagungen

Wir danken der OWASP Foundation für das Sponsoring des OWASP Application Security Verification Standard Projekts im Rahmen des OWASP Summer of Code 2008.

Projektleitung: ⁹ Mike Boberski (Booz Allen Hamilton)

Autoren: 10 Mike Boberski (Booz Allen Hamilton), Jeff Williams

(Aspect Security), Dave Wichers (Aspect Security)

Deutsche Übersetzung¹¹: Matthias Rohr (SEC Consult), Olaf Schulz (T-Systems)

Projekt-Sponsoren:







Besonderer Dank geht an die Beiträge von: Pierre Parrend, welcher beim OWASP Summer of Code 2008 als Reviewer unterstützt hat; Andrew van der Stock (Aspect Security); Nam Nguyen (Blue Moon Consulting); John Martin (Boeing); Gaurang Shah (Booz Allen Hamilton); Theodore Winograd (Booz Allen Hamilton); Stan Wisseman (Booz Allen Hamilton); Barry Boyd (CGI Federal); Steve Coyle (CGI Federal); Paul Douthit (CGI Federal); Ken Huang (CGI Federal); Dave Hausladen (CGI Federal); Mandeep Khera (Cenzic); Scott Matsumoto (Cigital); John Steven (Cigital); Stephen de Vries (Corsaire); Dan Cornell (Denim Group); Shouvik Bardhan (Electrosoft), Dr. Sarbari Gupta (Electrosoft); Eoin Keary (Ernst & Young); Richard Campbell (Federal Deposit Insurance Corporation); Matt Presson (FedEx); Jeff LoSapio (Fortify Software); Liz Fong (National Institute of Standards and Technology); George Lawless (Noblis); Dave van Stein (ps_testware); Terrie Diaz (SAIC); Ketan Dilipkumar Vyas (Tata Consultancy Services); Bedirhan Urgun (TURKCELL); Dr. Thomas Braun (United Nations); Colin Watson (Watson Hall); Jeremiah Grossman (WhiteHat Security); und schließlich sei der Application Security Verification Community sowie allen anderen am Thema Trusted Web Computing Interessierten für ihren enthusiastischen Rat und Unterstützung während des gesamten Projektes gedankt.

⁹ E-Mail: mike.boberski@owasp.org

¹⁰ E-Mail: jeff.williams@owasp.org, dave.wichers@owasp.org

¹¹ E-Mail: mail@matthiasrohr.de



Anmerkungen zur Deutschen Übersetzung

Im Rahmen der Übersetzung dieses Standards wurde darauf Wert gelegt, wenn dies möglich war, Fachausdrücke in deutscher Sprache zu verwenden. Dennoch liegt der Fokus dieses Standards in erster Linie in seiner Praxistauglichkeit, weshalb an vielen Stellen Englischen Ausdrücken der Vorzug gegeben wurde, wenn sich diese auch im deutschen Sprachgebrauch gefestigt haben (z.B. "Session Management" oder "Logging").

Der Begriff "Security Control" wird in diesem Dokument mit Sicherheitsmerkmal übersetzt.

Für jede Anmerkung zur Übersetzung bin ich sehr dankbar. Zu erreichen bin ich unter mail@matthiasrohr.de.

Besonders danke ich Olaf Schulz (T-Systems) für den Review der Deutschen Übersetzung dieses Standards.

Matthias Rohr, Hamburg, 10. Oktober 2010



Verifikationslevel für Anwendungssicherheit

Der ASVS-Standard definiert insgesamt vier Verifikationsstufen ("Level"), die sich hinsichtlich der Verifikationsabdeckung sowie der Verifikationsstrenge unterscheiden. Die Strenge ergibt sich dabei aus der für die Verifikation einer Sicherheitsanforderung anwendeten Methode und den Grad an Strenge der durchgeführten Verifikation. Tools stellen einen wichtigen Bestandteil jedes ASVS-Levels dar. Allerdings ist bei dem Einsatz solcher Hilfsmittel zu beachten, dass diese stets an die jeweilige Anwendung und das eingesetzte Framework angepasst und konfiguriert werden müssen um den gewünschten Nutzen zu erzielen. Zudem sind bei allen Leveln, die Resultate der eingesetzten Tools stets manuell zu verifizieren.

Es liegt in der Verantwortung eines Testers festzustellen, ob eine TOV die für ein angestrebtes Level geforderten Anforderungen erfüllt. Nur wenn die Anwendung die geforderten Anforderungen eines bestimmten Levels erfüllt darf diese als OWASP ASVS Level N Applikation bezeichnet werden. Wobei N hier für den Verifikationslevel, dem die Anwendung erfolgreich unterzogen wurde, steht. Sollte eine Anwendung zwar nicht sämtliche Anforderungen eines geprüften Levels, jedoch all jene eines niedrigeren Levels erfüllen, kann für die betreffende Anwendung das Bestehen des jeweilig niedrigen Levels festgestellt werden. Dieser Standard verwendet den Begriff Prüfer ("Verifier") um eine bestimmte Person oder Team zu bezeichnen, welches ein Review der Anwendung gegen diese Anforderungen durchgeführt.

Auch wenn die Spezifikation einer Anwendung ein bestimmtes OWASP ASVS-Level vorsieht, ist es trotzdem möglich diese um einzelne Detailanforderungen höherer ASVS-Level zu erweitern. Beispielsweise mag eine Organisation aus dem Finanzsektor die Verifikation einer Anwendung mit niedrigem Risiko nur ein OWASP ASVS Level 2 verlangen, jedoch zusätzlich die Verifikation von Schadcode (siehe V13, ausschließlich Level 4) fordern. Auch andere organisatorische oder geschäftliche Anforderungen, wie etwa im Bezug auf Compliance zu bestimmten Sicherheitsanforderungen oder Gesetzen, lassen sich so adressieren.

Ein Verifikationslevel 0 existiert nicht. Desweiterhin müssen zur Erlangung eines bestimmten Levels alle identifizierten Schwachstellen behoben und die Anwendung erneut verifiziert werden.

Level 1 - Automatische Verifikation

Level 1 ("Automatische Verifikation") ist üblicherweise für Anwendungen zweckmäßig, bei denen ein nur ein eingeschränkter Vertrauensgrad in den korrekten Einsatz bestimmter Sicherheitsmechanismen erforderlich ist. Als Sicherheitsbedrohungen¹² werden hierbei vor allem Viren und Würmer (Ziele werden hierbei eher zufällig durch weitangelegte Scans und Angriff der verwundbarsten Systeme) angenommen. Zu verifizieren ist sämtlicher Code, welcher zur Erstellung einer Anwendung entwickelt oder modifiziert wurde.

Bei Level 1 sieht die Verifikation den Einsatz automatisierter Tools in Verbindung mit manueller Verifizierung vor. Die manuelle Verifizierung ist dabei aber nicht dazu vorgesehen, eine vollständige Verifikation der Anwendungssicherheit für diesen Level durchzuführen. Stattdessen soll diese Aktivität lediglich zur Verifizierung der Korrektheit der durch die automatisierten Tests gelieferten Ergebnisse dienen.

Wie bereits erwähnt wurde, ist Level 1 aus zwei Verifikationsstufen aufgebaut. Level 1A bezieht sich dabei auf den Einsatz von Schwachstellen-Scannern (Dynamische Analyse) wogegen sich Level 1B auf den Einsatz von Tools zur automatisierten Sourcecodeanalyse (Statische Analyse) richtet. Bei der Verifikation kann jeder dieser Teile unabhängig voneinander oder gemeinsam geprüft werden, um

¹² Für mehr Informationen zur Durchführung von manuellen Verifikationen mittels Penetrationstests, siehe das *OWASP Testing Guide* (OWASP, 2008).



ein ASVS-Level 1-Rating zu erhalten. Der strukturelle Aufbau von Level 1 ist in Abbildung 3 dargestellt.

Auch wenn es in Situationen angebracht sein kann, eine Anwendung nur gegen 1A oder 1B zu verifizieren, bietet doch keines dieser beiden Teillevel alleine denselben Verifikationsgrad hinsichtlich Prüfgenauifkeit und Prüfstrenge, wie eine vollständige Prüfung nach Level 1 (1A und 1B).

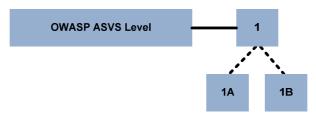


Abbildung 3 - OWASP ASVS Level 1, 1A und 1B

Im Folgenden sind die minimalen Anforderungen an Anwendungen gemäß Level 1, 1A oder 1B aufgeführt:

Verifikationsumfang

L1.1 Die Verifikation umfasst sämtlichen Code, welcher zur Erstellung einer Anwendung entwickelt oder modifiziert wurde.

Entscheidungsanforderungen an Sicherheitsmechanismen

Keine Level 1 definiert keine Anforderungen, wie Sicherheitsmechanismen bestimmte Entscheidungen zu treffen haben.

Einsatzanforderungen an Sicherheitsmechanismen

Keine Level 1 definiert keine Anforderungen, wie Sicherheitsmechanismen einzusetzen sind.

Implementierungsanforderungen an Sicherheitsmechanismen

Keine Level 1 definiert keine Anforderungen, wie Sicherheitsmechanismen zu entwickeln sind.

Verifikationsanforderungen an Sicherheitsmechanismen

- L1.2 Führe einen dynamischen Scan der Anwendung gemäß Anforderungen zu Level 1A, Abschnitt "Anforderungen an detaillierte Verifikation" durch.
- L1.3 Führe eine Sourcecodeanalyse der Webanwendung gemäß Anforderungen zu Level 1B, Abschnitt "Anforderungen an detaillierte Verifikation" durch.

Anforderungen zu Level 1, welche nur eines der beiden Teillevel (1A oder 1B) fordern, müssen auch nur für eines dieser Level geprüft werden. Zudem muss das vom Prüfer eingesetzte Tool nicht



notwendigerweise eine bestimmte Anforderung auch explizit prüfen können. Stattdessen kann dies in diesem Fall durch eine manuelle Verifikation des Prüfers erfolgen. ¹³

Anforderungen an den Verifikationsbericht

L1.4 Erstelle einen Prüfbericht welcher die Sicherheitsarchitektur der Anwendung darstellt. Dies beinhaltet die Auflistung der Anwendungskomponenten sowie das Ergebnis der Verifikation gemäß "Anforderungen an den Prüfbericht".

Wie in Abbildung 4 dargestellt, kann es sich bei Anwendungskomponenten im Sinne von Level 1 sowohl um einzelne Sourcedateien, spezifische Gruppen dergleichen, Bibliotheken und/oder auch ausführbare Dateien handeln.

Anwendungskomponenten sind zu identifizieren und Anwendung oder IT-Umgebung zuzuordnen. Eine Spezielle Sortierung oder Organisation von diesen ist allerdings nicht erforderlich. Die Anwendung kann so als Gruppe von Komponenten einer einzelnen monolithischen Entität betrachtet werden. Der oder die Pfade, über welche eine Benutzerangabe die Anwendung durchläuft muss aber nicht identifiziert oder dokumentiert werden.

