

OWASP Application Security Verification Standard 2009

Web Application Standard



Wersja polska Polish Version



Creative Commons (CC) Attribution Share-Alike Free version at http://www.owasp.org



Przedmowa

Niniejszy dokument opisuje cztery poziomy weryfikacji bezpieczeństwa aplikacji Web. Bezpieczeństwo na poziomie aplikacji skupia się na analizie komponentów składających się na warstwę aplikacji modelu OSI nie skupiając się np. na systemie operacyjnym poniżej lub podłączonych sieciach. Każdy z poziomów opisanych w niniejszym dokumencie zawiera szereg wymagań dotyczących weryfikacji efektywności mechanizmów bezpieczeństwa chroniących aplikacje Web.

Wymagania te zostały stworzone z myślą o następujących celach:

- *Użycie jako miara dostarczenie* twórcom i właścicielom aplikacji miary do oceny stopnia zaufania do ich aplikacji Web,
- Użycie jako wytyczne dostarczenie twórcom mechanizmów bezpieczeństwa wytycznych, w jaki sposób budować mechanizmy bezpieczeństwa, by spełnić wymagania bezpieczeństwa aplikacji, ¹
- Użycie podczas pozyskiwania dostarczenie podstaw do określenia wymagań weryfikacji bezpieczeństwa aplikacji w umowach.

Wymagania zostały opracowane by osiągnąć powyższe cele poprzez zapewnienie kontroli, w jaki sposób mechanizmy bezpieczeństwa są projektowane, wdrażane i używane przez aplikację. Wymagania te zapewniają, że używane w aplikacji mechanizmy bezpieczeństwa funkcjonują - zgodnie ze strategią domyślnego zabronienia (deny-by-default), są scentralizowane, zlokalizowane po stronie serwera i używane wszędzie tam, gdzie są potrzebne.

Prawa Autorskie I Licencja

Copyright © 2008 - 2009 The OWASP Foundation.



Niniejszy dokument jest wydany zgodnie z licencją Creative Commons Attribution ShareAlike 3.0.W przypadku ponownego użycia lub dystrybucji niniejszego dokumentu lub jego fragmentów musisz jasno poinformować innych o warunkach licencji.

¹ Więcej informacji o tworzeniu i użyciu mechanizmów bezpieczeństwa spełniających wymagania ASVS można znaleźć w *Enterprise Security API (ESAPI)* (OWASP, 2009).

² Więcej informacji o użyciu ASVS w umowach można znaleźć w *Contract Annex* (OWASP, 2009).



Spis treści

Wstęp	1
Podejście	1
Podziękowania	4
Poziomy weryfikacji bezpieczeństwa aplikacji	5
Poziom 1 - Weryfikacja automatyczna	5
Poziom 1A - Skanowanie dynamiczne (częściowa weryfikacja automatyczna)	7
Poziom 1B - Skanowanie kodu źródłowego (częściowa weryfikacja automatyczna)	8
Poziom 2 - Weryfikacja ręczna	8
Poziom 2A - Test bezpieczeństwa (częściowa weryfikacja ręczna)	11
Poziom 2B - Przegląd kodu (częściowa weryfikacja ręczna)	11
Poziom 3 - Weryfikacja projektu	
Poziom 4 - Weryfikacja wewnętrzna	15
Interpretacje wymagań i precedensy	
Szczegółowe wymagania weryfikacyjne	18
V1 - Wymagania dotyczące dokumentacji architektury bezpieczeństwa	
V2 - Wymagania weryfikacyjne dotyczące uwierzytelniania	
V3 - Wymagania weryfikacyjne dotyczące zarządzania sesją	
V4 - Wymagania weryfikacyjne dotyczące kontroli dostępu	
V5 - Wymagania weryfikacyjne dotyczące walidacji wejścia	
V6 - Wymagania weryfikacyjne dotyczące enkodowania/escapowania wyjścia	
V7 - Wymagania weryfikacyjne dotyczące kryptografii	
V8 - Wymagania weryfikacyjne dotyczące obsługi błędów i logowania	
V9 - Wymagania weryfikacyjne dotyczące ochrony danych	32
V10 - Wymagania weryfikacyjne dotyczące bezpieczeństwa komunikacji	
V11 - Wymagania weryfikacyjne dotyczące bezpieczeństwa HTTP	
V12 - Wymagania weryfikacyjne dotyczące konfiguracji zabezpieczeń	
V13 - Wymagania weryfikacyjne dotyczące wyszukiwania złośliwego kodu	
V14 - Wymagania weryfikacyjne dotyczące bezpieczeństwa wewnętrznego	37
Wymagania dotyczące raportowania weryfikacji	38
R1 - Wstęp do raportu	38
R2 - Opis apiikacji	
R3 - Architektura bezpieczeństwa aplikacji	
R4 - Wyniki weryfikacji	
Słowniczek	
Gdzie ieszcze możesz sie udać?	42



Rysunki	
Rysunek 1 - Poziomy OWASP ASVS	2
Rysunek 2 - Jeden ze sposobów wprowadzenia weryfikacji, jako czynności w Twoim SDLC	
Rysunek 3 - Poziomy 1, 1A i 1B OWASP ASVS	6
Rysunek 4 - Przykład architektury bezpieczeństwa dla poziomu 1 OWASP ASVS	7
Rysunek 5 - Poziomy 2, 2A i 2B OWASP ASVS	9
Rysunek 6 - Przykład architektury bezpieczeństwa dla poziomu 2 OWASP ASVS	11
Rysunek 7 - Poziom 3 OWASP ASVS	13

Tabele

Tabela 1 - Wymagania dotyczące dokumentacji architektury bezpieczeństwa w OWASP ASVS (V1))19
Tabela 2 - Wymagania dotyczące uwierzytelniania w OWASP ASVS (V2)	21
Tabela 3 - Wymagania dotyczące zarządzania sesją w OWASP ASVS (V3)	22
Tabela 4 - Wymagania dotyczące kontroli dostępu w OWASP ASVS (V4)	25
Tabela 5 - Wymagania dotyczące walidacji wejścia w OWASP ASVS (V5)	26
Tabela 6 - Wymagania dotyczące enkodowania/escapowania wyjścia w OWASP ASVS (V6)	27
Tabela 7 - Wymagania dotyczące kryptografii w OWASP ASVS (V7)	29
Tabela 8 - Wymagania dotyczące obsługi błędów i logowania w OWASP ASVS (V8)	30
Tabela 9 - Wymagania dotyczące ochrony danych w OWASP ASVS (V9)	32
Tabela 10 - Wymagania dotyczące bezpieczeństwa komunikacji w OWASP ASVS (V10)	33
Tabela 11 - Wymagania dotyczące bezpieczeństwa HTTP w OWASP ASVS (V11)	34
Tabela 12 - Wymagania dotyczące konfiguracji zabezpieczeń w OWASP ASVS (V12)	35
Tabela 13 - Wymagania dotyczące wyszukiwania złośliwego kodu w OWASP ASVS (V13)	36
Tabela 14 - Wymagania dotyczące bezpieczeństwa wewnętrznego w OWASP ASVS (V14)	
Tabela 15 - Zawartość wyników weryfikacji w raporcje OWASP ASVS	39



Wstęp

Open Web Application Security Project (OWASP) jest otwartą społecznością, której celem jest umożliwienie organizacjom tworzenia, nabywania oraz utrzymywania aplikacji, którym można zaufać. Wszystkie narzędzia, dokumenty, fora i lokalne oddziały OWASP są dostępne za darmo i otwarte dla wszystkich zainteresowanych poprawą bezpieczeństwa aplikacji. Zalecamy podejście do bezpieczeństwa aplikacji, jako problemu obejmującego ludzi, procesy i technologię, ponieważ najbardziej efektywne podejścia do bezpieczeństwa aplikacji zawierają usprawnienia we wszystkich tych obszarach. Możesz nas znaleźć na www.owasp.org

OWASP jest nowym rodzajem organizacji. Nasza niezależność od nacisków komercyjnych pozwala nam dostarczać pozbawione tendencyjności, praktyczne i efektywne kosztowo informacje na temat bezpieczeństwa aplikacji. OWASP nie jest powiązany z żadną firmą technologiczną, wspiera jednak świadome użycie komercyjnych rozwiązań bezpieczeństwa. Podobnie do innych projektów opensource OWASP tworzy wiele materiałów różnego typu w drodze otwartej współpracy. OWASP Foundation jest jednostką typu non-profit, co zapewnia długoterminowy sukces projektu.

Głównym celem projektu OWASP Application Security Verification Standard (ASVS) jest normalizacja zasięgu weryfikacji bezpieczeństwa aplikacji Web w odniesieniu do pokrycia i rygorystyczności możliwej do osiągnięcia w warunkach rynkowych poprzez stworzenie otwartego standardu przystosowanego do zastosowań komercyjnych. Niniejszy standard dostarcza podstaw do testowania technicznych mechanizmów bezpieczeństwa aplikacji, jak również innych technicznych mechanizmów bezpieczeństwa środowiska, od których zależy ochrona przed podatnościami takimi jak Cross-Site Scripting (XSS) i SQL injection. ³ Standard może być używany w celu uzyskania poziomu pewności, co do bezpieczeństwa aplikacji Web.

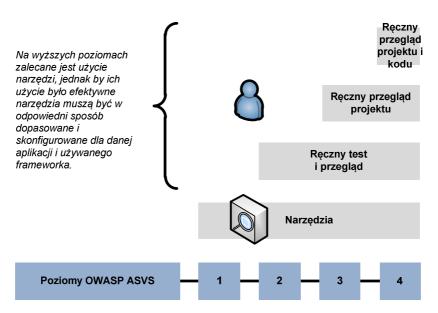
Podejście

OWASP ASVS określa wymagania względem weryfikacji i dokumentowania pogrupowane na podstawie pokrycia i rygorystyczności weryfikacji. Standard definiuje cztery poziomy ułożone w hierarchii (np. poziom 2 wymaga większego pokrycia i rygorystyczności niż poziom 1) jak pokazano na diagramie poniżej.

Dostępna jest już nowa wersja OWASP Top Ten 2010 [przyp. tłumacza]

³ Więcej informacji na temat częstych podatności w aplikacjach Web można znaleźć w *OWASP Top Ten* (OWASP, 2007).





Rysunek 1 - Poziomy OWASP ASVS

Weryfikacja bezpieczeństwa aplikacji Web z logicznego punktu widzenia jest wykonywana poprzez podążanie (lub próbę podążania) ścieżkami wejść i wyjść z danej aplikacji (nazywanej celem weryfikacji lub TOV (skrót od Target of Verification)) i przeprowadzanie analizy zgodnie z tymi ścieżkami. Aplikacje o większym stopniu złożoności zwykle wymagają większej ilości czasu na analizę, co sprawia, że weryfikacja jest dłuższa i bardziej kosztowna. Liczba linii kodu nie jest jedynym czynnikiem określającym złożoność aplikacji - różne technologie zwykle wymagają różnej wielkości analiz. Proste aplikacje mogą na przykład zawierać biblioteki i frameworki. Aplikacje o średnim stopniu złożoności mogą zawierać proste aplikacje Web 1.0. Aplikacje złożone mogą zawierać aplikacje Web 2.0 i nowe/unikalne technologie Web.

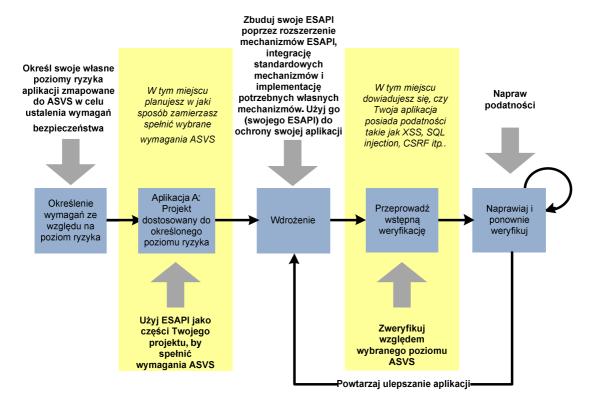
ASVS definiuje składowe komponenty dla poziomów 1 i 2 (np. weryfikacja na poziomie 1 wymaga spełnienia wymagań poziomów 1A i 1B). Dla przykładu, aplikacje mogą wymagać zgodności do poziomu 1A lub 1B zamiast poziomu 1, ale poprzez to stawiane wymagania są mniejsze od wymagań poziomu 1. Wymagania względem weryfikacji i dokumentowania zdefiniowane w tym standardzie dzielą się na trzy typy wymagań: wymagania wysoko-poziomowe, wymagania szczegółowe i wymagania raportowe. Wymagania wysoko-poziomowe określają całościowe wymagania względem implementacji i weryfikacji aplikacji. Wymagania szczegółowe określają niskopoziomowe wymagania względem implementacji i weryfikacji aplikacji (tj. konkretne elementy, które należy zweryfikować). Wymagania raportowe określają sposób dokumentacji wyników weryfikacji aplikacji przeprowadzonej zgodnie z OWASP ASVS.

