OWASP Open Web Application Security Project | Standard

Mobile AppSec Verification

Version 1.1

(French Translation)

Project leaders: Sven Schleier and Jeroen Willemsen

Table des matières

Exigences Concernant Securite

Annexes

Avant-propos

Par Bernhard Mueller, OWASP Mobile Project. Les révolutions technologiques peuvent arriver vite. Il y a moins de dix ans, les smartphones n'étaient que des appareils poussifs avec de petits claviers - des jouets onéreux pour hommes d'affaire passionés par la tech. Pourtant, aujourd'hui, les smartphones font partie intégrante de notre vie. Nous dépendons d'eux pour les informations, la navigation et la communication, et ils sont omniprésents aussi bien dans notre vie professionnelle que dans notre vie sociale.

Chaque nouvelle technologie introduit de nouveaux risques liés à la sécurité, et se maintenir à l'état de l'art est l'un des principaux défis de l'industrie de la sécurité. Le côté défensif est toujours un peu en retard. Par exemple, le réflexe naturel de beaucoup est d'appliquer toujours les mêmes recettes : les smartphones ressemblant à de petits ordinateurs et les applications mobiles étant du logiciel assez classique, les exigences de sécurité doivent donc être à peu près identiques? Hélas, cela ne marche pas comme ça. Les systèmes d'exploitation des smartphones sont différents de ceux des ordinateurs traditionnels et les applications mobiles sont différentes des applications web. Par exemple, la méthode classique consistant à scanner un appareil à la recherche de virus en se basant sur leur signature n'a pas de sens dans les environnements mobiles modernes : ceci est non seulement incompatible avec le modèle de distribution des applications mobiles, mais c'est aussi techniquement impossible à réaliser en raison des restrictions "bac à sable". Par ailleurs, certaines classes de vulnérabilité telles que les débordements de tampon et autres failles XSS, sont moins pertinentes dans le contexte des applications mobiles ordinaires que, disons, pour les applications PC ou web (toutefois avec certaines exceptions).

Au fil du temps, notre industrie a progressé sur la connaissance de l'écosystème de menaces dans le mobile. En fait, la sécurité mobile est centrée sur la protection des données : les applications stockent nos données personnelles, nos photos, nos enregistrements, nos notes, les données concernant nos différents comptes, de l'information professionnelle, notre localisation et bien plus. Elles se comportent comme des clients qui nous connectent aux services que nous utilisons chaque jour et centralisent tous les messages que nous échangeons avec les autres. Parvenir à pirater un smartphone revient à avoir un accès illimité à la vie d'une personne. En prenant en compte que les appareils mobiles sont plus couramment perdus ou volés et que les logiciels malveillants sur mobile sont en augmentation, le besoin de protection des données devient d'autant plus important.

Un standard de sécurité pour les applications mobiles doit donc se focaliser sur la manière dont les applications mobiles gèrent, stockent et protectent les informations sensibles. Même si les systèmes d'exploitation mobiles modernes comme iOS et Android offrent de bonnes APIs concernant la sécurité du stockage des données et la communication, celles-ci doivent toutefois être correctement implémentées et utilisées afin d'être efficaces. Le stockage de données, la communication inter-processus, la bonne utilisation des API de cryptographie et la sécurité des communications réseau sont seulement quelques aspects sur lesquels il convient de se concentrer.

Une question importante en souffrance d'un consensus de la part de l'industrie de la sécurité est de savoir jusqu'à quel point il faut protéger la confidentialité et l'intégrité des données. Par exemple, la plupart d'entre nous serait d'accord sur la nécessité pour l'application mobile de vérifier le certificat délivré par un serveur lors d'un échange TLS. Mais qu'en est-il concernant l'épinglage des certificats? Ne pas le faire entraîne t-il une vulnérabilité? Cela doit-t-il être une exigence si une application comporte des données sensibles, ou est-ce même contre-productif? A-t-on besoin de crypter les données stockées dans des bases de données SQLite, même si le système d'exploitation exécute l'application dans un bac à sable? Ce qui est pertinent pour une application peut être irréaliste pour une autre. Le MASVS est un essai pour standardiser ces exigences en utilisant des niveaux de validation différents pour différents scenarii de menaces.

De plus, l'apparition de logiciels malveillants de niveau système et autres outils d'administration à distance a entraîné la prise de conscience que les systèmes d'exploitation mobiles ont eux aussi des failles de sécurité exploitables, contribuant à la hausse de l'utilisation de stratégies de compartementalisation pour la mise en place de protections supplémentaires pour la sauvegarde des données sensibles et contre la manipulation côté client. Et c'est là que les

choses deviennent compliquées : les mécanismes de sécurité matériels et les solutions de compartementalisation au niveau du système d'exploitation, telles que Android for Work et Samsung Knox, existent, mais ne sont pas disponibles pour tous les types d'appareils. Une solution de secours est d'implémenter des mesures de protection logicielles, mais il n'y a malheureusement pas de standard ou de processus de test pour valider ce genre de protections.

Par conséquent, les rapports de test de sécurité des applications mobiles vont dans tous les sens : par exemple, certains testeurs considèrent un manque d'obscurcissement ou de détection de rootage dans une application Android en tant que "faille de sécurité". Par ailleurs, des mesures telles que le cryptage des chaînes de texte, la détection de déboggage ou l'obscurcissement du contrôle du flux ne sont pas considérées obligatoires. Pourtant, cette façon binaire de voir les choses n'a pas de sens dans la mesure où la résilience n'est pas une proposition binaire : elle dépend des menaces spécifiques côté client contre lesquelles se défendre. Les protections de niveau logiciel ne sont pas inutiles mais peuvent toujours être contournées ; elles ne doivent donc pas être utilisées en tant que substitut aux contrôles de sécurité.

Le but général du MASVS est de proposer un cadre de base pour la sécurité des applications mobiles (MASVS- L1), tout en permettant le rajout de mesures de défense en profondeur (MASVS-L2) et de protections contre les menaces côté client (MASVS-R). Le MASVS a les buts suivants :

- Fournir des exigences aux architectes logiciels et aux développeurs ayant pour but de développer des applications mobiles avec un bon niveau de sécurité ;
- Proposer un standard pouvant être utilisé comme référence lors de la revue de la sécurité des applications mobiles ;
- Clarifier le rôle des mécanismes de protection logiciels pour la sécurité mobile et fournir des exigences pour valider leur efficacité ;
- Fournir des recommendations spécifiques concernant le niveau de sécurité à viser en fonction du cas d'utilisation.

Nous sommes conscients du fait qu'un consensus à 100% de la part de l'industrie est impossible à atteindre. Néanmoins, nous espérons que le MASVS sera utile dans les directions qu'il fournit le long de toutes les phases de développement et de test des applications mobiles. En tant que standard ouvert, le MASVS évoluera au fil du temps et toutes les contributions et les suggestions sont les bienvenues.

Changelog

Ce document est généré automatiquement à Wed Jan 09 2019 21:44:21 GMT+0100 (Midden-Europese standaardtijd)

1.1.3 Quelques Améliorations

Les changements suivants font partie de la livraison 1.1.3 :

- Corrections d'erreurs de traduction de l'exigence 7.1 dans la version espagnole
- Nouvelle présentation des traducteurs dans les Remerciements
- Petites corrections pour la version japonaise.

1.1.2 Parrainage et internationalisation

Les changements suivants font partie de la livraison 1.1.2 :

- Ajout d'une note de remerciement pour les acheteurs de l'e-book.
- Ajout d'un lien d'authentification manquant & mise à jour d'un lien d'authentification obsolète dans la V4.
- Correction d'une inversion de 4.7 et de 4.8 en anglais.
- Premières livraisons internationales!
  - Corrections dans la traduction en espagnol. La traduction est dorénavant en phase avec la version anglaise (1.1.2).
  - Corrections dans la traduction en russe. La traduction est dorénavant en phase avec la version anglaise (1.1.2).
  - Ajout des premières livraisons en chinois (ZHTW), français, allemand et japonais!
- Simplification du document afin de faciliter la traduction.
- Ajout d'instructions pour les livraisons automatisées.

