

OWASP Application Security Verification Standard 2009

Web Application Standard
 Versi Bahasa Indonesia





Kata Pembuka

Dokumen ini mendefinisikan empat level verifikasi keamanan untuk aplikasi web. Keamanan di tingkat aplikasi berfokus pada analisa komponen yang berada di lapisan aplikasi (Application Layer) seperti yang digambarkan di Open Systems Interconnection Reference Model (OSI Model), daripada berfokus pada misalnya sistem operasi atau instalasi jaringan yang mendasari aplikasi tersebut. Setiap level yang dijelaskan dalam dokumen ini terdiri dari satu set persyaratan untuk memverifikasi efektivitas kontrol keamanan yang melindungi aplikasi web.

Persyaratan tersebut dikembangkan dengan tujuan-tujuan berikut:

- Untuk digunakan sebagai metrik (alat ukur) Menyediakan tongkat ukur bagi pengembang aplikasi dan pemilik aplikasi untuk menilai tingkat kepercayaan yang dapat diberikan kepada aplikasi Web mereka,
- Untuk digunakan sebagai pedoman Memberikan bimbingan untuk para pengembang tentang kontrol keamanan seperti apa saja yang harus dibangun dalam rangka untuk memenuhi persyaratan aplikasi keamanan,¹ dan
- *Untuk digunakan selama pengadaan* Menyediakan dasar untuk menetapkan persyaratan verifikasi keamanan aplikasi yang harus tertulis di dalam kontrak pengadaan.²

Persyaratan dirancang untuk memenuhi tujuan di atas dengan memastikan validasi akan bagaimana kontrol keamanan dirancang, diimplementasikan, dan digunakan oleh sebuah aplikasi. Persyaratan tersebut memastikan bahwa kontrol keamanan yang digunakan oleh aplikasi beroperasi dengan menggunakan strategi *deny-by-default*, terpusat, dilakukan di server, dan semuanya berjalan dengan baik saat diperlukan.

Lisensi dan Hak Cipta

Copyright © 2008 - 2009 The OWASP Foundation.



Dokumen ini dirilis di bawah lisensi Creative Commons Attribution ShareAlike 3.0. Untuk penggunaan atau distribusi, Anda harus membuat jelas kepada orang lain akan persyaratan lisensi dari karya tulis ini.

¹ Untuk informasi lebih lanjut tentang membangun dan menggunakan kontrol keamanan yang memenuhi persyaratan ASVS, lihat *Enterprise Security API (ESAPI)* (OWASP, 2009).

² Untuk informasi lebih lanjut tentang menggunakan ASVS dalam kontrak, lihat *Kontrak Lampiran* (OWASP, 2009).



Daftar Isi

Perkenalan	. 1
Pendekatan	. 1
Penghargaan	. 3
Level Verifikasi Keamanan Aplikasi	. 4
Level 1 - Verifikasi Otomatis Level 1A - Pemindaian Dinamis (Otomatisasi Verifikasi Parsial). Level 1B - Pemindaian Kode Sumber (Verifikasi Terotomatisasi Parsial) Level 2 - Verifikasi Manual Level 2A - Uji Keamanan (Verifikasi Manual Parsial) Level 2B - Code Review (Verifikasi Manual Parsial) Level 3 - Verifikasi Desain Tingkat 4 - Verifikasi Internal Interpretasi Persyaratan dan Preseden	. 7 . 7 . 7 10 11 11
Persyaratan Verifikasi Terperinci	17
V1 - Persyaratan Dokumentasi Arsitektur Keamanan. V2 - Persyaratan Verifikasi Otentikasi V3 - Persyaratan Verifikasi Manajemen Sesi V4 - Persyaratan Verifikasi Kontrol Akses V5 - Persyaratan Verifikasi Validasi Input V6 - Persyaratan Verifikasi Pengkodean Keluaran V7 - Persyaratan Verifikasi Kriptografi. V8 - Persyaratan Verifikasi Penanganan dan Pencatatan Error V9 - Persyaratan Verifikasi Proteksi Data V10 - Persyaratan Verifikasi Keamanan Komunikasi V11 - Persyaratan Verifikasi Keamanan HTTP V12 - Persyaratan Verifikasi Konfigurasi Keamanan V13 - Persyaratan Verifikasi Pencarian Kode Berbahaya V14 - Persyaratan Verifikasi Keamanan Internal	19 20 21 23 24 25 26 28 29 30 31 32
Persyaratan Pelaporan Hasil Verifikasi	34
R1 - Pembukaan Laporan R2 - Deskripsi Aplikasi R3 - Arsitektur Keamanan Aplikasi R4 - Hasil Verifikasi	34 34
Glosarium	36
Dari Sini Ke Mana ?	38



	-
Gambi	- 11

Gambar 1 - Level ASVS OWASP	1
Gambar 2 - Salah satu cara untuk menyisipkan proses verifikasi sebagai aktivitas dalam SDLC And	a . 2
Gambar 3 - OWASP ASVS Level 1, 1A, dan 1B	5
Gambar 4 - Contoh Arsitektur Kemanan OWASP ASVS Level 1	
Gambar 5 - OWASP ASVS Level 2, 2A, and 2B	8
Gambar 6 - Contoh Arsitektur Kemanaan OWASP ASVS Level 2	10
Gambar 7 - OWASP ASVS Level 3	11
Gambar 8 - Contoh Arsitektur Keamanan OWASP ASVS Level 3	13
Gambar 9 - OWASP ASVS Level 4	14
Gambar 10 - Contoh Kode Aplikasi yang belum Diperiksa di OWASP ASVS Level 4	16
Gambar 11 - Isi dari Laporan	

Tabel

Tabel 1 - Persyaratan Arsitektur Keamanan OWASP (V1)	18
Tabel 2 - Persyaratan Keamanan Otentikasi OWASP (V2)	19
Tabel 3 - Persyaratan Keamanan Manajemen Sesi OWASP (V3)	20
Tabel 4 - Persyaratan Kontrol Akses OWASP (V4)	22
Tabel 5 - Persyaratan Validasi Masukan OWASP (V5)	23
Tabel 6 - Persyaratan Verifikasi Pengkodean Keluaran OWASP (V6)	
Tabel 7 - Persyaratan Verifikasi Kriptografi OWASP (V7)	25
Tabel 8 - Persyaratan Verifikasi Penanganan Error OWASP (V8)	26
Tabel 9 - Persyaratan Verifikasi Perlindungan Data (V9)	
Tabel 10 - Persyaratan Keamanan Komunikasi (V10)	29
Tabel 11 - Persyaratan Verifikasi Keamanan HTTP OWASP ASVS (V11)	30
Tabel 12 - Persyaratan Konfigurasi Keamanan (V12)	31
Tabel 13 - Persyaratan Verifikasi Pencarian Kode Berbahaya OWASP ASVS (V13)	
Tabel 14 - Persyaratan Keamanan Internal OWASP (V14)	33
Tabel 15 - Isi dari Laporan Hasil Verifikasi OWASP ASVS	



Perkenalan

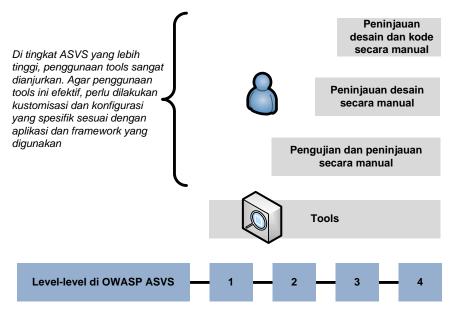
Open Web Application Security Project (OWASP) adalah sebuah komunitas terbuka yang bertujuan untuk memberdayakan organisasi untuk mengembangkan, membeli, dan memelihara aplikasi yang dapat dipercaya keamanannya. Semua *tool*, dokumen, forum, dan bab yang berasal dari OWASP adalah bebas dan terbuka untuk semua orang yang tertarik memperbaiki keamanan aplikasi. Pendekatan yang kami lakukan adalah mendefinisikan keamanan aplikasi sebagai manusia, proses, dan teknologi (*people*, *process*, *and technology*), karena perbaikan di wilayah-wilayah tersebut menghasilkan peningkatan paling efektif terhadap keamanan aplikasi. Kunjungi kami di www.owasp.org.

OWASP adalah organisasi jenis baru. Kami bebas dari tekanan komersil, sehingga dapat menghasilkan informasi tentang keamanan aplikasi yang tidak bias, praktis, dan relatif murah. OWASP tidak memiliki hubungan dengan perusahaan teknologi apapun, namun demikian kami mendukung penggunaan teknologi keamanan komersil selama itu tidak asal beli. Mirip dengan proyek-proyek *software open-source*, OWASP memproduksi berbagai jenis materi dengan cara yang terbuka dan kolaboratif. OWASP Foundation adalah entitas nirlaba yang menjamin keberhasilan jangka panjang proyek ini.

Tujuan utama dari Proyek Standar Verifikasi Keamanan Aplikasi OWASP (ASVS) adalah untuk menormalkan rentang cakupan dan tingkat kekakuan yang tersedia di pasar soal verifikasi keamanan aplikasi Web menggunakan standar terbuka yang tetap bisa diterapkan secara komersial. Standar ini memberikan dasar untuk pengujian kontrol keamanan aplikasi secara teknis, serta setiap kontrol keamanan teknis di lingkungan, yang diandalkan untuk melindungi terhadap kerentanan seperti *Cross-Site Scripting* (XSS) dan *SQL injection*. Standar ini dapat digunakan untuk membangun tingkat kepercayaan terhadap keamanan aplikasi Web.

Pendekatan

OWASP ASVS mendefinisikan persyaratan verifikasi dan dokumentasi yang dikelompokkan berdasarkan cakupan yang terkait dan tingkat keketatannya. Standar ini mendefinisikan empat tingkat yang hierarkis (misalnya Level 2 mempunyai cakupan dan ketetatan yang lebih dari Level 1) seperti yang digambarkan dalam gambar di bawah ini.



Gambar 1 - Level ASVS OWASP

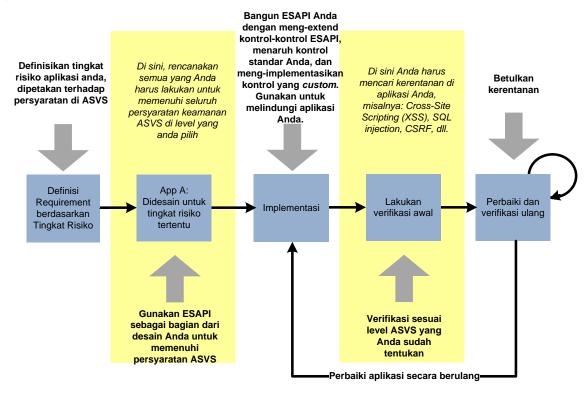
Verifikasi keamanan Aplikasi Web dilakukan dari sudut pandang logika dengan mengikuti (atau mencoba untuk mengikuti) jalan menuju dan keluar dari aplikasi yang ditargetkan (disebut sebagai Target Verifikasi



atau TOV) dan melakukan analisis di sepanjang jalan. Aplikasi yang lebih kompleks biasanya memerlukan lebih banyak waktu untuk dianalisis dan mengakibatkan proses verifikasi lebih lama dan lebih mahal. Jumlah baris kode bukan satu-satunya faktor yang menentukan kompleksitas aplikasi - berbagai teknologi biasanya akan membutuhkan jumlah analisis yang berbeda. Aplikasi sederhana contohnya adalah *library* dan *framework*. Aplikasi dengan kompleksitas moderat contohnya adalah aplikasi Web 1.0 yang sederhana. Aplikasi yang kompleks contohnya aplikasi Web 2.0 dan teknologi Web yang baru/unik.

