OWASP Top Ten 2013

Les dix risques de sécurité applicatifs Web les plus critiques

OSSIR Paris / 14 janvier 2014

Guillaume Lopes – Consultant Sécurité
Guillaume.Lopes@Intrinsec.com

Qui suis-je ?

- Guillaume Lopes
  - → Consultant Sécurité
    - ✓ Tests d'intrusion
    - ✓ Audits (architecture, configuration et organisationnel)
  - → OWASP addict depuis 2008
    - ✓ OWASP Testing Guide et Top Ten
  - → Participation à la traduction française de l'OWASP Top Ten 2013
- Intrinsec : acteur historique de la sécurité des SI (1995)
  - → Hébergement et infogérance des SI
  - → Sécurité de l'information
    - ✓ Pentest / Audit / Conseil / SOC

A propos de l'OWASP Top Ten

- L'OWASP Top Ten est un document de sensibilisation
  - → Il ne s'agit pas d'un standard / checklist / norme / etc.
  - → Même s'il est référencé par de nombreuses normes ou organisations
    - ✓ MITRE, PCI DSS, DISA, etc.
- La première version date de 2003
  - → Les versions 2004 et 2007 sont des mises à jour mineures
  - → La version 2010 a été complètement réorganisée afin de parler de risques et non de vulnérabilités
  - → La version 2013 est une mise à jour mineure
- Cette version reste dans la continuité de la version 2010

A qui s'adresse ce document ?

- Ce document a pour objectif de sensibiliser sur les 10 risques les plus critiques rencontrés sur des applications Web

- Le public concerné est varié
  - Développeurs
  - Designers
  - Architectes
  - Managers
  - Décideurs
  - Auditeurs
  - ...

Comment est conçu le Top Ten ?

- Le Top Ten se base sur les statistiques de vulnérabilités des organisations suivantes
  - Aspect Security
  - HP (Fortify and WebInspect)
  - Minded Security
  - Softtek
  - Trustwave SpiderLabs
  - Veracode
  - WhiteHat Security
- En chiffres
  - + 500 000 vulnérabilités
  - + 100 organisations
  - + 1000 applications Web

Comment est conçu le Top Ten ?

- Les données 2011/2012 du MITRE n'étaient pas disponibles lors de la parution de la version 2013 de l'OWASP Top Ten

- La démarche de création du Top Ten 2013 est présentée à
  - → www.owasp.org/index.php/Top_10_2013/ProjectMethodology

- Si vous souhaitez fournir des statistiques, vous pouvez contacter Dave Wichers (dave.wichers@owasp.org)

Différences entre 2010 et 2013

- 1 risque a été ajouté
  - A9 – Utilisation de composants avec des vulnérabilités connues
  - Ce problème était mentionné dans le risque A6 – Mauvaise configuration de sécurité de la version 2010
- 2 risques ont été fusionnés
  - A7 – Stockage cryptographique non sécurisé
  - A9 – Protection insuffisante de la couche de transport
  - Ces risques deviennent A6 – Exposition de données sensibles
- 1 risque a été élargi
  - A8 – Manque de restriction d'accès à une URL devient A7 – Manque de contrôle d'accès au niveau fonctionnel
- 7 risques ont été réorganisés

Différences entre 2010 et 2013

OWASP Top 10 – 2010 (Précédent)	OWASP Top 10 – 2013 (Nouveau)
2010-A1 – Injection	2013-A1 – Injection
2010-A2 – Cross Site Scripting (XSS)	2013-A2 – Violation de gestion d'authentification et de session
2010-A3 – Violation de gestion d'authentification et de session	2013-A3 – Cross Site Scripting (XSS)
2010-A4 – Références directes non sécurisées à un objet	2013-A4 – Références directes non sécurisées à un objet
2010-A5 – Falsification de requête intersites (CSRF)	2013-A5 – Mauvaise configuration sécurité
2010-A6 – Mauvaise configuration sécurité	2013-A6 – Exposition de données sensibles
2010-A7 – Stockage cryptographique non sécurisé	2013-A7 – Manque de contrôle d'accès au niveau fonctionnel
2010-A8 – Manque de restriction d'accès à une URL	2013-A8 – Falsification de requête intersites (CSRF)
2010-A9 – Protection insuffisante de la couche de transport	2013-A9 – Utilisation de composants avec des vulnérabilités connues (Nouveau)
2010-A10 – Redirections et renvois non validés (Nouveau)	2013-A10 – Redirections et renvois non validés