1.1.0 14 juillet 2018 :

Les changements suivants font partie de la livraison 1.1 :

- L'exigence 2.6 "Le presse-papier est désactivé sur les champs de texte qui peuvent contenir des données sensibles." a été enlevée.
- L'exigence 2.2 "Aucune donnée sensible ne devrait être stockée hors du container de l'application ou des services de stockage de références fournis par le système." a été ajouté.
- L'exigence 2.1 a été reformulée vers "Les services de stockage de références fournis par le système sont utilisés de façon appropriée pour le stockage de données sensibles telles que les PII, les références des utilisateurs ou les clés de cryptographie.".

1.0 12 Janvier 2018 :

Les changements suivants font partie de la livraison 1.0 :

- Suppression de 8.9 en raison d'une redondance avec 8.12
- Reformulation de 4.6 pour la rendre plus générique
- Quelques améliorations (typos, etc.)

Standard de Validation de la Sécurité des Applications Mobiles (Mobile Application Security Verification Standard - MASVS)

A Propos du Standard

Bienvenue sur le Mobile Application Security Verification Standard (MASVS) 1.1, le Standard de Validation de la Sécurité des Applications Mobiles. Le MASVS est un effort communautaire dont le but est d'établir un cadre concernant les exigences de sécurité pour le design, le développement et le test de sécurité des applications mobiles sur iOS et Android.

Le MASVS est l'aboutissement d'un effort communautaire et de retours d'expérience venant du monde de l'industrie. Le but de ce standard est d'évoluer dans le temps et, dans ce cadre, tout retour est le bienvenu. Le meilleur moyen de rentrer en contact avec nous est via le canal Slack OWASP Mobile Project: https://owasp.slack.com/messages/project-mobile_omtg/details/

Un compte peut être créé à l'endroit suivant: http://owasp.herokuapp.com/.

Copyright et Licence

Remerciements

Chef de Projet	Auteur Principal	Contributeurs et Relecteurs
Sven Schleier & Jeroen Willemsen	Bernhard Mueller	Alexander Antukh, Mesheryakov Aleksey, Bachevsky Artem, Jeroen Beckers, Vladislav Chelnokov, Ben Cheney, Peter Chi, Lex Chien, Stephen Corbiaux, Manuel Delgado, Ratchenko Denis, Ryan Dewhurst, Tereshin Dmitry, Christian Dong, Oprya Egor, Ben Gardiner, Rocco Gränitz, Henry Hu, Sjoerd Langkemper, Vinícius Henrique Marangoni, Martin Marsicano, Roberto Martelloni, Gall Maxim, Riotaro Okada, Abhinav Sejpal, Stefaan Seys, Yogesh Shamrma, Prabhant Singh, Sven Schleier, Nikhil Soni, Anant Shrivastava, Francesco Stillavato, Romuald SZKUDLAREK, Abdessamad Temmar, Koki Takeyama, Chelnokov Vladislav, Leo Wang, Jeroen Willemsen

La langue	Traducteurs et relecteurs
Chinois	Peter Chi, and Lex Chien, Henry Hu, Leo Wang
Français	Romuald SZKUDLAREK, Christian Dong (Review)
Allemand	Rocco Gränitz, Sven Schleier (Review)
Espanol	Martin Marsicano
Japonais	Koki Takeyama, Riotaro Okada (Review)
Russe	Gall Maxim, Chelnokov Vladislav (Review), Oprya Egor (Review), Tereshin Dmitry (Review)

Ce document est basé sur le Standard de Validation de la Sécurité Applicative de l'OWASP, le OWASP Application Security Verification Standard écrit par Jim Manico.

Sponsors

Tant le MASVS que le MSTG ont été créés et sont maintenus par la communauté sur le principe du bénévolat ; ceci dit, un peu d'aide extérieure est parfois nécessaire. Par conséquent, nous remercions nos sponsors de nous avoir fourni les fonds pour pouvoir employer des éditeurs techniques. Il est toutefois important de souligner que leur aide financière n'influence pas le contenu des documents MASVS et MSTG de quelque manière que ce soit. Les conditions de parrainage sont décrites sur le OWASP Project Wiki.

Notable Benefactors

Nous aimerions ensuite remercier le chapitre de la région OWASP Bay Area pour son parrainage. Enfin, nous voudrions remercier toutes les personnes qui ont acheté le livre à Leanpub et qui nous ont parrainé de cette manière.

Le Standard de Validation de la Sécurité des Applications Mobiles

Le MASVS peut être utilisé pour établir un niveau de confiance dans la sécurité des applications mobiles. Les exigences ont été développées avec les objectifs suivants à l'esprit :

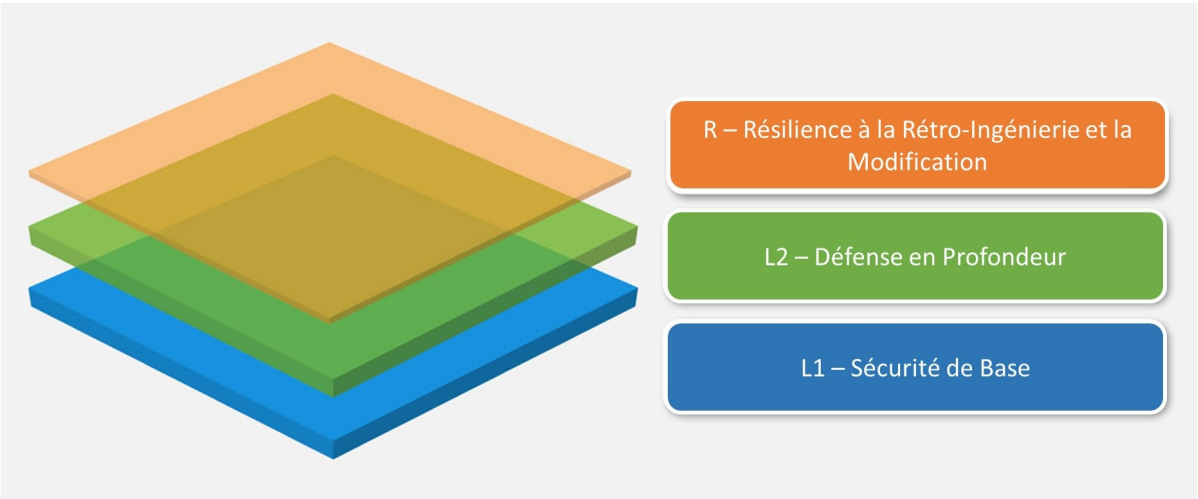
- Utilisation en tant que métrique - Fournir un standard de sécurité pouvant servir de mesure aux développeurs d'applications mobiles et à ceux qui en sont responsables ;
- Utilisation en tant que référence - Donner des points de repère pendant toutes les phases de développement de test et de développement d'applications mobiles ;
- Utilisation en phase d'achat - Permettre de définir un niveau de sécurité minimal attendu lors de la validation de la sécurité d'applications mobiles.

Modèle de Sécurité Applicative Mobile

Le MASVS définit deux niveaux distincts de validation de la sécurité (L1 et L2) ainsi qu'un ensemble flexible d'exigences concernant la protection contre la rétro-ingénierie (MASVS-R), c'est-à-dire adaptable au modèle de menaces d'une application. MASVS-L1 et MASVS-L2 comprennent des exigences de sécurité génériques et sont recommandés pour toute application mobile (L1) ainsi que pour celles qui gèrent des données hautement sensibles (L2). MASVS-R couvre des contrôles de protection supplémentaires qui peuvent être mis en oeuvre dans le cas où la protection contre les menaces côté client est l'un des buts du design.