ASVS mendefinisikan komponen konstituen untuk Level 1 dan 2 (misalnya verifikasi di Level 1 membutuhkan pencapaian kedua persyaratan Level 1A dan 1B). Sebagai contoh,sebuah aplikasi dapat mengklaim kepatuhan baik kepada 1A atau 1B, tetapi membuat klaim seperti itu adalah lebih lemah dari mengklaim kepatuhan kepada Level 1. Persyaratan verifikasi dan dokumentasi ditentukan dalam Standar ini dalam tiga jenis persyaratan: persyaratan tingkat-tinggi (high-level), persyaratan terperinci, dan persyaratan pelaporan. Persyaratan tingkat-tinggi mendefinisikan implementasi aplikasi secara keseluruhan. Persyaratan terperinci mendefinisikan implementasi tingkat-rendah (low-level) aplikasi dan persyaratan verifikasi (yaitu, menjabarkan hal-hal yang harus diperiksa). Persyaratan pelaporan menentukan bagaimana cara mendokumentasikan hasil verifikasi terhadap sebuah aplikasi sesuai dengan ASVS OWASP.

OWASP menyediakan banyak sumber daya, termasuk ASVS, untuk membantu organisasi-organisasi mengembangkan dan merawat aplikasi yang aman. OWASP ASVS, OWASP Contract Annex,³ dan OWASP ESAPI⁴ dapat digunakan untuk mendukung siklus pengembangan piranti lunak (SDLC) seperti digambarkan pada gambar di bawah.



Gambar 2 - Salah satu cara untuk menyisipkan proses verifikasi sebagai aktivitas dalam SDLC Anda⁵

³ Untuk informasi mengenai bagaimana untuk menyinggung tingkat ASVS dalam kontrak, lihat *OWASP Contract Annex*.

⁴ Untuk informasi lebih lanjut, lihat proyek OWASP *ESAPI* (OWASP 2009).

⁵ For more information about introducing security-related activities into your existing SDLC, see the *OWASP CLASP* (OWASP 2008) or *OWASP SAMM* Projects (OWASP 2009).



Penghargaan

Terima kasih kepada OWASP Foundation dalam mensponsori Projek OWASP Application Security Verification Standard di OWASP Summer of Code 2008.

Pimpinan Proyek: ⁶ Mike Boberski (Booz Allen Hamilton)

Penulis: ⁷ Mike Boberski (Booz Allen Hamilton), Jeff Williams

(Aspect Security), Dave Wichers (Aspect Security)

Disponsori

oleh:





Booz | Allen | Hamilton

Pengakuan dan penghargaan diberikan kepada: Pierre Parrend, yang menjadi *reviewer* di OWASP Summer of Code 2008 Reviewer; 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); dan akhirnya, banyak terima kasih untuk komunitas verifikasi keamanan aplikasi untuk saran-saran dan antusiasme mereka.

Penerjemah versi Bahasa Indonesia: Reggio Nurtanio Hartono, Charles Lim, Irwan Prasetyo, Hogan Kusnadi. Penyunting akhir: Zaki Akhmad.

⁶ Email: mike.boberski@owasp.org

Halaman 3

⁷ Email: jeff.williams@owasp.org, dave.wichers@owasp.org



Level Verifikasi Keamanan Aplikasi

ASVS mendefinisikan empat tingkat verifikasi yang meningkat baik luas dan kedalamannya seiring dengan bergeraknya TOV naik tingkat. Luas didefinisikan di masing-masing tingkat dengan satu set persyaratan keamanan yang harus diatasi. Kedalaman verifikasi didefinisikan dalam pendekatan dan tingkat keketatan yang diperlukan dalam memverifikasi setiap persyaratan keamanan. Tools yang dipakai merupakan bagian penting dari setiap tingkat ASVS. Pada tingkat yang lebih tinggi di ASVS, penggunaan *tools* dianjurkan. Tapi untuk menjadi efektif, alat harus sangat disesuaikan dan dikonfigurasi untuk aplikasi dan framework yang digunakan. Dan, di semua tingkatan, hasil verifikasi menggunakan tools harus diverifikasi secara manual.

Adalah tanggung jawab pemeriksa untuk menentukan apakah TOV memenuhi semua persyaratan di tingkat yang ditargetkan. Jika aplikasi memenuhi semua persyaratan untuk tingkat itu, maka dapat dianggap sebagai aplikasi berperingkat OWASP ASVS Level-N, dimana N adalah level verifikasi yang dipenuhi aplikasi. Jika aplikasi tidak memenuhi semua persyaratan untuk tingkat tertentu, tetapi memenuhi semua persyaratan untuk tingkat yang lebih rendah menurut Standar ini, maka dapat dianggap telah lulus verifikasi tingkat yang lebih rendah tersebut. Standar ini menggunakan istilah 'verifier' untuk menunjukkan orang atau tim yang meninjau penerapan aplikasi terhadap persyaratan ini.

Spesifikasi untuk aplikasi tertentu membutuhkan OWASP ASVS Level-N, tetapi juga dapat memasukkan persyaratan rinci tambahan dari tingkat ASVS yang lebih tinggi. Sebagai contoh, sebuah organisasi keuangan mungkin memiliki beberapa aplikasi yang berisiko lebih rendah dan cukup diverifikasi untuk OWASP ASVS Level-2 tetapi juga ingin memastikan bahwa tidak ada kode berbahaya (lihat V13, Level-4 saja) yang dimasukkan. Organisasi atau bisnis lainnya bisa menerapkan saja persyaratan tambahan, seperti kepatuhan terhadap kebijakan keamanan informasi tertentu dan peraturan-peraturan pemerintah tertentu.

Tidak ada verifikasi tingkat 0. Juga, untuk mendapatkan pengakuan, kerentanan harus diperbaiki(atau dimitigasikan), dan aplikasi diperiksa ulang.

Level 1 - Verifikasi Otomatis

Level 1 ("Verifikasi Otomatis") biasanya cocok untuk aplikasi di mana diperlukan kepercayaan pada penggunaan kendali keamanan (security controls) yang benar. Ancaman terhadap keamanan biasanya berbentuk virus dan worm, yang korbannya pada umumnya terungkap melalui pemindaian (scanning) tanpa pandang bulu dan berdampak serius pada aplikasi yang paling rawan (vulnerable). Ruang lingkup verifikasi termasuk kode yang dikembangkan atau dimodifikasi dalam rangka untuk menciptakan aplikasi.

Pada Level 1, verifikasi melibatkan penggunaan *tools* otomatis yang ditunjang dengan verifikasi manual. Level ini hanya memberikan verifikasi parsial atas keamanan aplikasi. Verifikasi manual di sini tidak dimaksudkan untuk melengkapi verifikasi keamanan aplikasi di tingkat ini, tapi hanya untuk memastikan bahwa setiap temuan otomatis adalah benar dan bukan laporan palsu.

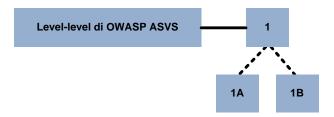
Ada dua komponen konstituen untuk Level 1 ini. Level 1A adalah untuk penggunaan *tools* pemindai kerawanan otomatis (analisa dinamis), dan Level 1B adalah untuk penggunaan *tools* yang memindai kode sumber secara otomatis (analisa statis). Upaya verifikasi dapat menggunakan salah satu dari komponen individu, atau dapat melakukan kombinasi dari pendekatan-pendekatan untuk mencapai kelengkapan peringkat Level 1. Struktur level-level ini digambarkan sebagai berikut.

Meskipun dapat ditentukan bahwa sebuah aplikasi telah memenuhi baik Level 1A atau 1B, namun jika baru salah satu yang terpenuhi, aplikasi belum bisa disebut mencapai Level 1. Dengan demikian, sebuah aplikasi yang memenuhi Level 1 harus memenuhi persyaratan Level 1A <u>dan</u> 1B.

Halaman 4

⁸ Untuk informasi lebih lanjut tentang mengidentifikasi risiko dan memperkirakan risiko yang terkait dengan kerentanan, lihat *OWASP Testing Guide* (OWASP, 2008).





Gambar 3 - OWASP ASVS Level 1, 1A, dan 1B

Berikut ini adalah persyaratan tingkat-tinggi minimal untuk aplikasi Level 1, 1A, 1B:

Lingkup Verifikasi

L1.1 Ruang lingkup verifikasi mencakup semua kode yang dikembangkan atau dimodifikasi dalam rangka untuk menciptakan aplikasi.

Persyaratan Perilaku Kontrol Keamanan

Tidak ada Tidak ada persyaratan untuk bagaimana aplikasi kontrol keamanan membuat keputusan di Level 1.

Persyaratan Penggunaan Kontrol Keamanan

Tidak ada Tidak ada persyaratan untuk mana aplikasi kontrol keamanan yang digunakan dalam aplikasi di Level 1.

Persyaratan Implementasi Kontrol Keamanan

Tidak ada Tidak ada persyaratan untuk bagaimana kontrol keamanan aplikasi yang dibangun di Level 1.

Persyaratan Verifikasi Kontrol Keamanan

- L1.2 Memindai aplikasi secara dinamis sesuai dengan persyaratan Level 1A di bagian "Persyaratan Verifikasi Terperinci".
- L1.3 Lakukan pemindaian kode sumber aplikasi sesuai dengan persyaratan Level 1B di bagian "Persyaratan Verifikasi Terperinci".

Persyaratan di Level 1 yang memungkinkan penggunaan kedua teknik verifikasi (antara manual dan otomatis) cukup diverifikasi dengan satu teknik. Selain itu, jika alat uji yang dipilih penguji tidak memiliki kemampuan untuk memverifikasi persyaratan verifikasi yang ditentukan, verifikasi dapat melakukan secara manual untuk mengisi kesenjangan ini. ⁹ 10

⁹ Untuk informasi lebih lanjut tentang melakukan verifikasi manual oleh melakukan pengujian penetrasi manual, lihat *OWASP Panduan Pengujian* (OWASP, 2008).

Halaman 5

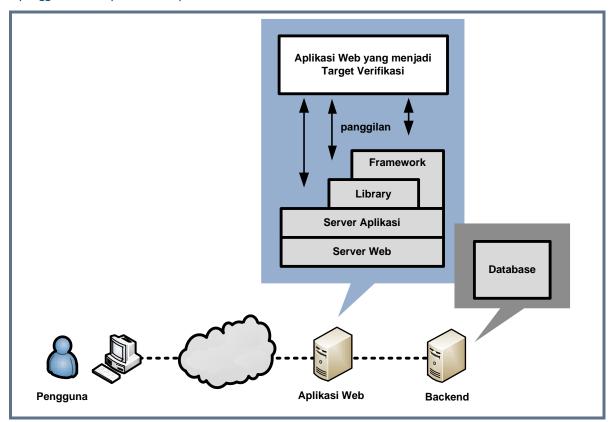
¹⁰ Untuk informasi lebih lanjut tentang melakukan verifikasi manual dengan melakukan review kode manual, lihat *Kode OWASP Panduan Tinjauan* (OWASP, 2008).



Persyaratan Pelaporan

L1.4 Membuat laporan verifikasi yang menjelaskan rincian arsitektur keamanan aplikasi dengan daftar komponen-komponennya, dan termasuk hasil verifikasi sesuai dengan persyaratan di bagian "Persyaratan Pelaporan Verifikasi".

Di Level 1, komponen aplikasi dapat didefinisikan dari segi baik kelompok atau satuan file kode sumber, perpustakaan (libraries), dan/atau hasil kompilasi (executables), seperti yang digambarkan dalam gambar di bawah ini. Di Level 1, daftar tidak perlu diurutkan atau terorganisir secara rumit selain daripada mengidentifikasi komponen-komponen yang merupakan bagian dari aplikasi, dan mana yang merupakan bagian dari lingkungan TI institusi dimana aplikasi itu berada. Aplikasi kemudian dapat diperlakukan sebagai kumpulan komponen dalam entitas tunggal yang monolitik. Path yang tertulis di *request* (URL) pengguna akhir aplikasi tidak perlu diidentifikasi dan didokumentasikan.



