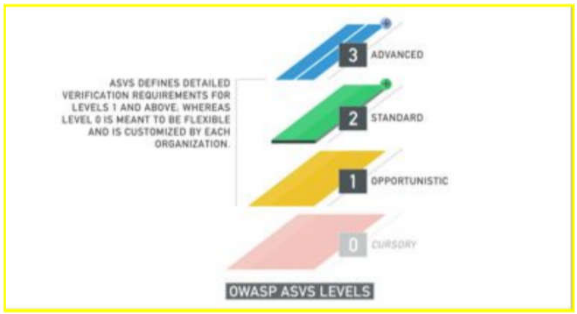
OWASP ASVS

Application Security Verification Standard

Antonio Fontes

Chapter Meeting - 19 octobre 2015

OWASP Geneva Chapter



Bio

Antonio Fontes

Sécurité et protection des données dans les opérations de développement et acquisition de logiciels

L7 Sécurité Sarl, Directeur

OWASP Suisse, membre du Comité

OWASP Genève, *co-chapter leader*

sur Twitter: @starbuck3000

Agenda

- Contexte et problématique
- Présentation de l'outil
- Exemples de rapports
- Opportunités

Contexte

Le projet ASVS adresse la problématique de **l'évaluation de la sécurité** des applications s'exécutant au-dessus du protocole HTTP (web, mobile, service web, etc.).

Problématique

- Quel critère pour évaluer la sécurité d'une application?
  - Qu'est-ce qui définit une application répondant aux exigences de sécurité?
  - Comment justifier l'évaluation de X critères pour l'application A et X-Y critères pour l'application B?
  - Peut-on se contenter du Top 10?

Problématique

- Standardiser quantitativement et qualitativement l'évaluation de sécurité d'une application **dans le temps**:
  - En quoi l'évaluation J est-elle comparable à l'évaluation J+1?

Problématique

- Standardiser quantitativement et qualitativement l'évaluation de sécurité d'une application **dans l'organisation**:
  - En quoi l'évaluation de l'application A est-elle comparable à celle de l'application B?
  - Comment gérer le cas des applications à impact/risque différent?
    - Si A est moins risqué, considérer un critère «allégé»?

Problématique

- Standardiser quantitativement et qualitativement l'évaluation de sécurité d'une application **par les fournisseurs**:
  - Cas des fournisseurs multiples (portefeuille d'applications) et des rotations de fournisseurs (application testée par divers fournisseurs).
  - S'assurer que tous les fournisseurs s'appuient sur un critère **commun** et **plus fourni** que 10 critères (Top 10)

Problématique

- Standardiser quantitativement et qualitativement **les différentes méthodes d'évaluation** de sécurité d'une application :
  - Revue / contrôle du concept
  - Revue / contrôle du code source
  - Revue / contrôle du système opérationnel

Problématique

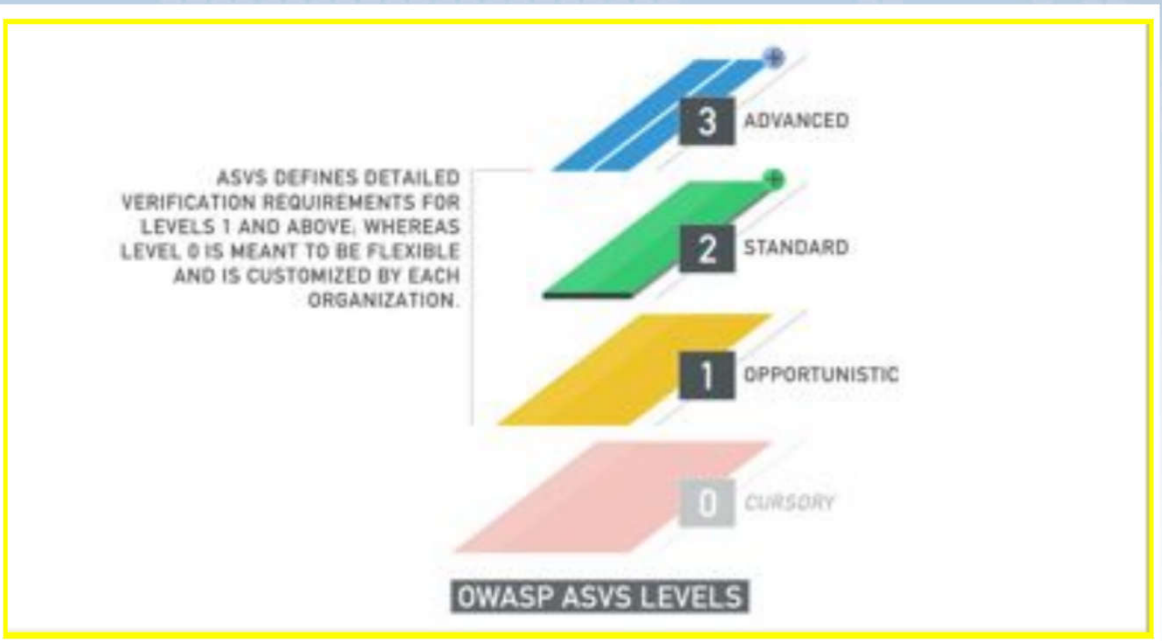
- Mais encore:
  - Augmenter la finesse des contrôles tout en maintenant la composante «agnostique»
  - Définir le critère de recette de l'application (cas de l'externalisation du développement ou des tests)