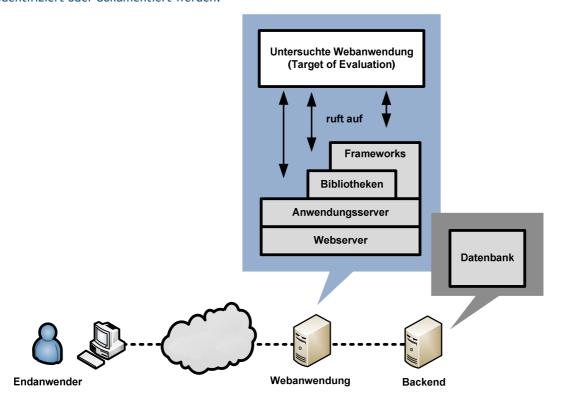


Abbildung 4 - Beispiel einer Sicherheitsarchitektur bei OWASP ASVS Level 1

¹³ Für mehr Informationen zur Durchführung von Manuellen Verifikationen mittels Penetrationstests, siehe das *OWASP Testing Guide* (OWASP, 2008).



Level 1A - Dynamischer Scan (partiell automatische Verifikation)

Anforderungen an die Prüfung von Sicherheitsmechanismen mittels dynamischer Scans

Dynamisches Scannen (auch bekannt als "Application Vulnerability Scanning") basiert auf der Verwendung automatisierter Tools mit denen die Schnittstellen einer laufenden Anwendung auf Schwachstellen untersucht werden. Zu beachten ist dabei, dass hierdurch keine ausreichende Verifikation der Korrektheit von Design, Implementierung oder Einsatz bestimmter Sicherheitsmechanismen durchgeführt wird, dies aber dennoch eine im Sinne von Level 1 hinreichende Verifikation darstellt. Die Verifikation hat dabei die folgenden Anforderungen an die Sicherheitsarchitektur zu berücksichtigen:

- L1A.1 Führe dynamische Scans nach Anforderungen zu Level 1A aus Abschnitt "Anforderungen an detaillierte Verifikation" durch.
- L1A.2 Verifiziere die Ergebnisse aller dynamischen Scans mittels manuellem Penetrationstest oder Code Review. Nicht verifizierte Findings automatisierter Tools stellen kein qualifiziertes Ergebnis dar und sind damit hier nicht verwertbar.

Sofern dies nicht bereits durch das eingesetzte Tool automatisch erfolgt, sind mehrere Instanzen einer Schwachstelle, welche auf dieselbe Ursache ("Root Cause") zurückgeführt werden können, als ein einzelnes Finding zu betrachtet.

Level 1B - Sourcecodeanalyse (partiell automatische Verifikation)

Anforderungen an die Prüfung von Sicherheitsmechanismen mittels Sourcecodeanalyse

Sourcecodeanalyse (auch bekannt als "Statische Analyse") basiert auf der Verwendung automatisierter Tools. Mit diesen wird der Sourcecode einer Anwendung auf Muster untersucht, welche auf das Vorhandensein bestimmter Schwachstelle hindeuten. Zu beachten ist dabei, dass hierdurch keine ausreichende Verifikation des korrekten Designs, der Implementierung oder der Verwendung eines bestimmten Sicherheitsmechanismus durchgeführt wird, dies jedoch eine für Level 1 hinreichende Verifikation darstellt. Für die Verifikation sind dabei die folgenden Anforderungen an die Sicherheitsarchitektur zu berücksichtigen:

- L1B.1 Führe eine Sourcecodeanalyse der Anwendung gemäß Anforderungen zu Level 1B, aus Abschnitt "Anforderungen an detaillierte Verifikation" durch.
- Verifiziere alle Ergebnisse der durchgeführten Sourcecodeanalyse mittels manuellem Penetrationstest oder Code Review. Nicht verifizierte Findings automatisierter Tools stellen kein qualifiziertes Ergebnis dar und sind damit hier nicht verwertbar.

Sofern dies nicht bereits durch das eingesetzte Tool automatisch erfolgt, sind mehrere Instanzen einer Schwachstelle, welche auf dieselbe Ursache (Root Cause) zurückgeführt werden können, als ein einzelnes Finding zu betrachtet.

Level 2 - Manuelle Verifikation

Level 2 ("Manuelle Verifikation") ist üblicherweise für solche Anwendungen zweckmäßig, die Benutzer- oder Business-to-Business-Transaktionen, Kreditkarten- oder personenbezogene Daten verarbeitet. Auf Basis von Level 2 lässt sich ein gewisser Vertrauensgrad an den korrekten Einsatz wie auch die korrekte Arbeitsweise der verwendeten Sicherheitsmechanismen ermitteln.

Als relevante Sicherheitsbedrohungen sind hier typischerweise Viren, Würmer sowie Gelegenheitstäter zu berücksichtigen.



Die Verifikation umfasst sämtlichen Code der zur Erstellung der Anwendung entwickelt oder modifiziert wurde. Zusätzlich sind eingesetzte Komponenten von Drittanbietern in die Verifikation mit einzubeziehen welche Sicherheitsfunktionen für die Anwendung bereitstellen. Wie aus der folgenden Abbildung hervorgeht, ist Level 2 aus Level 2A und 2B aufgebaut.



Abbildung 5 - OWASP ASVS Level 2, 2A und 2B

Auch wenn es in Situationen angebracht sein kann, eine Anwendung nur gegen 2A oder 2B zu verifizieren, so bietet keines dieser beiden Level alleine denselben Verifikationsgrad, hinsichtlich Prüfgenauigkeit und Prüfstrenge, wie eine vollständige Prüfung nach Level 2 (also 2A auch 2B). Zudem stellt Level 2 eine Obermenge von Level 1 dar. Daher existieren für das Erreichen von Level 2 auch keinerlei Vorgaben hinsichtlich des Einsatzes automatisierter Tools.

Stattdessen hat der Prüfer hier die Möglichkeit, für sämtliche der genannten Anforderungen ausschließlich auf manuelle Techniken zurückzugreifen. Sofern Ergebnisse automatisierter Tools vorliegen, hat der Prüfer allerdings die Möglichkeit diese in die Analyse mit einfließen zu lassen. Aus dem Bestehen einer Anforderung nach Level 1 erfolgt nicht, dass diese auch automatisch für Level 2 erfüllt ist. Dies ergibt sich daraus, dass automatisierte Tools keinen mit einer manuellen Verifikation vergleichbaren Zusicherungsgrad hinsichtlich der Erfüllung einer Vorgabe liefern.

Für manuelle Techniken wird allerdings stets von der Nutzung solcher Tools ausgegangen. Dies schließt die Anwendung von Test- oder Analysetools mit ein, wobei es sich durchaus auch um automatisierte Tools im Sinne von Level 1 handeln kann. Auch ist es möglich, dass Tools über bestimmte Logik verfügen mit der sie automatisch Schwachstellen in einer Anwendung identifizieren können. Dennoch stellen diese Tools aber stets nur ein Mittel zur Unterstützung des Prüfers dar, mit dem er Sicherheitsmechanismen identifizieren oder untersuchen kann.

Im Folgenden sind die minimalen Anforderungen für Anwendungen für Level 2 aufgeführt:

Verifikationsumfang

- L2.1 Die Verifikation umfasst sämtlichen Code, welcher zur Erstellung einer Anwendung entwickelt oder modifiziert wurde. Diese Anforderung wurde mit Level 1 eingeführt.
- L2.2 Die Verifikation umfasst sämtlichen Code von Drittanbieter-Frameworks, Bibliotheken sowie Sicherheitsfunktionen von Webservices, welche zur Abbildung von Sicherheitsaspekten innerhalb der Anwendung eingebunden sind. Diese Anforderung wird mit Level 2 neu eingeführt.

$Ents cheidungs an forderungen\ an\ Sicherheitsmechanismen$

- Verifiziere, dass sämtliche technischen Sicherheitsmechanismen Sicherheitsprüfungen ausschließlich auf Basis einer Positivliste ("Whitelisting") durchführen. Diese Anforderung wird mit Level 2 neu eingeführt.
- Verifiziere, dass alle Sicherheitsmechanismen welche Sicherheitsprüfungen durchführen, die in einer Sicherheitsentscheidung resultieren, nicht umgangen werden können, wie dies durch Level 2A und 2B gefordert und in Abschnitt "Anforderungen an detaillierte Verifikation" spezifiziert ist. Diese Anforderung wird mit Level 2 neu eingeführt.



Einsatzanforderungen an Sicherheitsmechanismen

L2.5 Verifiziere, dass alle Sicherheitsmechanismen überall dort in der Anwendung zum Einsatz kommen wo dies erforderlich ist, dass diese zentral und serverseitig implementiert sind und die Anforderungen zu Level 2 aus Abschnitt "Anforderungen an detaillierte Verifikation" erfüllen. Diese Anforderung wird mit Level 2 neu eingeführt.

Implementierungsanforderungen an Sicherheitsmechanismen

Keine	Level 2 definiert keine Anforderungen hinsichtlich der Entwicklung von
	Sicherheitsmechanismen.

Verifikationsanforderungen an Sicherheitsmechanismen

- L2.6 Prüfe die Anwendung mittels manueller Penetrationstests gemäß Anforderungen von Level 2A aus Abschnitt "Anforderungen an detaillierte Verifikation". Diese Anforderung wird mit Level 2 neu eingeführt.
- L2.7 Prüfe die Anwendung mittels manuellem Code Review gemäß Anforderungen von Level 2B aus Abschnitt "Anforderungen an detaillierte Verifikation". Diese Anforderung wird mit Level 2 neu eingeführt.

Anforderungen von Level 2, welche sich nur auf 2A oder 2B beziehen, erfordern auch nur die Anwendung eines dieser Verifikationsschritte.

Der Prüfer kann auch Gebrauch von Ergebnissen automatisierter Scanner oder Codeanalysen als Bestandteil einer Verifikation nach Level 2 machen. Eine automatisierte Verifikation darf jedoch nicht anstelle eines manuellen Reviews durchgeführt werden, wie dieses für jede der Anforderung aus Level 2 erforderlich ist.

Anforderungen an den Verifikationsbericht

L2.8 Erstelle einen Prüfbericht der Sicherheitsarchitektur. Hierzu ist es erforderlich, dass die relevanten Komponenten mitsamt den Resultaten der Verifikation gemäß der Anforderungen aus Abschnitt "Anforderungen an den Prüfbericht" in einer konzeptionellen Architektursicht dargestellt werden. Dies erweitert die Berichtsvorgabe welche mit Level 1 eingeführt wurde.

In Level 2 werden Anwendungskomponenten sowohl als einzelne Sourcedateien, spezifische Gruppen dergleichen, Bibliotheken und/oder ausführbare Dateien betrachtet, die in einer konzeptionellen Architektursicht (z.B. Komponenten für Model-View-Controller (MVC), Geschäftslogik, oder Data Layer) organisiert sind.

Für Level 2 ist es erforderlich, die Pfade einer bestimmten Benutzeranfrage innerhalb der Anwendung so zu dokumentieren, wie dies in der unten dargestellten Abbildung gezeigt ist. Die Untersuchung sämtlicher Pfad ist hierbei jedoch nicht erforderlich.