Gambar 4 - Contoh Arsitektur Kemanan OWASP ASVS Level 1



Level 1A - Pemindaian Dinamis (Otomatisasi Verifikasi Parsial)

Persyaratan Verifikasi Kontrol Keamanan dengan Pemindaian Dinamis

Pemindaian dinamis (juga dikenal sebagai "pemindaian kerentanan aplikasi ") terdiri dari menggunakan *tools* otomatis untuk mengakses interface aplikasi, selagi aplikasi sedang berjalan, untuk mendeteksi kerentanan dalam kontrol keamanan aplikasi. Harap perhatikan bahwa ini tidak cukup untuk memverifikasi desain yang tepat, implementasi, dan penggunaan kontrol keamanan, tetapi sudah cukup diterima di Level 1. Ruang lingkup verifikasi didefinisikan oleh persyaratan arsitektur keamanan ini.

- L1A.1 Memindai aplikasi secara dinamis sesuai persyaratan di Level 1A yang terdefinisi dalam bagian "Persyaratan Verifikasi Terperinci".
- L1A.2 Memverifikasi hasil pemindaian dinamis, entah secara manual melalui uji penetrasi atau pengecekan ulang kode sumber. Hasil verifikasi terotomatis yang tidak diuji ulang secara manual tidak dianggap memenuhi persyaratan Level 1.

Kerentanan yang berulang kali ditemukan tapi masih dalam rumpun penyebab yang sama dapat digabung dalam satu temuan jika *tools* pemindai tidak melakukannya secara demikian.

Level 1B - Pemindaian Kode Sumber (Verifikasi Terotomatisasi Parsial)

Persyaratan Verifikasi Pemindaian Source Code Kontrol Keamanan

Source code scanning (juga dikenal sebagai "analisa statis", karena aplikasi tidak berjalan) terdiri dari menggunakan tools otomatis untuk mencari melalui kode sumber aplikasi untuk menemukan pola yang mungkin menunjukkan kerentanan. Harap perhatikan bahwa ini tidak cukup untuk memverifikasi desain yang tepat, implementasi, dan penggunaan kontrol keamanan, tetapi sudah cukup diterima di Level 1. Ruang lingkup verifikasi didefinisikan oleh persyaratan arsitektur di level ini.

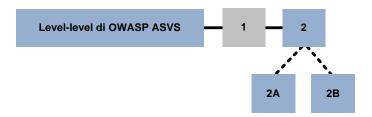
- L1B.1 Lakukan pemindaian kode sumber aplikasi sesuai dengan persyaratan level 1B yang ditentukan di bagian "Persyaratan Verifikasi Terperinci".
- L1B.2 Memverifikasi hasil pemindaian kode sumber, entah secara manual melalui uji penetrasi atau pengecekan ulang kode sumber. Hasil verifikasi terotomatis yang tidak diuji ulang secara manual tidak dianggap memenuhi persyaratan Level 1.

Kerentanan yang berulang kali ditemukan tapi masih dalam rumpun penyebab yang sama dapat digabung dalam satu temuan jika *tools* pemindai tidak melakukannya secara demikian.

Level 2 - Verifikasi Manual

Level 2 ("Verifikasi Manual") biasanya sesuai untuk aplikasi yang menangani transaksi pribadi, melakukan transaksi bisnis-ke-bisnis, memproses informasi kartu kredit, atau memproses informasi yang dapat digunakan untuk mengenali seseorang. Level 2 menyediakan kepercayaan yang cukup pada penggunaan kontrol keamanan yang benar dan keyakinan bahwa kontrol keamanan sudah bekerja dengan benar. Ancaman terhadap keamanan biasanya berwujud virus, worm, dan oportunis pemula seperti peretas dengan *tools* penyerang profesional atau *open-source*. Ruang lingkup verifikasi mencakup semua kode yang dikembangkan atau diubah untuk membuat aplikasi serta memeriksa keamanan semua komponen pihak ketiga yang menyediakan fungsi keamanan untuk aplikasi. Ada dua komponen konstituen untuk Level 2, seperti yang digambarkan dalam gambar di bawah ini.





Gambar 5 - OWASP ASVS Level 2, 2A, and 2B

Meskipun dapat ditentukan bahwa sebuah aplikasi telah memenuhi baik Level 2A atau 2B, namun jika baru salah satu yang terpenuhi, aplikasi belum bisa disebut mencapai Level 2. Selanjutnya, walaupun Level 2 adalah superset dari Level 1, tidak ada persyaratan untuk menjalankan alat otomatis untuk memenuhi persyaratan Level 2. Sebaliknya, verifikasi memiliki pilihan untuk hanya menggunakan teknik manual untuk memverifikasi semua persyaratan. Jika hasil dari alat otomatis tersedia, verifikasi dapat menggunakannya untuk mendukung analisa. Namun, bahkan melewati persyaratan di Level 1 tidak otomatis menandakan bahwa aplikasi tersebut melewati persyaratan yang sama di Level 2. Hal ini karena alat otomatis tidak memberikan bukti yang cukup bahwa sebuah persyaratan telah dipenuhi.

Teknik manual masih tetap diasumsikan memanfaatkan penggunaan *tools*. Hal ini dapat mencakup penggunaan setiap jenis analisa keamanan atau alat pengujian, termasuk alat bantu otomatis yang digunakan untuk verifikasi Level 1. Namun, alat tersebut hanya alat bantu untuk analis untuk menemukan dan menilai keamanan kontrol diverifikasi. Alat tersebut mungkin atau mungkin tidak berisi logika untuk secara otomatis mendeteksi kerentanan aplikasi.

Berikut ini adalah persyaratan tingkat-tinggi minimal untuk aplikasi Level 2, 2A atau 2B:

Lingkup Verifikasi

- L2.1 Ruang lingkup verifikasi mencakup semua kode yang dikembangkan atau dimodifikasi dalam rangka untuk menciptakan aplikasi. Persyaratan ini diperkenalkan di Level 1.
- L2.2 Ruang lingkup verifikasi termasuk kode untuk semua *framework* pihak ketiga, *library*, dan fungsi layanan keamanan yang dipanggil oleh atau mendukung keamanan aplikasi. Ini adalah persyaratan baru di Level 2.

Persyaratan Perilaku Kontrol Keamanan

- L2.3 Verifikasi bahwa semua kontrol keamanan teknis yang melakukan pemeriksaan keamanan membuat keputusan menggunakan pendekatan *whitelist*. Ini adalah persyaratan baru di Level 2.
- Verifikasi bahwa semua kontrol keamanan yang melakukan pemeriksaan keamanan dan kontrol tidak dapat dilewati menurut persyaratan Level 2A dan 2B yang ditetapkan dalam bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 2.

Persyaratan Penggunaan Kontrol Keamanan

Verifikasi bahwa semua kontrol keamanan digunakan dimanapun diperlukan dalam aplikasi, dan menggunakan implementasi yang terpusat, dilakukan pada sisi server, sesuai dengan persyaratan yang ditetapakan untuk Level 2 dalam bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 2.

Persyaratan Implementasi Kontrol Keamanan

Tidak ada Tidak ada persyaratan dalam bagaimana aplikasi kontrol keamanan dibangun di Level 2.



Persyaratan Verifikasi Kontrol Keamanan

- L2.6 Lakukan pengujian penetrasi manual pada aplikasi sesuai dengan persyaratan yang ditentukan untuk Level 2A di bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 2.
- L2.7 Lakukan review kode sumber secara manual pada aplikasi sesuai dengan persyaratan yang ditentukan untuk Level 2B di bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 2.

Persyaratan di Level 2 yang memungkinkan penggunaan kedua teknik verifikasi (antara manual dan otomatis) cukup diverifikasi dengan satu teknik.

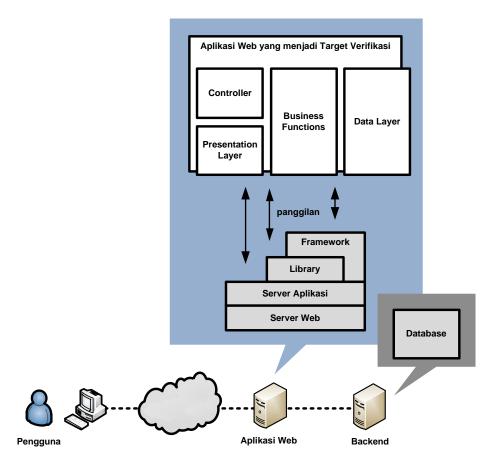
Verivikator mungkin menggunakan pemindaian otomatis atau analisis kode sebagai bagian dari upaya verifikasi di Level 2, tetapi verifikasi otomatis tidak dapat digunakan untku menggantikan verifikasi manual yang diperlukan untuk setiap persyaratan Level 2. Jika hasil scan membantu penguji dalam melakukan pekerjaan mereka atau dapat memperkaya hasil dari bagian manual, dapat digunakan untuk membantu dalam melakukan verifikasi Level 2.

Persyaratan Pelaporan

L2.8 Membuat laporan verifikasi yang menggambarkan arsitektur keamanan aplikasi dengan mengelompokkan komponen-komponen ke dalam sebuah arsitektur tingkat tinggi, dan termasuk hasil verifikasi sesuai dengan persyaratan di bagian "Persyaratan Pelaporan Verifikasi". Hal ini menambah persyaratan pelaporan dibandingkan di Level 1.

Di Level 2, komponen aplikasi dapat didefinisikan dari segi baik kelompok atau satuan file kode sumber, perpustakaan (libraries), dan/atau hasil kompilasi (executables) yang diatur dalam sebuah arsitektur tingkat tinggi (misalnya Model-View-Controller (MVC), komponen fungsi bisnis, dan komponen lapisan data). Sebagai contoh, diagram di bawah ini menggambarkan sebuah aplikasi yang terdiri dari aplikasi server, aplikasi server aplikasi, kode kustom, *libraries*, dan aplikasi database yang dikelompokkan sesuai dengan arsitektur MVC. Di Level 2, jalur atau path yang mungkin digunakan di *request* pengguna-akhir aplikasi tersebut harus didokumentasikan, seperti yang digambarkan dalam gambar di bawah ini. Namun, tidak semua jalur tersebut harus diperiksa.





Gambar 6 - Contoh Arsitektur Kemanaan OWASP ASVS Level 2

Level 2A - Uji Keamanan (Verifikasi Manual Parsial)

Persyaratan Verifikasi Penetrasi Kontrol Keamanan Aplikasi

Verifikasi keamanan aplikasi secara manual terdiri dari menciptakan pengujian dinamis untuk memverifikasi desain, implementasi, dan penggunaan kontrol keamanan sebuah aplikasi yang tepat. Ruang lingkup verifikasi didefinisikan oleh persyaratan arsitektur di level ini.

L2A.1 Lakukan pengujian keamanan aplikasi secara manual sesuai dengan persyaratan yang ditentukan Level 2A di bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 2.

Bila harus dilakukan, verifikasi dapat menggunakan *sampling* untuk memeriksa bahwa penggunaan control keamanan memang efektif. Penguji dapat memilih untuk mendokumentasikan pola kerentanan yang akan memungkinkan para pengembang secara yakin menemukan dan memperbaiki semua pola tersebut dalam perangkat lunak. Kenyataan kerentanan yang berulang kali ditemukan tapi masih dalam rumpun kerentanan yang sama dapat digabung dalam satu temuan tunggal.



Level 2B - Code Review (Verifikasi Manual Parsial)

Pedoman Tinjauan Manual Persyaratan Verifikasi Kontrol Keamanan

Memeriksa kode sumber secara manual terdiri dari pencarian oleh manusia dan analisis dari kode sumber aplikasi untuk memverifikasi desain aplikasi, implementasi, dan penggunaan yang tepat dari kontrol keamanan. Analisis tersebut diharapkan menggunakan alat bantu, tetapi bias juga hanya melibatkan alat-alat yang umum tersedia seperti editor kode sumber atau IDE. Ruang lingkup verifikasi didefinisikan oleh persyaratan arsitektur di level ini.