OWASP dostarcza wiele materiałów, włączając w to ASVS, by pomóc organizacjom tworzyć i utrzymywać bezpieczne aplikacje. OWASP ASVS, OWASP Contract Annex⁴ oraz OWASP ESAPI⁵ mogą być używane do wsparcia cyklu życia rozwoju oprogramowania SDLC (skrót od Software Development Life Cycle) w sposób przedstawiony na rysunku poniżej.

⁴ Więcej informacji na temat wyboru poziomu ASVS przy tworzeniu umów można znaleźć w *OWASP Contract Annex*.

⁵ Więcej informacji na temat jak wdrożyć ESAPI w Twojej aplikacji można znaleźć w projekcie OWASP ESAPI (OWASP 2009).





Rysunek 2 - Jeden ze sposobów wprowadzenia weryfikacji, jako czynności w <u>Twoim</u> SDLC⁶

⁶ Więcej informacji na temat wprowadzania czynności związanych z bezpieczeństwem do Twojego istniejącego SDLC można znaleźć w projektach *OWASP CLASP* (OWASP 2008) lub *OWASP SAMM* (OWASP 2009).



Podziękowania

Dziękujemy OWASP Foundation za sponsorowanie projektu OWASP ASVS w ramach OWASP Summer of Code 2008.

Lider projektu: ⁷ Mike Boberski (Booz Allen Hamilton)

Autorzy: ⁸ Mike Boberski (Booz Allen Hamilton), Jeff Williams (Aspect Security), Dave Wichers (Aspect Security)

Sponsorzy projektu:





Booz | Allen | Hamilton

Tłumaczenie: Tomasz Polański (jmsblond@gmail.com)

Przegląd i korekta: dr inż. Mariusz Stawowski

Dziękujemy również za wkład:Pierre Parrend, który był recenzentem OWASP Summer of Code 2008; 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); oraz na koniec składamy podziękowania społeczności związanej z weryfikacją bezpieczeństwa aplikacji i innym zainteresowanym zaufanym przetwarzaniem Web (trusted Web computing) za ich entuzjastyczne porady i wsparcie w trakcie realizacji tego projektu.

⁷ Email: mike.boberski@owasp.org

⁸ Email: jeff.williams@owasp.org, dave.wichers@owasp.org



Poziomy weryfikacji bezpieczeństwa aplikacji

ASVS definiuje cztery poziomy weryfikacji, gdzie przechodząc na wyższe poziomy rośnie szerokość i głębokość weryfikacji. Szerokość dla każdego poziomu jest definiowana przez zestaw wymagań bezpieczeństwa, które muszą być spełnione. Głębokość weryfikacji jest zdefiniowana przez podejście oraz wymaganą rygorystyczność weryfikacji każdego z wymagań bezpieczeństwa. Narzędzia są istotną częścią każdego poziomu ASVS. Na wyższych poziomach ASVS zalecane jest używanie narzędzi, jednak aby było to efektywne narzędzia te muszą być w odpowiedni sposób dopasowane i skonfigurowane dla danej aplikacji i używanego frameworka. Na każdym poziomie wyniki uzyskane przy pomocy narzędzi muszą być zweryfikowane ręcznie.

Weryfikujący jest odpowiedzialny za stwierdzenie czy TOV spełnia wszystkie wymagania na wskazanym poziomie przeglądu. Aplikacja spełniająca wszystkie wymagania danego poziomu jest aplikacją ASVS poziomu N, gdzie N jest poziomem, z którym dana aplikacja jest zgodna. Jeśli aplikacja nie spełnia wszystkich wymagań dla danego poziomu, lecz spełnia wszystkie wymagania dla poziomu niższego uznaje się, że przeszła weryfikację na poziomie niższym. W standardzie używany jest termin "weryfikujący" oznaczający osobę lub zespół dokonujący przeglądu aplikacji pod kątem niniejszych wymagań.

Specyfikacja aplikacji może wymagać zgodności z OWASP ASVS na poziomie N, jednak może także zawierać dodatkowe szczegółowe wymagania z wyższych poziomów ASVS. Na przykład firma z sektora finansowego może dla aplikacji niskiego ryzyka wymagać weryfikacji na poziomie 2 OWASP ASVS oraz dodatkowo weryfikacji pod kątem złośliwego kodu (patrz V13, wyłącznie dla poziomu 4). Mogą mieć również zastosowanie inne wymagania organizacyjne lub biznesowe takie jak zgodność z określonymi politykami bezpieczeństwa lub regulacjami.

tnieje poziom 0
ïkacji. Do
nania konkretnego
mu konieczne jest
ęcie (lub
nalizacja)
ności i ponowna
ïkacja aplikacji.

Poziom 1 - Weryfikacja automatyczna

Poziom 1 ("Weryfikacja automatyczna") jest zwykle odpowiedni dla aplikacji, dla których wymagany jest pewien poziom zaufania, co do poprawnego użycia mechanizmów bezpieczeństwa. Typowymi zagrożeniami dla bezpieczeństwa⁹ będą w tym przypadku wirusy i robaki (cele są wybierane przypadkowo w trakcie skanowania o szerokim zakresie, a atakowane są te najbardziej podatne). Zakres weryfikacji obejmuje kod stworzony lub zmodyfikowany przy tworzeniu aplikacji.

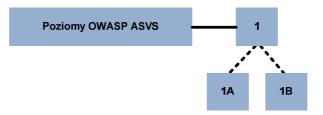
Na poziomie 1 weryfikacja jest przeprowadzana z użyciem automatycznych narzędzi i rozszerzona o ręczną weryfikację. Ten poziom weryfikacji obejmuje tylko częściowe pokrycie bezpieczeństwa aplikacji. Wykonywana na tym poziomie weryfikacja ręczna nie ma na celu zapewnienia kompletnej weryfikacji bezpieczeństwa aplikacji, tylko upewnienie się, że wyniki uzyskane w sposób automatyczny są poprawne i nie stanowią fałszywych wskazań.

Występują dwa elementy składowe poziomu 1. Poziom 1A gdzie wykorzystuje się narzędzia automatycznego skanowania podatności aplikacji (analiza dynamiczna) oraz poziom 1B, na którym wykorzystuje się narzędzia automatycznego skanowania kodu źródłowego (analiza statyczna). Weryfikacja może wykorzystywać poszczególne elementy składowe oddzielnie lub ich kombinację w celu osiągnięcia pełnego poziomu 1. Struktura tych poziomów jest przedstawiona na rysunku poniżej.

⁹ Więcej informacji na temat identyfikacji i szacowania ryzyka związanego z podatnościami można znaleźć w OWASP Testing Guide (OWASP, 2008).



Możliwe jest stwierdzenie czy aplikacja spełnia wymagania poziomu 1A lub 1B, jednak żaden z tych poziomów osobno nie zapewnia takiego poziomu rygorystyczności oceny ani pokrycia, jak aplikacja, która spełnia wymagania poziomu 1. Aplikacja poziomu 1 musi spełniać wymagania poziomu 1A i1B.



Rysunek 3 - Poziomy 1, 1A i 1B OWASP ASVS

Występują następujące minimalne wysoko-poziomowe wymagania dla aplikacji poziomów 1, 1A lub 1B:

Zakres weryfikacji

L1.1 Zakres weryfikacji obejmuje całość kodu napisanego lub zmodyfikowanego w celu stworzenia aplikacji.

Wymagania dotyczące zachowania mechanizmów bezpieczeństwa

Brak Na poziomie 1 nie ma wymagań dotyczących sposobu podejmowania decyzji przez mechanizmy bezpieczeństwa aplikacji.

Wymagania dotyczące użycia mechanizmów bezpieczeństwa

Brak Na poziomie 1 nie ma wymagań określających gdzie w aplikacji muszą być używane mechanizmy bezpieczeństwa.

Wymagania dotyczące implementacji mechanizmów bezpieczeństwa

Brak Na poziomie 1 nie ma wymagań mówiących o sposobie budowy mechanizmów bezpieczeństwa aplikacji.

Wymagania dotyczące weryfikacji mechanizmów bezpieczeństwa

- L1.2 Wykonaj dynamiczne skanowanie aplikacji zgodnie z wymaganiami dla poziomu 1A opisanymi w sekcji "Dokładne wymagania weryfikacyjne".
- L1.3 Wykonaj skanowanie kodu źródłowego aplikacji zgodnie z wymaganiami dla poziomu 1B opisanymi w sekcji "Dokładne wymagania weryfikacyjne".

Wymagania na poziomie 1 pozwalające na użycie dowolnej techniki weryfikacji muszą być zweryfikowane przy użyciu tylko jednej z technik. Dodatkowo, jeśli zestaw narzędzi wybrany przez

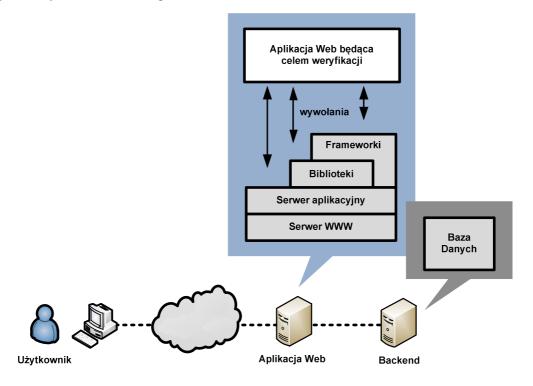


weryfikującego nie posiada możliwości sprawdzenia danego wymagania, weryfikujący może wypełnić tę lukę przez wykonanie weryfikacji ręcznej.^{10 11}

Wymagania raportowe

L1.4 Stwórz raport z weryfikacji opisujący architekturę bezpieczeństwa aplikacji poprzez wymienienie jej komponentów, zawierający wyniki weryfikacji zgodnie z wymaganiami opisanymi w sekcji "Wymagania raportowe z weryfikacji".

Na poziomie 1 komponenty aplikacji mogą być określone jako pojedyncze lub grupy plików źródłowych, bibliotek i/lub plików wykonywalnych, jak pokazano to na rysunku poniżej. Na poziomie 1 ta lista nie musi być posortowana ani zorganizowana w inny sposób, jak przez określenie, które komponenty są częścią aplikacji, a które są częścią środowiska IT. Aplikacja może być więc traktowana jako grupy komponentów wewnątrz pojedynczego monolitycznego obiektu. Nie jest konieczna identyfikacja i dokumentacja ścieżek, jakimi w aplikacji może przechodzić dane zapytanie użytkownika końcowego.



Rysunek 4 - Przykład architektury bezpieczeństwa dla poziomu 1 OWASP ASVS

¹⁰ Więcej informacji na temat przeprowadzania weryfikacji ręcznej można znaleźć w *OWASP Testing Guide* (OWASP, 2008).

¹¹ Więcej informacji na temat przeprowadzania weryfikacji ręcznej poprzez ręczny przegląd kodu można znaleźć w *OWASP Code Review Guide* (OWASP, 2008).



Poziom 1A - Skanowanie dynamiczne (częściowa weryfikacja automatyczna)

Wymagania dla weryfikacji mechanizmów bezpieczeństwa przy pomocy skanowania dynamicznego

Skanowanie dynamiczne (znane również jako "skanowanie podatności aplikacji") polega na użyciu automatycznych narzędzi uzyskujących dostęp do interfejsów aplikacji podczas jej pracy w celu wykrycia podatności w mechanizmach bezpieczeństwa aplikacji. Należy zauważyć, że nie jest to wystarczające do weryfikacji poprawności projektu, implementacji i zastosowania mechanizmów bezpieczeństwa, jednak takie sprawdzenie jest akceptowalne dla poziomu 1. Zakres weryfikacji jest określony przez wymagania dotyczące architektury bezpieczeństwa dla tego poziomu.