Atteindre les exigences du niveau MASVS-L1 amène à une application qui suit les bonnes pratiques de sécurité et ne souffre pas des vulnérabilités couramment rencontrées. MASVS-L2 ajoute de nouveaux contrôles en profondeur tels que le SSL pinning, permettant de rendre l'application résistante à des attaques plus sophistiquées - en supposant que les contrôles de sécurité du système d'exploitation mobile sont intacts et que l'utilisateur final n'est pas considéré en tant que l'attaquant potentiel. Implémenter tout ou partie des exigences de protection logicielles de la partie MASVS-R aide à réduire les menaces côté client où l'utilisateur final serait malicieux et/ou lorsque le système d'exploitation mobile serait corrompu.

Il convient de noter que les contrôles de protection logiciels listés dans le MASVS-R et décrits dans le Guide de Test Mobile de l'OWASP (OWASP Mobile Testing Guide) peuvent toujours être contournés et ne doivent jamais être utilisés en tant que substituts aux contrôles de sécurité. Leur but est d'apporter des contrôles de protection supplémentaires spécifiques aux menaces d'applications qui remplissent aussi les exigences MASVS L1 ou L2.



Structure du Document

La première partie du MASVS contient une description du modèle de sécurité et des niveaux de validation disponibles, suivis par des recommandations sur l'utilisation du standard en pratique. Les exigences de sécurité détaillées, ainsi que leurs associations aux niveaux de validation, sont listées dans la seconde partie. Les exigences ont été groupées en huit catégories (V1 à V8) en fonction des objectifs et des périmètres techniques. La nomenclature suivanteest utilisée tout au long du MASVS et du MSTG:

- *Catégorie de l'exigence :* MASVS-Vx, e.g. MASVS-V2: Stockage de Données et Vie Privée
- *Exigence :* MASVS-Vx.y, e.g. MASVS-V2.2: "Aucune donnée sensible n'est écrite dans les journaux applicatifs."

Niveaux de Validation en Détail

MASVS-L1: Sécurité Standard

Une application mobile qui remplit les exigences de niveau MASVS-L1 implémente les bonnes pratiques de sécurité de développement d'applications mobiles. Elle implémente les exigences de base en termes de qualité de code, de gestion de données sensibles et d'intéraction avec l'environnement mobile. Un processus de test doit être en place pour valider la bonne implémentation des contrôles de sécurité. Ce niveau convient à tout type d'application mobile.

MASVS-L2: Défense en Profondeur

MASVS-L2 introduit des contrôles de sécurité plus poussés qui vont au delà des exigences courantes. Pour atteindre le niveau L2, un modèle de menaces doit exister et la sécurité doit faire partie intégrale de l'architecture de l'application et de son design. Ce niveau est approprié pour des applications qui manipulent des données sensibles telles que les applications mobiles bancaires.

MASVS-R: Résistance à la Rétro-Ingénierie et à la Manipulation

L'application implémente des contrôles au niveau de l'état de l'art en matière de sécurité, et est aussi résistante à des attaques clairement spécifiques au côté client telles que la manipulation, le moddage ou la rétro-ingénierie visant à extraire des parties de code ou des données sensibles. Une telle application met en oeuvre des fonctions de sécurité matérielles ou des techniques de protection logicielles vérifiables et offrant un bon niveau de robustesse. MASVS-R est applicable aux applications qui gèrent des données sensibles et peut servir de moyen de protection de propriété intellectuelle ou contre la manipulation d'une application.

Utilisation Recommandée

Les applications peuvent être validées par rapport à MASVS L1 ou L2 en fonction des évaluations de risque précédentes et du niveau de sécurité requis. L1 s'applique à tout type d'application mobile tandis que L2 est généralement recommandé pour des applications qui gèrent des données ou des fonctionnalités plus sensibles. MASVS-R (ou du moins une partie) peut être appliqué pour la validation de la résistance à des menaces spécifiques telles que le ré-empaquetage ou l'extraction de données sensibles, *en plus* d'une validation de sécurité appropriée.

En résumé, les typologies de validation suivantes sont disponibles :

- MASVS-L1
- MASVS-L1+R
- MASVS-L2
- MASVS-L2+R

Les différents combinaisons reflètent différents niveaux de sécurité et de résistance. Le but est de permettre une certaine flexibilité : par exemple, un jeu sur mobile pourrait se permettre de ne pas ajouter les contrôles de sécurité MASVS-L2 tels que l'authentification à 2 facteurs pour des considérations de facilité d'utilisation mais pourrait avoir de forts besoins commerciaux concernant la protection contre la manipulation.

Quel Typologie de Validation Utiliser?

L'implémentation des exigences MASVS L2 améliore la sécurité, mais peut en même temps augmenter les coûts de développement et dégrader potentiellement l'expérience utilisateur (un compromis classique). En général, L2 devrait être utilisé pour toute application où l'analyse risques vs coûts démontre le besoin d'atteindre ce niveau (i.e. lorsque les pertes potentielles causées par la compromission de la confidentialité ou de l'intégrité sont supérieures au coût induit par l'implémentation des contrôles de sécurité supplémentaires). L'évaluation des risques devrait être la première étape avant l'implémentation du MASVS.

Exemples

MASVS-L1

- Pour tout type d'application mobile. MASVS-L1 liste les bonnes pratiques de sécurité qui peuvent être suivies avec un impact raisonnable sur les coûts de développement et l'expérience utilisateur. Il est conseillé d'appliquer les exigences de MASVS-L1 pour toute application qui n'a pas vocation à implémenter les exigences des niveaux supérieurs.

MASVS-L2

- Industrie de la Santé : Applications mobiles qui stockent des données personnelles (PII) pouvant être utilisées pour du vol d'identité, des paiements frauduleux ou tout autre type de fraude. Pour le secteur de la santé américain, les considérations de conformité incluent Health Insurance Portability and Accountability Act (HIPAA) ainsi que les règlementations sur le respect de la vie privée, la sécurité, la notification des pertes de données et la sûreté des patients.
- Industrie Financière : Applications mobiles qui donnent accès à des informations hautement sensibles telles que des numéros de cartes de crédit, des informations personnelles ou permettent des mouvements de fonds. Ces applications justifient l'implémentation de contrôles de sécurité additionnels pour contrer la fraude. Les applications du monde de la finance doivent être conformes au standard Payment Card Industry Data Security Standard (PCI DSS), au Gramm Leech Bliley Act et au Sarbanes-Oxley Act (SOX).

MASVS L1+R

- Applications mobiles où la protection de la propriété intellectuelle est un objectif commercial. Les contrôles contribuant à la résistance listés dans MASVS-R peuvent être utilisés pour augmenter la quantité d'effort requis pour obtenir le code source d'origine et pour entraver les possibilités de manipulation / de piratage.
- Industrie du Jeu : Jeux présentant un fort besoin d'empêcher le moddage et la triche, tels que les jeux de compétition en ligne. La triche est un problème majeur pour les jeux en ligne dans la mesure où un nombre important de tricheurs peut amener à mécontenter la majorité des joueurs et peut entraîner l'échec d'un jeu. MASVS-R fournit des contrôles de base contre la manipulation pour rendre la possibilité de triche plus compliquée.

MASVS L2+R

- Industrie Financière : Applications de banque en ligne permettant aux utilisateurs de transférer des fonds et où les techniques d'injection de code et d'instrumentation sur des appareils mobiles compromis entraînent un risque. Dans ce cas, les contrôles du MASVS-R peuvent être mis en oeuvre pour empêcher la manipulation et rendre la vie des créateurs de logiciels malveillants (malwares) plus compliquée.

- Toute application mobile qui, par design, doit stocker des données sensibles sur l'appareil mobile tout en devant fonctionner pour une large game d'appareils et de versions de systèmes d'exploitation. Dans ce cas, des contrôles de résistance peuvent être utilisés en tant que mesures de défense en profondeur pour augmenter la quantité d'effort que doivent fournir les attaquants voulant extraire les données sensibles.

Evaluation et Certification

Position de l'OWASP Concernant la Certification selon le MASVS et les Labels de Confiance

L'OWASP, en tant qu'organisation à but non-lucratif et indépendante de toute influence commerciale, ne certifie aucun fournisseur, aucun organisme de référence ni aucun logiciel.