L2B.1 Lakukan review kode sumber secara manual pada aplikasi sesuai dengan persyaratan Level 2B di bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 2.

Bila dibutuhkan, verifikasi dapat menggunakan metode sampling yang tepat untuk memeriksa penggunaan efektif dari kontrol keamanan. Penguji dapat memilih untuk mendokumentasikan pola kerentanan yang akan memungkinkan para pengembang secara yakin menemukan dan memperbaiki semua pola tersebut dalam perangkat lunak. Kenyataan kerentanan yang berulang kali ditemukan tapi masih dalam rumpun kerentanan yang sama dapat digabung dalam satu temuan tunggal.

Level 3 - Verifikasi Desain

Tingkat 3 ("Verifikasi Desain") biasanya sesuai untuk aplikasi yang menangani transaksi bisnis-ke-bisnis secara signifikan, termasuk pengolahan informasi layanan kesehatan, menerapkan fungsi-fungsi bisnis yang kritis atau sensitif, atau memproses aset sensitif lainnya. Ancaman terhadap keamanan biasanya adalah virus dan worm, oportunis, dan penyerang serius (penyerang terampil dan termotivasi yang berfokus pada target tertentu menggunakan *tools* yang memang sengaja dibuat untuk menyerang). Ruang lingkup verifikasi mencakup semua kode sumber yang dikembangkan atau diubah untuk membuat aplikasi, serta memeriksa keamanan semua komponen pihak ketiga yang menyediakan fungsionalitas keamanan untuk aplikasi. Level 3 memastikan bahwa kontrol keamanan sendiri bekerja dengan benar, dan bahwa kontrol keamanan yang digunakan di mana-mana dalam aplikasi mereka harus digunakan untuk menegakkan kebijakan yang spesifik terhadap aplikasi. Tingkat 3 tidak dibagi menjadi beberapa komponen, seperti yang digambarkan dalam gambar di bawah ini.



Gambar 7 - OWASP ASVS Level 3

Berikut ini adalah persyaratan tingkat-tinggi minimal untuk aplikasi Level 3:

Lingkup Verifikasi

- L3.1 Ruang lingkup verifikasi mencakup semua kode yang dikembangkan atau dimodifikasi dalam rangka untuk menciptakan aplikasi. Persyaratan ini diperkenalkan di Level 1.
- L3.2 Ruang lingkup verifikasi termasuk kode untuk semua *framework* pihak ketiga, *library*, dan fungsi layanan keamanan yang dipanggil oleh atau mendukung keamanan aplikasi. Persyaratan ini diperkenalkan di Level 2.
- L3.3 Ruang lingkup verifikasi termasuk kode untuk semua *framework* pihak ketiga, *library*, dan *service* yang terkait dengan aplikasi tersebut. Ini adalah persyaratan baru di Level 3.

Persyaratan Perilaku Kontrol Keamanan

L3.4 Verifikasi bahwa semua kontrol keamanan yang melakukan pemeriksaan keamanan membuat keputusan menggunakan pendekatan *whitelist*. Persyaratan ini diperkenalkan di Level 2.



L3.5 Verifikasi bahwa semua kontrol keamanan yang melakukan pemeriksaan keamanan dan kontrol keamanan tidak dapat dilewati sesuai dengan persyaratan Level 3 yang ditentukan di bagian "Persyaratan Verifikasi Terperinci". Persyaratan ini diperkenalkan di Level 2.

Persyaratan Penggunaan Kontrol Keamanan

L3.6 Verifikasi bahwa semua kontrol keamanan digunakan dimana diperlukan dalam aplikasi, dan implementasi yang terpusat dalam aplikasi, dilakukan pada sisi server, dan sesuai dengan persyaratan level 3 yang ditentukan di bagian "Persyaratan Verifikasi Terperinci". Persyaratan ini diperkenalkan di Level 2.

Persyaratan Implementasi Kontrol Keamanan

Tidak ada Tidak ada persyaratan terhadap bagaimana kontrol keamanan aplikasi diimplementasikan di Level 3.

Persvaratan Verifikasi Kontrol Keamanan

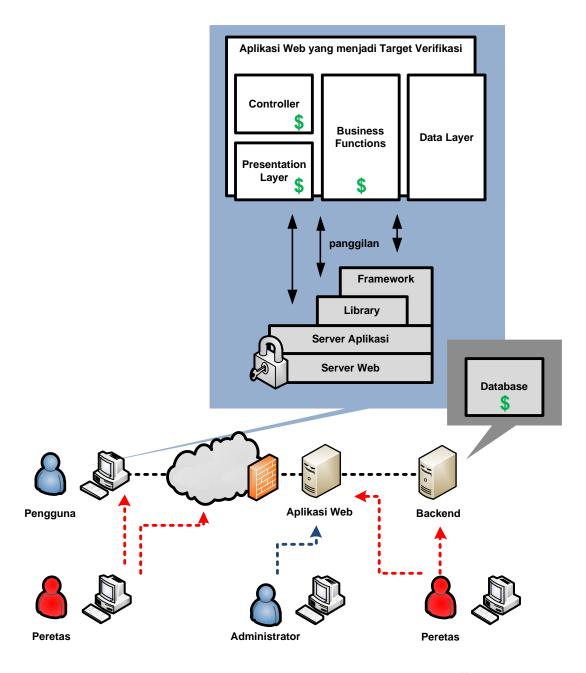
- L3.7 Memverifikasi secara manual aplikasi sesuai dengan kebutuhan Level 3 yang ditentukan di bagian "Persyaratan Verifikasi Terperinci". Hal ini menambah persyaratan verifikasi manual yang diperkenalkan di Level 2.
- L3.8 Mendokumentasikan arsitektur keamanan dan menggunakannya untuk memverifikasi bahwa desainnya sudah tepat dan penggunaan semua kontrol keamanan sudah benar dengan melakukan pemodelan ancaman (Threat modelling). Ini adalah persyaratan baru di Level 3.

Persyaratan Pelaporan

L3.9 Membuat laporan verifikasi yang menggambarkan arsitektur keamanan aplikasi dengan mengelompokkan komponen-komponen ke dalam sebuah arsitektur tingkat tinggi yang mencakup informasi pemodelan ancaman, dan memasukan hasil verifikasi sesuai dengan persyaratan di bagian "Persyaratan Pelaporan Verifikasi ". Hal ini menambah persyaratan pelaporan dari Level 2.

Di Level 3, komponen aplikasi dapat didefinisikan dari segi baik kelompok atau satuan file kode sumber, perpustakaan (libraries), dan/atau hasil kompilasi (executables) yang diatur dalam sebuah arsitektur tingkat tinggi (misalnya Model-View-Controller (MVC), komponen fungsi bisnis, dan komponen lapisan data). Pada Level 3, pemodelan ancaman harus ditambahi dengan informasi agen ancaman (threat agents) dan aset. Di Level 3, jalur atau path yang mungkin digunakan di *request* pengguna-akhir aplikasi tersebut harus didokumentasikan, seperti yang digambarkan dalam gambar di bawah ini. Pada Level 3, *semua* jalur potensial melalui tampilan tingkat-tinggi dari aplikasi harus diperiksa.





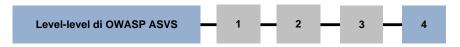
Gambar 8 - Contoh Arsitektur Keamanan OWASP ASVS Level 3¹¹

¹¹ Tanda menunjukkan aset dalam diagram.



Tingkat 4 - Verifikasi Internal

Tingkat 4 ("Verifikasi internal") biasanya sesuai untuk aplikasi kritis yang melindungi kehidupan dan keselamatan, infrastruktur kritis, atau fungsi pertahanan. Level 4 juga mungkin cocok untuk aplikasi yang mengolah aset sensitif. Level 4 memastikan bahwa kontrol keamanan sendiri bekerja dengan benar, bahwa semua kontrol keamanan digunakan dimanapun diperlukan dalam aplikasi dalam menegakkan kebijakan yang spesifik terhadap aplikasi, dan bahwa praktek pemrograman yang aman diikuti. Ancaman terhadap keamanan adalah peretas serius (penyerang terampil dan termotivasi yang berfokus pada target tertentu menggunakan *tools* yang memang sengaja dibuat untuk menyerang). Ruang lingkup verifikasi memperluas lingkup Level 3 untuk memasukkan juga semua kode sumber yang digunakan oleh aplikasi. Level 4 tidak dipecahkan menjadi komponen penyusun, seperti yang digambarkan dalam gambar di bawah ini



Gambar 9 - OWASP ASVS Level 4

Berikut ini adalah persyaratan tingkat tinggi untuk Level 4 aplikasi:

Lingkup Verifikasi

- L4.1 Ruang lingkup verifikasi mencakup semua kode yang dikembangkan atau dimodifikasi dalam rangka untuk menciptakan aplikasi. Persyaratan ini diperkenalkan di Level 1.
- L4.2 Ruang lingkup verifikasi termasuk kode untuk semua *framework* pihak ketiga, *library*, dan fungsi layanan keamanan yang dipanggil oleh atau mendukung keamanan aplikasi. Persyaratan ini diperkenalkan di Level 2.
- L4.3 Ruang lingkup verifikasi termasuk kode untuk semua *framework* pihak ketiga, *library*, dan layanan terkait dengan aplikasi tersebut. Persyaratan ini diperkenalkan di Level 3.
- L4.4 Ruang lingkup verifikasi mencakup semua kode yang tersisa berhubungan dengan aplikasi, termasuk framework, library, runtime environment, development tools, build tools, dan deployment tools. Ruang lingkup tidak termasuk kode untuk perangkat lunak platform, seperti server aplikasi, server database, mesin virtual, atau sistem operasi, yang telah menerima banyak pemeriksaan publik. Ini adalah persyaratan baru di Level 4.

Persyaratan Perilaku Kontrol Keamanan

- L4.5 Verifikasi bahwa semua kontrol keamanan yang melakukan pemeriksaan keamanan membuat keputusan menggunakan pendekatan *whitelist*. Persyaratan ini diperkenalkan di Level 2.
- Verifikasi bahwa semua kontrol keamanan yang melakukan pemeriksaan keamanan dan kontrol keamanan tidak dapat dilewati sesuai dengan persyaratan Level 4 yang ditetapkan dalam bagian "Persyaratan Verifikasi Terperinci". Persyaratan ini diperkenalkan di Level 2.

Persyaratan Penggunaan Kontrol Keamanan

Verifikasi bahwa semua kontrol keamanan digunakan dimana mereka diperlukan dalam aplikasi, dan bahwa implementasinya terpusat dalam aplikasi, dilakukan pada sisi server, sesuai dengan persyaratan Level 4 yang ditetapkan dalam bagian "Persyaratan Verifikasi Terperinci". Persyaratan ini diperkenalkan di Level 3.



Persyaratan Implementasi Kontrol Keamanan

L4.8 Pastikan bahwa aplikasi tidak berisi kode berbahaya sesuai dengan persyaratan Level 4 yang ditetapkan dalam bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 4.

Persyaratan Verifikasi Kontrol Keamanan

- L4.9 Secara manual memverifikasi aplikasi terhadap persyaratan level 4 yang ditentukan dalam bagian "Persyaratan Verifikasi Terperinci". Hal ini menambah persyaratan dari Level 3.
- L4.10 Mendokumentasikan arsitektur keamanan dan menggunakannya untuk memverifikasi desain yang tepat dan penggunaan semua kontrol keamanan dengan melakukan pemodelan ancaman. Persyaratan ini diperkenalkan di Level 3.
- L4.11 Secara manual meninjau semua kode yang dikembangkan atau dimodifikasi untuk aplikasi ini untuk kode berbahaya¹² sesuai dengan persyaratan Level 4 yang ditetapkan dalam bagian "Persyaratan Verifikasi Terperinci". Ini adalah persyaratan baru di Level 4.