- L1A.1 Wykonaj dynamiczne skanowanie aplikacji zgodnie z wymaganiami dla poziomu 1A opisanymi w sekcji "Dokładne wymagania weryfikacyjne"
- L1A.2 Zweryfikuj wszystkie wyniki skanowania dynamicznego poprzez manualne testy penetracyjne aplikacji lub przegląd kodu. Niezweryfikowane wyniki skanowania automatycznego nie dają żadnej pewności i nie są wystarczające do osiągnięcia poziomu 1.

Wiele wystąpień danego typu podatności, które mogą być sprowadzone do jednej głównej przyczyny ich występowania powinno być zgrupowane w pojedynczy wynik, jeśli narzędzie skanujące już tego nie zrobiło.

Poziom 1B - Skanowanie kodu źródłowego (częściowa weryfikacja automatyczna)

Wymagania dla weryfikacji mechanizmów bezpieczeństwa przy pomocy skanowania kodu źródłowego

Skanowanie kodu źródłowego (znane również jako "analiza statyczna") polega na użyciu automatycznych narzędzi do wyszukiwania w kodzie źródłowym aplikacji wzorców wskazujących na podatności. Należy zauważyć, że nie jest to wystarczające do weryfikacji poprawności projektu, implementacji i zastosowania mechanizmów bezpieczeństwa, jednak takie sprawdzenie jest akceptowalne dla poziomu 1. Zakres weryfikacji jest określony przez wymagania dotyczące architektury bezpieczeństwa dla tego poziomu.

- L1B.1 Wykonaj skanowanie kodu źródłowego aplikacji zgodnie z wymaganiami dla poziomu 1B opisanymi w sekcji "Dokładne wymagania weryfikacyjne".
- L1B.2 Zweryfikuj wszystkie wyniki skanowania kodu źródłowego poprzez manualne testy penetracyjne aplikacji lub przegląd kodu. Niezweryfikowane wyniki skanowania automatycznego nie dają żadnej pewności i nie są wystarczające do osiągnięcia poziomu 1.

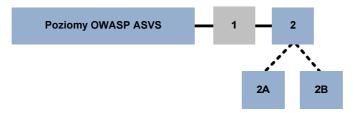
Wiele wystąpień danego typu podatności, które mogą być sprowadzone do jednej głównej przyczyny ich występowania powinno być zgrupowane w pojedynczy wynik, jeśli narzędzie do analizy kodu już tego nie zrobiło.

Poziom 2 - Weryfikacja ręczna

Poziom 2 ("Weryfikacja ręczna") jest zwykle odpowiedni dla aplikacji, które obsługują transakcje ludzi, wykonują transakcje biznesowe między firmami, przetwarzają dane kartowe lub dane osobowe. Poziom 2 daje pewien poziom zaufania, co do prawidłowego użycia mechanizmów bezpieczeństwa i ich prawidłowego działania. Typowymi zagrożeniami dla bezpieczeństwa będą w



tym przypadku wirusy, robaki i mało wymyślni, liczący na okazję napastnicy, używający profesjonalnych narzędzi ataku lub narzędzi open-source. Zakres weryfikacji obejmuje całość kodu napisanego lub zmodyfikowanego w celu stworzenia aplikacji oraz ocenę bezpieczeństwa wszystkich komponentów firm trzecich, dostarczających funkcjonalności bezpieczeństwa dla aplikacji. Występują dwa składowe komponenty poziomu 2, jak przedstawiono na rysunku poniżej.



Rysunek 5 - Poziomy 2, 2A i 2B OWASP ASVS

Możliwe jest stwierdzenie czy aplikacja spełnia wymagania poziomu 2A lub 2B, jednak żaden z tych poziomów osobno nie zapewnia takiego poziomu rygorystyczności oceny ani pokrycia jak poziom 2. Ponieważ poziom 2 zawiera wymagania poziomu 1, do spełnienia wymagań poziomu 2 nie jest wymagane uruchamianie narzędzi automatycznych. Zamiast tego weryfikujący posiada wybór użycia wyłącznie technik ręcznych dla wszystkich wymagań. Jeśli są dostępne wyniki narzędzi automatycznych mogą być użyte przez weryfikującego do wsparcia analizy. Jednakże nawet spełnienie danego wymagania na poziomie 1 nie oznacza automatycznie spełnienia tego samego wymagania na poziomie 2. Jest tak, ponieważ narzędzia automatyczne nie dostarczają wystarczającego dowodu na spełnienie wymagań pozytywnych.

Techniki ręczne wciąż zakładają użycie narzędzi. Mogą to być dowolnego rodzaju narzędzia do analizy lub testów bezpieczeństwa, włączając w to narzędzia automatyczne używane do weryfikacji w ramach poziomu 1. Jednakże takie narzędzia są jedynie pomocą dla analityka w wyszukiwaniu i ocenie poddawanych weryfikacji mechanizmów bezpieczeństwa. Takie narzędzia mogą zawierać (lub nie) logikę pozwalającą na automatyczne wykrywanie podatności w aplikacji.

Występują następujące minimalne wysoko-poziomowe wymagania dla aplikacji poziomu 2, 2A lub 2B:

Zakres weryfikacji

- L2.1 Zakres weryfikacji obejmuje całość kodu napisanego lub zmodyfikowanego w celu stworzenia aplikacji. To wymaganie zostało wprowadzone na poziomie 1.
- L2.2 Zakres weryfikacji obejmuje kod wszystkich pochodzących z firm trzecich Framework-ów, bibliotek i funkcjonalności bezpieczeństwa usług, realizujących lub wspomagających bezpieczeństwo aplikacji. Jest to nowe wymaganie na poziomie 2.

Wymagania dotyczące zachowania mechanizmów bezpieczeństwa

- L2.3 Zweryfikuj czy wszystkie techniczne mechanizmy bezpieczeństwa dokonujące kontroli bezpieczeństwa podejmują decyzję w oparciu o listę dozwolonych wyników (whitelist). Jest to nowe wymaganie na poziomie 2.
- L2.4 Zweryfikuj czy wszystkie mechanizmy bezpieczeństwa dokonujące kontroli bezpieczeństwa oraz mechanizmy bezpieczeństwa, których działanie ma wpływ na bezpieczeństwo nie mogą być ominięte zgodnie z wymaganiami dla poziomów 2A i 2B opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 2.



Wymagania dotyczące użycia mechanizmów bezpieczeństwa

L2.5 Zweryfikuj czy wszystkie mechanizmy bezpieczeństwa są używane w aplikacji wszędzie tam, gdzie są potrzebne, ich implementacje są scentralizowane i umieszczone po stronie serwera zgodnie z wymaganiami dla poziomu 2 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 2.

Wymagania dotyczące implementacji mechanizmów bezpieczeństwa

Brak Na poziomie 2 nie ma wymagań określających, w jaki sposób mają być zbudowane mechanizmy bezpieczeństwa aplikacji.

Wymagania dotyczące weryfikacji mechanizmów bezpieczeństwa

- L2.6 Wykonaj ręczny test penetracyjny aplikacji zgodnie z wymaganiami dla poziomu 2A opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 2.
- L2.7 Wykonaj ręczny przegląd kodu źródłowego aplikacji zgodnie z wymaganiami dla poziomu 2B opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 2.

Wymagania na poziomie 2 pozwalające na użycie dowolnej techniki weryfikacji muszą być zweryfikowane przy użyciu tylko jednej z technik.

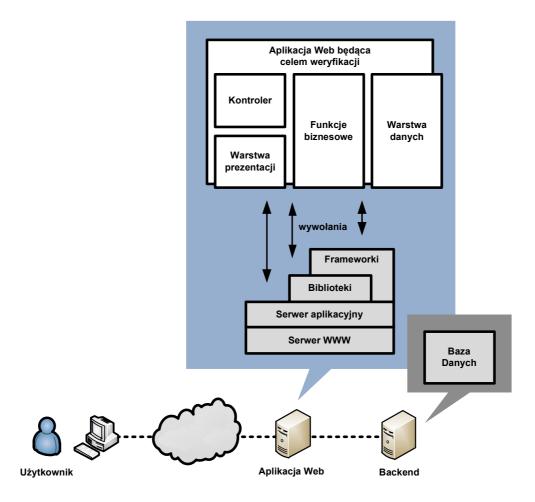
Weryfikujący może wykorzystać w swoich pracach automatyczne skanowanie lub analizę kodu, jednak automatyczna weryfikacja nie może zastąpić przeglądu ręcznego wymaganego dla każdego wymagania na poziomie 2. Jeśli wyniki skanowania pomagają weryfikującemu wykonywać pracę szybciej lub rozszerzają wyniki przeglądu ręcznego to oczywiście mogą być użyte do wspomagania weryfikacji na poziomie 2.

Wymagania Raportowe

L2.8 Stwórz raport z weryfikacji opisujący architekturę bezpieczeństwa aplikacji poprzez pogrupowanie jej komponentów w wysoko-poziomową architekturę oraz zawierający wyniki weryfikacji zgodnie z wymaganiami wymienionymi w sekcji "Wymagania raportowe z weryfikacji". Jest to rozszerzenie wymagań raportowych wprowadzonych na poziomie 1.

Na poziomie 2 komponenty aplikacji mogą być określone jako pojedyncze lub grupy plików źródłowych, bibliotek i/lub plików wykonywalnych, zorganizowane w wysoko-poziomową architekturę (np. komponenty Model-Widok-Kontroler (Model-View-Controller, MVC), komponenty funkcji biznesowych, komponenty warstwy danych). Na przykład diagram poniżej przedstawia aplikację składającą się z aplikacji serwera, aplikacji serwera aplikacyjnego, własnego kodu, bibliotek i aplikacji bazy danych zgrupowanych zgodnie z architekturą MVC. Na poziomie 2 ścieżka lub ścieżki, jakimi w aplikacji może przechodzić dane zapytanie użytkownika końcowego muszą być udokumentowane, jak przedstawiono to na rysunku poniżej. Jednakże, nie wszystkie ścieżki muszą być przeanalizowane.





Rysunek 6 - Przykład architektury bezpieczeństwa dla poziomu 2 OWASP ASVS



Poziom 2A - Test bezpieczeństwa (częściowa weryfikacja ręczna)

Wymagania dotyczące weryfikacji mechanizmów bezpieczeństwa przy pomocy ręcznych testów penetracyjnych aplikacji

Ręczne testy bezpieczeństwa aplikacji składają się z tworzonych dynamicznie testów w celu weryfikacji poprawności projektu, implementacji i użycia mechanizmów bezpieczeństwa. Zakres weryfikacji jest zdefiniowany przez wymagania dotyczące architektury bezpieczeństwa dla tego poziomu.

L2A.1 Wykonaj ręczne testowanie bezpieczeństwa aplikacji zgodnie z wymaganiami dla poziomu 2A opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 2.

W miejscach, gdzie jest to uzasadnione, weryfikujący może użyć próbkowania do ustalenia, w jaki sposób używany jest określony mechanizm bezpieczeństwa. Weryfikujący może postanowić udokumentować wzorzec podatności, który pozwoli developerom na sprawne znalezienie i poprawienie wszystkich wystąpień tego wzorca w wersji bazowej oprogramowania. Wiele wystąpień wzorca podatności, które mogą być sprowadzone do jednej głównej przyczyny ich występowania powinno być zgrupowane w pojedynczy wynik.

Poziom 2B - Przegląd kodu (częściowa weryfikacja ręczna)

Wymagania dotyczące weryfikacji mechanizmów bezpieczeństwa przy pomocy ręcznego przeglądu kodu

Ręczny przegląd kodu polega na wyszukiwaniu i analizie (wykonywanych przez człowieka) kodu źródłowego aplikacji w celu weryfikacji projektu, implementacji i właściwego użycia mechanizmów bezpieczeństwa. Oczekuje się, że analiza tego typu jest wspomagana narzędziami, jednak może po prostu wykorzystywać powszechnie dostępne narzędzia, takie jak edytor kodu źródłowego lub IDE. Zakres weryfikacji jest określony przez wymagania dotyczące architektury bezpieczeństwa dla tego poziomu.

L2B.1 Ręczny przegląd kodu aplikacji zgodnie z wymaganiami dla poziomu 2B opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 2.