Toute sorte de garantie de confiance, label ou autre certification n'est pas cautionnée, soutenue ou donnée par l'OWASP ; ainsi toute organisation se reposant sur une telle approche doit être attentive à la confiance accordée à une tierce partie ou à un label de confiance affirmant avoir la certification ASVS.

Ceci ne doit pas empêcher les organisations d'offrir de tels services concernant la confiance, tant qu'elles ne revendiquent pas une certification officielle de la part de l'OWASP.

Conseils pour la Certification d'Applications Mobiles

La façon recommandée pour vérifier la conformité d'une application mobile par rapport au MASVS est d'efffectuer une revue en mode "livre ouvert", ce qui signifie que les testeurs ont accès aux ressources clés telles que les architectes et les développeurs de l'application, la documentation du projet, le code source et un accès authentifié aux points terminaux, comprenant au moins un accès à un compte pour chaque rôle.

Il est important de noter que le MASVS ne couvre que les (côtés clients des) applications mobiles et leur communication avec les points terminaux distants, ainsi que quelques exigences génériques liées à l'authentification des utilisateurs et à la gestion des sessions. Il ne contient pas d'exigence spécifique aux services distants (e.g. services web) associés à l'application, à l'exception d'un nombre limité d'exigences génériques ayant trait à l'authentification et à la gestion des sessions. Ceci dit, le MASVS V1 spécifie le fait que les services distants doivent être couverts par un modèle de menaces d'ensemble et doivent être validés par des standards appropriés tels que l'ASVS de l'OWASP.

Une organisation certifiante doit inclure dans tout rapport le périmètre de la validation (en particulier quand un composant clé est hors de ce périmètre), un résumé des points principaux, incluant les tests passés avec succès ou en échec, avec de claires indications sur la manière de résoudre les tests en échec. La conservation de documents de travail détaillés, de copies d'écran ou de vidéos, de scripts permettant d'exploiter une vulnérabilité de façon fiable et répétée ainsi que de documents électroniques liés aux tests, tels que les journaux d'interception de proxies et les notes associées telles que les listes d'actions, est considérée comme une pratique standard de l'industrie. Il n'est pas suffisant de lancer simplement un outil et de faire un rapport sur les défauts signalés ; ceci n'amène pas suffisamment de preuves que tous les points pour un niveau de certification donné ont été testés de façon adéquate. En cas de désaccord, il devrait assez de preuves pour démontrer que chaque exigence validée a bien été testée.

Utiliser le Guide de Test de Sécurité Mobile OWASP Mobile Security Testing Guide (MSTG)

Le MSTG de l'OWASP est un manuel pour tester la sécurité des applications mobiles. Il décrit les processus techniques pour valider les exigences listées dans le MASVS. Le MSTG comporte une liste de cas de tests, chacun relié à une exigence du MASVS. Tandis que les exigences du MASVS sont de haut niveau et génériques, le MSTG fournit des recommendations en profondeur et des procédures de test par type de système d'exploitation mobile.

Le Role des Outils de Test Automatique

L'utilisation d'outils d'analyse de code source ou de test en boîte noire est encouragé dans le but d'améliorer l'efficacité dès que cela est possible. Cependant, il n'est pas possible de mener à bien toute la validation proposée par le MASVS en utilisant seulement des outils automatiques : chaque application mobile a ses spécificités et la compréhension de l'architecture d'ensemble, la logique d'affaire et les limites techniques des technologies et frameworks utilisés est obligatoire pour valider la sécurité d'une application.

Autres Cas d'Utilisation

En tant que Source de Conseils Détaillés pour l'Architecture de Sécurité

L'une des utilisations les plus courantes du MASVS est en tant que ressource pour les arhitectes de sécurité. Les deux référentiels principaux d'architecture de sécurité, SABSA et TOGAF, ne fournissent pas beaucoup d'information qui serait nécessaire à la revue de l'architecture de sécurité des applications mobiles. Le MASVS peut être utilisé pour combler ces manques en permettant aux architectes de sécurité de choisir de meilleurs contrôles par rapport aux problèmes courants des applications mobiles.

En tant que Substitut aux Listes de Contrôle pour le Codage de Sécurité

Un certain nombre d'organisations peuvent bénéficier de l'adoption du MASVS en choisissant l'un des deux niveaux, ou en s'appropriant le MASVS et en adaptant ce qui est nécessaire à chaque niveau de risque en fonction du domaine d'application ciblé. Nous encourageons ce type d'appropriation tant que la traçabilité est maintenue, de telle manière que si une application a passé l'exigence 4.1 la signification reste la même pour chaque copie lorsque le standard évolue.

En tant que Base Méthodologique pour le Test de Sécurité

Une bonne méthodologie de test de sécurité pour application mobile devrait couvrir l'ensemble des exigences listées dans le MASVS. Le Mobile Security Testing Guide (MSTG) de l'OWASP fournit des cas de test en boîte noire et en boîte blanche pour la validation de chaque exigence.

En tant que Guide pour les Tests Automatisés et les Tests d'Intégration

Le MASVS a été créé pour être hautement testable, à la seule exception des exigences architecturales. Les tests unitaires, d'intégration et d'acceptation basés sur lex exigences du MASVS peuvent être intégrés dans le cycle de développement continu. Ceci permet d'une part d'améliorer la sensibilisation à la sécurité des développeurs, mais aussi d'autre part d'améliorer la qualité globale de l'application cible et de réduire la quantité de défauts détectés pendant la phase de tests de sécurité avant la mise sur le marché.

Pour la Formation au Développement de Sécurité

Le MASVS peut aussi être utilisé pour définir les caractéristiques de sécurité des applications mobiles. Un certain nombre de cours de "codage de sécurité" sont juste des cours de piratage éthique avec un soupçon de développement. Ceci n'est pas en faveur des développeurs. Au lieu de cela, les cours de développement de sécurité peuvent utiliser le MASVS, en se focalisant sur les contrôles proactifs documentés dans le MASVS, plutôt que par exemple sur les 10 principales erreurs de sécurité en développement.

V1: Exigences Concernant l'Architecture, le Design et le Modèle de Menaces

Objectif de Contrôle

Dans un monde parfait, la sécurité serait prise en compte tout au long du cycle de développement. Cependant, en réalité la sécurité n'est une considération que tardivement dans le SDLC. Au delà des contrôles techniques, le MASVS exige que des processus soient en place pour garantir que la sécurité a bien été explicitement prise en compte lors de la préparation de l'architecture de l'application mobile et que, pour l'ensemble des composants, les roles fonctionnels et liés à la sécurité sont connus. Dans la mesure où la plupart des applications mobiles agissent en tant que clients de services distants, il est nécessaire de s'assurer que des standards de sécurité pertinents sont aussi appliqués à ces services - tester l'application mobile de manière isolée n'est pas suffisant.

La catégorie “V1” liste les exigences relatives à l'architecture et au design de l'application. Par conséquent, c'est la seule catégorie qui n'est pas reliée à des cas de test techniques dans le Mobile Testing Guide de l'OWASP. Afin de traiter des sujets tels que le modèle de menaces, le SDLC de sécurité, la gestion des clés, le lecteur du MASVS est invité à consulter les projets de l'OWASP dédiés à ces sujets et/ou d'autres standards tel que ceux listés ci-dessous.

Exigences pour la Validation de la Sécurité

Les exigences pour MASVS-L1 et MASVS-L2 sont listées ci-dessous.