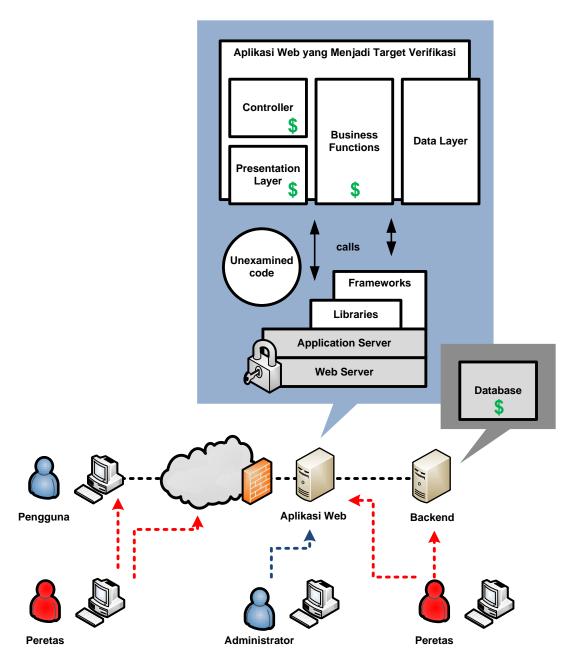
Persyaratan Pelaporan

L4.12 Membuat laporan verifikasi yang menggambarkan arsitektur keamanan aplikasi sesuai dengan persyaratan Level 3, yang meliputi semua kode aplikasi, dan termasuk hasil verifikasi sesuai dengan persyaratan di bagian "Persyaratan Pelaporan Verifikasi". Hal ini menambah persyaratan pelaporan dari Level 3.

Pada level 4, arsitektur aplikasi harus dituangkan seperti yang diminta di Level 3. Selanjutnya, Level 4 mensyaratkan bahwa semua kode aplikasi, termasuk kode yang tidak secara eksplisit diperiksa, diidentifikasi sebagai bagian dari definisi aplikasi, seperti yang digambarkan dalam gambar di bawah ini. Kode ini harus mencakup semua *library*, *frameworks*, dan kode pendukung yang mendukung aplikasi. verifikasi sebelumnya dari komponen ini dapat digunakan kembali sebagai bagian dari upaya verifikasi lanjutannya. Kode Platform, seperti sistem operasi, mesin virtual, atau *library* yang diterbitkan dengan lingkungan mesin virtual, Web server, atau server aplikasi tidak termasuk dalam Level 4. Sebagai contoh, perpustakaan yang terkait dengan runtime Java tidak akan perlu diuji di Level 4.

¹² Kode berbahaya tidak sama dengan *malware*. Lihat definisi kode berbahaya di *glossary*.





Gambar 10 - Contoh Kode Aplikasi yang belum Diperiksa di OWASP ASVS Level 4

Interpretasi Persyaratan dan Preseden

OWASP ASVS adalah dokumen hidup. Jika Anda melakukan verifikasi keamanan aplikasi sesuai dengan standar ini, maka Anda harus selalu meninjau artikel yang dapat ditemukan pada halaman proyek ASVS OWASP di lokasi berikut: http://www.owasp.org/index.php/ASVS # Articles_Below_-_More_About_ASVS_and_Using_It . Artikel-artikel pada halaman proyek ASVS OWASP memberikan klarifikasi persyaratan lebih lanjut, preseden putusan persyaratan, dan petunjuk pembantu.



Persyaratan Verifikasi Terperinci

Bagian ini mendefinisikan persyaratan verifikasi rinci yang diturunkan dari persyaratan tingkat tinggi untuk setiap tingkat verifikasi yang didefinisikan dalam standar ini. Setiap bagian di bawah ini mendefinisikan satu set persyaratan verifikasi rinci yang dikelompokkan ke dalam beberapa bidang terkait.

ASVS mendefinisikan area atau lingkup dari keamanan aplikasi sebagai berikut:

- V1. Arsitektur Keamanan
- V2. Otentikasi
- V3. Manajemen Sesi
- V4. Kontrol Akses
- V5. Validasi masukan
- V6. Pengkodean dan pemberian escape-character kepada keluaran
- V7. Kriptografi
- V8. Penanganan dan pencatatan kesalahan
- V9. Proteksi data
- V10. Keamanan komunikasi
- V11. Keamanan HTTP
- V12. Konfigurasi keamanan
- V13. Pencarian kode berbahaya (malicious)
- V14. Keamanan internal

Untuk setiap bidang tersebut, akan dijelaskan persyaratan yang harus dipenuhi di setiap tingkatan dibawah

- Level 1: Verifikasi Terotomatisasi
 - Level 1A Pemindaian Dinamis (Partial Automated Verification)
 - Level 1B Pemindaian Kode Sumber (Partial Automated Verification)
- Level 2: Verifikasi Manual
 - Level 2A Pengujian Keamanan (Partial Manual Verification)
 - Level 2B Code Review (Partial Manual Verification)
- Level 3: Verifikasi Desain
- Level 4: Verifikasi Internal



V1 - Persyaratan Dokumentasi Arsitektur Keamanan

Untuk semua tingkat ASVS, pendokumentasian beberapa informasi arsitektur keamanan dasar sangat diperlukan untuk memastikan baik kelengkapan dan akurasi (dan pengulangan saat remediasi diperlukan) dari verifikasi keamanan aplikasi yang dilakukan. Analisis dapat diarahkan dan hasil dapat ditelusuri kembali ke arsitektur keamanan tingkat tinggi aplikasi. Persyaratan ini mulai dengan tingkat dasar detail arsitektur keamanan yang harus diketahui dan tingkat kedetilan meningkat di tiap level. Tabel di bawah ini mendefinisikan persyaratan keamanan dokumentasi arsitektur keamanan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 1 - Persyaratan Arsitektur Keamanan OWASP (V1)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V1.1	Memeriksa bahwa semua komponen aplikasi (entah pecahan kode sumber, pustaka, atau hasil kompilasi) yang berada dalam aplikasi, sudah teridentifikasi.	✓	✓	✓	✓	✓	✓
V1.2	Memeriksa bahwa semua komponen yang bukan merupakan bagian dari aplikasi tersebut namun keberadaannya diperlukan untuk kelangsungan operasional aplikasi tersebut, sudah teridentifikasi.			✓	✓	✓	✓
V1.3	Memeriksa bahwa desain arsitektur high-level untuk aplikasi tersebut sudah terdefinisikan. ¹³			✓	✓	✓	✓
V1.4	Memeriksa bahwa komponen- komponen aplikasi terdefinisikan dalam hal fungsi bisnis-nya dan/atau fungsi keamanan yang mereka miliki.					✓	✓
V1.5	Memeriksa bahwa semua komponen yang bukan merupakan bagian dari aplikasi tersebut namun keberadaannya diperlukan untuk kelangsungan operasional aplikasi tersebut terdefinisikan dalam hal fungsi bisnis-nya dan/atau fungsi keamanan yang mereka miliki.					√	✓
V1.6	Memeriksa bahwa informasi pemodelan ancaman (threat) telah tersedia.					✓	✓

 $^{^{13}}$ Jika pengembang aplikasi tidak menyediakan dokumen ini, pihak yang hendak memeriksa dipersilakan membuat dokumen high-level tersebut.



V2 - Persyaratan Verifikasi Otentikasi

Bagian ini mendefinisikan persyaratan untuk menangani hak dan kuasa yang dimiliki akun secara aman. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 2 - Persyaratan Keamanan Otentikasi OWASP (V2)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V2.1	Memeriksa bahwa untuk mengakses semua halaman web dan sumber daya, dibutuhkan otentikasi, kecuali memang ditujukan kepada publik.	✓	✓	✓	✓	✓	✓
V2.2	Memeriksa bahwa semua ruas untuk sandi lewat tidak tertampilkan secara polos, dan bahwa semua ruas untuk sandi lewat (atau borang yang memuatnya) mematikan fitur pengisian otomatis (autocomplete).	√	✓	√	√	√	✓
V2.3	Memastikan bahwa jika terdapat upaya otentikasi yang melampaui jumlah maksimum, akun terkunci untuk jangka waktu yang cukup lama untuk mencegah serangan brute force.	✓		√	√	√	✓
V2.4	Memastikan bahwa semua penegakkan kontrol otentikasi dilakukan di sisi server.			✓	✓	✓	✓
V2.5	Memastikan bahwa semua kontrol otentikasi (termasuk pustaka yang memanggil layanan otentikasi eksternal) memiliki implementasi terpusat.				✓	✓	✓
V2.6	Memastikan bahwa semua kontrol otentikasi tetap aman walaupun terjadi kegagalan (prinsip <i>fail securely</i>).			✓	✓	✓	✓
V2.7	Memastikan bahwa kekuatan dari setiap unsur identitas otentikasi cukup untuk menahan serangan yang sering terjadi untuk setiap ancaman di lingkungan dimana aplikasi itu berada.			✓	✓	✓	✓
V2.8	Memastikan bahwa semua fungsi- fungsi manajemen akun tahan terhadap serangan setidaknya sama kuatnya dengan mekanisme otentikasi utama.			✓	✓	✓	✓



V2.9	Memastikan bahwa pengguna dapat dengan aman mengubah unsur identitasnya dengan menggunakan mekanisme yang setidaknya tahan terhadap serangan setidaknya sama kuatnya dengan mekanisme otentikasi utama.		√	√	√	✓
V2.10	Memastikan bahwa otentikasi ulang diperlukan sebelum mengizinkan operasi sensitif yang spesifik terhadap aplikasi.		✓	✓	✓	✓
V2.11	Memastikan bahwa setelah periode waktu yang bisa diatur, ada unsur identitas otentikasi yang harus diperbaharui.		✓	✓	✓	✓
V2.12	Memastikan bahwa semua keputusan dalam otentikasi dicatatkan.			✓	✓	✓
V2.13	Memastikan bahwa sandi lewat akun ditambahkan salt yang unik terhadap akun tersebut (misalnya, ID internal pengguna, waktu pembuatan account) dan di-hash sebelum disimpan.			✓	✓	✓
V2.14	Memastikan bahwa unsur identitas otentikasi untuk mengakses semua layanan eksternal untuk operasional aplikasi harus dienkripsi dan disimpan di lokasi yang dilindungi (tidak dalam kode sumber).			✓	✓	√
V2.15	Memastikan bahwa di semua implementasi kontrol otentikasi tidak terdapat kode berbahaya (<i>malicious code</i>).					✓

V3 - Persyaratan Verifikasi Manajemen Sesi

Bagian ini mendefinisikan persyaratan untuk menggunakan HTTP requests, responses, sessions, cookies, headers, and logging dalam mengatur sesi ssecara aman. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 3 - Persyaratan Keamanan Manajemen Sesi OWASP (V3)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V3.1	Memastikan bahwa aplikasi menggunakan kontrol manajemen <i>session</i> bawaan dari <i>framework</i> yang dipakai.	✓		✓	✓	✓	✓