W miejscach gdzie jest to uzasadnione weryfikujący może użyć właściwego próbkowania do ustalenia, w jaki sposób używany jest określony mechanizm bezpieczeństwa. Weryfikujący może postanowić udokumentować wzorzec podatności, który pozwoli developerom na sprawne znalezienie i poprawienie wszystkich wystąpień tego wzorca w wersji bazowej oprogramowania. Wiele wystąpień wzorca podatności, które mogą być sprowadzone do jednej głównej przyczyny ich występowania powinno być zgrupowane w pojedynczy wynik

Poziom 3 - Weryfikacja projektu

Poziom 3 ("Weryfikacja projektu") jest zwykle odpowiedni dla aplikacji obsługujących ważne transakcje biznesowe pomiędzy organizacjami, włączając to aplikacje przetwarzające dane o stanie zdrowia, aplikacje spełniające krytyczne dla biznesu lub wrażliwe funkcje lub przetwarzające inne wrażliwe aktywa. Typowymi zagrożeniami dla bezpieczeństwa będą w tym przypadku wirusy, robaki, napastnicy liczący na okazję oraz zdeterminowani napastnicy (wykwalifikowani i zmotywowani, skupieni na określonych celach, używający m.in. specjalnie przystosowanych narzędzi skanujących). Zakres weryfikacji obejmuje całość kodu napisanego lub zmodyfikowanego w celu stworzenia aplikacji, a także ocenę bezpieczeństwa komponentów firm trzecich, dostarczających funkcjonalności bezpieczeństwa dla aplikacji. Poziom 3 zapewnia, że mechanizmy bezpieczeństwa funkcjonują w sposób prawidłowy i są użyte w aplikacji wszędzie tam, gdzie powinny być użyte by



egzekwować polityki specyficzne dla aplikacji. Poziom 3 nie jest podzielony na wiele części, jak to przedstawiono na rysunku poniżej.



Rysunek 7 - Poziom 3 OWASP ASVS

Występują następujące minimalne wysoko-poziomowe wymagania dla aplikacji poziomu 3:

Zakres Weryfikacji

- L3.1 Zakres weryfikacji obejmuje całość kodu napisanego lub zmodyfikowanego w celu stworzenia aplikacji. To wymaganie zostało wprowadzone na poziomie 1.
- L3.2 Zakres weryfikacji obejmuje kod wszystkich pochodzących z firm trzecich framework-ów, bibliotek i funkcjonalności bezpieczeństwa usług, realizujących lub wspomagających bezpieczeństwo aplikacji. To wymaganie zostało wprowadzone na poziomie 2.
- L3.3 Zakres weryfikacji obejmuje kod wszystkich pochodzących z firm trzecich framework-ów, bibliotek i usług związanych z aplikacją. Jest to nowe wymaganie na poziomie 3.

Wymagania dotyczące zachowania mechanizmów bezpieczeństwa

- L3.4 Zweryfikuj czy wszystkie techniczne mechanizmy bezpieczeństwa dokonujące kontroli bezpieczeństwa podejmują decyzję w oparciu o listę dozwolonych wyników (whitelist). To wymaganie zostało wprowadzone na poziomie 2.
- L3.5 Zweryfikuj czy wszystkie mechanizmy bezpieczeństwa dokonujące kontroli bezpieczeństwa oraz mechanizmy bezpieczeństwa, których działanie ma wpływ na bezpieczeństwo nie mogą być ominięte zgodnie z wymaganiami dla poziomu 3 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". To wymaganie zostało wprowadzone na poziomie 2.

Wymagania dotyczące użycia mechanizmów bezpieczeństwa

L3.6 Zweryfikuj czy wszystkie mechanizmy bezpieczeństwa są używane w aplikacji wszędzie tam, gdzie są potrzebne, ich implementacje są scentralizowane i umieszczone po stronie serwera, zgodnie z wymaganiami dla poziomu 3 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". To wymaganie zostało wprowadzone na poziomie 2.

Wymagania dotyczące implementacji mechanizmów bezpieczeństwa

Brak Na poziomie 3 nie ma wymagań określających, w jaki sposób mają być zbudowane mechanizmy bezpieczeństwa aplikacji.

Wymagania dotyczące weryfikacji mechanizmów bezpieczeństwa

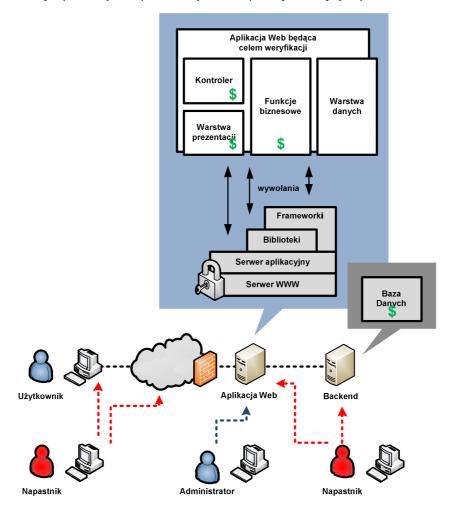
- L3.7 Zweryfikuj ręcznie aplikację zgodnie z wymaganiami dla poziomu 3 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to rozszerzenie wymagań dla weryfikacji ręcznej wprowadzonych na poziomie 2.
- L3.8 Udokumentuj architekturę bezpieczeństwa i użyj jej do weryfikacji poprawności projektu i użycia wszystkich mechanizmów bezpieczeństwa poprzez wykonanie modelowania zagrożeń. Jest to nowe wymaganie na poziomie 3.



Wymagania Raportowe

L3.9 Napisz raport z weryfikacji opisujący architekturę bezpieczeństwa aplikacji poprzez pogrupowanie jej komponentów w wysoko-poziomową architekturę, obejmującą wyniki modelowania zagrożeń oraz zawierający wyniki weryfikacji zgodnie z wymaganiami opisanymi w sekcji "Wymagania raportowe z weryfikacji". Jest to rozszerzenie wymagań raportowych poziomu 2.

Na poziomie 3 komponenty aplikacji mogą być określone jako pojedyncze lub grupy plików źródłowych, bibliotek i/lub plików wykonywalnych, zorganizowane w wysoko-poziomową architekturę (np. komponenty MVC, komponenty funkcji biznesowych, komponenty warstwy danych). Na poziomie 3 wymagane są dodatkowo informacje wspomagające modelowanie zagrożeń takie jak czynniki stwarzające zagrożenia oraz aktywa. Ścieżka lub ścieżki, jakimi w wysoko-poziomowym widoku aplikacji przechodzi dane zapytanie użytkownika końcowego muszą być udokumentowane, jak przedstawiono to na rysunku poniżej. Na poziomie 3 wszystkie potencjalne ścieżki przechodzące przez wysoko-poziomowy widok aplikacji muszą być sprawdzone.



Rysunek 8 - Przykład architektury bezpieczeństwa dla poziomu 3 OWASP ASVS¹²

¹² Znaki dolara na diagramie oznaczają aktywa.



Poziom 4 - Weryfikacja wewnętrzna

Poziom 4 ("Weryfikacja wewnętrzna") jest zwykle odpowiedni dla krytycznych aplikacji, od których zależy życie i bezpieczeństwo, krytyczna infrastruktura lub zadania związane z obronnością. Poziom 4 może być również odpowiedni dla aplikacji służących do przetwarzania wrażliwych aktywów. Poziom 4 zapewnia, że mechanizmy bezpieczeństwa pracują prawidłowo, są użyte w aplikacji wszędzie tam gdzie są potrzebne do egzekwowania polityk specyficznych dla aplikacji oraz przestrzegano praktyk pisania bezpiecznego kodu. Zagrożeniami dla bezpieczeństwa w tym przypadku będą zdeterminowani napastnicy (wykwalifikowani i zmotywowani napastnicy skupieni na określonych celach, używający m.in. specjalnie przystosowanych narzędzi skanujących). Zakres weryfikacji wykracza poza zakres poziomu 3 by objąć cały kod używany przez aplikację. Poziom 4 nie jest podzielony na element składowe, jak to przedstawiono na rysunku poniżej.



Rysunek 9 - Poziom 4 OWASP ASVS

Występują następujące minimalne wysoko-poziomowe wymagania dla aplikacji poziomu 4:

Zakres Weryfikacji

- L4.1 Zakres weryfikacji obejmuje całość kodu napisanego lub zmodyfikowanego w celu stworzenia aplikacji. To wymaganie zostało wprowadzone na poziomie 1.
- L4.2 Zakres weryfikacji obejmuje kod wszystkich pochodzących z firm trzecich framework-ów, bibliotek i funkcjonalności bezpieczeństwa usług, realizujących lub wspomagających bezpieczeństwo aplikacji. To wymaganie zostało wprowadzone na poziomie 2.
- L4.3 Zakres weryfikacji obejmuje kod wszystkich pochodzących z firm trzecich framework-ów, bibliotek i usług związanych z aplikacją. To wymaganie zostało wprowadzone na poziomie 3.
- L4.4 Zakres weryfikacji obejmuje całość pozostałego kodu związanego z aplikacją, włączając w to frameworki, biblioteki, środowiska uruchomieniowe oraz narzędzia służące do rozwoju, budowania i wdrażania aplikacji. Zakres nie obejmuje kodu oprogramowania platformy, takich jak serwer aplikacyjny, serwer bazy danych, wirtualna maszyna lub system operacyjny, które są poddawane stałej publicznej kontroli. Jest to nowe wymaganie na poziomie 4.

Wymagania dotyczące zachowania mechanizmów bezpieczeństwa

- L4.5 Zweryfikuj czy wszystkie mechanizmy bezpieczeństwa dokonujące kontroli bezpieczeństwa podejmują decyzje w oparciu o listę dozwolonych wyników (model pozytywny). To wymaganie zostało wprowadzone na poziomie 2.
- L4.6 Zweryfikuj czy wszystkie mechanizmy bezpieczeństwa dokonujące kontroli bezpieczeństwa oraz mechanizmy bezpieczeństwa, których działanie ma wpływ na bezpieczeństwo nie mogą być ominięte zgodnie z wymaganiami dla poziomu 4 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". To wymaganie zostało wprowadzone na poziomie 2.

Wymagania dotyczące użycia mechanizmów bezpieczeństwa

L4.7 Zweryfikuj czy wszystkie mechanizmy bezpieczeństwa są używane w aplikacji wszędzie tam, gdzie są potrzebne, ich implementacje są scentralizowane i umieszczone po stronie serwera zgodnie z wymaganiami dla poziomu 4 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". To wymaganie zostało wprowadzone na poziomie 3.



Wymagania dotyczące implementacji mechanizmów bezpieczeństwa

L4.8 Zweryfikuj czy aplikacja nie zawiera złośliwego kodu zgodnie z wymaganiami dla poziomu 4 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 4.

Wymagania dotyczące weryfikacji mechanizmów bezpieczeństwa

- L4.9 Ręcznie zweryfikuj aplikację zgodnie z wymaganiami dla poziomu 4 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to rozszerzenie wymagania z poziomu 3.
- L4.10 Udokumentuj architekturę bezpieczeństwa i użyj jej do weryfikacji poprawności projektu i użycia mechanizmów bezpieczeństwa poprzez wykonanie modelowania zagrożeń. To wymaganie zostało wprowadzone na poziomie 3.
- L4.11 Przejrzyj ręcznie całość kodu napisanego lub zmodyfikowanego w aplikacji pod kątem złośliwego kodu¹³ zgodnie z wymaganiami dla poziomu 4 opisanymi w sekcji "Dokładne wymagania weryfikacyjne". Jest to nowe wymaganie na poziomie 4.

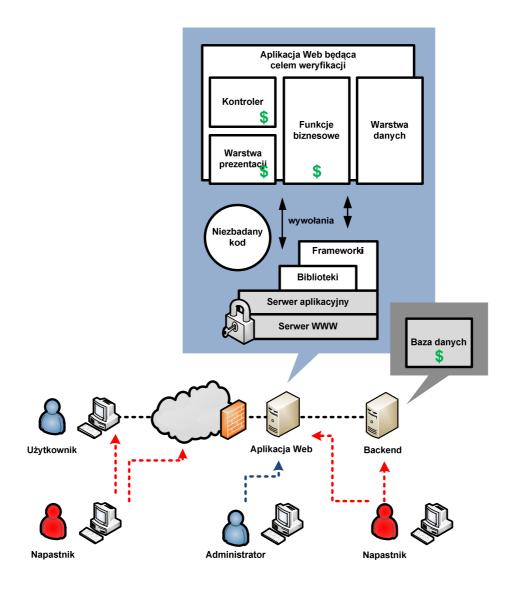
Wymagania Raportowe

L4.12 Napisz raport z weryfikacji opisujący architekturę bezpieczeństwa aplikacji zgodnie z wymaganiami dla poziomu 3, obejmujący całość kodu aplikacji i zawierający wyniki weryfikacji zgodnie z wymaganiami opisanymi w sekcji "Wymagania raportowe z weryfikacji". Jest to rozszerzenie wymagań raportowych z poziomu 3.