Evolution de l'OWASP Top Ten

- Une comparaison des différentes versions de l'OWASP Top Ten a été réalisée par Christian Heinrich
  - → https://github.com/cmlh/OWASP-Top-Ten-2013/

OWASP Top Ten Entries (Unordered)	Releases				
	2003	2004	2007	2010	2013
Unvalidated Input	A1	A1[9]	x	x	x
Buffer Overflows	A5	A5	x	x	x
Denial of Service	x	A9[2]	x	x	x
Injection	A6	A6[3]	A2	A1[10]	A1
Cross Site Scripting (XSS)	A4	A4	A1	A2	A3
Broken Authentication and Session Management	A3	A3	A7	A3	A2
Insecure Direct Object Reference	x	A2	A4[11]	A4	A4
Cross Site Request Forgery (CSRF)	x	x	A5	A5	A8
Security Misconfiguration	A10	A10[3][5]	x	A6	A5
Missing Functional Level Access Control	A2	A2[1]	A10[13]	A8	A7[16]
Unvalidated Redirects and Forwards	x	x	x	A10	A10
Information Leakage and Improper Error Handling	A7	A7[14][4]	A6	A6[8]	x
Malicious File Execution	x	x	A3	A6[8]	x
Sensitive Data Exposure	A8	A8[6][5]	A8	A7	A6[17]
Insecure Communications	x	A10	A9[7]	A9	x
Remote Administration Flaws	A9	x	x	x	x
Using Known Vulnerable Components	x	x	x	x	A9 [18][19]

Présentation des risques

- Chaque risque défini dans l'OWASP Top Ten est présenté en 5 parties
  - Le niveau de risque selon
    - ✓ L'exploitabilité, la prévalence et la détection de la vulnérabilité
    - ✓ Les agents de menace et les impacts métier sont à déterminer au cas par cas
  - Suis-je vulnérable ?
    - ✓ Fournit des indicateurs sur les moyens d'identifier les vulnérabilités associées
  - Comment s'en prémunir ?
    - ✓ Indique comment se protéger des vulnérabilités associées
  - Exemples de scénarios d'attaque
  - Références
    - ✓ propres à l'OWASP ou externes

Présentation des risques

RISQUES	Agents de Menace	Vecteurs d'attaque	Vulnérabilité		Impacts Techniques	Impacts Métier
		Exploitabilité	Prévalence	Détection	Impact	
A1-Injection	Spécifique à l'Application	FACILE	COMMUNE	MOYENNEMENT	SÉVÈRE	Spécifique à l'Application
A2-Auth et Sess	Spécifique à l'Application	MOYENNE	RÉPANDUE	MOYENNEMENT	SÉVÈRE	Spécifique à l'Application
A3-XSS	Spécifique à l'Application	MOYENNE	TRÈS RÉPANDUE	FACILEMENT	MODÉRÉ	Spécifique à l'Application
A4-Réf non sécu.	Spécifique à l'Application	FACILE	COMMUNE	FACILEMENT	MODÉRÉ	Spécifique à l'Application
A5-Config	Spécifique à l'Application	FACILE	COMMUNE	FACILEMENT	MODÉRÉ	Spécifique à l'Application
A6-Données	Spécifique à l'Application	DIFFICILE	RARE	MOYENNEMENT	SÉVÈRE	Spécifique à l'Application
A7-ACL Fonc.	Spécifique à l'Application	FACILE	COMMUNE	MOYENNEMENT	MODÉRÉ	Spécifique à l'Application
A8-CSRF	Spécifique à l'Application	MOYENNE	COMMUNE	FACILEMENT	MODÉRÉ	Spécifique à l'Application
A9-Composants	Spécifique à l'Application	MOYENNE	RÉPANDUE	DIFFICILEMENT	MODÉRÉ	Spécifique à l'Application
A10-Redirection	Spécifique à l'Application	MOYENNE	RARE	FACILEMENT	MODÉRÉ	Spécifique à l'Application