V3.2	Memastikan bahwa sesi ditutup ketika pengguna log out.	✓		✓	✓	✓	✓
V3.3	Pastikan bahwa sesi mengalami timeout setelah tidak aktif selama jangka waktu tertentu.	✓		✓	✓	✓	✓
V3.4	Pastikan bahwa sesi mengalami timeout setelah jangka waktu tertentu yang bisa ditentukan oleh administrasi, terlepas dari adanya aktivitas atau tidak.					✓	✓
V3.5	Pastikan bahwa semua halaman yang membutuhkan otentikasi untuk diakses, memiliki tautan untuk logout.	✓		√	√	✓	✓
V3.6	Pastikan bahwa nomor sesi tidak pernah diungkapkan selain di header cookie, khususnya di URL, pesan error, atau log. Ini termasuk memastikan bahwa aplikasi tidak memperbolehkan penulisan ulang session cookies melalui URL.		✓		✓	✓	√
V3.7	Pastikan bahwa ID <i>session</i> diubah setiap kali login.			✓	✓	✓	✓
V3.8	Pastikan bahwa ID <i>session</i> diubah setiap kali otentikasi ulang.			✓	✓	✓	✓
V3.9	Pastikan bahwa sesi id berubah atau dibersihkan pada saat logout.			✓	✓	✓	✓
V3.10	Memastikan bahwa id sesi yang diakui sebagai sah oleh aplikasi adalah hanya id yang dihasilkan oleh framework aplikasi.			✓		✓	✓
V3.11	Pastikan bahwa token sesi yang telah terotentikasi cukup panjang dan acak untuk bertahan dari jenis serangan yang sering terjadi berdasarkan ancaman-ancaman di lingkungan.					✓	✓
V3.12	Pastikan bahwa cookie yang mengandung token/id sesi terontetikasi, memiliki domain dan path ke lokasi terbatas untuk situs tersebut.					✓	✓
V3.13	Verifikasi bahwa semua kode yang menggunakan control manajemen sesi tidak terinfeksi oleh kode malicious.						✓

V4 - Persyaratan Verifikasi Kontrol Akses

Bagian ini mendefinisikan bagaimana sebuah aplikasi dapat menegakkan kontrol akses secara aman. Pada kebanyakan kasus, kontrol akses harus dilakukan diberbagai lapisan dan lokasi di aplikasi. Kontrol akses



yang diverifikasi adalah untuk URLs, business functions, data, services, and files. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 4 - Persyaratan Kontrol Akses OWASP (V4)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V4.1	Pastikan bahwa pengguna hanya dapat mengakses fungsi dimana mereka memiliki wewenang khusus.	✓	✓	✓	√	✓	✓
V4.2	Pastikan bahwa pengguna hanya dapat mengakses URL dimana mereka memiliki wewenang khusus.	✓		✓	✓	✓	✓
V4.3	Pastikan bahwa pengguna hanya dapat mengakses file data dimana mereka memiliki wewenang khusus.	✓		✓	✓	✓	✓
V4.4	Pastikan bahwa akses langsung kepada objek terlindungi, dimana hanya dapat dilakukan oleh mereka yang memiliki wewenang khusus.	✓		✓	✓	✓	√
V4.5	Pastikan bahwa penjelajahan direktori dinonaktifkan kecuali diinginkan secara sengaja.	✓		✓		✓	✓
V4.6	Pastikan bahwa pengguna hanya dapat mengakses layanan dimana mereka memiliki wewenang khusus.			✓	✓	✓	✓
V4.7	Pastikan bahwa pengguna hanya dapat mengakses data dimana mereka memiliki wewenang khusus.			✓	✓	✓	✓
V4.8	Memastikan bahwa semua kontrol akses tetap aman walaupun terjadi kegagalan (prinsip <i>fail securely</i>).			✓	✓	✓	✓
V4.9	Pastikan bahwa aturan kontrol akses yang dilakukan di lapisan presentasi/tampilan dilakukan juga pada sisi server.			✓	✓	√	✓
V4.10	Verifikasi bahwa semua atribut pengguna dan data dan informasi kebijakan yang digunakan oleh kontrol akses tidak dapat dimanipulasi oleh pengguna akhir kecuali diberi wewenang secara khusus.			✓	✓	√	✓
V4.11	Verifikasi bahwa semua kontrol akses dilakukan pada sisi server.			✓	✓	✓	✓



V4.12	Pastikan bahwa ada mekanisme terpusat (termasuk <i>library</i> yang menggunakan layanan otorisasi eksternal) untuk melindungi akses ke setiap jenis sumber daya yang dilindungi.			✓	✓	✓
V4.13	Pastikan bahwa batasan-batasan pada input dan akses yang dikenakan oleh bisnis pada aplikasi (seperti batas transaksi harian atau tugas yang telah terurut) tidak dapat dilewati.		✓	✓	✓	✓
V4.14	Pastikan bahwa semua keputusan kontrol akses dapat dicatat dan semua kegagalan dicatat.			✓	✓	✓
V4.15	Verifikasi bahwa semua kode yang menerapkan atau menggunakan kontrol akses tidak terdapat kode berbahaya/malicious.					✓

V5 - Persyaratan Verifikasi Validasi Input

Bagian ini mendefinisikan bagaimana masukan harus divalidasi agar aman dipakai dalam aplikasi. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 5 - Persyaratan Validasi Masukan OWASP (V5)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V5.1	Pastikan bahwa lingkungan runtime tidak rentan terhadap buffer overflow, atau kontrol keamanan yang ada mampu mencegah serangan-serangan buffer overflow.	✓	✓	✓	✓	✓	✓
V5.2	Pastikan bahwa pola validasi positif didefinisikan dan diterapkan untuk semua input.	✓	✓	✓	✓	✓	✓
V5.3	Pastikan bahwa semua kegagalan validasi input berujung pada penolakan input atau pembersihan input.	✓		✓	✓	✓	✓
V5.4	Verifikasi adanyap penentuan sebuah set karakter, seperti UTF-8, untuk semua sumber input.			✓	✓	✓	✓
V5.5	Verifikasi bahwa semua validasi input dilakukan pada sisi server.			✓	✓	✓	✓



V5.6	Pastikan bahwa aplikasi menggunakan kontrol validasi input yang seragam untuk setiap jenis data yang diterima.		✓	✓	✓
V5.7	Pastikan bahwa semua kegagalan validasi input dicatat.		✓	✓	✓
V5.8	Pastikan bahwa semua data input dikanonikalisasikan (canonicalized) sebelum validasi.			✓	✓
V5.9	Pastikan bahwa tidak terdapat kode berbahaya di semua kontrol validasi input.				✓

V6 - Persyaratan Verifikasi Pengkodean Keluaran

Bagian ini mendefinisikan bagiamana keluaran dari sebuah aplikasi harus dikodekan dengan benar atau diberikan *escape character*, sehingga aman digunakan oleh aplikasi lainnya. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 6 - Persyaratan Verifikasi Pengkodean Keluaran OWASP (V6)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V6.1	Pastikan bahwa semua data yang tidak dipercaya yang berupa output ke HTML (termasuk elemen HTML, atribut HTML, nilai data javascript, blok CSS, dan atribut URI) sudah dipasangi escape character yang sesuai untuk konteks yang berlaku.		✓	✓	✓	✓	✓
V6.2	Verifikasi bahwa semua penandaan dan pemberian escape character terhadap output diimplementasikan pada sisi server.			✓	✓	✓	✓
V6.3	Pastikan bahwa mekanisme penandaan / pemberian escape character pada output, mengkodekan semua karakter yang dikenal tidak aman untuk interpreter yang menjadi tujuan.				√	✓	✓
V6.4	Pastikan bahwa semua data output yang tidak dipercaya menuju ke interpreter SQL menggunakan parameterized interfaces, prepared statements, atau diberi penandaan escape character dengan benar.				✓	✓	✓



V6.5	Pastikan bahwa semua data output yang tidak dipercaya dalam bentuk XML, menggunakan <i>parameterized interfaces</i> atau diberi penandaan escape character dengan benar.		✓	✓	✓
V6.6	Pastikan bahwa semua data yang tidak dipercaya yang digunakan dalam query LDAP diberi penandaan escape character dengan benar.		✓	✓	✓
V6.7	Pastikan bahwa semua data tidak dipercaya yang disertakan sebagai parameter perintah kepada sistem operasi, diberi tanda escape character dengan benar.		✓	✓	✓
V6.8	Pastikan bahwa semua data tidak dipercaya yang menjadi output untuk diumpan kepada <i>interpreter</i> apapun yang tidak secara khusus tercantum di atas, diberi penanda escape character dengan benar.		✓	✓	✓
V6.9	Verifikasi bahwa untuk setiap jenis pengkodean/pemberian escape character kepada output yang dilakukan oleh aplikasi, ada kontrol keamanan tunggal untuk jenis output tertentu dan tujuan tertentu.			✓	✓
V6.10	Verifikasi bahwa semua kode yang bertujuan mengontrol validasi keluaran tidak terpengaruh oleh kode berbahaya.				✓

V7 - Persyaratan Verifikasi Kriptografi

Bagian ini mendefinisikan cara mem-verifikasi sistem enkripsi yang digunakan aplikasi, key management, random number, and hashing operations. Aplikasi harus menggunakan modul enkripsi yang terakreditasi FIPS 140-2, atau oleh standar lain yang setara. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 7 - Persyaratan Verifikasi Kriptografi OWASP (V7)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V7.1	Verifikasi bahwa semua fungsi kriptografi yang digunakan untuk melindungi rahasia dari pengguna aplikasi diimplementasikan di sisi server.			✓	✓	✓	✓
V7.2	Pastikan bahwa semua modul kriptografi, ketika mengalami kegagalan, mengikuti prinsip <i>fail</i> <i>securely</i> .			✓	✓	✓	✓



V7.3	Pastikan bahwa akses ke rahasia utama dilindungi dari akses yang tidak sah (Sebuah rahasia utama adalah komponen identitas aplikasi yang tersimpan sebagai file text di harddrive yang digunakan untuk melindungi keamanan akses ke informasi konfigurasi konfigurasi).		✓	√	✓
V7.4	Pastikan bahwa hash password diberi salt ketika mereka diciptakan.		✓	✓	✓
V7.5	Pastikan bahwa setiap kegagalan modul kriptografi dicatat.		✓	✓	✓
V7.6	Verifikasi bahwa semua nomor acak, nama file acak, GUIDs acak, dan string acak dihasilkan dengan menggunakan generator nomor yang disetujui modul kriptografi jika nilai-nilai acak dimaksudkan untuk tidak dapat diterka oleh penyerang.		✓	✓	✓
V7.7	Pastikan bahwa modul kriptografi yang digunakan oleh aplikasi telah divalidasi terhadap FIPS 140-2 atau standar lain yang setara. (Lih. http://csrc.nist.gov/groups/STM/cmvp/validati on.html).			✓	✓
V7.8	Pastikan bahwa modul kriptografi beroperasi dalam modus yang sesuai dengan publikasi kebijakan keamanan masing-masing modul (Lih. http://csrc.nist.gov/groups/STM/cmvp/validati on.html).			√	✓
V7.9	Pastikan bahwa ada kebijakan yang eksplisit tentang bagaimana kunci kriptografi dikelola (misalnya, dihasilkan, didistribusikan, dicabut, kedaluwarsa). Pastikan bahwa kebijakan ini diterapkan dengan benar.			✓	✓
V7.10	Verifikasi bahwa semua kode yang mendukung atau menggunakan modul kriptografi tidak terpengaruh oleh kode berbahaya.				✓

V8 - Peryaratan Verifikasi Penanganan dan Pencatatan Error

Bagian ini mendefinisikan cara-cara untuk mem-verifikasi pencatatan yang digunakan untuk pelacakan dan mengenali kejadian-kejadian mencurigakan. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 8 - Persyaratan Verifikasi Penanganan Error OWASP (V8)

Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
------------------------	----------	----------	----------	----------	---------	---------