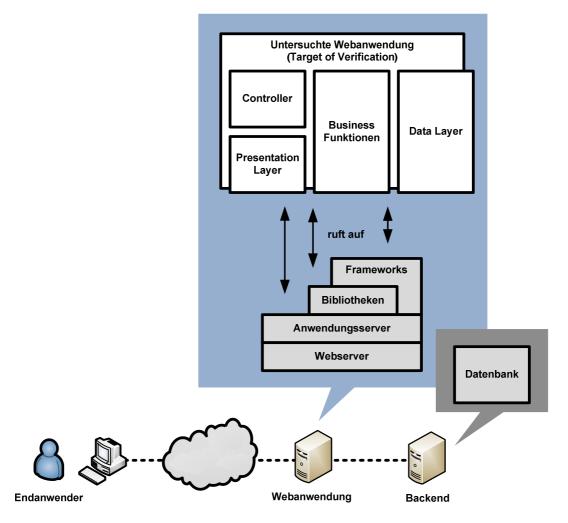


Abbildung 6 - OWASP ASVS Level 2 Beispiel einer Sicherheitsarchitektur



Level 2A - Security Test (partiell manuelle Verifikation)

Verifikationsanforderungen für manuelle Penetrationstests von Sicherheitsmechanismen

Manuelle anwendungsseitige Sicherheitstests basieren auf der Erstellung von dynamischen Tests zur Verifikation der Korrektheit von Design, Implementierung und Einsatz bestimmter Sicherheitsmechanismen.

Der Umfang der Verifikation wird anhand der Anforderungen an die Sicherheitsarchitektur dieses Levels festgelegt.

L2A.1 Führe einen manuellen Penetrationstests auf Anwendungsebene gemäß Anforderungen von Level 2A aus Abschnitt "Anforderungen an detaillierte Verifikation" durch. Diese Anforderung wird mit Level 2 neu eingeführt.

Dort wo es angebracht ist, kann der Prüfer mittels Stichproben den korrekten Einsatz eines bestimmten Sicherheitsmechanismus verifizieren. Der Prüfer kann Schwachstellenmuster (Vulnerability Patterns) dokumentieren, so dass Entwickler damit entsprechende Vorkommen im Code identifizieren und korrigieren können. Mehrere Instanzen eines Musters, welche auf dieselbe Ursache (Root Cause) zurückgeführt werden können, sollten dabei als ein einzelnes Finding betrachtet werden.

Level 2B - Code Review (partiell manuelle Verifikation)

Verifikationsanforderungen für Code Reviews von Sicherheitsmechanismen

Manuelle Code Reviews basierend auf von einem Menschen durchgeführten Suche und Analyse vom Sourcecode mit dem Ziel, die Korrektheit von Design, Implementierung sowie den Einsatz bestimmter Sicherheitsmechanismen zu verifizieren. Diese Analyse wird gewöhnlich durch den Einsatz bestimmter Analysetools unterstützt, wobei es sich allerdings auch um allgemein verwendete Tools wie Sourcecodeeditoren oder IDEs handeln kann.

Der Umfang der Verifikation wird anhand der Anforderungen an die Sicherheitsarchitektur dieses Levels festgelegt.

L2B.1 Führe einen manuellen Code Review auf Basis der Anforderungen zu Level 2B aus Abschnitt "Anforderungen an detaillierte Verifikation". Diese Anforderung wird mit Level 2 neu eingeführt.

Dort wo es angebracht ist, kann der Prüfer mittels Stichproben den korrekten Einsatz eines bestimmten Sicherheitsmechanismus verifizieren. Der Prüfer kann Schwachstellenmuster (Vulnerability Patterns) dokumentieren, so dass Entwickler damit entsprechende Vorkommnisse im Code identifizieren und korrigieren können. Mehrere Instanzen desselben Musters, welche auf dieselbe Ursache (Root Cause) zurückgeführt werden können, sollten dabei als ein einzelnes Finding betrachtet werden.

Level 3 - Design Verifizierung

Level 3 ("Design Verifizierung") ist üblicherweise für all jene Anwendungen zweckmäßig, bei denen Business-to-Business-Transaktionen, medizinische Daten, geschäftskritische oder sensible Funktionen implementiert sind oder welche jegliche sonstige Form sensibler Informationen verarbeiten.

Als Sicherheitsbedrohungen sind hierbei vor allem Viren oder Würmer aber auch Gelegenheitstäter sowie möglicherweise gezielt agierende Angreifer (sachkundig, motivierte, die es auf konkrete Ziele abgesehen haben und hierfür auch über Spezialtools verfügen) zu berücksichtigen.



Die Verifikation umfasst sämtlichen neu entwickelten oder modifizierten Code einer Anwendung sowie die Untersuchung aller Komponenten von Drittanbietern die Sicherheitsfunktionen für die Anwendung bereitstellen. Level 3 stellt sicher, dass Sicherheitsmechanismen korrekt arbeiten und überall dort in der Anwendung eingesetzt werden, wo dies notwendig ist um eine vorhanden anwendungsseitige Sicherheitsrichtlinie umzusetzen. Wie aus der folgenden Abbildung hervorgeht, ist Level 3 in keine weiteren Verifikationsstufen unterteilt.

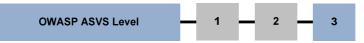


Abbildung 7 - OWASP ASVS Level 3

Im Folgenden werden die allgemeinen Minimalanforderungen zu Level-3-Anwendungen beschrieben:

Verifikationsumfang

- L3.1 Die Verifikation umfasst sämtlichen Code, welcher zur Erstellung einer Anwendung entwickelt oder modifiziert wurde. Diese Anforderung wurde mit Level 1 eingeführt.
- L3.2 Die Verifikation umfasst sämtlichen Code, alle Drittanbieter-Frameworks, Bibliotheken sowie Sicherheitsfunktionen von Webservices, welche von der Anwendung für die Abbildung von Sicherheitsaspekten genutzt werden. Diese Anforderung wurde durch Level 2 eingeführt.
- L3.3 Die Verifikation umfasst sämtlichen Code, alle Drittanbieter-Frameworks, Bibliotheken und Services, die von dieser Anwendung genutzt werden. Diese Vorgabe wird durch Level 3 neu eingeführt.

Entscheidungsanforderungen an Sicherheitsmechanismen

- L3.4 Verifiziere für alle technischen Sicherheitsmechanismen, dass diese Sicherheitsprüfungen ausschließlich gegen eine Positivliste ("Whitelisting") durchführen. Diese Anforderung wurde mit Level 2 eingeführt.
- Verifiziere, dass alle Sicherheitsmechanismen welche Sicherheitsprüfungen durchführen, die in einer Sicherheitsentscheidung resultieren, nicht umgangen werden können, wie dies durch Level 3A und 3B gefordert und in Abschnitt "Anforderungen an detaillierte Verifikation" spezifiziert ist. Diese Anforderung wurde mit Level 2 eingeführt.

Einsatzanforderungen an Sicherheitsmechanismen

L3.6 Verifiziere, dass alle Sicherheitsmechanismen überall dort in der Anwendung zum Einsatz kommen wo dies erforderlich ist, dass diese zentral und serverseitig implementiert sind und die Anforderungen zu Level 3 aus Abschnitt "Anforderungen an detaillierte Verifikation" erfüllen. Diese Anforderung wird mit Level 3 neu eingeführt.

Implementierungsanforderungen an Sicherheitsmechanismen

Keine Level 3 definiert keine Anforderungen hinsichtlich der Entwicklung von Sicherheitsmechanismen.

Verifikationsanforderungen an Sicherheitsmechanismen

L3.7 Prüfe die Anwendung mittels manueller Penetrationstests gemäß Anforderungen zu Level 3A aus Abschnitt "Anforderungen an detaillierte Verifikation". Diese Anforderung wurde mit Level 2 eingeführt.



L3.8 Dokumentiere die Sicherheitsarchitektur und nutze dies zur Verifikation des korrekten Designs und des Einsatzes von Sicherheitsmechanismen mittels Bedrohungsmodellierung (Threat Modeling). Diese Anforderung wird mit Level 3 neu eingeführt.

Anforderungen an den Verifikationsbericht

L3.9 Erstelle einen Prüfbericht der Sicherheitsarchitektur. Hierzu ist es erforderlich, dass die relevanten Komponenten mitsamt der Informationen aus der Bedrohungsmodellierung sowie der Resultate der Verifikation gemäß der Anforderungen aus Abschnitt "Anforderungen an den Prüfbericht" in einer konzeptionellen Architektursicht dargestellt werden. Dies erweitert die entsprechende Berichtsvorgabe aus Level 2.

Bei Level 3 werden unter Anwendungskomponenten sowohl einzelne Sourcedateien, separate Gruppen dergleichen, Bibliotheken und/oder ausführbaren Dateien verstanden. Diese werden in einer konzeptionellen Architektursicht (z.B. Komponenten für Model-View-Controller (MVC), Geschäftslogik, oder Data Layer) organisiert. Auch die aus der Bedrohungsmodellierung gewonnen Informationen (z.B. Threat Agents und Assets) sind bei Level 3 zu beschreiben.

Die folgende Abbildung zeigt ein Beispiel hierzu. Die dort dargestellte MVC-Anwendung ist dabei aus den Anwendungskomponenten Anwendungsserver, Serveranwendung, eigenem Code, verschiedenen Bibliotheken sowie einer Datenbankanwendung aufgebaut. Für Level 2 ist es erforderlich, die Pfade einer bestimmten Benutzeranfrage innerhalb der Anwendung so zu dokumentieren, wie dies in der unten gezeigten Abbildung gezeigt ist. Bei Level 3 sind außerdem sämtliche potenziellen Pfade, die auf konzeptioneller Ebene identifiziert wurden, zu untersuchen.



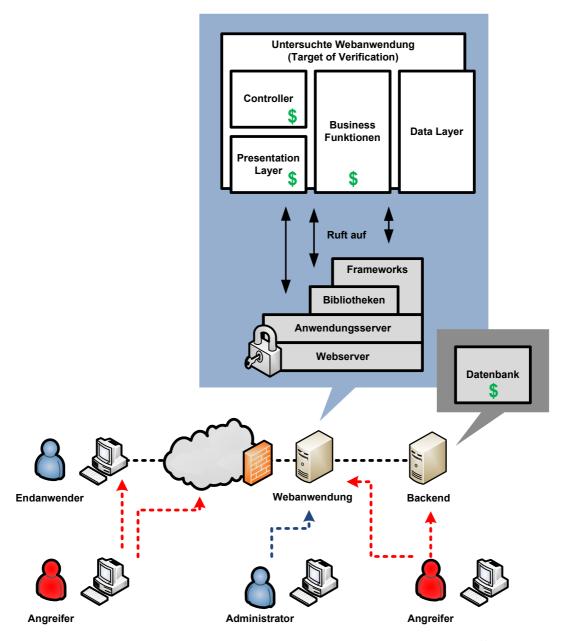


Abbildung 8 - OWASP ASVS Level 3 Beispiel einer Sicherheitsarchitektur¹⁴

¹⁴ Assets werden in dieser Abbildung durch Dollar-Zeichen ausgewiesen.



Level 4 - Interne Verifikation

Level 4 ("Interne Verifikation") ist üblicherweise für kritische Anwendungen zweckmäßig, die Leib und Leben, kritische Infrastrukturen oder Informationen zur nationalen Verteidigung schützen oder allgemein sehr sensible Assets verarbeiten. Level 4 stellt dazu sicher, dass Sicherheitsmechanismen korrekt arbeiten, überall dort in der Anwendung genutzt werden, wo dies erforderlich ist um anwendungsspezifische Sicherheitsrichtlinien umzusetzen und dass Secure-Coding-Anforderungen eingehalten werden.