Na poziomie 4 architektura aplikacji powinna być odwzorowana tak jak dla poziomu 3. Dodatkowo poziom 4 wymaga by całość kodu aplikacji, włączając w to kod, który nie podlegał wyraźniej ocenie, był zidentyfikowany, jako część definicji aplikacji, jak to przedstawiono na diagramie poniżej. Ten kod musi zawierać wszystkie bliblioteki, frameworki i kod wspomagający, od którego zależy praca aplikacji. Poprzednie weryfikacje tych komponentów mogą być użyte ponownie, jako część innej weryfikacji. Kod platform takich jak system operacyjny, maszyna wirtualna lub bazowe biblioteki dostarczone ze środowiskiem wirtualnej maszyny, serwer Web lub serwer aplikacyjny nie jest uwzględniany na poziomie 4. Na przykład biblioteki związane ze środowiskiem uruchomieniowym Java nie muszą podlegać ocenie na poziomie 4.

¹³ Złośliwy kod nie jest dokładnie tym samym pojęciem co malware. Patrz definicja złośliwego kodu w słowniczku.





Rysunek 10 - Przykład niezbadanego kodu na poziomie 4 OWASP ASVS

Interpretacje wymagań i precedensy

OWASP ASVS jest "żyjącym" dokumentem. Jeśli przeprowadzasz weryfikację bezpieczeństwa aplikacji zgodnie z tym standardem powinieneś zapoznać się z artykułami na stronie projektu OWASP ASVS pod adresem:

http://www.owasp.org/index.php/ASVS#Articles_Below_-_More_About_ASVS_and_Using_It

Artykuły na stronie projektu OWASP ASVS zawierają objaśnienia wymagań, precedensy werdyktów weryfikacji i pomocne podpowiedzi.



Szczegółowe wymagania weryfikacyjne

Niniejsza sekcja ASVS określa dokładne wymagania weryfikacyjne wywodzące się z wymagań wysoko-poziomowych dla każdego z poziomów weryfikacji zdefiniowanych w tym standardzie. Każda z poniższych sekcji określa zestaw dokładnych wymagań weryfikacyjnych pogrupowanych w spokrewnione obszary.

ASVS określa następujące obszary wymagań bezpieczeństwa:

- V1. Architektura bezpieczeństwa
- V2. Uwierzytelnianie
- V3. Zarządzanie sesją
- V4. Kontrola dostępu
- V5. Walidacja wejścia
- V6. Enkodowanie/escapowanie wyjścia
- V7. Kryptografia
- V8. Obsługa błędów i logowanie
- V9. Ochrona danych
- V10. Bezpieczeństwo komunikacji
- V11. Bezpieczeństwo HTTP
- V12. Konfiguracja zabezpieczeń
- V13. Wyszukiwanie złośliwego kodu
- V14. Bezpieczeństwo wewnętrzne

Poniżej wymienione są wymagania dla każdego z tych obszarów, które muszą być spełnione dla każdego z poziomów weryfikacji:

Poziom 1: Weryfikacja automatyczna

Poziom 1A - Skanowanie dynamiczne (częściowa weryfikacja automatyczna)

Poziom 1B - Skanowanie kodu źródłowego (częściowa weryfikacja automatyczna)

Poziom 2: Weryfikacja ręczna

Poziom 2A - Test bezpieczeństwa (częściowa weryfikacja ręczna)

Poziom 2B - Przegląd kodu (częściowa weryfikacja ręczna)

Poziom 3: Weryfikacja projektu

Poziom 4: Weryfikacja wewnętrzna



V1 - Wymagania dotyczące dokumentacji architektury bezpieczeństwa

Dla zapewnienia kompletności i dokładności (oraz powtarzalności w przypadku konieczności wykonania działań naprawczych) przeprowadzanej weryfikacji bezpieczeństwa aplikacji wszystkie poziomy ASVS wymagają udokumentowania pewnych podstawowych informacji dotyczących architektury bezpieczeństwa. Analiza może wrócić do wysoko-poziomowej architektury bezpieczeństwa aplikacji, gdzie wyniki mogą zostać ponownie prześledzone. Wymagania zaczynają się od podstawowego poziomu ilości informacji na temat architektury bezpieczeństwa, które muszą być zgromadzone, a ich ilość rośnie z każdym poziomem. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji odnośnie dokumentacji architektury bezpieczeństwa.

Tabela 1 - Wymagania dotyczące dokumentacji architektury bezpieczeństwa w OWASP ASVS (V1)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V1.1	Zweryfikuj czy zostały zidentyfikowane wszystkie komponenty występujące w aplikacji (zarówno pojedyncze jak i grupy plików źródłowych, bibliotek i/lub plików wykonywalnych).	√	√	√	✓	√	✓
V1.2	Zweryfikuj czy zostały zidentyfikowane wszystkie komponenty, które nie są częścią aplikacji, ale są potrzebne do jej działania.			✓	✓	✓	✓
V1.3	Zweryfikuj czy została zdefiniowana wysoko-poziomowa architektura aplikacji. ¹⁴			✓	✓	✓	✓
V1.4	Zweryfikuj czy wszystkie komponenty aplikacji zostały zdefiniowane w kategoriach funkcji biznesowych i/lub funkcji bezpieczeństwa, dostarczanych przez te komponenty.					✓	✓
V1.5	Zweryfikuj czy wszystkie komponenty, które nie są częścią aplikacji, ale zależy od nich jej działanie zostały zdefiniowane w kategoriach funkcji biznesowych i/lub funkcji bezpieczeństwa, dostarczanych przez te komponenty.					✓	✓

¹⁴ Weryfikujący może opracować lub udokumentować wysoko-poziomowy projekt, jeśli nie został dostarczony przez developera aplikacji.



Standard Aplikacji Web



V1.6 Zweryfikuj czy są dostępne informacje wynikające z modelowania zagrożeń.



V2 - Wymagania weryfikacyjne dotyczące uwierzytelniania

Wymagania weryfikacyjne dotyczące uwierzytelniania definiują zestaw wymagań względem bezpiecznego generowania i obsługi danych służących do logowania. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 2 - Wymagania dotyczące uwierzytelniania w OWASP ASVS (V2)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V2.1	Zweryfikuj czy wszystkie strony oraz zasoby wymagają uwierzytelnienia za wyjątkiem tych, które mają być dostępne publicznie.	✓	✓	✓	✓	✓	✓
V2.2	Zweryfikuj czy wszystkie pola służące do wprowadzania haseł nie pokazują haseł użytkowników w czasie ich wpisywania i mają wyłączoną funkcję automatycznego uzupełniania (to samo odnosi się do formularzy, w których znajdują się pola służące do wprowadzania haseł).	✓	√	√	√	√	✓
V2.3	Zweryfikuj czy po przekroczeniu maksymalnej liczby prób uwierzytelnienia konto zostaje zablokowane na okres czasu pozwalający na powstrzymanie ataków typu brute force.	✓		√	√	√	✓
V2.4	Zweryfikuj czy wszystkie mechanizmy uwierzytelniania są egzekwowane po stronie serwera.			✓	✓	✓	✓
V2.5	Zweryfikuj czy wszystkie mechanizmy uwierzytelniania (również biblioteki wywołujące zewnętrzne usługi uwierzytelniania) są zaimplementowane centralnie.				√	✓	✓
V2.6	Zweryfikuj czy wszystkie mechanizmy uwierzytelniania, które kończą pracę niepowodzeniem robią to sposób bezpieczny.			✓	✓	✓	✓
V2.7	Zweryfikuj czy siła danych wykorzystywanych w procesie uwierzytelniania jest wystarczająca do wytrzymania ataków typowych dla zagrożeń we wdrożonym środowisku.			~	√	- ✓	✓



V2.8	Zweryfikuj czy wszystkie funkcje zarządzania kontami są odporne na ataki przynajmniej w takim stopniu jak podstawowy mechanizm uwierzytelniania.		√	✓	✓	✓
V2.9	Zweryfikuj czy użytkownicy mogą w sposób bezpieczny zmienić swoje dane do uwierzytelniania przy użyciu mechanizmu, który jest odporny na ataki przynajmniej w takim stopniu jak podstawowy mechanizm uwierzytelniania.		✓	✓	✓	✓
V2.10	Zweryfikuj czy przed zezwoleniem na wykonanie wrażliwych operacji w aplikacji wymagane jest ponowne uwierzytelnienie.		✓	✓	✓	✓
V2.11	Zweryfikuj czy ważność danych do uwierzytelniania wygasa w konfigurowalnym przez administratora czasie.		✓	✓	✓	✓
V2.12	Zweryfikuj czy są logowane wszystkie decyzje dotyczące wyników uwierzytelnienia.			✓	✓	✓
V2.13	Zweryfikuj czy hasła do kont są tworzone z użyciem soli (salt) unikalnej dla danego konta (np. wewnętrzny identyfikator użytkownika, czas utworzenia konta) i przekształcane przy użyciu mechanizmu hashowania przed ich zapisaniem.			√	√	√
V2.14	Zweryfikuj czy wszystkie dane uwierzytelniania służące do uzyskiwania dostępu do usług zewnętrznych względem aplikacji są zaszyfrowane i przechowywane w zabezpieczonej lokalizacji (nie w kodzie źródłowym).			✓	√	√
V2.15	Zweryfikuj czy kod implementujący mechanizmy uwierzytelniania lub wykorzystujący te mechanizmy nie zawiera kodu złośliwego.					✓

V3 - Wymagania weryfikacyjne dotyczące zarządzania sesją

Wymagania weryfikacyjne dotyczące zarządzania sesją definiują zestaw wymagań względem bezpiecznego użycia zapytań HTTP, odpowiedzi, sesji, cookie, nagłówków i logowania w celu zapewnienia odpowiedniego zarządzania sesjami. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 3 - Wymagania dotyczące zarządzania sesją w OWASP ASVS (V3)



	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V3.1	Zweryfikuj czy aplikacja wykorzystuje domyślną implementację mechanizmu zarządzania sesją, dostarczaną przez framework.	✓		✓	✓	✓	✓
V3.2	Zweryfikuj czy sesje są unieważniane przy wylogowaniu użytkownika.	✓		✓	✓	✓	✓
V3.3	Zweryfikuj czy sesje wygasają po określonym czasie bezczynności.	✓		✓	✓	✓	✓
V3.4	Zweryfikuj czy sesje wygasają po administracyjnie konfigurowalnym maksymalnym okresie czasu, niezależnie od aktywności (bezwarunkowy okres ważności).					✓	✓
V3.5	Zweryfikuj czy wszystkie strony, do których dostęp wymaga uwierzytelnienia zawierają linki umożliwiające wylogowanie.	✓		√	√	√	✓
V3.6	Zweryfikuj czy identyfikatory sesji nie są nie są ujawniane poza nagłówkami cookie, w szczególności nie są ujawniane w URL, komunikatach błędów lub logach. Należy również zweryfikować, czy aplikacja nie pozwala na przepisywanie w URL cookie		√		√	√	✓
V3.7	zweryfikuj czy identyfikator sesji jest zmieniany przy logowaniu.			✓	✓	✓	✓
V3.8	Zweryfikuj czy identyfikator sesji jest zmieniany przy ponownym uwierzytelnieniu.			✓	✓	✓	✓
V3.9	Zweryfikuj czy identyfikator sesji jest zmieniany lub usuwany przy wylogowaniu.			✓	✓	✓	✓
V3.10	Zweryfikuj czy tylko identyfikatory sesji wygenerowane przez framework aplikacji są uznawane przez aplikację za poprawne.			√		✓	✓
V3.11	Zweryfikuj czy tokeny uwierzytelnionych sesji są odpowiednio długie i losowe by wytrzymać ataki typowe dla zagrożeń we wdrożonym środowisku.					√	✓



V3.12	Zweryfikuj czy cookie zawierające uwierzytelnione tokeny/identyfikatory sesji mają swoje atrybuty domain i path ustawione odpowiednio do tej lokalizacji.		- ✓	→
V3.13	Zweryfikuj czy cały kod implementujący lub wykorzystujący mechanizmy zarządzania sesją nie zawiera kodu złośliwego.			✓

V4 - Wymagania weryfikacyjne dotyczące kontroli dostępu

Wymagania weryfikacyjne dotyczące kontroli dostępu definiują, w jaki sposób aplikacja może bezpiecznie egzekwować kontrolę dostępu. W większości aplikacji kontrola dostępu musi być realizowana w wielu miejscach i różnych warstwach aplikacyjnych. Niniejsze wymagania definiują wymagania weryfikacyjne dotyczące kontroli dostępu do URL, funkcji biznesowych, danych, usług i plików. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.