A1 - Injection

- L'injection consiste à envoyer des données non prévues à une application afin de détourner le comportement attendu
- Il existe différents types d'injections
  - → Injection SQL / LDAP / Xpath / NoSQL / XML / SMTP / OS / …
- Aujourd'hui, l'injection SQL reste la vulnérabilité la plus rencontrée et la plus critique
  - → De nombreuses applications sont vulnérables
  - → Les développeurs ne sont pas assez sensibilisés ou se reposent sur des solutions tierces non validées (bibliothèques externes, WAF, etc.)
- Impact
  - → Accès complet en lecture / écriture à la base de données
  - → Voire compromission de l'équipement

A1 - Injection

Recommandations

- Il est préférable d'utiliser une API fournissant une interface paramétrable, notamment
  - ✓ Requêtes paramétrées
  - ✓ Procédures stockées
- En l'absence d'API paramétrable, il est recommandé d'échapper les caractères spéciaux
- Si possible effectuer une validation par « whitelist » des données utilisateurs
- Restreindre les privilèges de la base de données

Plus d'informations

www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

A2 – Violation de gestion d'authentification et de session

- Ce risque couvre différents aspects
  - L'identifiant de session de l'utilisateur
  - Les fonctionnalités de gestion de compte (oubli du mot de passe, etc.)
  - Le transport des informations d'authentification ou de session sur des canaux non chiffrés
- Les vulnérabilités souvent rencontrées
  - Identifiant de session prédictible (cookie = login utilisateur)
  - Identifiant de session transitant dans l'URL
  - Authentification ou envoi du cookie de session sur flux HTTP
  - Stockage des mots de passe des utilisateurs en clair
  - Fonctionnalité de récupération de mot de passe vulnérable
    - ✓ Il suffit de fournir le login de l'utilisateur pour changer de mot de passe
    - ✓ La question secrète est trop faible

A2 – Violation de gestion d'authentification et de session

- Recommandations
  - Le processus d'authentification doit être simple, centralisé et standardisé
  - Utiliser les mécanismes standards de gestion de session des langages sélectionnés
    - ✓ Il ne faut pas réinventer la roue
  - Assurer la confidentialité des identifiants de session et de connexion de l'utilisateur (SSL / TLS)
    - ✓ Il faut chiffrer l'ensemble des communications
  - Valider les fonctionnalités relatives au processus d'authentification
  - Stocker les mots de passe des utilisateurs de manière non réversible
  - Se prémunir contre les failles XSS
- Plus d'informations

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

A3 – Cross Site Scripting (XSS)

- Avec l'injection SQL, le XSS est la faille la plus répandue dans les applications Web
  - → Quasiment toutes les applications testées possèdent au moins un paramètre vulnérable au XSS
- Un XSS permet d'exécuter du code (JavaScript ou HTML) dans le contexte du navigateur de l'utilisateur
  - → Lié à un mauvais filtrage des entrées utilisateur de l'application
- Deux grands types de XSS
  - → Reflected : L'attaque s'effectue via l'envoi d'un lien forgé à l'utilisateur
  - → Stored : Le code malveillant est déposé directement sur l'application et exécuté lorsqu'un utilisateur visualise la page

A3 – Cross Site Scripting (XSS)