Als Sicherheitsbedrohungen sind hier entschlossene Angreifer (sachkundige und motivierte Personen, die auf konkrete Ziele fokussiert sind und auch auf eigens erstellte Spezialtools zurückgreifen können) zu berücksichtigen.

Die Verifikation auf Basis von Level 4 umfasst sämtlichen von der Anwendung verwendeten Code. Wie in der folgenden Abbildung dargestellt, unterteilt sich Level 4 in keine weiteren Verifikationsstufen.



Abbildung 9 - OWASP ASVS Level 4

Im Folgenden werden die allgemeinen Minimalanforderungen zu Level-4-Anwendungen beschrieben:

Verifikationsumfang

- L4.1 Die Verifikation beinhaltet sämtlichen Code, welcher zur Erstellung einer Anwendung entwickelt oder modifiziert wurde. Diese Anforderung wurde mit Level 1 eingeführt.
- L4.2 Die Verifikation beinhaltet sämtlichen Code aller Drittanbieter-Frameworks, Bibliotheken sowie Sicherheitsfunktionen von Webservices, welche zur Abbildung von Sicherheitsaspekten innerhalb der Anwendung eingebunden sind. Diese Anforderung wurde durch Level 2 eingeführt.
- L4.3 Die Verifikation umfasst sämtlichen Code aller Drittanbieter-Frameworks, Bibliotheken und Services, die mit dieser Anwendung verbunden sind. Diese Vorgabe wurde durch Level 3 eingeführt.
- L4.4 Die Verifikation umfasst sämtlichen Code der Anwendung (Frameworks, Bibliotheken, Laufzeitumgebungen, Entwickler-, Build- sowie Deployment-Tools). Die Verifikation umfasst jedoch nicht den Sourcecode der Softwareplattform, wie z.B. den des Applikationsservers, der Datenbank, einer Virtuellen Maschine oder des Betriebssystems. Bei letztgenannten Komponenten ist davon auszugehen, dass diese bereits einer ausgiebigen Analyse unterzogen worden sind. Diese Vorgabe wird mit Level 4 neu eingeführt.

Entscheidungsanforderungen an Sicherheitsmechanismen

- Verifiziere für alle technischen Sicherheitsmechanismen, dass diese Sicherheitsprüfungen ausschließlich gegen eine Positivliste durchführen. Diese Anforderung wurde mit Level 2 eingeführt.
- Verifiziere, dass alle Sicherheitsmechanismen, welche Sicherheitsprüfungen durchführen, die in einer Sicherheitsentscheidung resultieren, nicht umgangen werden können, wie dies durch Level 3A und 3B gefordert und in Abschnitt "Anforderungen an detaillierte Verifikation" spezifiziert ist. Diese Anforderung wurde mit Level 2 eingeführt.



Einsatzanforderungen an Sicherheitsmechanismen

Verifiziere, dass alle Sicherheitsmechanismen überall dort in der Anwendung zum Einsatz kommen wo dies erforderlich ist, dass diese zentral und serverseitig implementiert sind und die Anforderungen zu Level 3 aus Abschnitt "Anforderungen an detaillierte Verifikation" erfüllen. Diese Anforderung wurde durch Level 3 eingeführt.

Implementierungsanforderungen an Sicherheitsmechanismen

Verifiziere, dass die Anwendung keinen Schadcode gemäß der Anforderungen zu Level 4 aus Abschnitt "Anforderungen an detaillierte Verifikation" enthält. Diese Anforderung wird durch Level 4 neu eingeführt.

Verifikationsanforderungen an Sicherheitsmechanismen

- L4.9 Führe eine manuelle Verifikation gemäß Anforderungen zu Level 4 aus Abschnitt "Anforderungen an detaillierte Verifikation" durch. Dies erweitert die entsprechende Vorgabe aus Level 3.
- L4.10 Dokumentiere die Sicherheitsarchitektur und nutze dies zur Verifikation des korrekten Designs und des Einsatzes von Sicherheitsmechanismen mittels Bedrohungsmodellierung (Threat Modeling). Diese Anforderung wurde durch Level 3 eingeführt.
- L4.11 Untersuche sämtlichen Code der für die Anwendung entwickelt oder modifiziert wurde manuell nach Schadcode¹⁵ gemäß Anforderungen zu Level 4 aus Abschnitt "Anforderungen an detaillierte Verifikation". Diese Anforderung wird durch Level 4 neu eingeführt.

Anforderungen an den Verifikationsbericht

L4.12 Erstelle einen Prüfbericht der Sicherheitsarchitektur gemäß Anforderungen zu Level 3. Dieser Umfass sämtlichen Anwendungscode sowie die Ergebnisse der Verifikation gemäß Anforderungen aus Abschnitt "Anforderungen an detaillierte Verifikation". Dies erweitert die entsprechende Vorgabe aus Level 3.

Bei Level 4 ist die Anwendungsarchitektur so zu erfassen, wie dies in den Anforderungen zu Level 3 beschrieben wurde. Weiterhin erfordert Level 4 dass sämtlicher Anwendungscode, inklusive solchen der nicht explizit analysiert wurde, als Teil der Anwendungsdefinition identifiziert wurde (siehe hierzu die Abbildung unten). Dieser Code muss sämtliche Bibliotheken, Frameworks und sonstigen Code umfassen, welche die Anwendung einbezieht. Die Verifikation umfasst jedoch nicht den Code der Softwareplattform, z.B. den des Applikationsservers, der Datenbank, der Virtuellen Maschine oder des Betriebssystems. So sind beispielsweise auch keine Bibliotheken der Java Laufzeitumgebung bei Level 4 zu untersuchen.

¹⁵ Schadcode (Malicious Code) ist nicht dasselbe wie Malware, siehe hierzu die Definition aus dem Glossar zu Schadcode.



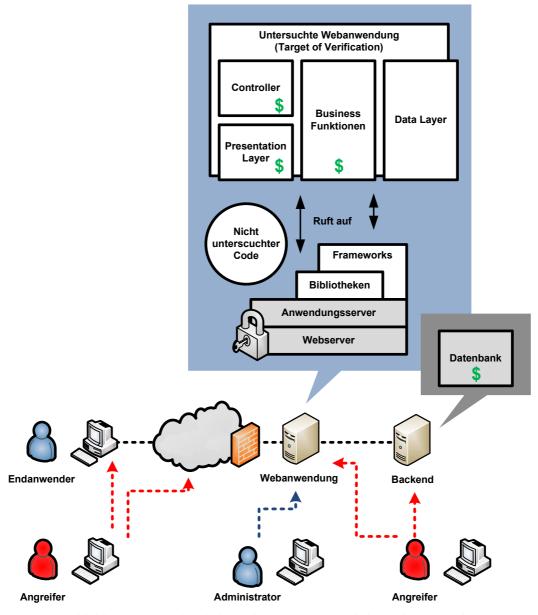


Abbildung 10 - OWASP ASVS Level 4 Beispiel für nichtbetrachteten Code

Interpretation und Praxisbeispiele von Anforderungen

Der OWASP ASVS Standard ist ein lebendiges Dokument. Sollten Sie die Sicherheit einer Anwendung auf Basis dieses Standards verifizieren, ist es stets empfehlenswert, vorab einen Blick auf bereits hierzu veröffentlichten Artikel zu werfen. Diese sind von der Projektseite des ASVS-Standards unter folgender URL abrufbar: http://www.owasp.org/index.php/ASVS#Articles_Below_-_More_About_ASVS_and_Using_It . Artikel der ASVS-Seite bieten Erläuterungen zu Anforderungen, Praxisbeispiele sowie zahlreiche Hilfestellungen.



Anforderungen an detaillierte Verifikation

Dieser Abschnitt des OWASP Application Security Verification Standard (ASVS) definiert Anforderungen an die detaillierte Verifikation welche aus den allgemeinen Anforderungen jedes Levels abgeleitet sind. Jeder untenstehende Abschnitt definiert ein bestimmtes Set von Anforderungen die detaillierte Verifikation.

Der ASVS-Standard definiert die folgenden Bereiche von Sicherheitsanforderungen:

- V1. Sicherheitsarchitektur und Dokumenation
- V2. Authentisierung
- V3. Session Management
- V4. Zugriffskontrollen
- V5. Eingabevalidierung
- V6. Ausgabeenkodierung /-Escaping
- V7. Kryptographie
- V8. Fehlerbehandlung und Logging
- V9. Datensicherheit
- V10. Kommunikationssicherheit
- V11. HTTP-Sicherheit
- V12. Sicherheitskonfiguration
- V13. Identifikation von Schadcode
- V14. Interne Sicherheit

Für jeden dieser Bereiche sind die Anforderungen, welche für jedes der Verifikationslevel erfüllt sein müssen, in folgender Form aufgeführt:

Level 1: Automatische Verifikation

Level 1A - Dynamischer Scan (partiell automatische Verifikation)

Level 1B - Sourcecode Scan (partiell automatische Verifikation)

Level 2: Manuelle Verifikation

Level 2A - Security Test (partiell manuelle Verifikation)

Level 2B - Code Review (partiell manuelle Verifikation)

Level 3: Design Verifikation

Level 4: Interne Verifikation



V1 - Verifikationsanforderungen zur Dokumentation der Sicherheitsachitektur

Die Dokumentation einiger Basisinformationen zur Sicherheitsarchitektur ist für alle ASVS-Level erforderlich um sowohl Vollständigkeit als auch Korrektheit (und auch Wiederholbarkeit, wenn eine Behebung erforderlich ist) einer durchgeführten Verifikation zu gewährleisten. Analysen können zielgenau adressiert und auf ein allgemeineres Level innerhalb der Sicherheitsarchitektur bezogen werden. Bei niedrigeren Leveln fordern die Anforderungen dabei zunächst nur einen recht allgemeinen Detaillierungsgrad der Sicherheitsarchitektur, der jedoch mit steigendem Level stetig erhöht wird. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zur Dokumentation der Sicherheitsarchitektur für jedes der vier Verifikationslevel.

Tabelle 1 -Verifikationsanforderungen zur Dokumentation und Sicherheitsarchitektur (V1)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V1.1	Verifiziere, dass alle Komponenten (Sourcedateien oder -Gruppen, Bibliotheken und/oder ausführbare Dateien), die einen Bestandteil der Anwendung darstellen, identifiziert wurden.	√	√	✓	✓	✓	✓
V1.2	Verifiziere, dass alle Komponenten welche keinen Bestandteil der Anwendung darstellen, aber für deren Funktionieren benötigt werden, identifiziert wurden.			✓	✓	√	✓
V1.3	Verifiziere, dass eine konzeptionelle Architektursicht erstellt wurde. 16			✓	✓	✓	✓
V1.4	Verifiziere, dass die Anwendungskomponenten im Hinblick auf die von ihnen bereitgestellten Business- und Security-Funktionen identifiziert wurden.					✓	✓
V1.5	Verifiziere, dass für alle Komponenten welche keinen Bestandteil der Anwendung darstellen aber für deren Ausführung erforderlich sind, die Business- und Sicherheits-Funktionen identifiziert wurden.					✓	✓
V1.6	Verifiziere, dass ein Bedrohungsmodell erstellt wurde.					✓	✓

¹⁶ Der Prüfer kann eine High-Level-Sicht des Designs erstellen bzw. dokumentieren, sollte der Anwendungsentwickler keine solche Darstellung bereitstellen.