#	Description	L1	L2
1.1	Tous les composants de l'application sont identifiés et leur besoin est confirmé.	✓	✓
1.2	Les contrôles de sécurité ne sont jamais mis en oeuvre seulement côté client, mais aussi sur les points terminaux distants.	✓	✓
1.3	Une architecture de haut niveau concernant l'application mobile et tous les services distants utilisés a été définie et la sécurité a été prise en compte dans cette architecture.	✓	✓
1.4	Les données considérées comme sensibles dans le contexte de l'application mobile sont clairement identifiées.	✓	✓
1.5	Tous les composants de l'application sont définis en termes des fonctions métier et/ou de sécurité qu'ils apportent.		✓
1.6	Un modèle de menaces pour l'application mobile et les services distants associés a été livré et définit les menaces potentielles et les contre-mesures associées.		✓
1.7	Tous les contrôles de sécurité ont une implémentation centralisée.		✓
1.8	Il existe une politique explicite sur la façon de gérer les clés de cryptographie (dès qu'elles existent) tout au long de leur cycle de vie. Idéalement, un standard de gestion des clés est suivi (tel que NIST SP 800-57).		✓
1.9	Un mécanisme pour permettre les mises à jour de l'application mobile existe.		✓
1.10	La sécurité est prise en compte tout au long du cycle de développement.		✓

Références

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M10 - Fonctions superflues : https://www.owasp.org/index.php/Mobile_Top_10_2016-M10-Extraneous_Functionality
- OWASP Security Architecture cheat sheet: https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet
- OWASP Thread modelling: https://www.owasp.org/index.php/Application_Threat_Modeling
- OWASP Secure SDLC Cheat Sheet: https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet
- Microsoft SDL: https://www.microsoft.com/en-us/sdl/
- NIST SP 800-57: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

V2: Exigences Concernant le Stockage des Données et le Respect de la Vie Privée

Objectif de Contrôle

La protection des données sensibles telles que les références utilisateurs et autres informations privées est un point d'attention central dans la sécurité mobile. Tout d'abord, les données sensibles peuvent être accidentellement exposéesà d'autres applications fonctionnant sur le même appareil si des mécanismes du système d'exploitation tels que la communication inter-processus sont utilisés d'une mauvaise manière. Des données peuvent aussi accidentellement fuiter vers le stockage en nuage, les sauvegardes ou le cache. Aussi, les appareils mobiles peuvent être perdus ou volés plus facilement que les autres types d'appareils, renforçant la probabilité qu'un attaquant ait un accès physique à l'appareil. Dans ce cas, des protections supplémentaires peuvent être implémentées pour rendre l'accès aux données sensibles plus difficile.

Il faut noter que, dans la mesure où le MASVS se concentre sur l'application, il ne couvre pas les politiques du niveau de l'appareil telles que celles mises en oeuvre par les solutions de MDM (Mobile Device Management). L'utilisation de telles politiques est encouragée dans le contexte de l'entreprise pour améliorer la sécurité des données.

Définition des Données Sensibles

Dans le contexte du MASVS, les données sensibles ont trait aux deux aspects que sont les références utilisateurs ainsi que toute autre donnée considérée comme sensible dans un contexte particulier tel que :

- Information personnellement identifiable (PII) qui peut être utilisée pour un vol d'identité : numéros de sécurité sociale, numéros de carte de crédit, numéros de comptes bancaires, information de santé ;
- Information hautement sensible pouvant amener à une perte de réputation et/ou un coût financier si elle était divulguée : information contractuelle, information sous le coup de clauses de non-divulgation, information de gestion ;
- Toute information devant être protégée légalement ou pour des raisons de conformité.

Exigences pour la Validation de la Sécurité

La grande majorité des problèmes de divulgation de données peuvent être empêchés en suivant des règles simples. La plupart des contrôles listés dans ce chapitre sont obligatoires pour tous les niveaux de validation.

#	Description	L1	L2
2.1	Les fonctions de stockage sécurisées proposées par les systèmes sont utilisées de manière appropriée pour stocker les données sensibles tels que les informations personnellement identifiables (PII), les références des utilisateurs ou les clés cryptographiques.	✓	✓
2.2	Aucune donnée sensible ne devrait être stockée hors du conteneur de l'application ou des fonctions de stockage sécurisées proposées par le système.	✓	✓
2.3	Aucune donnée sensible n'est écrite dans les journaux applicatifs.	✓	✓
2.4	Aucune donnée sensible n'est partagée avec des tierces parties à moins que cela ne soit un besoin de l'architecture.	✓	✓
2.5	Le cache du clavier est désactivé sur les champs d'entrée textuels qui traitent de données sensibles.	✓	✓
2.6	Aucune donnée sensible n'est exposée par les mécanismes d'IPC.	✓	✓
2.7	Aucune donnée sensible, tels que les mots de passe ou les codes PIN, n'est exposée à travers l'interface utilisateur.	✓	✓
2.8	Aucune donnée sensible n'est incluse dans les sauvegardes générées par le système d'exploitation mobile.		✓
2.9	L'application enlève les données sensibles des vues lors de son passage en arrière-plan.		✓
2.10	L'application ne garde pas les données sensibles en mémoire plus longtemps que nécessaire et la mémoire est explicitement nettoyée après son utilisation.		✓
2.11	L'application met en oeuvre un minimum de politique concernant la sécurité de l'accès à l'appareil tel que l'obligation pour l'utilisateur de définir un code d'accès à l'appareil.		✓
2.12	L'application instruit l'utilisateur sur les types d'information personnellement identifiable traités ainsi que sur les bonnes pratiques que l'utilisateur devrait suivre en utilisant l'application.		✓

Références

Le Mobile Security Testing Guide de l'OWASP donne des instructions détaillées pour valider les exigences listées dans cette section.

- Pour Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05d-Testing-Data-Storage.md
- Pour iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06d-Testing-Data-Storage.md

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M2 - Stockage de données non-sécurisé: https://www.owasp.org/index.php/Mobile_Top_10_2016-M2-Insecure_Data_Storage
- CWE: https://cwe.mitre.org/data/definitions/922.html

V3: Exigences Concernant la Cryptographie

Objectif de Contrôle

La cryptographie est un ingredient essentiel pour la protection des données stockées sur un appareil mobile. C'est aussi une catégorie pour laquelle les choses peuvent très mal se passer, en particulier quand les conventions standards ne sont pas suivies. Le but des contrôles de ce chapitre est de garantir que les applications à valider implémentent la cryptographie en suivant les bonnes pratiques de l'industrie, notamment :

- L'utilisation de librairies cryptographiques éprouvées ;
- Le choix et la configuration pertinents des primitives cryptographiques ;
- L'utilisation d'un générateur de nombres aléatoires convenable lorsque cela est nécessaire.

Exigences pour la Validation de la Sécurité

#	Description	L1	L2
3.1	L'application n'utilise pas la cryptographie symétrique avec des clés codées en dur comme seule méthode de cryptage.	✓	✓
3.2	L'application utilise des implémentations de primitives cryptographiques éprouvées.	✓	✓
3.3	L'application utilise des primitives cryptographiques appropriées au cas d'utilisation, configurées en adéquation avec les bonnes pratiques de l'industrie.	✓	✓
3.4	L'application n'utilise pas de protocole ou d'algorithme de cryptographie considéré par la communauté comme déprécié pour des raisons de sécurité.	✓	✓
3.5	L'application ne ré-utilise pas la même clé de cryptographie à des fins différentes.	✓	✓
3.6	Toute valeur aléatoire est générée par un générateur de nombre aléatoire offrant un bon niveau de sécurité.	✓	✓

Références

Le Mobile Security Testing Guide de l'OWASP donne des instructions détaillées pour valider les exigences listées dans cette section.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05e-Testing-Cryptography.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06e-Testing-Cryptography.md

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M5 - Cryptographie Insuffisante: https://www.owasp.org/index.php/Mobile_Top_10_2016-M5-Insufficient_Cryptography
- CWE: https://cwe.mitre.org/data/definitions/310.html

V4: Exigences Concernant l'Authentification et la Gestion des Sessions

Objectif de Contrôle

Dans la plupart des cas, la connexion des utilisateurs à un service distant doit être appréhendée au niveau de l'architecture générale des applications mobiles. Même si la majorité de la logique se passe sur le point terminal, le MASVS définit des exigences de base concernant la manière de gérer les comptes et les sessions des utilisateurs.