- Recommandations
  - Echapper toutes les données non fiables
  - Effectuer une validation positive des données (whitelist)
    - ✓ Quand cela est possible
  - Utiliser des bibliothèques externes pour filtrer les données
    - ✓ OWASP AntiSamy
    - ✓ Java HTML Sanitizer Project
  - Mettre en place Content Security Policy (CSP) sur votre application
    - ✓ Voir dernier MISC 71
- Plus d'informations

https://www.owasp.org/index.php/XSS_(Cross Site Scripting) Prevention Cheat Sheet

A4 – Références directes non sécurisées à un objet

- Risque lié à un mauvais contrôle d'accès aux ressources
  - Rencontré encore assez souvent sur les applications
  - L'accès à une ressource est contrôlé par un identifiant prédictible (identifiant numérique, nom de fichier, etc.)
- L'application ne s'assure pas que l'utilisateur possède les droits nécessaires afin d'accéder à la ressource qu'il demande
  - Lorsqu'il s'agit d'un identifiant numérique, il suffit d'itérer sur la valeur
  - Lorsqu'il s'agit d'un nom, il est possible de deviner les valeurs attendues

A4 – Références directes non sécurisées à un objet

- Recommandations
  - Implémenter des références indirectes par utilisateur ou par sessions
    - ✓ Via la bibliothèque ESAPI par exemple
  - Contrôler l'accès aux ressources
    - ✓ L'application doit s'assurer que l'utilisateur possède les droits nécessaires pour accéder à l'objet

A5 – Mauvaise configuration sécurité

- Ce risque couvre toutes les vulnérabilités liées à une configuration par défaut ou un manque de mises à jour
  - Présence d'une interface d'administration avec des comptes par défauts
    - ✓ JMX console / Drupal / Joomla / Typo3 / etc.
  - Présence d'une interface d'installation du CMS
  - Listage des répertoires activé
    - ✓ Identification de fichiers sensibles
  - Affichage d'erreur techniques verbeuses
  - Manque de mises à jour systèmes ou applicatives

- L'impact est varié et dépend de la vulnérabilité identifiée
  - De l'accès à certaines informations à la compromission de l'équipement

A5 – Mauvaise configuration sécurité

- Recommandations
  - → Mettre en place un durcissement des OS, applications, etc.
  - → Mettre en place une politique de mise à jour des différents composants de l'application

- L'utilisation de scans automatisés aide à détecter les mauvaises configurations et l'absence de correctifs de sécurité

A6 – Exposition de données sensibles

- Certaines données sensibles d'une application ne sont pas correctement stockées ou transmises

- Les cas classiques
  - Authentification de l'utilisateur sur des flux HTTP
  - Envoi du cookie de session sur des flux HTTP
  - Stockage en clair des données (mots de passe, données de cartes bleues, etc.)
  - Utilisation d'un algorithme aisément réversible (base64 par exemple)

A6 – Exposition de données sensibles

- Recommandations
  - Identifier les données sensibles manipulées par l'application
    - ✓ Types de données
    - ✓ Ou sont-elles stockées ? Conservées ?
  - Déterminer les méthodes de chiffrement à utiliser
    - ✓ Choisir des algorithmes éprouvés
    - ✓ Générer des clés robustes
    - ✓ Mise en place d'une gestion des clés
  - Stocker les mots de passe au moyen d'un algorithme adapté à cet usage
    - ✓ Bcryp / PBKDF2 / scrypt / etc.