V2 - Verifikationsanforderungen zur Authentisierung

Die Verifikationsanforderungen zur Authentisierung dienen der Sicherstellung, dass Zugangsdaten auf sichere Weise erstellt und verwendet werden. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 2 - Verifikationsanforderungen zur Authentisierung (V2)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V2.1	Verifiziere, dass alle nicht- öffentlichen Seiten und Ressourcen eine Authentisierung erforderlich machen.	✓	✓	✓	✓	✓	✓
V2.2	Verifiziere, dass alle Passwortfelder nicht das vom Anwender eingegebene Passwort anzeigen und dass alle Passwortfelder (sowie Formulare, welche diese enthalten) die Autocomplete-Funktion deaktiviert haben.	✓	✓	✓	✓	✓	✓
V2.3	Verifiziere, dass wenn eine maximale Anzahl an Authentisierungsversuchen überschritten wurde, das Benutzerkonto für eine ausreichende Zeit gesperrt wird, so dass Bruteforce-Angriffe verhindert werden.	✓		✓	✓	✓	✓
V2.4	Verifiziere, dass alle Authentisierungskontrollen serverseitig umgesetzt sind.			✓	✓	✓	✓
V2.5	Verifiziere, dass alle Authentisierungskontrollen (inklusive Bibliotheken, welche externe Authentisierungsdienste aufrufen) zentral implementiert wurden.				√	√	✓
V2.6	Verifiziere, dass alle Authentisierungskontrollen bei einem Fehlerfall in einen sichern Zustand fallen.			✓	✓	✓	✓
V2.7	Verifiziere, dass die Stärke der Authentisierungsdaten ausreichend ist um typischen Angriffen in der jeweiligen Umgebung wiederstehen zu können.			✓	√	√	√



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V2.8	Verifiziere, dass alle für die Nutzerverwaltung eingesetzten Sicherheitsmechanismen mindestens den gleichen Grad an Resistenz gegen Angriffe bieten wie die zur primären Authentisierung verwendeten.			✓	✓	√	✓
V2.9	Verifiziere, dass Anwender nur auf sichere Art ihre Zugangsdaten ändern können und hierzu ein Mechanismus verwendet wird, der mindestens über den gleichen Schutzlevel verfügt, wie bei der primäre Authentisierung.			✓	✓	√	✓
V2.10	Verifiziere, dass eine erneute Authentisierung (Re-Authentication) erfolgt, bevor der Aufruf einer sensiblen Operation zugelassen wird.			✓	✓	✓	✓
V2.11	Verifiziere, dass Zugangsdaten nach einer administrativ konfigurierbaren Zeit ungültig werden.			✓	✓	✓	✓
V2.12	Verifiziere, dass alle Entscheidungen in Bezug auf die Authentisierung protokolliert werden.				✓	✓	✓
V2.13	Verifiziere, dass Passwörter von Benutzern nur als Hash und mit einem spezifischen Salt (z.B. interne Account-ID oder dessen Erstellungsdatum) gespeichert werden.				√	✓	✓
V2.14	Verifiziere, dass alle Zugangsdaten für den Zugriff auf bzgl. der Anwendungs-externe Dienste, nur verschlüsselt und an einem geschützten Ort gespeichert werden (z.B. nicht im Sourcecode).				✓	✓	✓
V2.15	Verifiziere, dass sämtlicher Code nur solche Mechanismen für die Authentisierung verwendet, die frei von jeder Form von Schadcode sind.						✓

V3 - Verifikationsanforderungen zum Session Management

Die Verifikationsanforderungen zum Session Management dienen zur Sicherstellung, dass HTTP-Anfragen, -Antworten, -Sessions, -Cookies,- Header sowie -Logging so verwendet werden, dass ein



sicheres Session Management gewährleistet ist. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 3 -Verifikationsanforderungen zum Session Management (V3)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V3.1	Verifiziere, dass die Anwendung für ihr Session Management die Standardimplementierung des jeweiligen Frameworks verwendet.	✓		✓	✓	✓	✓
V3.2	Verifiziere, dass Sessions invalidiert werden, wenn sich der Benutzer abmeldet.	✓		✓	✓	✓	✓
V3.3	Verifiziere, dass Sessions nach einer bestimmten Zeit der Inaktivität invalidiert werden (einfaches Session Timeout).	✓		✓	✓	✓	✓
V3.4	Verifiziere, dass Sessions nach einer administrativ konfigurierbaren Zeit der Aktivität invalidiert werden (absolutes Session Timeout).					✓	✓
V3.5	Verifiziere, dass alle Seiten, deren Zugriff eine Authentisierung erfordert, über einen Logout-Link verfügen.	✓		✓	✓	✓	✓
V3.6	Verifiziere, dass die Session ID nirgendwo anders als in Cookies preisgegeben wird; insbesondere nicht in URLs, Fehlermeldungen oder Logdateien. Dies beinhaltet sicherzustellen, dass die Anwendung kein URL Rewriting von Session Cookies durchführt.		→		✓	→	√
V3.7	Verifiziere, dass die Session ID nach erfolgtem Login erneuert wird.			✓	✓	✓	✓
V3.8	Verifiziere, dass die Session ID nach erneutem Login erneuert wird.			✓	✓	✓	✓
V3.9	Verifiziere, dass die Session ID nach Logout geändert oder gelöscht wird.			✓	✓	✓	✓
V3.10	Verifiziere, dass nur Session IDs, die vom Application Framework erstellt wurden, von der Anwendung akzeptiert werden.			✓		✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V3.11	Verifiziere, dass im angemeldeten Bereich verwendete Session Tokens über eine ausreichende Länge und Zufälligkeit verfügen um typischen Angriffen für die jeweilige Umgebung fzu widerstehen, in denen die Anwendung installiert wurde.					✓	✓
V3.12	Verifiziere, dass im angemeldeten Bereich verwendete Session Tokens/IDs ein entsprechend restriktiv gesetztes Domain- und Pfad-Attribut verwenden.					√	✓
V3.13	Verifiziere, dass sämtlicher Code, welcher Funktion des Session Managements implementiert oder einbindet, frei von Schadcode ist.						✓

V4 - Verifikationsanforderungen an Zugriffskontrollen

Die Verifikationsanforderungen an Zugriffskontrollen (Access Controls) legen fest, wie sich in einer Anwendung auf sichere Weise ein Zugriffsschutz implementieren lässt. In den meisten Anwendungen müssen Zugriffskontrollen an mehreren Stellen der unterschiedlichen Anwendungsschichten implementiert werden. Diese Verifikationsanforderungen definieren Vorgaben an Zugriffskontrollen für URLs, Geschäftsfunktionen, Daten, Services und Dateien. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.



Tabelle 4 -Verifikationsanforderungen zu Zugriffskontrollen (V4)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V4.1	Verifiziere, dass Anwender nur dann auf geschützte Funktionen zugreifen können, wenn dieser Zugriff zuvor erfolgreich autorisiert wurde.	√	✓	✓	✓	✓	✓
V4.2	Verifiziere, dass Anwender nur dann auf geschützte URLs zugreifen können, wenn dieser Zugriff zuvor erfolgreich autorisiert wurde.	✓		✓	✓	✓	✓
V4.3	Verifiziere, dass Anwender nur dann auf geschützte Dateien zugreifen können, wenn dieser Zugriff zuvor erfolgreich autorisiert wurde.	√		✓	✓	√	✓
V4.4	Verifiziere, dass direkte Objektreferenzen so geschützt sind, dass Anwenden nur der Zugriff auf berechtigte Objekte möglich ist.	✓		✓	✓	✓	✓
V4.5	Verifiziere, dass Directory Browsing abgeschaltet ist sofern dies nicht ausdrücklich gewünscht sein sollte.	✓		✓		✓	✓
V4.6	Verifiziere, dass Anwender nur auf solche Dienste zugreifen können, wenn dieser Zugriff zuvor erfolgreich autorisiert wurde.			✓	✓	✓	✓
V4.7	Verifiziere, dass Anwender nur auf solche Daten zugreifen können, wenn dieser Zugriff zuvor erfolgreich autorisiert wurde.			✓	✓	✓	✓
V4.8	Verifiziere, dass Zugriffskonten im Fehlerfall in einen sicheren Zustand fallen.			✓	✓	✓	✓
V4.9	Verifiziere, dass die Zugriffsregeln aus dem Presentation Layer auch serverseitig umgesetzt werden.			✓	✓	✓	✓
V4.10	Verifiziere, dass die Änderung von Daten von Benutzern und Richtlinien nur angemeldeten Endanwendern mit entsprechender Autorisierung möglich ist			√	✓	✓	√
V4.11	Verifiziere, dass Zugriffskontrollen serverseitig umgesetzt sind.			✓	✓	✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V4.12	Verifiziere, dass für alle Arten von Zugriffen auf geschützte Ressourcen ein zentraler Mechanismus existiert (inklusive Bibliotheken, welche externe Autorisierungsdienste aufrufen).				√	- ✓	- ✓
V4.13	Verifiziere, dass Limitierungen von Eingaben, die durch die Geschäftslogik der Anwendung bedingt sind (z.B. tägliche Transaktionslimits oder Reihenfolgen von Tasks) nicht umgangen werden können.			✓	✓	✓	✓
V4.14	Verifiziere, dass das Loggen sämtlicher Entscheidungen von Zugriffskontrollen möglich ist und alle fehlerhaften Entscheidungen stets protokolliert werden.				✓	✓	✓
V4.15	Verifiziere, dass sämtlicher Code, welche Zugriffskontrollen implementiert oder nutzt, frei von Schadcode ist.						✓

V5 - Verifikationsanforderungen zur Eingabevalidierung

Die Verifikationsanforderungen an Eingabevalidierung dienen zur Sicherstellung, dass Eingaben so validiert werden, dass diese innerhalb der Anwendung gefahrlos verwendet werden können. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 5 - Verifikationsanforderungen zur Eingabevalidierung (V5)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V5.1	Verifiziere, dass in der Laufzeitumgebung keine Pufferüberlaufe existieren oder dass dort ein entsprechender Schutz gegen diese existiert.	✓	✓	✓	✓	✓	✓
V5.2	Verifiziere, dass alle Eingaben gegen ein positives Sicherheitsmodell geprüft werden.	✓	√	✓	✓	✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V5.3	Verifiziere, dass alle Fehler während der Eingabevalidierung zu einer Zurückweisung bzw. Bereinigung (Sanitization) der entsprechenden Eingaben führt.	✓		✓	✓	✓	✓
V5.4	Verifiziere, dass ein bestimmter Zeichensatz (z.B. UTF-8) für alle Eingabequellen definiert ist.			✓	✓	✓	✓
V5.5	Verifiziere, dass sämtliche Eingabevalidierung serverseitig durchgeführt wird.			✓	✓	✓	✓
V5.6	Verifiziere, dass die Anwendung für jeden akzeptierten Datentyp einen zentralen Mechanismus zur Eingabevalidierung verwendet.				✓	✓	✓
V5.7	Verifiziere, dass Fehler bei der Eingabevalidierung protokolliert werden.				✓	✓	✓
V5.8	Verifiziere, dass alle Eingabedaten vor jeglicher Validierung hinsichtlich aller bekannter Downstream Decoder und Interpreter kanonisiert werden.					✓	✓
V5.9	Verifiziere, dass sämtlicher Code, welche Eingabevalidierung implementiert oder nutzt, frei von Schadcode ist.						✓