V8.1	Memverifikasi bahwa aplikasi tidak mengeluarkan pesan atau jejak stack yang berisi data sensitif yang dapat membantu penyerang, termasuk id sesi dan informasi pribadi.	✓	✓	✓	✓	✓	✓
V8.2	Verifikasi bahwa kesalahan di sisi server semua ditangani di server.			✓	✓	✓	✓
V8.3	Verifikasi bahwa semua kontrol pencatatan dilaksanakan di server.			✓	✓	✓	✓
V8.4	Pastikan bahwa logika penanganan kesalahan dalam kontrol keamanan menolak akses pada pengaturan standar.			✓	✓	✓	✓
V8.5	Pastikan pencatatan kontrol keamanan menyediakan kemampuan untuk mencatat peristiwa keberhasilan dan kegagalan yang diidentifikasi sebagai yang terkait keamanan.				✓	✓	✓
V8.6	Pastikan bahwa setiap peristiwa dalam pencatatan meliputi:						
	 cap waktu dari sumber yang dapat dipercaya, 						
	2. tingkat keparahan insiden tersebut,						
	3. indikasi bahwa ini adalah peristiwa yang terkait keamanan (jika dicampur dengan catatan hal lain),				√	√	./
	4. identitas pengguna yang menyebabkan peristiwa (jika ada user yang berhubungan dengan acara tersebut),				·	·	·
	5. alamat IP sumber dari permintaan yang terkait dengan insiden tersebut,						
	6. apakah kejadian tersebut berhasil atau gagal, dan7. deskripsi kejadian.						
V8.7	Memastikan bahwa semua kejadian						
	yang menyangkut data tidak dipercaya tidak akan tereksekusi sebagai kode dalam perangkat lunak yang dimaksudkan untuk melihat catatan.				✓	✓	√
V8.8	Pastikan bahwa pencatatan keamanan dilindungi dari akses dan modifikasi yang tidak sah.				✓	✓	✓
V8.9	Pastikan bahwa ada implementasi pencatatan tunggal yang digunakan oleh aplikasi.				✓	✓	√



V8.10	Memverifikasi bahwa aplikasi tidak mencatat data sensitif yang spesifik terhadap aplikasi yang bisa membantu penyerang, termasuk sesi id pengguna dan informasi pribadi atau sensitif.		✓	✓	✓
V8.11	Pastikan ketersediaan alat analisa log yang memungkinkan analis untuk mencari peristiwa catatan berdasarkan kombinasi dari kriteria pencarian di semua kolom dalam format rekaman log yang didukung oleh sistem ini.		√	√	✓
V8.12	Verifikasi bahwa semua kode yang menerapkan atau menggunakan penanganan kesalahan dan kontrol pencatatan tidak terpengaruh oleh kode berbahaya.				✓

V9 - Persyaratan Verifikasi Proteksi Data

Bagian ini mendefinisikan cara mem-verifikasi bahwa aplikasi telah melindungi data-data yang sensitif (mis., nomor kartu kredit, nomor paspor, dan informasi-informasi lain yang dapat disangkut-pautkan kepada pribadi tertentu). Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 9 - Persyaratan Verifikasi Perlindungan Data (V9)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V9.1	Pastikan bahwa semua form yang berisi informasi sensitif telah menonaktifkan caching di sisi klien, termasuk fitur autocomplete.	✓	✓	√	✓	✓	√
V9.2	Pastikan bahwa daftar data sensitif yang diproses oleh aplikasi ini teridentifikasi, dan bahwa ada kebijakan eksplisit untuk bagaimana akses ke data ini harus dikontrol, dan kapan data ini harus dienkripsi (baik saat diam dan dalam transmisi). Pastikan bahwa kebijakan ini diterapkan dengan benar.				√	√	√
V9.3	Pastikan bahwa semua data sensitif dikirim ke server di bagian isi HTTP (yaitu, parameter URL tidak pernah digunakan untuk mengirim data sensitif).			✓		✓	✓



V9.4	Memastikan bahwa semua cache atau salinan sementara data sensitif yang dikirim ke klien dilindungi dari akses yang tidak sah atau dibersihkan / dirusak setelah pengguna yang berwenang mengakses data sensitif (misalnya, pengaturan no-cache dan no-store di Cache-Control header).		√	✓	✓
V9.5	Memastikan bahwa semua cache atau salinan sementara data sensitif yang tersimpan di server dilindungi dari akses yang tidak sah atau dibersihkan / dirusak setelah pengguna yang berwenang mengakses data sensitif.		✓	✓	✓
V9.6	Pastikan bahwa ada metode untuk menghapus setiap jenis data sensitif dari aplikasi pada akhir periode penyimpanan yang diperlukan nya.			✓	✓

V10 - Persyaratan Verifikasi Keamanan Komunikasi

Bagian ini mendefinisikan cara-cara untuk mem-verifikasi bahwa semua jalur komunikasi yang digunakan aplikasi adalah aman. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 10 - Persyaratan Keamanan Komunikasi (V10)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V10.1	Memastikan bahwa sebuah jalur dapat dibangun dari CA terpercaya untuk setiap sertifikat Transport Layer Security (TLS), dan bahwa setiap sertifikat server adalah benar.	✓		✓	✓	✓	✓
V10.2	Pastikan koneksi TLS yang gagal tidak masuk kembali kepada koneksi yang tidak aman.			✓		✓	✓
V10.3	Memastikan bahwa TLS digunakan untuk semua koneksi (termasuk koneksi eksternal dan backend) yang diotentikasi atau yang melibatkan fungsi atau data sensitif.				√	√	✓
V10.4	Memastikan bahwa kegagalan koneksi backend TLS dicatat.				✓	✓	✓
V10.5	Pastikan bahwa jalan sertifikat dibuat dan diverifikasi untuk semua sertifikat klien menggunakan trust anchors terkonfigurasi dan pencabutan informasi.				√	✓	✓



V10.6	Pastikan bahwa semua koneksi ke sistem eksternal yang melibatkan informasi sensitif atau fungsi sensitif diotentikasi.		✓	✓	✓
V10.7	Pastikan bahwa semua koneksi ke sistem eksternal yang melibatkan informasi sensitif atau fungsi sensitif menggunakan account yang telah diatur untuk memiliki hak minimum yang diperlukan untuk aplikasi berfungsi dengan baik.		✓	✓	✓
V10.8	Pastikan bahwa ada implementasi TLS standar tunggal yang digunakan oleh aplikasi yang dikonfigurasi untuk beroperasi dalam mode operasi yang disetujui (Lih. http://csrc.nist.gov/groups/STM/cm vp/documents/fips140- 2/FIPS1402IG.pdf).			√	√
V10.9	Memastikan bahwa pengkodean karakter yang spesifik didefinisikan untuk semua koneksi (misalnya UTF-8).			✓	✓

V11 - Persyaratan Verifikasi Keamanan HTTP

Bagian ini mendefinisikan cara-cara mem-validasi keamanan dalam penggunaan HTTP requests, responses, sessions, cookies, headers, and logging. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 11 - Persyaratan Verifikasi Keamanan HTTP OWASP ASVS (V11)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V11.1	Pastikan bahwa pengalihan tidak memasukkan data yang tidak tervalidasi.	✓	✓	✓	✓	✓	✓
V11.2	Pastikan bahwa aplikasi hanya menerima satu set metode request HTTP yang ditentukan, seperti GET dan POST.	✓	✓	✓	✓	✓	✓
V11.3	Pastikan bahwa setiap respon HTTP berisi content type header yang menetapkan set karakter yang aman (misalnya, UTF-8).	✓	✓	✓	✓	✓	✓
V11.4	Memastikan bahwa flag HTTPOnly digunakan pada semua cookie yang tidak secara spesifik memerlukan akses dari JavaScript.			✓	✓	✓	✓



V11.5	Pastikan bahwa flag tersebut digunakan pada semua cookie yang berisi data sensitif, termasuk session cookie.		✓	✓	✓	✓
V11.6	Memastikan bahwa header HTTP baik permintaan dan tanggapan hanya berisi karakter ASCII yang dapat ditampilkan.		✓	✓	✓	✓
V11.7	Pastikan bahwa aplikasi menghasilkan token acak yang kuat sebagai bagian dari semua link dan form yang berhubungan dengan transaksi atau pengaksesan data yang sensitif, dan bahwa aplikasi memverifikasi keberadaan token ini dengan nilai yang tepat untuk pengguna saat ini saat memproses permintaan ini. ¹⁴				✓	✓

V12 - Persyaratan Verifikasi Konfigurasi Keamanan

Bagian ini mendefinisikan cara-cara memverifikasi bahwa aplikasi menggunakan data atau file yang digunakan untuk menyimpan informasi konfigurasi penting secara aman. Perlindungan terhadap konfigurasi ini sangatlah penting. Tabel di bawah ini mendefinisikan persyaratan yang berlaku untuk setiap dari empat tingkat verifikasi.

Tabel 12 - Persyaratan Konfigurasi Keamanan (V12)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V12.1	Memastikan bahwa semua informasi konfigurasi yang relevan dengan keamanan disimpan di lokasi yang terlindung dari akses yang tidak sah.				✓	✓	✓
V12.2	Memastikan bahwa semua akses ke aplikasi ini ditolak jika aplikasi tidak dapat mengakses konfigurasi informasi keamanannya.				✓	✓	✓
V12.3	Pastikan bahwa semua perubahan pada konfigurasi pengaturan keamanan yang dikelola oleh aplikasi dicatat di event log keamanan.					✓	✓

¹⁴ Persyaratan ini mendekripsikan mekanisme yang dibutuhkan untuk melindungi diri dari serangan Cross Site Request Forgery (CSRF).



V12.4	Pastikan bahwa penyimpanan konfigurasi dapat dikeluarkan dalam format yang terbaca oleh manusia			✓
	untuk memfasilitasi audit.			

V13 - Persyaratan Verifikasi Pencarian Kode Berbahaya

Khusus untuk level 4, adalah untuk menjamin bahwa semua kode yang belum disentuh di level 3 harus dicari lagi akan adanya kemungkinan masuknya kode jahat.

Tabel 13 - Persyaratan Verifikasi Pencarian Kode Berbahaya OWASP ASVS (V13)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V13.1	Pastikan bahwa tidak ada kode berbahaya didalam kode yang dikembangkan atau dimodifikasi ketika membuat aplikasi. ¹⁵						✓
V13.2	Pastikan bahwa integritas kode interpreted, libraries, executable, dan file konfigurasi diverifikasi menggunakan checksum atau hash.						✓

V14 - Persyaratan Verifikasi Keamanan Internal

Bagian ini mendefinisikan cara-cara untuk memeriksa bahwa ada usaha dari aplikasi tersebut untuk melindungi diri sendiri dari kesalahan implementasi atau konfigurasi.

¹⁵ Mis. Periksa pemanggilan fungsi waktu untuk mencari bom waktu, pintu-belakang, Easter eggs, logika yang tidak benar dalam transaksi finansial, dan kode jahat tipe lainnya.



Tabel 14 - Persyaratan Keamanan Internal OWASP (V14)

	Persyaratan Verifikasi	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V14.1	Pastikan bahwa aplikasi melindungi atribut pengguna dan data dan informasi kebijakan yang digunakan oleh kontrol akses dari akses tanpa otorisasi atau modifikasi.					✓	✓
V14.2	Pastikan bahwa antarmuka kontrol keamanan cukup sederhana untuk digunakan sehingga pengembang cenderung untuk menggunakannya dengan benar.						✓
V14.3	Memastikan bahwa aplikasi dengan benar melindungi variabel dan sumber daya bersama dari akses bersamaan yang tidak sesuai.						✓



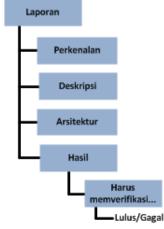
Persyaratan Pelaporan Hasil Verifikasi

Sebuah laporan OWASP ASVS berisi sebuah penjelasan tentang aplikasi yang dianalisa terhadap persyaratan

OWASP ASVS untuk tingkat tertentu. Laporan adalah juga dokumendokumen hasil analisa, termasuk remediasi dari kelemahan-kelemahan yang dibutuhkan.

Persyaratan laporan ASVS mendefiniskan jenis informasi yang dibutuhkan ada dalam laporan. Persyaratan laporan ASVS tidak mendefinisikan struktur, organisasi atau format dari laporan. Persyaratan laporan ASVS tidak membatasi tambahan informasi jika hendak dimasukkan dalam laporan.