Problématique

- Comment communiquer avec l'exécutif?
  - S'accorder sur le profil de risque:
    - Niveau *opportuniste*: requis pour tout système
    - Niveau *standard*: requis pour tout système connecté sur Internet
    - Niveau *avancé*: systèmes à haut impact / haut risque

ASVS

Builder Project



ASVS: A… S… V… S…?

- Application Security Verification Standard
- Traduction: *standard d'évaluation de sécurité logicielle*

Auteurs du projet

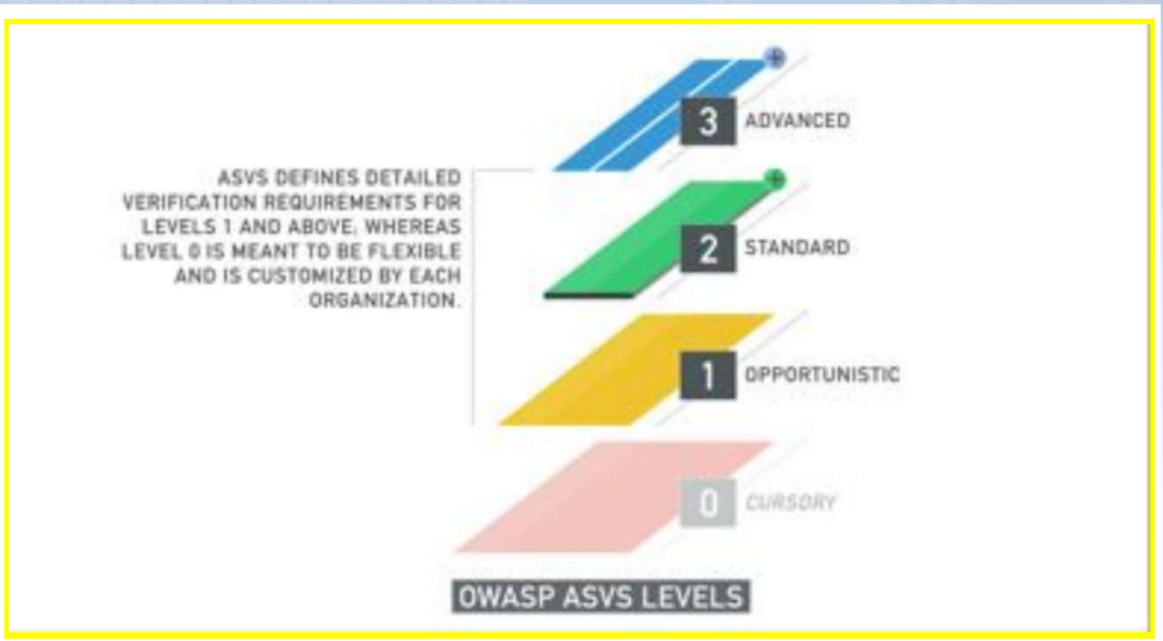
Project Leads	Lead Authors	Contributors and reviewers
Andrew van der Stock Daniel Cuthbert	Jim Manico	Boy Baukema Ari Kesäniemi Colin Watson François-Eric Guyomarc'h Cristinel Dumitru James Holland Gary Robinson Stephen de Vries Glenn Ten Cate Riccardo Ten Cate Martin Knobloch Abhinav Sejpal David Ryan Steven van der Baan Ryan Dewhurst Raoul Endres Roberto Martelloni

ASVS: comment cela se présente-t-il?