V6 - Verifikationsanforderungen zur Ausgabeenkodierung /-Escaping

Mit den Verifikationsanforderungen zur Ausgabeenkodierung und -Escaping lässt sich prüfen, ob Ausgaben auf für externe Anwendungen sichere Weise enkodiert worden. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 6 -Verifikationsanforderungen zur Ausgabeenkodierung /-Escaping (V6)

Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
--------------------------	----------	----------	----------	----------	---------	---------



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V6.1	Verifiziere, dass nicht vertrauenswürdige Daten, welche in HTML ausgegeben werden (inklusive HTML-Elemente, Attribute, Javascript-Datenobjekte, CSS-Blöcke und URI-Attribute) für den jeweiligen Kontext entsprechend escaped wurden.		√	√	√	√	✓
V6.2	Verifiziere, dass sämtliche Ausgabeenkodierung /-Escaping serverseitig erfolgt.			✓	✓	✓	✓
V6.3	Verifiziere, dass sämtliche Methoden zur Ausgabeenkodierung alle Zeichen enkodieren, die für den jeweiligen Interpreter als unsicher bekannt sind.				✓	✓	✓
V6.4	Verifiziere, dass die Verwendung von nicht vertrauenswürdigen Daten bei SQL-Interpretern nur mittels parametrisierter Interfaces, Prepared Statements oder zumindest korrektem Escaping erfolgt.				√	√	√
V6.5	Verifiziere, dass die Verwendung von vertrauenswürdigen Daten bei XML- Interpretern nur mittels parametrisierter Interfaces oder korrektem Escaping erfolgt.				√	√	✓
V6.6	Verifiziere, dass die Verwendung von nicht vertrauenswürdigen Daten bei LDAP-Interpretern nur mittels korrektem Escaping erfolgt.				✓	✓	✓
V6.7	Verifiziere, dass die Verwendung von vertrauenswürdigen Daten bei OS- Command-Interpretern nur mittels korrektem Escaping erfolgt.				✓	✓	✓
V6.8	Verifiziere, dass die Verwendung von nicht vertrauenswürdigen Daten bei Interpretern, die oben nicht genannt wurden, nur mittels korrektem Escaping erfolgt.				✓	✓	✓
V6.9	Verifiziere, dass für jede Art von Ausgabeenkodierung /-Escaping zu einem bestimmten Kontext ein dedizierter Sicherheitsmechanismus existiert.					✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V6.10	Verifiziere, dass sämtlicher Code, welcher Methoden für Ausgabevalidierung implementiert oder nutzt, frei von Schadcode ist.						✓

V7 - Verifikationsanforderungen zur Kryptographie

Mit den Verifikationsanforderungen zur Kryptographie lassen sich die Mechanismen für Verschlüsselung, Schlüsselmanagement, Zufallszahlengenerierung sowie Prüfsummenoperationen prüfen. Anwendungen sollten stets nur Kryptomodule einsetzen, die nach FIPS 140-2 (oder eines vergleichbaren anderen Standards) validiert wurden. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 7 - Verifikationsanforderungen zur Kryptographie (V7)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V7.1	Verifiziere, dass kryptographische Funktionen, die von der Anwendung für den Schutz sensible Informationen eingesetzt werden, serverseitig implementiert sind.			✓	✓	✓	✓
V7.2	Verifiziere, dass alle kryptographischen Module im Fehlerfall in einen sicheren Zustand fallen.			✓	✓	✓	✓
V7.3	Verifiziere, dass alle Arten von Master Secrets gegen nicht-autorisierten Zugriff geschützt sind. Master Secrets sind Zugangsdaten, welche im Klartext lokal gespeichert sind und für den Zugriff auf sicherheitsrelevante Konfigurationsdaten verwendet werden.				✓	✓	✓
V7.4	Verifiziere, dass Passwort-Hashes nur mit einem Salt erstellt ("gesaltet") wurden.				✓	✓	✓
V7.5	Verifiziere, dass Fehler von kryptographischen Modulen protokolliert werden.				✓	✓	✓
V7.6	Verifiziere, dass Zufälligkeiten für Zahlen, Dateinahmen, GUIDs und Dateinamen mittels eines zugelassenen Zufallsgenerators des kryptographischen Moduls erstellt wurden, wenn diese für Angreifer nicht ratbar sein sollen.				✓	✓	√
V7.7	Verifiziere, dass von der Anwendung eingesetzte kryptographische Module gemäß FIPS 140-2 oder eines entsprechenden andere Standards validiert worden sind (siehe http://csrc.nist.gov/groups/STM/cmvp/validati on.html).					✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V7.8	Verifiziere, dass kryptographische Module in ihrem Approved Mode gemäß ihrer Sicherheitsrichtlinie arbeiten (siehe http://csrc.nist.gov/groups/STM/cmvp/validati on.html).					✓	√
V7.9	Verifiziere, dass eine explizite Sicherheitsrichtlinie existiert, die beschreibt, wie kryptographische Schlüssel zu verwalten sind (z.B. Erzeugung, Verteilung, Sperrung, Ablauf). Verifiziere, dass diese Richtlinie korrekt durchgesetzt wird.					✓	✓
V7.10	Verifiziere, dass sämtlicher Code, welcher ein kryptographisches Modul implementiert oder nutzt, frei von Schadcode ist.						✓

V8 - Verifikationsanforderungen zur Fehlerbehandlung und Logging

Die Verifikationsanforderungen zur Fehlerbehandlung und Logging dienen zur Verifikation, dass sicherheitsrelevante Ereignisse erkannt und Angriffe identifiziert werden können. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 8 -Verifikationsanforderungen zur Fehlerbehandlung und Logging (V8)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V8.1	Verifiziere, dass die Anwendung keine Fehlermeldungen oder Stack Traces ausgibt, welche sensible Daten enthalten (inklusive Session IDs und personenbezogene Informationen).	√	√	✓	√	√	✓
V8.2	Verifiziere, dass alle serverseitigen Fehler auch serverseitig behandelt werden.			✓	✓	✓	✓
V8.3	Verifiziere, dass alle Logging- Mechanismen serverseitig implementiert sind.			✓	✓	✓	✓
V8.4	Verifiziere, dass Sicherheitsmechanismen im Fehlerfall den Zugriff verweigern.			✓	✓	✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V8.5	Verifiziere, dass für die zum Logging eingesetzten Sicherheitsmechanismen sowohl die Möglichkeit bieten, um fehlerhafte als auch erfolgreiche sicherheitsrelevant Events loggen zu können.				✓	✓	✓
V8.6	 Verifiziere, dass jedes geloggte Ereignis folgende Informationen enthält: Einen Zeitstempel (Timestamp) von einer vertrauenswürdigen Quelle, Das Sicherheitslevel des Ereignisses, Einen Hinweis darauf ob dieses Ereignis sicherheitsrelevant ist (wenn mit andern Logdateien vermischt), Die Identität des Benutzers, der das Ereignis ausgelöst hat (sofern bekannt), Die Quell-IP-Adresse, mit der das Ereignis assoziiert ist, Ob das Ereignis erfolgreich oder fehlerhaft ist Die Beschreibung des Ereignisses. 				→	→	✓
V8.7	Verifiziere, dass Ereignisse, die nicht vertrauenswürdige Informationen betreffen, keinen Code enthalten können, der im für die Logdateianzeige vorgesehenen Programm zur Ausführung gebracht werden kann.				√	√	✓
V8.8	Verifiziere, dass Sicherheitslogdateien vor unbefugten Zugriff und Modifikation geschützt sind.				✓	✓	✓
V8.9	Verifiziere, dass die Anwendung zum Logging nur eine Implementierung verwendet.				✓	✓	✓
V8.10	Verifiziere, dass die Anwendung keine sensiblen Daten der Anwendung protokolliert. Dies beinhaltet Session ID, personenbezogene oder andere sensible Informationen.				✓	✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V8.11	Verifiziere, dass ein Tool zur Logdateianalyse vorhanden ist, welches es erlaubt in den Logdateien nach Logevents auf Basis von Kombinationen von Suchkriterien aller Felder im verwendeten Logformat zu suchen.				√	√	✓
V8.12	Verifiziere, dass sämtlicher Code, welcher Controls zur Fehlerbehandlung oder Logging implementiert oder nutzt, frei von Schadcode ist.						✓

V9 - Verifikationsanforderungen zur Datensicherheit

Mit den Verifikationsanforderungen zur Datensicherheit lässt sich prüfen, dass sensible Daten (z.B. Kreditkartendaten, Pass- oder Ausweisnummern) in ausreichendem Maße geschützt werden. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 9 - Verifikationsanforderungen zur Datensicherheit (V9)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V9.1	Verifiziere, dass für alle Formulare, welche sensible Daten enthalten, das browserseitiges Caching deaktiviert ist (inklusive AutoComplete-Funktionen).	✓	✓	✓	✓	✓	✓
V9.2	Verifiziere, dass eine Liste der sensiblen Daten existiert, die von der Anwendung verarbeitet werden. Stelle sicher dass eine explizite Richtlinie existiert, wie Zugriffe auf diese Daten kontrolliert werden und wann diese zu verschlüsseln sind (sowohl für Speicherung als auch Übertragung). Verifiziere, dass diese Richtlinie korrekt umgesetzt ist.				√	√	✓
V9.3	Verifiziere, dass sensible Daten ausschließlich im HTTP-Body an den Sever gesendet werden können, also z.B. niemals als URL-Parameter.			✓		✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V9.4	Verifiziere, dass alle zwischengespeicherten oder temporären Kopien von sensible Daten, welche zum Client gesendet werden, vor unbefugtem Zugriff geschützt sind bzw. invalidiert oder zerstört werden, wenn dies der Fall ist (dies beinhaltet auch die Verwendung der korrekten Cache-Control-Header "no-cache" sowie "no-store").				√	√	√
V9.5	Verifiziere, dass alle zwischengespeicherten Daten oder temporären Kopien sensibler Daten vor unbefugtem Zugriff geschützt sind oder dass diese invalidiert bzw. gelöscht werden.				√	√	✓
V9.6	Verifiziere, dass ein Verfahren für die Löschung von sensiblen Daten existiert, wenn deren Aufbewahrungsfrist abgelaufen ist.					✓	✓

V10 - Verifikationsanforderungen zur Komunikationsicherheit

Die Verifikationsanforderungen zur Kommunikationssicherheit dienen zur Prüfung, dass sämtliche Kommunikation mit einer Anwendung angemessen gesichert erfolgt. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

Tabelle 10 -Verifikationsanforderungen zur Kommunikationssicherheit (V10)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V10.1	Verifiziere, dass ein Pfad zwischen einer vertrauenswürdigen CA und jedem TLS-Server-Zertifikat hergestellt werden kann und jedes Server-Zertifikat valide ist.	√		√	√	√	√
V10.2	Verifiziere, dass fehlgeschlagene TLS-Verbindungen nicht auf eine unsichere Verbindung zurückgesetzt werden.			✓		✓	✓



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V10.3	Verifiziere, dass TLS für alle solche Verbindungen verwendet wird (für sowohl externe wie auch Backend- Verbindungen), die entweder authentisiert sind oder sensible Daten bzw. Funktionen betreffen.				√	√	√
V10.4	Verifiziere, dass Fehler von TLS- Verbindungen zum Backend geloggt werden.				✓	✓	✓
V10.5	Verifiziere, dass Zertifikatpfade für alle Clientzertifikate mittels Vertrauensanker (Trust Anchor) und Sperrinformationen (Revocation Information) erstellt und verifiziert werden.				✓	√	√
V10.6	Verifiziere, dass alle Verbindungen zu externen Systemen authentisiert sind, wenn diese sensible Informationen oder Funktionen betreffen.				√	√	√
V10.7	Verifiziere, dass alle Verbindungen zu externen Systemen, welche sensible Informationen oder Funktionen betreffen, ein Nutzerkonto verwenden, welches nur über minimale Privilegien verfügt um korrekt zu funktionieren.				✓	✓	✓
V10.8	Verifiziere, dass nur eine einzige standardisierte TLS-Implementierung verwendet wird, welche so konfiguriert ist, dass diese in einem Approved Mode arbeitet (siehe http://csrc.nist.gov/groups/STM/cm vp/documents/fips140-2/FIPS1402IG.pdf).					√	√
V10.9	Verifiziere, dass ein spezifischer Zeichensatz für alle Verbindungen festgelegt ist (z.B. UTF-8).					✓	✓

V11 - Verifikationsanforderungen zur HTTP-Sicherheit

Die Verifikationsanforderungen zur HTTP-Sicherheit ermöglichen eine Prüfung der Sicherheit von HTTP-Requests, -Responses, -Sessions, -Cookies, -Header und -Logging. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.