Tabela 4 - Wymagania dotyczące kontroli dostępu w OWASP ASVS (V4)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V4.1	Zweryfikuj czy użytkownicy mogą uzyskać dostęp wyłącznie do chronionych funkcji, do których posiadają przyznane uprawnienia.	✓	✓	✓	✓	✓	√
V4.2	Zweryfikuj czy użytkownicy mogą uzyskać dostęp wyłącznie do URLi, do których posiadają przyznane uprawnienia.	✓		✓	✓	✓	✓
V4.3	Zweryfikuj czy użytkownicy mogą uzyskać dostęp wyłącznie do tych plików z danymi, do których posiadają przyznane uprawnienia.	✓		✓	✓	✓	✓
V4.4	Zweryfikuj czy bezpośrednie odwołania do obiektów są chronione w taki sposób, że tylko autoryzowane obiekty są osiągalne dla użytkowników.	√		√	√	√	√
V4.5	Zweryfikuj czy przeglądanie zawartości katalogów jest wyłączone, chyba że świadomie zadecydowano inaczej.	✓		✓		✓	√
V4.6	Zweryfikuj czy użytkownicy mogą uzyskać dostęp wyłącznie do usług, do których posiadają przyznane uprawnienia.			✓	✓	✓	✓
V4.7	Zweryfikuj czy użytkownicy mogą uzyskać dostęp wyłącznie do tych danych, do których posiadają przyznane uprawnienia.			✓	✓	✓	✓
V4.8	Zweryfikuj czy mechanizmy kontroli dostępu, które kończą pracę niepowodzeniem robią to w sposób bezpieczny.			✓	✓	✓	✓
V4.9	Zweryfikuj czy reguły kontroli dostępu warstwy prezentacji są egzekwowane po stronie serwera.			✓	✓	✓	✓
V4.10	Zweryfikuj czy wszystkie atrybuty użytkowników i danych oraz informacje o polityce wykorzystywane przez mechanizmy kontroli dostępu nie mogą być poddawane manipulacji przez użytkowników, chyba, że są oni do tego uprawnieni.			√	√	√	√



V4.11	Zweryfikuj czy wszystkie mechanizmy kontroli dostępu są egzekwowane po stronie serwera.		✓	✓	✓	✓
V4.12	Zweryfikuj czy istnieje scentralizowany mechanizm (również dla bibliotek wywołujących zewnętrzne usługi autoryzacji) zabezpieczający dostęp do każdego typu chronionych zasobów.			✓	✓	✓
V4.13	Zweryfikuj czy nie można ominąć ograniczeń dotyczących wejścia i dostępu nałożonych na aplikację przez wymagania biznesowe (takie jak dzienne limity transakcji lub kolejność zadań).		√	✓	✓	✓
V4.14	Zweryfikuj czy wszystkie decyzje dotyczące kontroli dostępu mogą być logowane a wszystkie decyzje zakończone niepowodzeniem są logowane.			✓	√	√
V4.15	Zweryfikuj czy cały kod implementujący lub wykorzystujący mechanizmy kontroli dostępu nie zawiera kodu złośliwego.					✓

V5 - Wymagania weryfikacyjne dotyczące walidacji wejścia

Wymagania weryfikacyjne dotyczące walidacji wejścia definiują zestaw wymagań względem walidacji danych wejściowych tak, aby mogły być bezpiecznie użyte w aplikacji. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 5 - Wymagania dotyczące walidacji wejścia w OWASP ASVS (V5)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V5.1	Zweryfikuj czy środowisko uruchomieniowe nie jest podatne na przepełnienia bufora lub mechanizmy bezpieczeństwa zapobiegają wystąpieniu takich błędów.	✓	√	√	√	✓	✓
V5.2	Zweryfikuj czy dla wszystkich wejść są zdefiniowane i zastosowane wzorce pozytywnej walidacji.	✓	√	✓	√	✓	✓
V5.3	Zweryfikuj czy wszystkie walidacje wejścia zakończone niepowodzeniem powodują odrzucenie lub oczyszczenie danych wejściowych.	✓		✓	✓	✓	✓



V5.4	Zweryfikuj czy dla wszystkich źródeł wejścia jest zdefiniowana strona kodowa, np. UTF-8.		✓	✓	✓	✓
V5.5	Zweryfikuj czy cała walidacja danych wejściowych jest realizowana po stronie serwera.		✓	✓	✓	✓
V5.6	Zweryfikuj czy aplikacja wykorzystuje jeden mechanizm walidacji danych wejściowych dla każdego typu przyjmowanych danych.			✓	✓	✓
V5.7	Zweryfikuj czy wszystkie walidacje wejścia zakończone niepowodzeniem są logowane.			✓	✓	✓
V5.8	Zweryfikuj czy dla wszystkich dekoderów lub interpreterów przed rozpoczęciem walidacji wszystkie dane wejściowe są ustandaryzowane.				✓	✓
V5.9	Zweryfikuj czy mechanizmy walidacji wejścia nie zawierają kodu złośliwego					✓

V6 - Wymagania weryfikacyjne dotyczące enkodowania/escapowania wyjścia

Wymagania weryfikacyjne dotyczące enkodowania/escapowania wyjścia definiują zestaw wymagań, które pozwalają upewnić się, że dane wyjściowe są poprawnie enkodowane tak, aby były bezpieczne dla zewnętrznych aplikacji. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 6 - Wymagania dotyczące enkodowania/escapowania wyjścia w OWASP ASVS (V6)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V6.1	Zweryfikuj czy wszystkie niezaufane dane, których wynikiem jest kod HTML (elementy HTML, atrybuty HTML, wartości danych javascript, bloki CSS i atrybuty URI) podlegają escapowaniu odpowiednio do kontekstu.		√	✓	✓	✓	✓
V6.2	Zweryfikuj czy wszystkie mechanizmy enkodowania/escapowania są zaimplementowane po stronie serwera.			✓	✓	√	✓



V6.3	Zweryfikuj czy mechanizmy enkodowania/escapowania danych wyjściowych enkodują wszystkie znaki, które nie są uważane za bezpieczne dla danego interpretera.		✓	✓	✓
V6.4	Zweryfikuj czy wszystkie niezaufane dane trafiające do interpreterów SQL używają parametryzowanych interfejsów (ang. prepared statements), przygotowanych instrukcji lub są właściwie escapowane.		✓	✓	✓
V6.5	Zweryfikuj czy wszystkie niezaufane dane, których wynikiem jest XML używają parametryzowanych interfejsów lub są właściwie escapowane.		✓	✓	✓
V6.6	Zweryfikuj czy wszystkie niezaufane dane używane w zapytaniach LDAP są właściwie escapowane.		✓	✓	✓
V6.7	Zweryfikuj czy wszystkie niezaufane dane używane jako parametry poleceń systemu operacyjnego są właściwie escapowane.		✓	✓	✓
V6.8	Zweryfikuj czy wszystkie niezaufane dane, które są przekazywane do interpreterów niewymienionych powyżej są właściwie escapowane.		✓	✓	√
V6.9	Zweryfikuj czy dla każdego typu enkodowania/escapowania wyjścia wykonywanego przez aplikację istnieje pojedynczy mechanizm bezpieczeństwa dla tego typu docelowych danych wyjściowych.			✓	✓
V6.10	Zweryfikuj czy kod implementujący lub wykorzystujący mechanizmy walidacji wyjścia nie zawiera kodu złośliwego.				✓

V7 - Wymagania weryfikacyjne dotyczące kryptografii

Wymagania weryfikacyjne dotyczące kryptografii definiują zestaw wymagań, które mogą być użyte do weryfikacji używanych w aplikacji operacji szyfrowania, zarządzania kluczami, generowania liczb losowych oraz hashowania. Aplikacje powinny zawsze korzystać z modułów kryptograficznych sprawdzonych pod kątem zgodności z FIPS 140-2 lub innym równoważnym standardem (np. niepochodzącym z USA). Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.



Tabela 7 - Wymagania dotyczące kryptografii w OWASP ASVS (V7)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V7.1	Zweryfikuj czy wszystkie funkcje kryptograficzne służące do zabezpieczenia danych poufnych przed użytkownikiem aplikacji są zaimplementowane po stronie serwera.			✓	✓	✓	✓
V7.2	Zweryfikuj czy wszystkie moduły kryptograficzne, które kończą pracę niepowodzeniem robią to sposób bezpieczny.			✓	✓	✓	✓
V7.3	Zweryfikuj czy dostęp do wszystkich kluczy master secret jest chroniony przed nieuprawnionym dostępem (klucz master secret to poufne dane aplikacji przechowywane na dysku w formie niezaszyfrowanego pliku, używane do ochrony dostępu do informacji o konfiguracji zabezpieczeń).				√	✓	✓
V7.4	Zweryfikuj czy hashe haseł są tworzone przy użyciu soli.				✓	✓	✓
V7.5	Zweryfikuj czy działania modułu kryptograficznego zakończone niepowodzeniem są logowane.				√	√	✓
V7.6	Zweryfikuj czy wszystkie liczby losowe, losowe nazwy plików, losowe GUID'y oraz losowe ciągi znaków, których napastnik nie powinien odgadnąć generowane są z wykorzystaniem zatwierdzonego generatora liczb losowych z modułu kryptograficznego.				✓	✓	✓
V7.7	Zweryfikuj czy moduły kryptograficzne wykorzystywane przez aplikację zostały zatwierdzone w zakresie zgodności ze standardem FIPS 140-2 lub jego odpowiednikiem (patrz http://csrc.nist.gov/groups/STM/cmvp/validati on.html).					✓	✓
V7.8	Zweryfikuj czy moduły kryptograficzne pracują w trybie zgodnym z opublikowanymi politykami bezpieczeństwa dla danego modułu (patrz http://csrc.nist.gov/groups/STM/cmvp/validati on.html).					✓	✓
V7.9	Zweryfikuj czy istnieje jasna polityka określająca jak są zarządzane klucze kryptograficzne (np. jak są generowane, rozprowadzane, unieważniane, w jaki sposób wygasają). Zweryfikuj czy ta polityka jest właściwie egzekwowana.					✓	✓
V7.10	Zweryfikuj czy cały kod wspierający lub wykorzystujący moduły kryptograficzne nie złośliwego zawiera kodu złośliwego.						✓



V8 - Wymagania weryfikacyjne dotyczące obsługi błędów i logowania

Wymagania weryfikacyjne dotyczące obsługi błędów i logowania definiują zestaw wymagań, które mogą być użyte do weryfikacji śledzenia zdarzeń istotnych dla bezpieczeństwa i identyfikacji zachowań ataku. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 8 - Wymagania dotyczące obsługi błędów i logowania w OWASP ASVS (V8)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V8.1	Zweryfikuj czy aplikacja nie zwraca komunikatów o błędach lub śladów stosu (stack traces), zawierających dane wrażliwe, które mogą pomóc napastnikowi, włączając w to identyfikatory sesji i dane osobowe.	√	✓	✓	✓	✓	✓
V8.2	Zweryfikuj czy wszystkie błędy po stronie serwera są obsługiwane przez serwer.			✓	✓	✓	✓
V8.3	Zweryfikuj czy wszystkie mechanizmy logowania są zaimplementowane na serwerze.			✓	✓	✓	✓
V8.4	Zweryfikuj czy logika obsługi błędów w mechanizmach bezpieczeństwa domyślnie zabrania dostępu.			✓	✓	✓	✓
V8.5	Zweryfikuj czy mechanizmy logowania zdarzeń bezpieczeństwa zapewniają możliwość logowania zdarzeń istotnych dla bezpieczeństwa zakończonych zarówno sukcesem jak i porażką.				✓	✓	✓