Exigences pour la Validation de la Sécurité

#	Description	L1	L2
4.1	Si l'application donne accès aux utilisateurs à un service distant, un certain niveau d'authentification, tel que l'authentification par nom d'utilisateur / mot de passe, est faite sur le point terminal distant.	✓	✓
4.2	Si des sessions avec état sont utilisées, le point terminal distant utilise des identifiants de session aléatoirement générés pour authentifier les requêtes des clients sans avoir à envoyer les références des utilisateurs.	✓	✓
4.3	Si l'authentification sans état basée sur des jetons est utilisée, le serveur fournit des jetons qui ont été signés par un algorithme à la sécurité éprouvée.	✓	✓
4.4	Le point terminal distant met fin à la session existante lorsque l'utilisateur se déconnecte.	✓	✓
4.5	Une politique de mot de passe existe et est appliquée sur le point terminal distant.	✓	✓
4.6	Le point terminal distant implémente un mécanisme permettant la protection contre les essais répétés de références utilisateurs.	✓	✓
4.7	Les sessions sont dévalidées sur le point terminal distant après une période d'inactivité donnée et les jetons d'accès associés expirent.	✓	✓
4.8	L'authentification biométrique, lorsqu'elle est utilisée, n'est pas basée sur des évènements (c'est-à-dire l'utilisation d'une API qui retourne simplement "vrai" ou "faux"). A la place, son utilisation est basée sur le déverrouillage du trousseau d'accès / du magasin de clé (keychain / keystore).		✓
4.9	Un second facteur d'authentification est disponible sur le point terminal distant et l'exigence d'authentification à deux facteurs est mise en application de façon systématique.		✓
4.10	Les transactions sensibles requièrent une authentification améliorée.		✓
4.11	L'application informe les utilisateurs de toutes les connexions sur leurs comptes. Les utilisateurs ont accès à la liste des appareils utilisés pour accéder à leurs comptes et peuvent en bloquer.		✓

Références

Le Mobile Security Testing Guide de l'OWASP (guide de test de la Sécurité mobile) fournit des instructions détaillées pour valider les exigences listées dans cette section.

- General - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md
- Pour Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05f-Testing-Local-Authentication.md
- Pour iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06f-Testing-Local-Authentication.md

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M4 - Authentification Non-Sécurisée: https://www.owasp.org/index.php/Mobile_Top_10_2016-M4-Insecure_Authentication
- OWASP Mobile Top 10: M6 - Autorisation Non-Sécurisée: https://www.owasp.org/index.php/Mobile_Top_10_2016-M6-Insecure_Authorization
- CWE: https://cwe.mitre.org/data/definitions/287.html

V5: Exigences Concernant la Communication Réseau

Objectif de Contrôle

Le but des contrôles listés dans cette section est de garantir la confidentialité et l'intégrité de l'information échangée entre l'application mobile et les services des points terminaux distants. A minima, une application mobile doit mettre en place un canal sécurisé crypté pour la communication réseau en utilisant le protocole TLS avec les bons paramètres. Le niveau 2 liste des mesures additionnelles de défense en profondeur telles que l'épinglage SSL (SSL pinning).

Exigences pour la Validation de la Sécurité

#	Description	L1	L2
5.1	Les données sont cryptées sur le réseau en utilisant TLS. Le canal sécurisé est utilisé systématiquement à travers toute l'application.	✓	✓
5.2	Les réglages de TLS sont en ligne avec les meilleures pratiques, ou aussi proches que possible dans le cas où le système d'exploitation ne supporte pas les standards recommandés.	✓	✓
5.3	L'application valide le certificat X.509 du point terminal distant lors de l'établissement du canal sécurisé. Seuls les certificats signés par une CA de confiance sont acceptés.	✓	✓
5.4	Soit l'application utilise son propre magasin de certificats, ou bien elle épingle le certificat du point terminal ou sa clé publique, et par là n'établit pas de connexion avec des points terminaux qui proposent des certificats ou des clés différents, même s'ils sont signés par des CA de confiance.		✓
5.5	L'application ne repose pas sur un canal de communication non-sécurisé unique (e-mail ou SMS) pour les opérations critiques telles que l'enregistrement ou la récupération de compte.		✓
5.6	L'application implémente l'état de l'art en termes de connectivité et de librairies de sécurité.		✓

Références

Le Mobile Security Testing Guide de l'OWASP (guide de test de la Sécurité mobile) fournit des instructions détaillées pour valider les exigences listées dans cette section.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05g-Testing-Network-Communication.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06g-Testing-Network-Communication.md

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M3 - Communication Non-Sécurisée : https://www.owasp.org/index.php/Mobile_Top_10_2016-M3-Insecure_Communication
- CWE: https://cwe.mitre.org/data/definitions/319.html
- CWE: https://cwe.mitre.org/data/definitions/295.html

V6: Exigences Concernant les Interactions avec la Plateforme

Objectif de Contrôle

Le but des contrôles de ce groupe est de garantir que l'application utilise les API de la plateforme ainsi que ses composants standards d'une façon compatible avec la sécurité. De plus, les contrôles couvrent la communication entre les applications (IPC).

Exigences pour la Validation de la Sécurité

#	Description	L1	L2
6.1	L'application ne demande qu'une série minimum de permissions nécessaires.	✓	✓
6.2	Toutes les entrées provenant de sources externes ainsi que des utilisateurs sont validées et si nécessaire assainies. Ceci inclut les données reçues via l'interface utilisateur, les mécanismes IPC tel que les intentions, les URL propres à l'application et les sources sur le réseau.	✓	✓
6.3	L'application n'exporte pas de fonctionnalité sensible via des schémas d'URL propres à l'application, à moins que ces mécanismes ne soient correctement protégés.	✓	✓
6.4	L'application n'exporte pas de fonctionnalité sensible à travers les possibilités IPC, à moins que ces mécanismes ne soient correctement protégés.	✓	✓
6.5	JavaScript est désactivé dans les WebViews à moins qu'il ne soit explicitement requis.	✓	✓
6.6	Les WebViews sont configurées pour ne permettre que le jeu minimum de gestionnaires de protocoles requis (idéalement, seul https est supporté). Les gestionnaires potentiellement dangereux, tels que ceux pour les fichiers, les appels téléphoniques ou l'identifiant de l'application sont désactivés.	✓	✓
6.7	Dans le cas où des méthodes natives de l'application sont exposées à une WebView, il convient de valider que la WebView ne rend que le JavaScript contenu dans le package de l'application.	✓	✓
6.8	La désérialisation des objets, s'il en existe, est implémentée à l'aide d'API de sérialisation de confiance.	✓	✓

Références

Le Mobile Security Testing Guide de l'OWASP (guide de test de la Sécurité mobile) fournit des instructions détaillées pour valider les exigences listées dans cette section.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05h-Testing-Platform-Interaction.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06h-Testing-Platform-Interaction.md

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M1 - Mauvais Utilisation de la Plateforme: https://www.owasp.org/index.php/Mobile_Top_10_2016-M1-Improper_Platform_Usage
- CWE: https://cwe.mitre.org/data/definitions/20.html
- CWE: https://cwe.mitre.org/data/definitions/749.html

V7: Exigences Concernant la Qualité du Code et les Paramètres de Génération

Objectif de Contrôle

Le but de ce contrôle est d'assurer que les pratiques de codage de base concernant la sécurité sont suivies pendant le développement de l'application et que les fonctionnalités de sécurité "gratuites" fournies par le compilateur sont activées.