Tabelle 11 - Anforderungen zur HTTP-Sicherheit (V11)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V11.1	Verifiziere, dass Weiterleitungen keine nicht validierten Daten enthalten.	✓	✓	✓	✓	✓	✓
V11.2	Verifiziere, dass die Anwendung nur eine definierte Menge von http- Methoden erlaubt, wie etwa GET und POST.	✓	√	✓	✓	✓	✓
V11.3	Verifiziere, dass jede HTTP-Antwort einen Content Type enthält, der einen sicheren Zeichensatz festlegt (z.B. UTF-8).	✓	✓	✓	√	✓	✓
V11.4	Verifiziere, dass das HTTPOnly-Flag für alle Cookies gesetzt wird, auf die nicht notwendigerweise mittels Javascript zugegriffen werden muss.			✓	✓	✓	✓
V11.5	Verifiziere, dass das Secure-Flags bei allen Cookies gesetzt wird, die sensible Informationen enthalten (inklusive Session Cookies).			✓	✓	✓	✓
V11.6	Verifiziere, dass HTTP-Header, sowohl in HTTP Requests als auch HTTP Responses, nur druckbare ASCII-Zeichen enthalten.			✓	✓	✓	✓
V11.7	Verifiziere, dass die Anwendung ein kryptographische starkes Zufallstoken generiert, welches Teil sämtlicher Links und Formulare ist, die Transaktionen oder Zugriffe auf sensible Daten bereitstellen. Zudem hat die Anwendung zu prüfen, ob das korrekte Token des jeweiligen Benutzers vorhanden ist, bevor sie eine Anfrage ausführt. ¹⁷					√	✓

V12 - Verifikationsanforderungen zur Sicherheitskonfiguration

Die Verifikationsanforderungen zur Sicherheitskonfiguration ermöglichen die Prüfung der sicheren Speicherung aller sicherheitsrelevanten Konfigurationsinformationen der Anwendung. Der Schutz der Konfigurationsinformationen ist für den sicheren Betrieb der Anwendung kritisch. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.

¹⁷ Diese Anforderung beschreibt einen Mechanismus der für die Abwehr von Cross Site Request Forgery (CSRF)-Angriffen dient.



Tabelle 12 - Verifikationsanforderungen zur Sicherheitskonfiguration (V12)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V12.1	Verifiziere, dass alle sicherheitsrelevanten Konfigurationsinformationen nur an Orten gespeichert werden, die vor unbefugten Zugriff geschützt sind.				✓	✓	✓
V12.2	Verifiziere, dass alle Zugriffe auf die Anwendung abgelehnt werden, wenn diese keinen Zugriff zu ihrer Sicherheitskonfiguration herstellen kann.				√	✓	✓
V12.3	Verifiziere, dass alle Änderungen der Sicherheitskonfiguration durch die Anwendung verwaltet und im Sicherheitsereignisprotokoll protokolliert werden.					✓	✓
V12.4	Verifiziere, dass die Konfiguration in einem für Menschen lesbaren und für ein Audit verwendbaren Format ausgegeben werden kann.						✓

V13 - Verifikationsanforderungen zur Identifikation von Schadcode

Für Level 4 muss jeglicher Code nach Schadcode geprüft werden, der nach einer Level-3-Verifikation noch nicht untersucht wurde. Die folgende Tabelle beschreibt die Anforderungen zur Schadcodesuche, die durch Level 4 eingeführt werden.

Tabelle 13 -Anforderungen zur Schadcodesuche (V13)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V13.1	Verifizieren, dass kein Schadcode in jeglichem zur Erstellung oder Modifikation der Anwendung verwendeten Code enthalten ist. ¹⁸						✓

¹⁸ Untersuche z.B. alle Aufrufe der Systemzeit auf Time Bombs; Funktionen, die keinen Bezug zu Geschäftsanforderungen haben auf Hintertüren (Backdoors); Ausführungspfade auf Easter Eggs; Finanztransaktionen auf inkorrekte Logik, die Hinweise auf einen Salami-Angriff liefern oder Formen von sonstigem Schadcode.



	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V13.2	Stelle sicher, dass die Integrität von interpretierten Code, Bibliotheken, ausführbaren Dateien und Konfigurationsdateien mittels Prüfsummen oder Hashwerten verifiziert wird.						√

V14 - Verifikationsanforderungen zur Internen Sicherheit

Die Verifikationsanforderungen zur internen Sicherheit ermöglichen die Anwendung darauf zu prüfen, dass diese einen bestimmten Grad an zusätzlichem Schutz gegen Implementierungsfehler besitzt. Die folgende Tabelle beschreibt die entsprechenden Anforderungen zu jedem der vier Verifikationslevel.



Tabelle 14 - Verifikationsanforderungen zur Internen Verifikation (V14)

	Verifikationsanforderung	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V14.1	Verifiziere, dass die Anwendung Benutzer- und Datenattribute sowie Richtlinieninformationen die von Zugriffskontrollen verwendet werden vor unbefugtem Zugriff oder Modifikation schützt.					√	✓
V14.2	Verifiziere, dass Schnittstellen von Sicherheitsmechanismen einfach genug sind, damit Entwickler diese mit hoher Wahrscheinlichkeit korrekt verwenden können.						✓
V14.3	Verifiziere, dass die Anwendung gemeinsam genutzte Variablen (Shared Variables) vor ungewünschten gleichzeitigen Zugriffen schützt.						✓



Anforderungen an den Verifikationsbericht

Ein OWASP-ASVS-Bericht enthält neben einer Beschreibung der verifizierten Anwendung, die Ergebnisse der Analyse, inklusive jede erforderliche Korrekturmaßnahme beschriebener Schwachstellen.

Art und Umfang der enthaltenen Informationen sind durch die geprüften Anforderungen festgelegt. Es existieren jedoch keine Vorgaben hinsichtlich der Struktur, Organisation oder des Formats des Berichtes. Zusätzliche Informationen sind ebenfalls nicht davon ausgeschlossen in den Bericht mit einzufließen.

Alle Arten von Informationen, die in den Anforderungen zum Verifikationsbericht genannt werden, sind zu benennen, formatieren und, entsprechend der Verifikationsanforderungen, zu strukturieren. Die ASVS-Berichtsanforderungen sind erfüllt, sobald alle geforderten Informationen im Bericht enthalten sind. Gleiches gilt in Bezug auf zusätzliche Informationen, die für den Leser erforderlich sind, um die durchgeführte Analyse und deren Ergebnisse nachvollziehen zu können. Dies schließt Informationen zur Konfiguration und Code Snippets mit ein. Die oben gezeigte Abbildung kann als Grundlage für die Gliederung des Berichtes verwendet dienen.

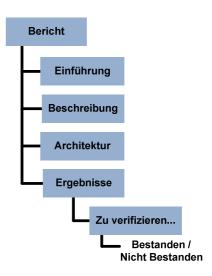


Abbildung 11 - Berichtsanforderungen

R1 - Berichtseinführung

- R1.1 Der Berichtseinführung hat alle erforderlichen Informationen zu enthalten, so dass sowohl der Bericht selbst, als auch die vom Bericht untersuchte Anwendung, verstanden wird.
- R1.2 Die Berichtseinführung hat eine Gesamtbewertung hinsichtlich des Vertrauens in die Sicherheit der Anwendung darzustellen.
- R1.3 Die Berichtseinführung hat die zentralen Geschäftsrisiken (Business Risks) darzustellen, die mit dem Betrieb der Anwendung verbunden sind.
- R1.4 Die Berichtseinführung hat die Rahmenbedingungen darzustellen, die mit der Durchführung der Analyse verbunden oder die deren Umfang beeinträchtigt haben.

R2 - Anwendungsbeschreibung

R2.1 Die Anwendungsbeschreibung muss alle erforderlichen Informationen enthalten um ihre Arbeitsweise sowie die Umgebung in der sie arbeitet darzustellen.

R3 - Sicherheitsarchitektur der Anwendung

R3.1 Die Sicherheitsarchitektur der Anwendung hat zusätzliche Details zu enthalten, so dass der Leser die Korrektheit und Vollständigkeit der Analyse erkennen kann. Dieser Teil des Berichts enthält Kontextinformationen zur Analyse. Die Informationen, die in diesem Abschnitt dargestellt sind



werden im Rahmen der Analyse zur Identifikation von Inkonsistenzen verwendet. Je nach angewendeten ASVS-Level sind in diesem Teil des Berichtes unterschiedliche Detailierungsgrade anzuwenden.

R4 - Verifikationsergebnisse

R4.1 Die Verifikationsergebnisse sind gemäß des Abschnitts "Verifikationsanforderungen" dieses Standards darzustellen. Dies beinhaltet die folgende Beschreibung sämtlicher erforderlichen Korrekturmaßnahmen zu Schwachstellen:

Tabelle 15 - OWASP ASVS Inhalte der Verifikationsergebnisse im Bericht

Level	Bestanden	Nicht Bestanden				
Level 1 Ergebnisse	 Bewertung Toolkonfiguration (wenn das Tool für die Durchführung der Prüfung geeignet ist) oder eine Begründung zu der getroffenen Beurteilung (Vollständigkeit und Korrektheit muss nachvollziehbar sein). Abbildung der Einsatzmöglichkeiten des automatischen Tools hinsichtlich der Anforderungen zur detaillierten Verifikation. Beschreibung der Konfiguration und die Abbildung der Einsatzmöglichkeiten des Tools muss nur an einer Stelle des Berichtes enthalten sein. 	 Bewertung Position (URL, Parameter und oder Pfad, Name und Zeilennummer(n) der Sourcedatei) eines Mangels Beschreibung (inklusive relevante Konfigurationsdaten) Risikobewertung¹⁹ Begründung zur Risikobewertung 				
	Eine Beschreibung der Toolkonfiguration und Abbildung der Einsatzmöglichkeiten des Tools ist ebenfalls in diesem Teil des Berichts darzustellen.					