A7 – Manque de contrôle d'accès au niveau fonctionnel

- Risque assez similaire à A4 – Références directes non sécurisées à un objet
  - → Focalisé sur l'accès à des fonctions non privilégiées
  - → Lié au contrôle d'accès / autorisation de l'application

- Les cas classiques
  - → Certaines fonctions ne sont pas affichées dans l'interface utilisateur
    - ✓ Il suffit de retrouver / connaitre le lien ou paramètre associé pour accéder à la fonction
  - → Certaines pages sont accessibles à des utilisateurs non authentifiés
    - ✓ L'application ne valide pas que l'utilisateur est authentifié

A7 – Manque de contrôle d'accès au niveau fonctionnel

- Recommandations
  - Restreindre les fonctions aux utilisateurs authentifiés si nécessaire
  - Vérifier les privilèges / droits de l'utilisateur lorsqu'il accède à une ressource
  - Interdire par défaut l'accès aux fonctionnalités de l'application

A8 – Falsification de requête intersite (CSRF)

- Cross Site Request Forgery (CSRF)
  - L'attaquant effectue une action à l'insu de l'utilisateur
    - ✓ Ajout d'une règle de redirection dans sa boite de messagerie par ex.
  - Il est possible de prévoir les actions possibles sur une application
  - Il faut forcer le navigateur à émettre une requête au nom de l'utilisateur
    - ✓ Via une balise image / une faille XSS / etc.

- Recommandations
  - Utilisation d'un jeton unique à chaque requête
  - Utilisation de CAPTCHAS dans certains cas
  - Projet OWAP CSRF Guard

- Plus d'informations

www.owasp.org/index.php/CSRF_Prevention_Cheat_Sheet

A9 – Utilisation de composants avec des vulnérabilités connues

- Nouveauté de cet OWASP Top Ten

- Aujourd'hui, lors du développement d'une application Web de nombreux composants tiers sont utilisés
  - → CMS / bibliothèques externes / frameworks / etc.

- Ces composants peuvent présenter des vulnérabilités connues et exploitables
  - → Il est simple d'identifier un manque de mise à jour via des outils automatisés

A9 – Utilisation de composants avec des vulnérabilités connues

Recommandations

- Utiliser du code uniquement validé et écrit par votre équipe de développement
  - ✓ Très peu réaliste ☺
- Mettre en place un processus de mise à jour sur les composants tiers
  - ✓ Identifier tous les composants installés, ainsi que leur version
  - ✓ Surveiller les bases de données publiques, les listes de diffusion, etc. afin d'identifier de nouvelles vulnérabilités
  - ✓ Effectuer des mises à jour sur les composants tiers

A10 – Redirections et renvois non validés

- Les applications Web utilisent des redirections et des renvois pour rediriger les utilisateurs
  - → La page de destination est spécifiée dans un paramètre
- Si le paramètre contenant la page de destination n'est pas vérifié par l'application, alors un attaquant peut rediriger l'utilisateur vers un site qu'il contrôle
  - → Principalement utilisé lors d'attaques de phishing
- Parfois, certains renvois permettent de contourner les autorisations mises en place sur l'application

A10 – Redirections et renvois non validés

- Recommandations
  - Dans la mesure du possible, éviter l'utilisation de redirections et de renvois
  - En cas d'utilisation de redirections ou de renvois
    - ✓ Ne pas utiliser de valeur de destination dans les paramètres utilisateur si possible
    - ✓ Sinon valider la valeur de destination

Conclusion

- Ne vous arrêtez pas à 10 !

- Développer de manière sécurisée
  - → Utiliser les guides de développement sécurisé de l'OWASP
    - ✓ https://www.owasp.org/index.php/Guide
    - ✓ https://www.owasp.org/index.php/Cheat_Sheets
  - → Former à la sécurité et aux attaques
    - ✓ WebGoat : https://www.owasp.org/index.php/Webgoat

Conclusion

- Effectuer des tests de sécurité et des revues de code
  - Utiliser le standard ASVS lors de vos audits
    - ✓ https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project
  - Utiliser l'OWASP Testing Guide (la version 4 devrait sortir bientôt)
    - ✓ https://www.owasp.org/index.php/OWASP_Testing_Project
  - Utiliser l'OWASP Code Review pour la revue
    - ✓ https://www.owasp.org/index.php/Code_Review_Guide

Merci de votre attention
Questions ?