- Document téléchargeable (gratuit): https://www.owasp.org/index.php/ASVS
- Ou imprimable (payant):
- Dernière version pas encore disponible en imprimé
- 4 niveaux de conformité
- 16 listes de contrôles, 180 contrôles
- Anglais, Espagnol, Japonais, Allemand

ASVS: niveaux de sécurité

- 3 niveaux:
  - Opportuniste
  - Standard
  - Avancé
- + niveau 0:
  - *Superficiel*



ASVS: niveaux de sécurité

- Une guidance sur la détermination du niveau de sécurité attendu est incluse.
- Superficiel (0): niveau par défaut, qualifie l'absence de toute approche formelle d'évaluation de sécurité.
- Attention: une application de niveau 0 n'est pas pour autant vulnérable!

ASVS: niveaux de sécurité

- Opportuniste (1): un effort minimal, mais formel, a été investi dans l'évaluation de la sécurité de l'application.
- Une application de niveau 1 bénéficie de contrôles destinés à la défense contre des adversaires opportunistes (généralement, usant d'outils de recherche automatisée de vulnérabilités).

ASVS: niveaux de sécurité

- Standard (2): des moyens sont formellement investis dans l'évaluation de sécurité de l'application.
- Une application de niveau 2 bénéficie de contrôles destinés à la défense contre des adversaires motivés mais ne disposant pas d'un avantage financier, logistique ou technologique.

ASVS: niveaux de sécurité

- Avancé (3): des moyens sont investis dans une évaluation formelle de la sécurité de l'application, de sa conception à son exploitation.
- Une application de niveau 3 bénéficie de contrôles destinés à la défense contre des adversaires organisés et disposant de soutiens financiers, logistiques ou/et technologiques.

ASVS: Fonctions évaluées

- V1: Architecture, design et menaces (10 contr.)
- V2: Authentification (27 contr.)
- V3: Sessions (13 contr.)
- V4: Contrôle d'accès (12 contr.)
- V5: Validation de données (21 contr.)
- V6: -
- V7: Cryptographie (10 contr.)
- V8: Erreurs et exceptions (12 contr.)
- V9: Protection de données (11 contr.)
- V10: Communications (13 contr.)
- V11: Configuration HTTP (8 contr.)
- V12: -
- V13: Développement malveillant (2 contr.)
- V14: -
- V15: Logique métier (2 contr.)
- V16: Fichiers et autres ressources (9 contr.)
- V17: Applications mobiles (11 contr.)
- V18: Services web (10 contr.)
- V19: Configuration générale (9 contr.)

ASVS: liste V2, authentification

#	Description	1	2	3	Since
2.1	Verify all pages and resources by default require authentication except those specifically intended to be public (Principle of complete mediation).	✓	✓	✓	1.0
2.2	Verify that all password fields do not echo the user's password when it is entered.	✓	✓	✓	1.0
2.4	Verify all authentication controls are enforced on the server side.	✓	✓	✓	1.0
2.6	Verify all authentication controls fail securely to ensure attackers cannot log in.	✓	✓	✓	1.0
2.7	Verify password entry fields allow, or encourage, the use of passphrases, and do not prevent long passphrases/highly complex passwords being entered.	✓	✓	✓	3.0
2.8	Verify all account identity authentication functions (such as update profile, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.	✓	✓	✓	2.0
2.9	Verify that the changing password functionality includes the old password, the new password, and a password confirmation.	✓	✓	✓	1.0
2.12	Verify that all suspicious authentication decisions are logged. This should include requests with relevant metadata needed for security investigations.		✓	✓	2.0
2.13	Verify that account passwords make use of a sufficient strength encryption routine and that it withstands brute force attack against the encryption routine.		✓	✓	3.0
2.16	Verify that credentials are transported using a suitable encrypted link and that all pages/functions that require a user to enter credentials are done so using an	✓	✓	✓	3.0