¹⁹ Für mehr Informationen zur Identifikation von Risiken und die Bewertung von mit Schwachstellen verbundenen Risiken, siehe das *Testing Guide* (OWASP, 2008).



Level	Bestanden	Nicht Bestanden
Levels 2 - 4 Ergebnisse	 Bewertung Begründung zu der getroffenen Bewertung (Vollständigkeit und Korrektheit muss nachvollziehbar sein). 	 Bewertung Position (URL w/Parameter und oder Pfad, Name und Zeilennummer(n) der Sourcedatei) eines Mangels Bewertung (inklusive Pfad durch die Anwendungskomponenten und Schritte zur Reproduzierung) Risikobewertung (siehe hierzu die OWASP Risk Rating Methodology)
		Begründung zur Risikobewertung



Glossar

Zugriffskontrolle ("Access Controls") - Eine Maßnahme um Zugriff auf Dateien, referenzierte Funktionen, URLs und Daten auf Basis der Identität von Benutzern und/oder Gruppen einzuschränken.

Applikationskomponente - Eine dedizierte Gruppe von Sourcecodedateien, Bibliotheken und/oder ausführbaren Dateien, die vom Prüfer für eine bestimmte Anwendung.

Application Security - Anwendungsseitige Sicherheit zielt weniger auf die Analyse des zugrundeliegenden Betriebssystems oder die Netzwerkebene als mehr auf die Analyse der Komponenten der Anwendungsschicht des OSI-Modells (Open Systems Interconnection Reference Model).

Application Security Verification - Die technische Prüfung einer Anwendung gegen den OWASP ASVS.

Application Security Verification Bericht - Ein Bericht der die gesamten Ergebnisse inklusive sämtlicher zusätzlicher Analysen, welche der Prüfer durchgeführt hat, beinhaltet.

Application Security Verification Standard (ASVS) - Ein OWASP-Standard der vier Level zur Verifikation der Anwendungssicherheit definiert.

Authentisierung -Die Verifikation, dass der Anwendungsbenutzer tatsächlich derjenige ist, der er vorgibt zu sein.

Automatische Verifikation - Die Verwendung automatisierter Tools (sowohl dynamische oder statische Analystetools), welche Schwachstellensignaturen verwenden, um Probleme zu identifizieren.

Hintertür ("Back Doors") - Eine Form von Schadcode welcher nicht autorisierten Anwendungszugriff ermöglicht.

Blacklist - Eine Liste von Daten oder Operationen welche nicht zulässig sind, z.B. eine Liste von Zeichen, die nicht in Eingaben enthalten sein können.

Common Criteria (CC) - Ein mehrteiliger Standard, der als Basis zur Verifikation von Design und Implementierung von Sicherheitsmechanismen in IT-Produkten genutzt werden kann.

Kommunikationssicherheit - Der Schutz von Anwendungsdaten während diese zwischen verschiedene Applikationskomponenten, Clients und Servern sowie zwischen externen Systemen und der Anwendung übertragen werden.

Design Verifikation - Die technische Prüfung der Sicherheitsarchitektur einer Anwendung.

Interne *Verifikation* - Die technische Prüfung spezifischer Aspekte der Sicherheitsarchitektur einer Anwendung, wie dies im OWASP ASVS definiert ist.

Kryptographische Module - Hardware, Software und/oder Firmware welche kryptographische Algorithmen implementiert und/oder kryptographische Schlüssel generiert.

Denial of Service (DOS)-Angriffe - Das Überfluten ("Flooding") der Anwendung mit mehr Anfragen als diese Verarbeiten kann.

Dynamische Verifikation - Die Verwendung von automatisierten Tools, welche Schwachstellensignaturen verwenden um während der Ausführung einer Anwendung Probleme identifizieren.

Easter Eggs - Eine Form von Schadcode, welcher erst dann ausgeführt wird, wenn eine bestimmte Benutzereingabe erfolgt.

Externes System - Eine serverseitige Anwendung oder Dienst der kein Bestandteil der Anwendung selbst ist.

FIPS 140-2 - Ein Standard, der zur Verifikation von Design und Implementierung von Kryptomodulen verwendet werden kann.



Eingabevalidierung - Die Kanonisierung und Validierung von nicht-vertrauenswerten Eingabedaten.

Schadcode (Malicious Code) - Code, der in eine Anwendung während deren Entwicklung eingebaut wurde, deren Existenz dem Anwendungseigner unbekannt ist und der die Sicherheitsrichtlinie der Anwendung versucht zu umgehen. Schadcode ist nicht dasselbe wie Schadprogramme, wie etwa Viren oder Würmern!

Schadprogramme - Ausführbare Code, der, ohne das Wissen des Benutzers oder Administrators, zur Laufzeit in eine Anwendung eingeschleust wird (Malware).

Open Web Application Security Project (OWASP) - Das Open Web Application Security Projekt (OWASP) ist eine weltweite, freie und offene Cummunity, die sich die Verbesserung der Sicherheit von Anwendungen zum Ziel setzt. Unsere Mission besteht darin, Anwendungssicherheit "sichtbar" zu machen, so dass Personen und Organisationen qualifizierte Aussagen über Sicherheitsrisiken von Anwendungen machen können. Siehe: http://www.owasp.org/

Ausgabevalidierung - Die Kanonisierung und Validierung von Anwendungsausgaben zu Webbrowsern oder externer Systeme.

OWASP Enterprise Security API (ESAPI) - Eine freie und offene Sammlung aller Sicherheitsmethoden, die Entwickler benötigen um sichere Webanwendungen zu bauen. Siehe: http://www.owasp.org/index.php/ESAPI

OWASP Risk Rating Methodology - Eine Vorgehensweise zur Risikoermittlung welche speziell auf den Bereich der Anwendungssicherheit zugeschnitten wurde. Siehe:

http://www.owasp.org/index.php/How_to_value_the_real_risk

OWASP Testing Guide - Ein Dokument welches erstellt wurde um Organisationen zu verstehen, welche Bestanteile ein Testprogram besitzt und welche Schritte benötigt werden um ein solche Testprogramm zu erstellen und zu betreiben. Siehe:

http://www.owasp.org/index.php/Category:OWASP_Testing_Project

OWASP Top Ten - Ein Dokument, welches den Konsens an kritischen Schwachstellen darstellt. Siehe: http://www.owasp.org/index.php/Top10

Positiv - Siehe Whitelist.

Prüfer - Die Person oder das Team, der bzw. die eine Anwendung gegen OWASP-ASVS-Anforderungen untersucht bzw. verifiziert.

 ${\it Salami~Angriff}$ - Eine Form von Schadcode, die unbemerkt versucht sehr geringe Geldmengen abzuzweigen.

Sicherheitsarchitektur - Die Abstraktion des Designs zur Identifikation und Beschreibung wie Sicherheitsmechanismen verwendet werden sowie Orte sensibler Benutzer- und Anwendungsdaten.

Sicherheitsmechanismus (Security Control) - Eine Funktion oder Komponente, welche Sicherheitsprüfungen durchführt (z.B. eine Zugriffsprüfung) oder beim Aufruf eine Sicherheitseffekt auslöst (z.B. die Generierung eines Audit Records).

Sicherheitskonfiguration - Die Laufzeitkonfiguration einer Anwendung, welche Einfluss auf die Funktionsweise der eingesetzten Sicherheitsmechanismen hat.

Statische Verifikation - Der Einsatz von automatisierten Tools, welche Schwachstellensignaturen einsetzen, um Probleme im Sourcecode zu identifizieren.

Target of Verification (TOV) - Eine Verifikation von Anwendungssicherheit auf Basis der Anforderungen von OWASP ASVS bezieht sich stets auf eine konkrete Anwendung. Diese Anwendung stellt den Untersuchungsgegenstand dar und wird als "Target of Verification" oder einfacher TOV bezeichnet.

Bedrohungsmodellierung (Threat Modeling) - Eine Technik basierend auf der schrittweisen Verfeinerung von Sicherheitsarchitekturen zur Identifikation von Threat Agents, Sicherheitszonen, Sicherheitsmechanismen und wichtigen technischen oder geschäftsbezogenen Assets.



 $\it Time\ Bomb$ - Eine Form von Schadcode, welcher erst zu einer vordefinierten Zeit oder Datum ausgeführt wird.

 $\it Whitelist$ - Eine Liste von erlaubten Daten oder Operationen, z.B. eine Liste von für eine Eingabevalidierung erlaubten Zeichen.



Weiterführende Informationen

Die Webseite der OWASP ist die führende Quelle für Informationen zur Webanwendungssicherheit. Auf der Seite sind zahlreiche Projekte, Foren, Blogs, Präsentationen, Tools und Paper abrufbar. Zusätzlich richtet die OWASP jährlich zwei große Konferenzen zur Webanwendungssicherheit aus und hat mehr als 80 lokale Chapter. Die ASVS-Projektseite ist unter http://www.owasp.org/index.php/ASVS zu finden.

Von besonderem Interesse für Nutzer und Anwender des Standards können die folgenden OWASP-Projekte sein:

- OWASP Top Ten Project http://www.owasp.org/index.php/Top_10
- OWASP Code Review Guide http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
- OWASP Testing Guide http://www.owasp.org/index.php/Testing_Guide
- OWASP Enterprise Security API (ESAPI) Project http://www.owasp.org/index.php/ESAPI
- OWASP Legal Project http://www.owasp.org/index.php/Category:OWASP_Legal_Project

Ebenfalls von Interesse für Nutzer und Anwender des Standards können die folgenden Webseiten sein:

- OWASP http://www.owasp.org
- MITRE Common Weakness Enumeration Vulnerability Trends, http://cwe.mitre.org/documents/vuln-trends.html
- PCI Security Standards Council Herausgeber des PCI-Standards, relevant für alle Organisationen die Kreditkartendaten verarbeiten oder speichern, https://www.pcisecuritystandards.org
- PCI Data Security Standard (DSS) v1.1 https://www.pcisecuritystandards.org/pdfs/pci_dss_v1-1.pdf

THE BELOW ICONS REPRESENT WHAT OTHER VERSIONS ARE AVAILABLE IN PRINT FOR THIS TITLE BOOK.

ALPHA: "Alpha Quality" book content is a working draft. Content is very rough and in development until the next level of publication.

BETA: "Beta Quality" book content is the next highest level. Content is still in development until the next publishing.

RELEASE: "Release Quality" book content is the highest level of quality in a books title's lifecycle, and is a final product.







ALPHA

BETA PUBLISHED

RELEASI

YOU ARE FREE:



to share - to copy, distribute and transmit the work



to Remix - to adapt the work

UNDER THE FOLLOWING CONDITIONS:



Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Share Alike. - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.



The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on improving the security of application software. Our mission is to make application security "visible," so that people and organizations can make informed decisions about application security risks. Everyone is free to participate in OWASP and all of our materials are available under a free and open software license. The OWASP Foundation is a 501c3 not-for-profit charitable organization that ensures the ongoing availability and support for our work.

On the cover: Braconid wasps are beneficial parasites. Braconids parasitize a broad range of hosts: caterpillars, flies, wasps, beetles, and aphids. After a female injects an egg into a host, the larva feeds slowly on that single host. By the time the host dies, the larva is fully grown. It pupates inside or near the dead host, sometimes in a silken cocoon, to emerge later as an adult wasp.