V8.6	Zweryfikuj czy każde zdarzenie w				
	logu zawiera: 1. znacznik czasu z wiarygodnego źródła,				
	2. poziom ważności zdarzenia,				
	3. informację czy oznaczenie że zdarzenie jest istotne dla bezpieczeństwa (jeśli różnego typu logi są wymieszane),				
	4. tożsamość użytkownika, który spowodował zdarzenie (jeśli ze zdarzeniem powiązany jest użytkownik),		√	√	✓
	5. źródłowy adres IP zapytania związanego ze zdarzeniem,				
	6. informację czy zdarzenie zakończyło się sukcesem czy porażką,				
	7. opis zdarzenia.				
V8.7	Zweryfikuj czy wszystkie zdarzenia zawierające niezaufane dane nie zostaną uruchomione jako kod w oprogramowaniu służącym do przeglądania logów.		✓	✓	✓
V8.8	Zweryfikuj czy logi bezpieczeństwa są chronione przed nieautoryzowanym dostępem i modyfikacją.		✓	√	✓
V8.9	Zweryfikuj czy aplikacja używa pojedynczej implementacji mechanizmu logowania.		✓	✓	✓
V8.10	Zweryfikuj czy aplikacja nie loguje specyficznych dla aplikacji wrażliwych danych, które mogłyby pomóc napastnikowi, włączając w to identyfikatory sesji, dane osobowe lub informacje wrażliwe.		√	✓	✓
V8.11	Zweryfikuj czy dostępne są narzędzia służące do analizy logów pozwalające analitykowi na wyszukiwanie zdarzeń odpowiadających kombinacjom kryteriów wyszukiwania, obejmujących wszystkie pola w logach w formacie wspieranym przez dany system.		~	~	✓
V8.12	Zweryfikuj czy kod implementujący lub wykorzystujący mechanizmy obsługi błędów i logowania nie złośliwego zawiera kodu złośliwego.				✓



V9 - Wymagania weryfikacyjne dotyczące ochrony danych

Wymagania weryfikacyjne dotyczące ochrony danych definiują zestaw wymagań, które mogą być użyte do weryfikacji zabezpieczania danych wrażliwych (np. numer karty kredytowej, numer paszportu, informacje umożliwiające identyfikację osób). Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 9 - Wymagania dotyczące ochrony danych w OWASP ASVS (V9)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V9.1	Zweryfikuj czy dla wszystkich formularzy zawierających dane wrażliwe wyłączone jest cachowanie po stronie klienta, włączając w to funkcjonalność automatycznego wypełniania pól.	✓	√	✓	√	√	✓
V9.2	Zweryfikuj czy zidentyfikowano listę danych wrażliwych przetwarzanych przez aplikację i istnieje jasna polityka określająca jak musi być kontrolowany dostęp do tych danych oraz kiedy te dane muszą podlegać szyfrowaniu (zarówno w trakcie przechowywania jak i przesyłania). Upewnij się, że polityka jest poprawnie egzekwowana.				√	√	✓
V9.3	Zweryfikuj czy wszystkie dane wrażliwe są przesyłane do serwera wewnątrz wiadomości HTTP (tzn. parametry URL nie są używane do przesyłania danych wrażliwych).			√		✓	✓
V9.4	Zweryfikuj czy wszystkie wrażliwe dane w cache lub kopiach tymczasowych przesyłane do klienta są chronione przed nieautoryzowanym dostępem lub czyszczone/unieważniane po uzyskaniu dostępu do danych wrażliwych przez upoważnionego użytkownika (np. są ustawiane odpowiednie nagłówki kontrolne Cache-Control: no-cache i no-store).				√	√	√
V9.5	Zweryfikuj czy wszystkie wrażliwe dane w cache lub kopiach tymczasowych przechowywane na serwerze są chronione przed nieautoryzowanym dostępem lub czyszczone/unieważniane po uzyskaniu dostępu do danych wrażliwych przez upoważnionego użytkownika.				√	✓	√



V9	.6 Zweryfikuj czy istnieje metoda usuniecia z aplikacji każdego typu				
	wrażliwych danych na końcu ich wymaganego okresu ważności.			√	√

V10 - Wymagania weryfikacyjne dotyczące bezpieczeństwa komunikacji

Wymagania weryfikacyjne dotyczące bezpieczeństwa komunikacji definiują zestaw wymagań, które mogą być użyte do weryfikacji czy cała komunikacja z aplikacją jest poprawnie zabezpieczona. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 10 - Wymagania dotyczące bezpieczeństwa komunikacji w OWASP ASVS (V10)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V10.1	Zweryfikuj czy możliwe jest zbudowanie ścieżki z zaufanego CA do każdego serwerowego certyfikatu Transport Layer Security (TLS) i każdy certyfikat serwera jest ważny.	√		√	√	√	✓
V10.2	Zweryfikuj czy nieudane połączenia TLS nie są powtórnie nawiązywane, jako połączenia niezabezpieczone.			√		✓	✓
V10.3	Zweryfikuj czy TLS używane jest dla wszystkich połączeń (zewnętrznych i backendowych), które są uwierzytelnione lub są związane z wrażliwymi danymi lub funkcjami.				√	✓	✓
V10.4	Zweryfikuj czy nieudane backendowe połączenia TLS są logowane.				✓	✓	✓
V10.5	Zweryfikuj czy ścieżki certyfikatów są budowane i weryfikowane dla wszystkich certyfikatów klienckich przy użyciu skonfigurowanych powiązań zaufania i informacji nt. unieważniania certyfikatów.				✓	✓	√
V10.6	Zweryfikuj czy wszystkie połączenia do zewnętrznych systemów, które dotyczą wrażliwych informacji lub funkcji są uwierzytelnione.				✓	✓	✓
V10.7	Zweryfikuj czy wszystkie połączenia do zewnętrznych systemów, które dotyczą wrażliwych informacji lub funkcji używają konta o minimalnych przywilejach koniecznych do właściwej pracy aplikacji.				√	√	→



V10.	2.8 Zweryfikuj czy aplikacja używa implementacji TLS opartej na pojedynczym standardzie, skonfigurowanej do działania w zatwierdzonym trybie pracy (patrz http://csrc.nist.gov/groups/STM/cm vp/documents/fips140-2/FIPS1402IG.pdf).		√	✓
V10.	9 Zweryfikuj czy dla wszystkich połączeń jest zdefiniowane kodowanie znaków (np. UTF-8).		✓	✓

V11 - Wymagania weryfikacyjne dotyczące bezpieczeństwa HTTP

Wymagania weryfikacyjne dotyczące bezpieczeństwa HTTP definiują zestaw wymagań, które mogą być użyte do weryfikacji bezpieczeństwa zapytań HTTP, odpowiedzi, sesji, cookie, nagłówków i logowania. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 11 - Wymagania dotyczące bezpieczeństwa HTTP w OWASP ASVS (V11)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V11.1	Zweryfikuj czy przekierowania nie zawierają danych, które nie podlegały walidacji.	✓	✓	✓	✓	✓	✓
V11.2	Zweryfikuj czy aplikacja akceptuje tylko zdefiniowany zestaw metod zapytań HTTP, takich jak GET i POST.	✓	✓	✓	✓	✓	✓
V11.3	Zweryfikuj czy każda odpowiedź HTTP zawiera nagłówek content type, określający zestaw bezpiecznych znaków (np. UTF-8).	√	√	✓	✓	√	✓
V11.4	Zweryfikuj czy flaga HTTPOnly jest wykorzystywana dla wszystkich cookie, do których nie jest wymagany dostęp z JavaScript.			✓	✓	✓	✓
V11.5	Zweryfikuj czy flaga secure jest wykorzystywana dla wszystkich cookie, które zawierają wrażliwe informacje, włączając w to cookie sesyjne.			√	→	√	✓
V11.6	Zweryfikuj czy nagłówki HTTP w zapytaniach i odpowiedziach zawierają wyłącznie drukowalne znaki ASCII.			✓	✓	✓	✓



V11.7	Zweryfikuj czy aplikacja generuje silne losowe tokeny będące częścią linków i formularzy powiązanych z transakcjami lub mających dostęp do				
	wrażliwych danych, a także w trakcie przetwarzania takich zapytań			✓	✓
	aplikacja sprawdza obecność tego tokena oraz poprawność jego				
	wartości odpowiednio dla bieżącego użytkownika. ¹⁵				

V12 - Wymagania weryfikacyjne dotyczące konfiguracji zabezpieczeń

Wymagania weryfikacyjne dotyczące konfiguracji zabezpieczeń definiują zestaw wymagań, które mogą być użyte do weryfikacji bezpiecznego przechowywania wszystkich informacji o konfiguracji, które kierują zachowaniem aplikacji związanym z bezpieczeństwem. Ochrona tych informacji o konfiguracji jest krytyczna dla bezpiecznego działania aplikacji. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 12 - Wymagania dotyczące konfiguracji zabezpieczeń w OWASP ASVS (V12)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V12.1	Zweryfikuj czy wszystkie informacje o konfiguracji związanej z bezpieczeństwem są przechowywane w miejscach chronionych przed nieautoryzowanym dostępem.				✓	✓	✓
V12.2	Zweryfikuj czy wszelki dostęp do aplikacji jest zablokowany w przypadku, gdy aplikacja nie może uzyskać dostępu do informacji o swojej konfiguracji zabezpieczeń.				√	√	✓
V12.3	Zweryfikuj czy wszystkie zmiany w zarządzanych przez aplikację ustawieniach konfiguracji zabezpieczeń są logowane w logu zdarzeń bezpieczeństwa.					√	✓
V12.4	Zweryfikuj czy przechowywana konfiguracja dla ułatwienia audytu może być pobrana w formacie czytelnym dla człowieka.						✓

¹⁵ Wymaganie opisuje mechanizm potrzebny do ochrony przed atakami Cross Site Request Forgery (CSRF).



V13 - Wymagania weryfikacyjne dotyczące wyszukiwania złośliwego kodu

Dla poziomu 4 wymagane jest przeszukiwanie każdego kodu nieprzeanalizowanego jeszcze po weryfikacji aplikacji na poziomie 3 pod kątem złośliwego kodu. Tabela poniżej określa wymagania dotyczące wyszukiwania złośliwego kodu, które zostały wprowadzone na poziomie 4.

Tabela 13 - Wymagania dotyczące wyszukiwania złośliwego kodu w OWASP ASVS (V13)

	Wymaganie werfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V13.1	Zweryfikuj czy w kodzie, który został napisany lub zmodyfikowany w celu stworzenia aplikacji nie ma złośliwego kodu. ¹⁶						✓
V13.2	Zweryfikuj czy integralność interpretowanego kodu, bibliotek, plików wykonywalnych i plików konfiguracyjnych jest weryfikowana przy użyciu sum kontrolnych lub hashy.						✓

¹⁶ Przykładowo: zbadaj wywołania do zegara systemowego pod kątem wyszukiwania bomb czasowych(time bomb), funkcje nie związane z wymaganiami biznesowymi pod kątem wyszukiwania tylnych wejść (back door), ścieżki uruchomieniowe pod kątem wyszukiwania jaj wielkanocnych (Easter egg), operacje finansowe pod kątem wyszukiwania błędnej logiki, która może wskazywać na atak salami, inne rodzaje złośliwego kodu.



V14 - Wymagania weryfikacyjne dotyczące bezpieczeństwa wewnętrznego

Wymagania weryfikacyjne dotyczące bezpieczeństwa wewnętrznego definiują zestaw wymagań, które mogą być użyte do weryfikacji czy aplikacja dodatkowo chroni samą siebie przed błędami implementacyjnymi. Tabela poniżej określa wymagania odpowiednie dla każdego z czterech poziomów weryfikacji.

Tabela 14 - Wymagania dotyczące bezpieczeństwa wewnętrznego w OWASP ASVS (V14)

	Wymaganie weryfikacyjne	Poziom 1A	Poziom 1B	Poziom 2A	Poziom 2B	Poziom 3	Poziom 4
V14.1	Zweryfikuj czy aplikacja chroni przed nieautoryzowanym dostępem lub modyfikacją atrybuty użytkownika i danych oraz informacje o polityce używane przez mechanizmy kontroli dostępu.					~	✓
V14.2	Zweryfikuj czy interfejsy mechanizmów bezpieczeństwa są wystarczająco proste w użyciu by były prawidłowo używane przez developerów.						✓
V14.3	Zweryfikuj czy aplikacja odpowiednio chroni przed nieodpowiednim jednoczesnym dostępem współdzielone zmienne i zasoby.						✓

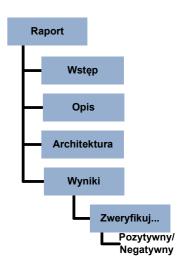


Wymagania dotyczące raportowania weryfikacji

Raport OWASP ASVS zawiera opis aplikacji, która została przeanalizowana względem wymagań OWASP ASVS dla danego poziomu. Raport dokumentuje również wyniki analizy, w tym wymagane poprawki dla podatności.

Wymagania raportowe ASVS określają rodzaj informacji, których obecność jest wymagana w raporcie. Wymagania raportowe ASVS nie określają struktury, organizacji ani formatu raportu. Wymagania raportowe ASVS nie wykluczają możliwości zamieszczania w raporcie dodatkowych informacji.