ASVS: liste V7, cryptographie

#	Description	1	2	3	Since
7.2	Verify that all cryptographic modules fail securely, and errors are handled in a way that does not enable oracle padding.	✓	✓	✓	1.0
7.6	Verify that all random numbers, random file names, random GUIDs, and random strings are generated using the cryptographic module's approved random number generator when these random values are intended to be not guessable by an attacker.		✓	✓	1.0
7.7	Verify that cryptographic algorithms used by the application have been validated against FIPS 140-2 or an equivalent standard.	✓	✓	✓	1.0
7.8	Verify that cryptographic modules operate in their approved mode according to their published security policies.			✓	1.0
7.9	Verify that there is an explicit policy for how cryptographic keys are managed (e.g., generated, distributed, revoked, and expired). Verify that this key lifecycle is properly enforced.		✓	✓	1.0
7.11	Verify that all consumers of cryptographic services do not have direct access to key material. Isolate cryptographic processes, including master secrets and consider the use of a hardware key vault (HSM).			✓	3.0
7.12	*Personally Identifiable Information* should be stored encrypted at rest and ensure that communication goes via protected channels.		✓	✓	3.0
7.13	Verify that where possible, keys and secrets are zeroed when destroyed.		✓	✓	3.0

ASVS: pour quel usage?

- Référentiel détaillé de contrôles:
  - Utile aux testeurs / auditeurs
  - Formalise ce qui est à considérer comme «standard» au niveau de la communauté

ASVS: pour quel usage?

- Mesure, indicateurs-clés:
  - Portefeuille d'applications soumises à la vérification standard / avancée / opportuniste
  - Risque corrélé:
    - Nb. Jours depuis vérification vs. dernière modification
    - Niveau de conformité atteint vs. attendu
    - Etc.

ASVS: pour quel usage?

- Standardisation des évaluations:
  - Standardisation inter-fournisseurs
  - Standardisation inter-applications
  - Standardisation intra-cycle (évaluation de concept, de code souce, des binaires, test d'intrusion)

ASVS: pour quel usage?

- Outil contractuel pour les fournisseurs / client:
  - Spécification des clauses d'acceptation / recette dans les contrats d'acquisition / location.
  - Mise en référentiel des tests de sécurité réalisés par le fournisseur

ASVS: points forts

- Recherche d'exhaustivité des contrôles tout en restant agnostique
- Réutilisable à toutes les étapes de contrôle ou d'exécution (concept, code, production)
- Suivi périodique de la conformité
- «clés en mains»:
  - Peu de contraintes / prérequis
  - Projet actif (maintenu)

ASVS: points faibles

- Liste de contrôles trop fournie vs. liste de contrôles insuffisante (p.ex.: cryptographie, protocoles d'authentification fédérée, etc.)
- Regroupement par fonctions au lieu de scénarios
- Aucune automatisation fournie

ASVS: qui l'adopte?

- Pas de liste "officielle"
- Les sociétés de test d'intrusion l'utilisent (ou utilisent le OWASP Testing Guide)

- Vous?

ASVS: un guide, pas un standard!

Re: [OWASP ASVS] ASVS3: 16.9 client side technologies

Jim Manico

To: Boy Baukema,owasp-application-security-verification-standard@lists.owasp.org

Oct 26, 2015, 1:23 PM

Then fork ASVS for your specific use case where Flash is ok.... ASVS is really made to be forked by every team.

Aloha,
- Jim

On 10/26/15 5:07 PM, Boy Baukema wrote:

> Hi,
>
> Could ASVS3s level 1 requirement 16.9 be weakened?
> Because:
>
> > Do not use Flash, Active-X, Silverlight, NACL, client-side Java or other client side technologies not supported natively via W3C browser standards.
>
> Makes most modern web applications that rely on Cross-Browser polyfills fail a simple level 1 verification.

Historique - Evolutions

- Publication originale: 2009, version 1
- Version 2: 2014
- Version 3: 9 octobre 2015

Questions?

?

antonio.fontes@owasp.org
@starbuck3000

Merci.

Liens

Projet ASVS

https://www.owasp.org/index.php/ASVS

Téléchargement (en Anglais):

https://www.owasp.org/images/6/67/OWASPApplicationSecurityVerificationStandard3.0.pdf

Checklist ASVS au format Excel (Dave Ryan)

https://docs.google.com/spreadsheets/d/108gxWfwNNpTB-kpgIgRSUacahreYPc5eSiG72zKQYmM/edit?usp=sharing