Jenis informasi yang disyaratkan oleh setiap set dari persyaratan pelaporan ASVS dapat diberi nama, diformat dan disusun sesuai keinginan yang melakukan verifikasi. Persyaratan pelaporan ASVS terpenuhi sepanjang informasi yang disyaratkan ada. Sebuah laporan wajib memasukkan semua materi yang penting untuk pembaca dapat mengerti analisa yang dilakukan dan hasil analisa, termasuk infromasi konfigurasi dan potongan-potongan kode program, seperti yang digambarkan dalam gambar disamping, yang mungkin dapat dipakai pada saat pembuatan kerangka laporan.



Gambar 11 - Isi dari Laporan

R1 - Pembukaan Laporan

- R1.1 Pembukaan laporan sebaiknya menyediakan informasi yang cukup untuk mengidentifikasikan baik laporan maupun aplikasi yang menjadi subjek dalam laporan.
- R1.2 Pembukaan laporan sebaiknya merangkum *confidence* secara menyeluruh terhadap keamanan dalam aplikasi yang dimaksud.
- R1.3 Pembukaan laporan sebaiknya mengidentifikasikan Risiko bisnis kunci yang berhubungan dengan pengoperasian aplikasi.
- R1.4 Pembukaan laporan sebaiknya mengidentifikasikan aturan main yang berhubungan dengan verifikasi yang dilakukan atau apapun yang menjadi batasan dalam cakupan verifikasi.

R2 - Deskripsi Aplikasi

R2.1 Deskripsi aplikasi sebaiknya memberikan deskripsi tentang aplikasi untuk membantu penjelasn yang cukup tentang pengoperasian dan lingkungan dimana aplikasi beroperasi.

R3 - Arsitektur Keamanan Aplikasi

R3.1 Arsitektur keamanan aplikasi sebaiknya menyediakan tambahan informasi yang lebih detil yang menjelaskan aplikasi sebagai langkah pertama dalam memberikan keyakinan kepada pembaca dari laporan analisa yang dilakukan adalah lengkap dan akurat. Bagian ini dari laporan menyediakan konteks untuk analisa. Informasi yang dipresentasikan dalam bagian ini akan digunakan untuk mengidentifikasikan ketidak-konsistenan. Bagian dari laporan ini sebaiknya menyediakan rincian pada berbagai tingkatan, tergantung pada tingkatan Level OWASP ASVS dimana analisa dilakukan. Detil akan berbeda sesuai dengan tingkatan.



R4 - Hasil Verifikasi

R4.1 Hasil verikasi ini sebaiknya menyajikan hasil analisa yang sudah dilakukan sesuai dengan bagian dari standar "Persyaratan Verifikasi", termasuk deskripsi remediasi dari kelemahan-kelemahan yang disyaratkan sebagai berikut:

Tabel 15 - Isi dari Laporan Hasil Verifikasi OWASP ASVS

Tingkat	LULUS	GAGAL
Hasil Tingkat 1	 Putusan Konfiguasi alat (tool) (jika alat dapat melakukan pemeriksaan)atau putusan pembenaran (argumentasi untuk kelengkapan dan keakuratan, penyediaan bukti spesifik) Sebuah pemetaan kapabilitas alat otomatis untuk persyaratan detil verifikasi terkait Sebuah deskripsi konfigurasi alat dan sebuah pemetaan kemampuan alat yang perlu disediakan sebagai bagian dari laporan 	 Putusan Lokasi (URL dengan parameter dan atau jalur sumber berkas, nama dan nomor baris) Deskripsi (termasuk informasi konfigurasi yang sesuai) Peringkat Risiko¹⁶ Pembenaran Risiko
Hasil Tingkat 2 - 4	 Sebuah deskripsi konfigurasi alat dan swajib disediakan sebagai bagian dari l Putusan Pembenaran Putusan (argumentasi untuk kelengkapan dan keakuratan, penyediaan bukti spesifik) 	 Putusan Lokasi (URL dengan parameter dan atau jalur sumber berkas, nama dan nomor baris) Deskripsi (termasuk jalur melalui komponen aplikasi dan tahapan untuk mereproduksi) Peringkat Risiko (lihat Methodologi Peringkat Risiko OWASP) Pembenaran Risiko

¹⁶ Untuk informasi tambahan mengenai pengidentifikasian risiko dan estimasi risiko terkait dengan kelemahan, lihat *Testing Guide* (OWASP, 2008).



Glosarium

Application Security Verification Standard (ASVS) - Standar OWASP yang mendefinisikan tingkat keamanan aplikasi menjadi 4 level.

Arsitektur Keamanan -Sebuah abstraksi dari desain aplikasi yang mengindentifikasikan dan menjelaskan dimana dan bagaimana kontrol keamanan digunakan, serta menjabarkan juga lokasi dan kesensitifan data aplikasi dan pengguna.

Bom Waktu (Time Bomb) - Tipe kode jahat yang berjalan ketika waktu atau tanggal tertentu.

Daftar Hitam (*Blacklist*) - Daftar yang memuat misalnya data atau operasi yang tidak boleh dijalankan, misalnya daftar alamat IP yang tidak boleh membuka koneksi.

Daftar Putih (Whitelist) - Daftar yang memuat misalnya data atau operasi yang boleh dijalankan, misalnya daftar alamat IP yang boleh membuka koneksi.

FIPS 140-2 - Sebuah standar yang dapat digunakan untuk memverifikasi desain dan implementasi sebuah modul kriptografi.

Keamanan Aplikasi - Yang dimaksud disini adalah kemanan yang berfokus pada lapisan Aplikasi, jika Anda melihat sistem sesuai dengan pemisahan OSI Model, dibanding berfokus ke platform atau jaringan dibawahnya.

Keamanan Komunikasi- Pengamanan terhadap data aplikasi ketika dalam perjalanan antara komponen-komponen aplikasi, klien dan server, serta dengan sistem eksternal.

Kode Jahat (Malicious Code) - Kode yang disisipkan kedalam aplikasi pada tahap pengembangan tanpa sepengetahuan pemilik aplikasi untuk mengakali kebijakan keamanan yang diinginkan. Tidak sama dengan malware!

Komponen Aplikasi - Masing-masing atau kumpulan dari kode sumber, libraries, dan hasil kompilasi, yang didefinisikan oleh pemverifikasi untuk setiap aplikasi.

Konfigurasi Keamanan - Konfigurasi di aplikasi berjalan yang mempengaruhi bagaimana kontrol keamanan ditegakkan.

Kontrol Akses (Access Control)- Sebuah cara untuk membatasi akses kepada dokumen, fungsi, URL, dan data berdasarkan identitas dari pengguna, atau dari grup dimana pengguna tersebut termasuk didalamnya.

Kontrol Keamanan - Sebuah fungsi atau komponen yang melakukan pemeriksaan keamanaan (mis. Pemeriksaan kontrol akses) atau menghasilkan sebuah tindakan keamanan ketika dipanggil (mis. Menghasilkan rekaman jejak audit).

Kriteria Umum (Common Criteria) - Standar yang bisa dipakai sebagai basis untuk memverifikasi desain dan implementasi kontrol-kontrol keamanan dalam sebuah produk IT.

Laporan Verifikasi Keamanan Aplikasi - Laporan yang ditulis pemverifikasi, berisikan hasil pengujian secara umum dan analisa pendukungnya.

Malware - Executable yang disisipkan kedalam aplikasi ketika berjalan tanpa sepengetahuan pemilik atau administrator aplikasi.

Modul Kriptografi - Hardware, software, dan/atau firmware yang melakukan operasi kriptografi dan/atau menghasilkan kunci kriptografis.

Open Web Application Security Project (OWASP) - Adalah sebuah komunitas global dan terbuka yang peduli dengan kualitas keamanan aplikasi. Misi kami adalah memberi informasi tentang keamanan yang dapat dibaca semua orang, sehingga keputusan dapat diambil dengan dasar informasi yang benar.

Otentikasi (Authentication)- pemeriksaan bahwa identitas seorang pengguna aplikasi adalah benar.

OWASP Enterprise Security API (ESAPI) - metode-metode keamanan yang dapat digunakan secara bebas dan terbuka oleh pengembang. Lihat: http://www.owasp.org/index.php/ESAPI



OWASP Risk Rating Methodology - Sebuah metodologi pemeringkatan risiko yang sudah dibuat khusus untuk keamanan aplikasi. Lihat: http://www.owasp.org/index.php/How_to_value_the_real_risk

OWASP Testing Guide - Dokumen yang menjabarkan langkah-langkah yang diperlukan dalam membuat atau mengoperasikan sebuah program untuk pengujian keamanan. Lihat: http://www.owasp.org/index.php/Category:OWASP_Testing_Project

OWASP Top Ten - Dokumen yang menjabarkan cacat-cacat keamanan aplikasi Web yang paling kritis. Lihat: http://www.owasp.org/index.php/Top10

Pemodelan Ancaman - Sebuah teknik untuk menyempurnakan arsitektur kemanan aplikasi dengan cara mengidentifikasi ancaman, agen ancaman, zona kemanaan, kontrol keamanan, dan aset teknis dan bisnis.

Pintu Belakang (*Back Doors*) - Sebuah potongan kode yang membuat lubang untuk masuk ke aplikasi secara tidak benar, biasanya ditanam secara diam-diam oleh pengembang.

Serangan Denial of Service (DOS) - Membanjiri aplikasi dengan permintaan diluar kapasitasnya.

Serangan Salami (Salami Attack) - Tipe kode jahat yang mencuri uang dengan cara mengutip dalam jumlah kecil sehingga tidak terdeteksi.

Sistem Eksternal - Aplikasi server atau layanan yang bukan menjadi bagian dari aplikasi.

Target Verifikasi - Aplikasi yang diuji keamanannya, disebut "TOV" di OWASP ASVS.

Telur Paskah (Easter Eggs) - Sebuah jenis kode jahat yang berjalan hanya ketika menemui input tertentu.

Validasi Keluaran - Kanonikalisasi dan validasi terhadap keluaran yang dihasilkan aplikasi, biasanya terhadap peramban atau sistem eksternal.

Validasi Masukan - Kanonikalisasi dan validasi terhadap masukan dari pengguna.

Verifikasi Desain - Pemeriksaan secara teknis terhadap arsitektur keamanan sebuah aplikasi.

Verifikasi Dinamis - Penggunaan tools yang mencari tanda kerentanan, saat aplikasi berjalan.

Verifikasi Internal - Pemeriksaan secara teknis terhadap aspek tertentu dari arsitektur keamanan sebuah aplikasi seperti yang terdefinisikan di OWASP ASVS.

Verifikasi Keamanan Aplikasi - Pengujian secara teknis keamanan sebuah aplikasi menggunakan OWASP ASVS.

Verifikasi Statis - Penggunaan tools yang mencari tanda kerentanandi kode sumber, bukan saat aplikasi berjalan.

Verifikasi Terotomatisasi - Penggunaan dari tools terotomatisasi (entah dinamis atau statis) yang mencari karakteristik-karakteristik khusus dari kerentanan.

Verifikatur - Orang atau sekelompok orang yang menguji aplikasi sesuai persyaratan OWASP ASVS.



Dari Sini Ke <u>Mana ?</u>

OWASP adalah situs paling terkemuka untuk informasi keamanan aplikasi internet. Situs OWASP menjadi sarang bagi banyak proyek, forum, blog, presentasi, tools, dan paper. OWASP juga mengadakan konferensi besar mengenai keamanan aplikasi berbasis web, dan memiliki lebih dari 80 kelompok pendukung regional. Halaman situs OWASP ASVS dapat ditemukan di http://www.owasp.org/index.php/ASVS

Projek OWASP berikut ini hampir pasti akan berguna untuk siapapun yang menggunakan standar ini:

- 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

Begitu juga, situs-situs web dibawah ini hampir pasti berguna buat pengguna standar ini:

- OWASP http://www.owasp.org
- MITRE Common Weakness Enumeration Vulnerability Trends, http://cwe.mitre.org/documents/vuln-trends.html
- PCI Security Standards Council publishers of the PCI standards, relevant to all organizations
 processing or holding credit card data, 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.