Rodzaj informacji wymaganych przez każdy z zestawów wymagań raportowych ASVS może być nazywany, formatowany i organizowany zgodnie z wymaganiami weryfikującego. Wymagania raportowe ASVS są spełnione, jeśli obecne są wszystkie wymagane informacje. Raport powinien zawierać wszelkie materiały niezbędne czytelnikowi do zrozumienia przeprowadzonej analizy i jej wyników, włączając w to informacje o konfiguracji oraz fragmenty kodu jak przedstawiono na przyległym rysunku, który może być używany przy tworzeniu zarysu raportu.



Rysunek 11 - Zawartość raportu

R1 - Wstęp do raportu

- R1.1 Wstęp do raportu powinien dostarczać informacji wystarczających do identyfikacji zarówno raportu, jak i aplikacji, która jest tematem raportu.
- R1.2 Wstęp do raportu powinien podsumowywać całkowite zaufanie do bezpieczeństwa aplikacji.
- R1.3 Wstęp do raportu powinien identyfikować kluczowe ryzyka biznesowe związane z działaniem aplikacji.
- R1.4 Wstęp do raportu powinien opisywać zasady zaangażowania związane z wykonywaniem weryfikacji lub to, co mogło ograniczać zakres weryfikacji.

R2 - Opis Aplikacji

R2.1 Sekcja opis aplikacji powinna dostarczyć odpowiedniego opisu aplikacji pomagającego w zrozumieniu jej działania i środowiska, w którym działa.

R3 - Architektura bezpieczeństwa aplikacji

R3.1 Sekcja architektura bezpieczeństwa aplikacji powinna dostarczać dodatkowe szczegóły opisujące aplikację, stanowiąc pierwszy krok w zapewnieniu czytelnikowi raportu pewności, że wykonana analiza jest kompletna i dokładna. Ta część raportu dostarcza kontekst dla analizy. Informacje przedstawione w tej sekcji będą służyć w trakcie analizy do identyfikacji niespójności. Ta część raportu powinna zawierać detale o szczegółowości zależnej od poziomu OWASP ASVS, zgodnie z którym została przeprowadzona analiza. Szczegóły będą różnić się w zależności od poziomu (ASVS).



R4 - Wyniki weryfikacji

R4.1 Te wyniki weryfikacji powinny przedstawiać rezultat analizy przeprowadzonej zgodnie z sekcją "Wymagania weryfikacyjne" niniejszego standardu, włącznie z opisem wymaganych poprawek dla podatności, w następujący sposób:

Tabela 15 - Zawartość wyników weryfikacji w raporcie OWASP ASVS

Poziom	Pozytywny	Negatywny
Wyniki poziomu 1	 Werdykt. Konfiguracja narzędzia (jeśli narzędzie może służyć sprawdzeniu) lub uzasadnienie werdyktu (argument przemawiający za kompletnością i poprawnością, zapewniający specyficzny dowód). Mapowanie możliwości narzędzi automatycznych na dające się zastosować szczegółowe wymagania weryfikacyjne. Opis konfiguracji narzędzia i mapowanie możliwości narzędzia muszą być dostarczone jako część raportu tylko jednorazowo. 	 Werdykt Lokalizacja (URL z parametrami i/lub ścieżką do kodu źródłowego, nazwa i numer(y) linii) Opis (uwzględniający informacje konfiguracyjne tam, gdzie jest to odpowiednie) Ocena ryzyka ¹⁷ Uzasadnienie ryzyka
	Opis konfiguracji narzędzia oraz mapo również dostarczone jako część raport	owanie możliwości narzędzia muszą być tu.
Wyniki poziomów	• Werdykt	Werdykt
2 - 4	 Uzasadnienie werdyktu (argument przemawiające za kompletnością i poprawnością, zapewniający specyficzny dowód) 	 Lokalizacja (URL z parametrami i/lub ścieżką do kodu źródłowego, nazwa i numer(y) linii)
		 Opis (uwzględniający ścieżkę komponentów aplikacji oraz kroki potrzebne do odtworzenia)
		Ocena ryzyka (patrz OWASP Risk Rating Methodology)
		Uzasadnienie ryzyka

¹⁷ Więcej informacji na temat identyfikacji i szacowania ryzyk związanych z podatnościami można znaleźć w *Testing Guide* (OWASP, 2008).



Słowniczek

Application Security Verification Standard (ASVS) - standard OWASP definiujący cztery poziomy weryfikacji bezpieczeństwa aplikacji.

Architektura bezpieczeństwa - abstrakcyjna reprezentacja projektu aplikacji, która identyfikuje i opisuje gdzie i jak są użyte mechanizmy bezpieczeństwa oraz lokalizację i wrażliwość danych użytkownika oraz aplikacji.

Ataki DoS - zalewanie aplikacji większą liczbą zapytań, niż ta jest w stanie obsłużyć.

Atak salami - rodzaj złośliwego kodu, który jest wykorzystywany do przekierowywania małych sum pieniędzy niezauważalnych w transakcjach finansowych.

Bezpieczeństwo aplikacji - bezpieczeństwo na poziomie aplikacji skupia się na analizie komponentów składających się na warstwę aplikacji modelu OSI, nie skupiając się np. na systemie operacyjnym poniżej lub podłaczonych sieciach.

Bezpieczeństwo komunikacji - ochrona danych aplikacji podczas transmisji pomiędzy komponentami aplikacji, pomiędzy klientami i serwerami oraz pomiędzy systemami zewnętrznymi a aplikacją.

Biała lista - lista dozwolonych danych lub operacji, na przykład lista znaków, które są dozwolone w celu sprawdzania poprawności danych wejściowych.

Bomba czasowa - rodzaj złośliwego kodu, który nie uruchamia się przed zaprogramowanym czasem lub datą.

Cel weryfikacji (TOV) - jeśli wykonujesz weryfikację bezpieczeństwa aplikacji zgodnie z wymaganiami OWASP ASVS, weryfikacja ta będzie dotyczyć określonej aplikacji. Ta aplikacja jest nazywana celem weryfikacji (TOV).

Common Criteria (CC) - wieloczęściowy standard, który może być używany, jako podstawa do weryfikacji projektu i wdrożenia mechanizmów bezpieczeństwa w produktach IT.

Czarna lista - lista danych lub operacji, które nie są dozwolone - na przykład lista znaków, które nie są dozwolone jako dane wejściowe.

FIPS 140-2 - standard, który może być używany jako podstawa weryfikacji projektu i implementacji modułów kryptograficznych.

Jaja wielkanocne - rodzaj złośliwego kodu uruchamianego poprzez specyficzne dane wejściowe użytkownika.

Kontrola dostępu - środki ograniczające dostęp do plików, wskazanych funkcji, adresów URL i danych w oparciu o tożsamość użytkowników i/lub grup, do których należą.

Kod złośliwy - kod włączony do aplikacji podczas jej tworzenia, bez wiedzy właściciela aplikacji, który omija przyjętą politykę bezpieczeństwa aplikacji. Nie mylić z programami złośliwymi (malware), takimi jak wirus czy robak!

Komponent aplikacji - pojedynczy lub grupa plików źródłowych, bibliotek i/lub plików wykonywalnych, zdefiniowane dla danej aplikacji przez weryfikującego.

Konfiguracja zabezpieczeń - konfiguracja uruchomieniowa aplikacji wpływająca na sposób użycia mechanizmów bezpieczeństwa.

Mechanizm bezpieczeństwa - funkcja lub komponent dokonujący sprawdzenia bezpieczeństwa (np. kontroli dostępu) lub w przypadku wywołania wpływająca na działanie zabezpieczeń (np. generowanie wpisu audytowego).

Modelowanie zagrożeń - technika polegająca na tworzeniu coraz bardziej wyrafinowanych architektur bezpieczeństwa w celu identyfikacji czynników stwarzających zagrożenia, stref bezpieczeństwa, mechanizmów bezpieczeństwa oraz ważnych aktywów technicznych i biznesowych.



Moduł kryptograficzny - sprzęt, oprogramowanie i/lub firmware implementujące algorytmy kryptograficzne i/lub generujące klucze kryptograficzne.

Open Web Application Security Project (OWASP) - Open Web Application Security Project (OWASP) jest darmową i otwartą społecznością o zasięgu światowym, skupioną na poprawie bezpieczeństwa aplikacji. Naszą misją jest sprawianie, że bezpieczeństwo aplikacji jest "widoczne" tak, aby ludzie i organizacje mogły podejmować świadome decyzje odnośnie ryzyk związanych z bezpieczeństwem aplikacji. Zobacz http://www.owasp.org/

OWASP Enterprise Security API (ESAPI) - darmowa i otwarta kolekcja wszystkich metod bezpieczeństwa potrzebnych developerom do tworzenia bezpiecznych aplikacji Web. Zobacz http://www.owasp.org/index.php/ESAPI

OWASP Testing Guide - dokument mający na celu pomóc organizacjom zrozumieć, z czego składa się program testowania oraz pomóc im ustalić kroki potrzebne do stworzenia i uruchomienia tego programu testów. Zobacz http://www.owasp.org/index.php/Category:OWASP_Testing_Project

OWASP Top Ten - dokument reprezentujący szeroki konsensus w kwestii najbardziej krytycznych błędów bezpieczeństwa aplikacji. Zobacz http://www.owasp.org/index.php/Top10

OWASP Risk Rating Methodology - metodyka oceny ryzyka dostosowana do bezpieczeństwa aplikacji. Zobacz http://www.owasp.org/index.php/How_to_value_the_real_risk

Pozytywny - patrz biała lista.

Program złośliwy - kod uruchamialny włączony do aplikacji podczas jej pracy bez wiedzy użytkownika aplikacji lub administratora.

Raport z weryfikacji bezpieczeństwa aplikacji - raport napisany dla danej aplikacji przez weryfikującego, dokumentujący całościowe wyniki i wspomagające je badania.

System zewnętrzny - aplikacja serwerowa lub usługa, które nie są częścią aplikacji.

Tylne wejścia - rodzaj złośliwego kodu umożliwiającego nieuprawniony dostęp do aplikacji.

Uwierzytelnienie - weryfikacja tożsamości deklarowanej przez użytkownika aplikacji.

Walidacja wejścia - normalizacja i walidacja niezaufanych danych wejściowych użytkownika.

Walidacja wyjścia - normalizacja i walidacja danych wyjściowych z aplikacji do przeglądarek Web i systemów zewnętrznych.

Weryfikacja automatyczna - użycie automatycznych narzędzi (narzędzi do analizy dynamicznej, analizy statycznej lub obu typów) wykorzystujących wzorce podatności do wyszukiwania problemów.

Weryfikacja bezpieczeństwa aplikacji - techniczna ocena zgodności aplikacji z OWASP ASVS

Weryfikacja dynamiczna - użycie automatycznych narzędzi wykorzystujących wzorce podatności do zlokalizowania problemów podczas pracy aplikacji.

Weryfikacja projektu - techniczna ocena architektury bezpieczeństwa aplikacji.

Weryfikacja statyczna - użycie automatycznych narzędzi wykorzystujących wzorce podatności do zlokalizowania problemów w kodzie źródłowym aplikacji.

Weryfikacja wewnętrzna - techniczna ocena specyficznych aspektów architektury bezpieczeństwa aplikacji zgodnie z OWASP ASVS.

Weryfikujący - osoba lub zespół wykonujący przegląd zgodności aplikacji z wymaganiami OWASP ASVS.



Gdzie jeszcze możesz się udać?

OWASP jest główną stroną dotyczącą bezpieczeństwa aplikacji Web. Na stronach OWASP znajduje się wiele projektów, for, blogów, prezentacji, narzędzi i dokumentów. Dodatkowo OWASP jest organizatorem dwóch w roku głównych konferencji dotyczących bezpieczeństwa aplikacji Web i posiada 80 lokalnych oddziałów. Stronę projektu OWASP ASVS możesz znaleźć: http://www.owasp.org/index.php/ASVS

Następujące projekty OWASP najprawdopodobniej będą użyteczne dla użytkowników niniejszego standardu:

- 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

Podobnie następujące strony Web najprawdopodobniej będą użyteczne dla użytkowników niniejszego standardu:

- OWASP http://www.owasp.org
- MITRE Common Weakness Enumeration Trendy podatności,
- PCI Security Standards Council wydawcy standardów PCI, odpowiednich dla wszystkich organizacji przetwarzających lub przechowujących dane kart kredytowych, 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.



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.