Exigences pour la Validation de la Sécurité

#	Description	L1	L2
7.1	L'application est signée et livrée avec un certificat en cours de validité, dont la clé privée est correctement protégée.	✓	✓
7.2	L'application a été générée en mode release avec des réglages appropriés à ce mode (c.a.d. sans les possibilités de déboggage).	✓	✓
7.3	Les symboles pour le déboggage ont été enlevés des binaires natifs.	✓	✓
7.4	Le code de déboggage a été enlevé de l'application et celle-ci ne journalise ni de messages d'erreur inutilement longs ni de messages de déboggage.	✓	✓
7.5	Tous les composants utilisés par l'application provenant de sources externes, notamment les librairies et les frameworks, ont été identifiés et analysés à la recherche de vulnérabilités connues.	✓	✓
7.6	L'application intercepte et gère les exceptions potentielles.	✓	✓
7.7	La logique de gestion des erreurs dans les contrôles de sécurité refuse tout accès par défaut.	✓	✓
7.8	Dans le code non-géré, la mémoire est allouée, libérée et utilisée de façon sécurisée.	✓	✓
7.9	Les fonctionnalités de sécurité intégrées dans les outils de la chaîne de génération, par exemple ceux pour la minification de byte-code, pour la protection de la pile, pour le support PIE ou le comptage de références automatiques, sont activées.	✓	✓

Références

Le Mobile Security Testing Guide de l'OWASP (guide de test de la Sécurité mobile) fournit des instructions détaillées pour valider les exigences listées dans cette section.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M7 - Qualité du Code Client: https://www.owasp.org/index.php/Mobile_Top_10_2016-M7-Poor_Code_Quality
- CWE: https://cwe.mitre.org/data/definitions/119.html
- CWE: https://cwe.mitre.org/data/definitions/89.html
- CWE: https://cwe.mitre.org/data/definitions/388.html
- CWE: https://cwe.mitre.org/data/definitions/489.html

V8: Exigences Concernant la Résilience

Objectif de Contrôle

Cette section couvre des mesures de défense en profondeur recommandées dans les cas d'applications qui traitent ou donnent accès à des données ou des fonctionnalités sensibles. L'absence de l'un de ces contrôles n'entraîne pas une vulnérabilité dans la mesure où ils ont pour seul but d'améliorer la résilience de l'application envers la rétro-ingénierie et autres attaques spécifiques côté client.

Les contrôles de cette section devraient être mis en oeuvre autant qu'il est nécessaire, c'est-à-dire tel que requis par une évaluation des risques dûs à une modification non-autorisée de l'application et /ou une rétro-ingénierie du code. Nous ne pouvons que suggérer la consultation du document de l'OWASP "Technical Risks of Reverse Engineering and Unauthorized Code Modification Reverse Engineering and Code Modification Prevention" ("Risques Techniques Liés à la Rétro-Ingénierie et Modification de Code Non-Autorisée Rétro-Ingénierie et Prévention de Modification de Code", voir les références ci-dessous) pour une liste des risques liés aux affaires et des menaces techniques associées.

Pour que les contrôles de la liste ci-dessous soient effectifs, l'application doit implémenter au moins toutes les exigences de niveau MASVS-L1 (c'est-à-dire que des contrôles de sécurité robustes doivent être en place) ainsi que toutes les exigences de la partie V8. Par exemple, les contrôles concernant l'obscurcissement listés dans la partie "Entraver la Compréhension" doivent être combinés avec "Isolation de l'Application", "Entraver l'Analyse Dynamique et la Modification" et "Liaison avec un Appareil".

Il convient de noter que les mécanismes de protection logiciels ne doivent jamais être utilisés en tant que substituts aux contrôles de sécurité. Les contrôles listés dans le MASVR-R ont pour intention de rajouter aux applications des mécanismes de protection supplémentaires, spécifiques à une menace donnée, qui sont compatibles avec les exigences de sécurité du MASVS.

Il convient de prendre en compte les considérations suivantes :

1. Un modèle de menace doit exister et souligner clairement les menaces côté client contre lesquelles il faut se défendre. De plus, le niveau de protection à atteindre doit être spécifié. Par exemple, un but bien défini pourrait être de forcer les auteurs d'un logiciel malveillant ciblant une application et cherchant à l'instrumenter de devoir investir un effort significatif de rétro-ingénierie manuelle.
2. Le modèle de menaces doit être porteur de sens. Par exemple, cacher une clé de cryptographie dans une implémentation en boîte-blanche n'est pas cohérent dans le cas où l'attaquant a la possiblité de récupérer l'ensemble du code et de l'analyser dans son ensemble.
3. L'efficacité de la protection devrait toujours être validée par un expert humain ayant l'expérience du test des moyens particuliers mis en oeuvre contre la modification du code ou son obscurcissement (voir aussi les chapitres liés à la "rétro-ingénierie" et à la "validation des protections logicielles" dans le Mobile Security Testing Guide).

Entraver l'Analyse Dynamique et la Modification

#	Description	R
8.1	L'application détecte et réagit à la présence d'appareils rootés ou jailbreakés soit en alertant l'utilisateur ou en mettant fin à l'application.	✓
8.2	L'application empêche le déboggage et / ou réagit à la présence d'un déboggeur. Tous les protocoles de déboggage disponibles doivent être couverts.	✓
8.3	L'application détecte et réagit à la modification de fichiers exécutables et de données critiques au sein de son bac à sable.	✓
8.4	L'application détecte et réagit à la présence d'outils et de frameworks de rétro-ingénierie courants sur l'appareil.	✓
8.5	L'application détecte et réagit à son exécution dans un émulateur.	✓
8.6	L'application détecte et réagit à la modification de code et de données dans son espace mémoire.	✓
8.7	L'application implémente plusieurs mécanismes parmi les catégories de défense (8.1 à 8.6). Il convient de noter que la résilience augmente avec la quantité et la diversité de l'originalité des mécanismes utilisés.	✓
8.8	Les mécanismes de détection déclenchent des réponses de différents types, notamment des réponses invisibles de retardement de l'attaque.	✓
8.9	L'obscurcissement est mis en oeuvre par des défenses programmatiques qui, à leur tour, entravent le dé-obscurcissement via l'analyse dynamique.	✓

Liaison avec un Appareil

#	Description	R
8.10	L'application implémente un mécanisme de 'liaison avec l'appareil' utilisant une empreinte de l'appareil dérivée de multiples propriétés uniques à cet appareil.	✓

Entraver la Compréhension

#	Description	R
8.11	Tous les fichiers exécutables et les librairies appartenant à l'application sont soit chiffrés au niveau du fichier et / ou le code important et les segments de données à l'intérieur des exécutables sont chiffrés ou compactés. Une analyse statique triviale ne révèle pas de code important ou de données.	✓
8.12	Si le but de l'obscurcissement est de protéger des traitements sensibles, le schéma d'obscurcissement utilisé est à la fois approprié à la tâche considérée et est résistant envers les méthodes de dé-obscurcissement manuelles et automatiques, en prenant en considération les recherches disponibles. L'efficacité du schéma d'obscurcissement doit être validée à travers du test manuel. Il convient de noter que les fonctionnalités d'isolation au niveau matériel doivent être mises en pratique de préférence à l'obscurcissement toutes les fois que cela est possible.	✓

Références

Le Mobile Security Testing Guide de l'OWASP (guide de test de la Sécurité mobile) fournit des instructions détaillées pour valider les exigences listées dans cette section.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md

Pour de plus amples informations, il est possible de consulter aussi :

- OWASP Mobile Top 10: M8 - Modification du Code: https://www.owasp.org/index.php/Mobile_Top_10_2016-M8-Code_Tampering
- OWASP Mobile Top 10: M9 - Rétro-Ingénierie: https://www.owasp.org/index.php/Mobile_Top_10_2016-M9-Reverse_Engineering
- OWASP Reverse Engineering Threats - https://www.owasp.org/index.php/Technical_Risks_of_Reverse_Engineering_and_Unauthorized_Code_Modification
- OWASP Reverse Engineering and Code Modification Prevention - https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project

Annexe A: Glossaire

2FA

L'Authentification à Deux Facteurs ajoute un second niveau d'authentification lors de la connexion à un compte.

Distribution Aléatoire de l'Espace D'adressage (ASLR)

Une technique rendant plus difficiles l'exploitation de défauts permettant la corruption de la mémoire.

Sécurité Applicative

La sécurité de niveau applicatif se focalise sur l'analyse des composants qui constituent la couche applicative du modèle OSI (the Open Systems Interconnection Reference Model), plutôt que de se focaliser par exemple sur le système d'exploitation sous-jacent ou sur les réseaux connectés.

Validation de la Sécurité Applicative

Evaluation technique d'une application par rapport au MASVS de l'OWASP.

Rapport de Validation de Sécurité Applicative

Rapport qui documente les résultats dans leur ensemble ainsi que l'analyse sous-jacente réalisée par le validateur de l'application considérée.

Authentification

Vérification de l'identité clamée par l'utilisateur de l'application.

Validation Automatique

Utilisation d'outils automatisés (outils d'analyse dynamique ou outils d'analyse statique, ou les deux) qui utilisent les signatures des vulnérabilités pour identifier les problèmes.

Test en Boîte Noire

Méthode de test logiciel permettant l'analyse d'une application via ses fonctionnalités, sans avoir connaissance de ses structures internes ou de son fonctionnement.

Composant

Unité de code autonome avec ses interfaces de stockage et de réseau et qui communique avec d'autres composants.

Cross-Site Scripting (XSS)

Vulnérabilité de sécurité typiquement présente dans les applications web et permettant l'injection de script dans le contenu côté client.

Module Cryptographique

Matériel, logiciel et/ou micrologiciel qui implémente des algorithmes cryptographiques et/ou qui génère des clés de cryptographie.

CWE

Liste communautaire des faiblesses en sécurité communes dans le logiciel. Sert à établir un langage commun, de barème aux outils de sécurité logiciels et en tant que référentiel aux efforts d'identification, de réduction et de prévention des faiblesses.

DAST

Les technologies de test de sécurité des applications en mode dynamique sont conçues pour détecter les conditions indicatives d'une vulnérabilité de sécurité dans une application lors de son exécution.

Validation du Design

Evaluation technique de la sécurité de l'architecture d'une application.

Validation Dynamique

Utilisation d'outils automatisés qui utilisent les signatures des vulnérabilités pour identifier les problèmes pendant l'exécution d'une application.

Globally Unique Identifier (GUID)

Numéro de référence unique utilisé en tant qu'identifiant en logiciel.

Hyper Text Transfer Protocol (HTTP)

Prototole de niveau applicatif pour les systèmes d'information distribués, collaboratifs et hypermedia. Fondateur pour la communication de données sur le World Wide Web.

Clés Codées en Dur

Clés de cryptographie stockées sur l'appareil même.

IPC

Acronyme de Inter Process Communications. Sert aux processus à communiquer entre eux et avec le noyau pour coordonner leurs activités.

Validation des Entrées

Canonicalisation et validation de données considérées comme non-fiables saisies par l'utilisateur.

Bytecode JAVA

Le bytecode Java correspond au jeu d'instructions de la machine virtuelle Java (JVM). Chaque bytecode est composé d'un, ou parfois deux, octet(s) qui représente(nt) l'instruction (opcode), accompagné(s) de zéro octet ou plus pour le passage de paramètres.

Code Malicieux

Code introduit dans une application pendant son développement à l'insu du responsable de l'application et qui a pour finalité de contourner la politique de sécurité visée. Différent des logiciels malveillants tels que les virus ou les vers!

Logiciel Malveillant (Malware)

Code exécutable introduit dans une application au cours de son fonctionnement sans le consentement de l'utilisateur de l'application ou de son administrateur.

Open Web Application Security Project(OWASP)

L'Open Web Application Security Project (OWASP) est une communauté mondiale indépendante et gratuite oeuvrant pour l'amélioration de la sécurité des applications logicielles. Notre mission est de rendre la sécurité applicative "visible" afin que les particuliers et les organisations puissent prendre les bonnes décisions en incluant les risques liés à la sécurité applicative . Voir : http://www.owasp.org/

PIE

Position-independent executable (PIE) is a body of machine code that, being placed somewhere in the primary memory, executes properly regardless of its absolute address.

Informations d'Identification Personnelles (PII)

Informations pouvant, directement ou indirectement, mener à l'identification d'un individu ou permettre de contacter ou de localiser une personne dans un contexte donné.

PKI

Une PKI est une infrastructure permettant de lier une clé publique à l'identité de l'entité la possédant. Le lien est établi grâce à un processus d'enregistrement et de délivrance de certificats par une autorité de certification (CA).

SDLC

Software development lifecycle.

SAST

Les technologies de test de sécurité des applications en mode statique (SAST) ont pour but d'analyser le code source d'une application, son bytecode ou son binaire afin de repérer les conditions indicatives de vulnérabilités de sécurité dans le code ou le design. Les solutions de SAST analysent les applications de l'intérieur et au repos.

Architecture de Sécurité

Abstraction du design de l'application qui identifie et décrit où et comment les contrôles de sécurité sont mis en oeuvre et identifie et décrit l'emplacement et le niveau de sensibilité des données de l'utilisateur et de l'application.

Configuration de Sécurité

Configuration de l'application lors de son exécution impactant directement la façon dont les contrôles de sécurité sont mis en oeuvre.

Contrôle de Sécurité

Fonction ou composant exécutant une vérification liée à la sécurité (par exemple une validation des droits d'accès) ou ayant un impact sur la sécurité (par exemple génération d'un enregistrement dans un journal).

Injection SQL (SQLi)

Technique d'injection de code utilisée dans l'attaque d'applications fortement consommatrices de données où des expressions SQL malicieuses sont insérées à des points d'entrée.

Authentification par SSO

L'authentification unique (SSO) intervient lorsqu'un utilisateur s'authentifie sur un client donné et est ensuite authentifié automatiquement sur d'autres clients indépendemment de la plateforme, de la technologie ou du domaine utilisé par l'utilisateur. Par exemple, lorsque vous vous authentifiez sous Google, vous êtes automatiquement authentifiés par la suite sous les services YouTube, Docs et Gmail.

Modèle de Menaces

Technique consistant dans le développement incrémental d'architectures de sécurité de plus en plus raffinées ayant pour but l'identification d'agents menaçants, de zones de sécurité, de contrôles de sécurité et des éléments importants concernant la technique et les affaires.

Transport Layer Security

Protocoles de cryptographie pour les communications sécurisées à travers Internet.

Fragments URI/URL

Un Uniform Resource Identifier (URI) est une chaîne de caractères utilisée pour l'identification d'un nom ou d'une ressource sur le web. Un Uniform Resource Locator (URL) est en général utilisé pour faire référence à une ressource.

Tests d'Acceptance Utilisateur (UAT)

Correspond traditionnellement à un environnement de test ayant les mêmes comportements que l'environnement de production et où les tests logiciels sont réalisés avant le passage en production.

Validateur

La personne ou l'équipe qui évalue une application par rapport aux exigences du MASVS de l'OWASP.

Liste Blanche

Liste de données ou d'opérations permises, par exemple une liste de caractères acceptés lors de la validation des entrées.

Certificat X.509

Un certificat X.509 est un certificat digital qui implémente le standard international largement utilisé concernant les infrastructures à clés publiques (PKI) X.509 dans le but de vérifier qu'une clé publique correspond bien à l'identité de l'utilisateur, de l'ordinateur ou du service indiqué sur le certificat.

Annexe B: Références

Les projets de l'OWASP suivants sont susceptibles d'être utiles aux utilisateurs / ceux qui vont adopter ce standard :

- OWASP Mobile Security Project - https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- OWASP Mobile Security Testing Guide - https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide
- OWASP Mobile Top 10 Risks - https://www.owasp.org/index.php/Projects/OWASP*Mobile_Security_Project*-_Top_Ten_Mobile_Risks
- OWASP Reverse Engineering and Code Modification Prevention - https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project

De la même manière, les sites web suivants sont susceptibles d'être utiles aux utilisateurs / ceux qui vont adopter ce standard :

- MITRE Common Weakness Enumeration - http://cwe.mitre.org/
- PCI Security Standards Council - https://www.pcisecuritystandards.org
- PCI Data Security Standard (DSS) v3.0 Requirements and Security Assessment Procedures https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf