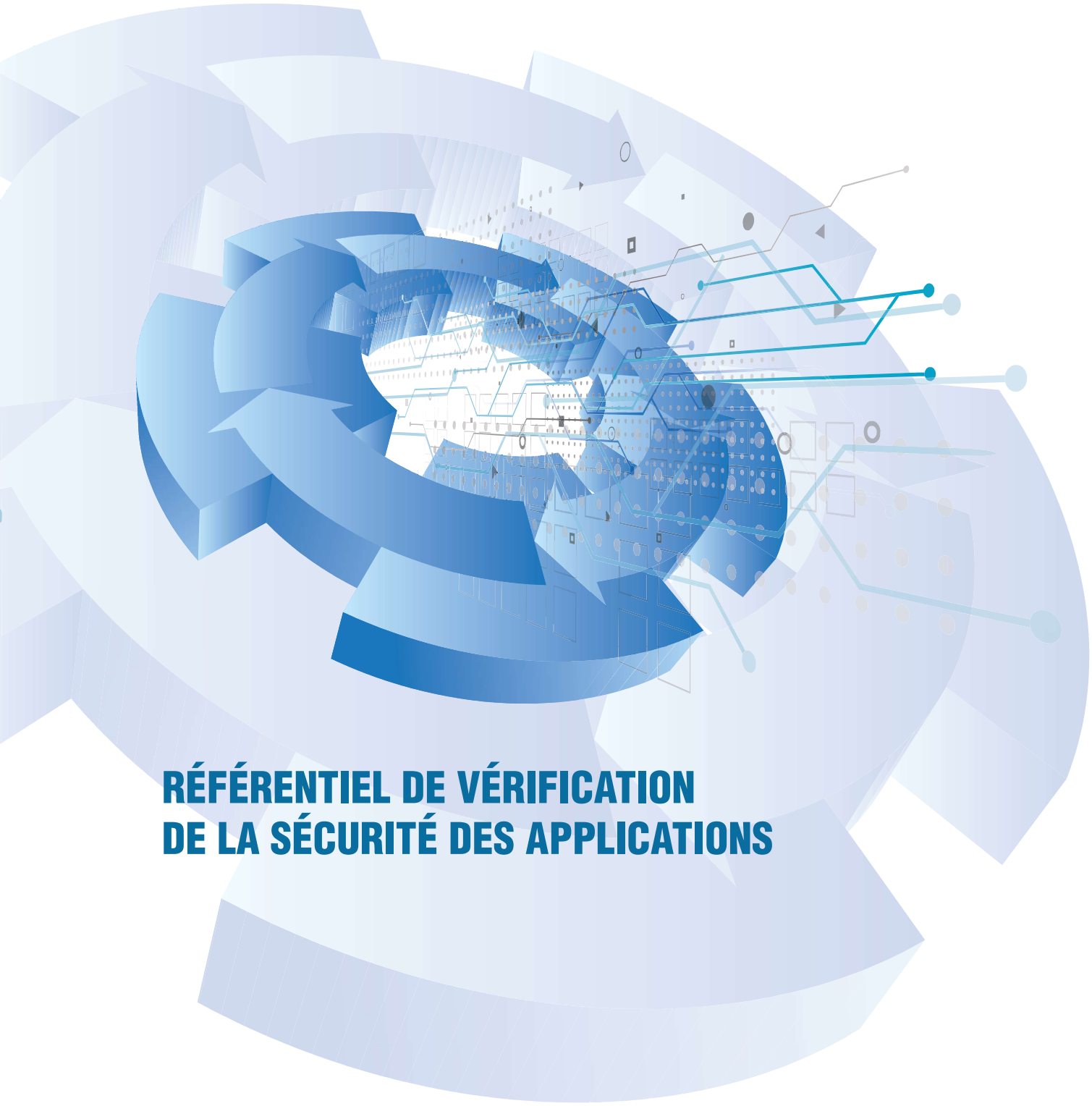
ROYAUME DU MAROC
ADMINISTRATION DE LA DÉFENSE NATIONALE
DIRECTION GÉNÉRALE DE LA SÉCURITÉ
DES SYSTÈMES D'INFORMATION

المملكة المغربية
إدارة الدفاع الوطني
المديرية العامة لأمن
نظم المعلومات



RÉFÉRENTIEL DE VÉRIFICATION DE LA SÉCURITÉ DES APPLICATIONS

UNE PUBLICATION DE LA DIRECTION GÉNÉRALE
DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION

2022

Référentiel De Vérification De La Sécurité Des Applications

UNE PUBLICATION DE LA
DIRECTION GENERALE DE LA SECURITE DES SYSTEMES D'INFORMATION
2022

TABLE DES MATIERS

Introduction

Le présent référentiel de vérification de la sécurité des applications comporte un ensemble d'exigences et de tests susceptibles de révéler des défauts et d'identifier des lacunes. Ce référentiel peut être utilisé, notamment pour définir, construire, tester et vérifier la sécurité des applications.

Le cadre d'exigences et de contrôle de sécurité est basé sur des tests fonctionnels et non fonctionnels qui doivent être appliqués lors de la conception, du développement et des tests des logiciels.

Ce référentiel, qui se base sur la version ASVS[1] 4.0.3 publiée en octobre 2021 par la communauté OWASP[2], s'applique à tous les modèles de développement logiciel et vise à atteindre deux objectifs principaux:

1. Aider les organismes à développer et à maintenir des applications sécurisées (guide de tests unitaires et d'intégration automatisés, guide de formation au développement sécurisé) ;
2. Guider les parties prenantes dans le choix des meilleures offres en matière d'acquisition de logiciels sécurisés auprès des sociétés de développement. Le cadre d'exigences permettra en effet, de mettre à la disposition des parties prenantes, une solution basée sur les bonnes pratiques de développement sécurisé et permettant de comparer les exigences exprimées avec les offres proposées par les soumissionnaires.

Ce référentiel de vérification de la sécurité des applications intègre plusieurs normes de sécurité, dont les directives NIST[3] 800-63-3 sur l'identité numérique, les recommandations NIST SP 800-57 relatives à la gestion des clés, le Top 10 2021 de l'OWASP, les contrôles proactifs 2018 de l'OWASP, les sections 6.5 de la norme PCI-DSS[4] v3.2.1 ainsi qu'un mappage vers le CWE[5].

La synthèse des normes précitées dans un seul document vise à réduire et à unifier les exigences de sécurité d'une part et d'autre part à couvrir les différentes applications et services Web, les architectures d'applications traditionnelles et modernes, les pratiques de sécurité agiles ainsi que la culture DevSecOps[6]. A cet effet, trois niveaux de vérification de la sécurité ont été retenus. Le niveau trois représente le niveau d'assurance le plus élevé, avec 286 pratiques de sécurité.

→ **Niveau 1** : est un niveau d'assurance faible. Il comporte des tests d'intrusions classiques. Ce niveau constitue une première étape pour sécuriser d'une manière progressive les applications d'une entité. Il est aussi parfois suffisant pour les applications qui ne stockent pas ou qui ne traitent pas de données sensibles et partant ne nécessitent pas des contrôles rigoureux contenus dans les niveaux 2 ou 3. Les contrôles du niveau 1 peuvent être lancés automatiquement par des outils ou effectués manuellement sans accès au code source.

→ **Niveau 2** : est nécessaire pour les applications qui contiennent des données sensibles et qui nécessitent une protection adaptée. Ce niveau constitue généralement le niveau recommandé pour la plupart des applications.

→ **Niveau 3** : est destiné aux applications critiques, qui traitent des données hautement sensibles, ou qui requièrent un haut niveau de confiance.

En fonction de l'analyse du risque et des exigences métiers, chaque organisme doit déterminer le niveau d'exigence approprié.

Les exigences de sécurité (le niveau du référentiel approprié, N1, N2 ou bien N3), dépendent de divers facteurs notamment, la nature de l'organisme (exemple. Secteur bancaire, Administration publique, etc.), les exigences légales (PCI-DSS) ou conformité par rapport aux normes de sécurité. En fonction de ces exigences, les processus informatiques (exemple. Gestion) doivent être adaptés en conséquence. L'exemple ci-après (figure 1), décrit comment les tests de sécurité des logiciels peuvent être intégrés dans un modèle de développement logiciel tel que le DevSecOps.

Le niveau 1 est destiné aux niveaux d'assurance faibles. A la différence des autres niveaux qui exigent l'accès à la documentation, au code source, à la configuration et aux personnes impliquées dans le processus de développement, le niveau 1 autorise la réalisation de tests boîte noire (pas de documentation ni de code source), Les tests en boîte noire, souvent effectués en fin de développement, ne sont pas suffisant pour se protéger contre des attaquants déterminés.

---

[1] ASVS : Cadre d'exigences des contrôles de sécurité fonctionnels et non fonctionnels requis lors de la conception, du développement et du test d'applications Web.
[2] OWASP : Organisme à but non lucratif mondial qui milite pour l'amélioration de la sécurité des logiciels.
[3] NIST : Institut national des normes et de la technologie, est une agence du département du Commerce des États-Unis.
[4] PCI-DSS : Norme de sécurité de l'industrie des cartes de paiement.
[5] CWE : Une base de données contenant les vulnérabilités logicielles et les faiblesses matérielles publiquement connues.
[6] DevSecOps : Approche qui permet d'intégrer la sécurité des données dès le début d'un projet.

Les tests de type boîte noire ont prouvé à maintes reprises qu'ils ne couvrent pas les problèmes de sécurité critiques. A cet effet, il est fortement conseillé d'utiliser un large éventail de tests d'assurances et de vérifications de sécurité, y compris le remplacement des tests de pénétration (boîte noire) par des tests de pénétration (hybrides) qui puisent les informations à partir de code source et de la documentation réalisée tout au long du processus de développement.

La DGSSI encourage fortement l'utilisation d'outils de sécurité dans le processus de développement lui-même. Les outils DAST et SAST peuvent être utilisés en continu par le pipeline de génération des jeux de tests pour enrichir l'éventail de problèmes de sécurité probables.

Les outils automatisés et les analyses en ligne sont insuffisant pour effectuer la totalité des vérifications de la sécurité des applications en l'absence de l'assistance humaine. Si une automatisation complète des tests pour chaque build est requise, une combinaison de tests unitaires et d'intégration personnalisés, ainsi que des analyses en ligne initiées par le build sont recommandées. Les failles de la logique métier et les tests de contrôle d'accès ne sont généralement faisables qu'avec une assistance humaine.

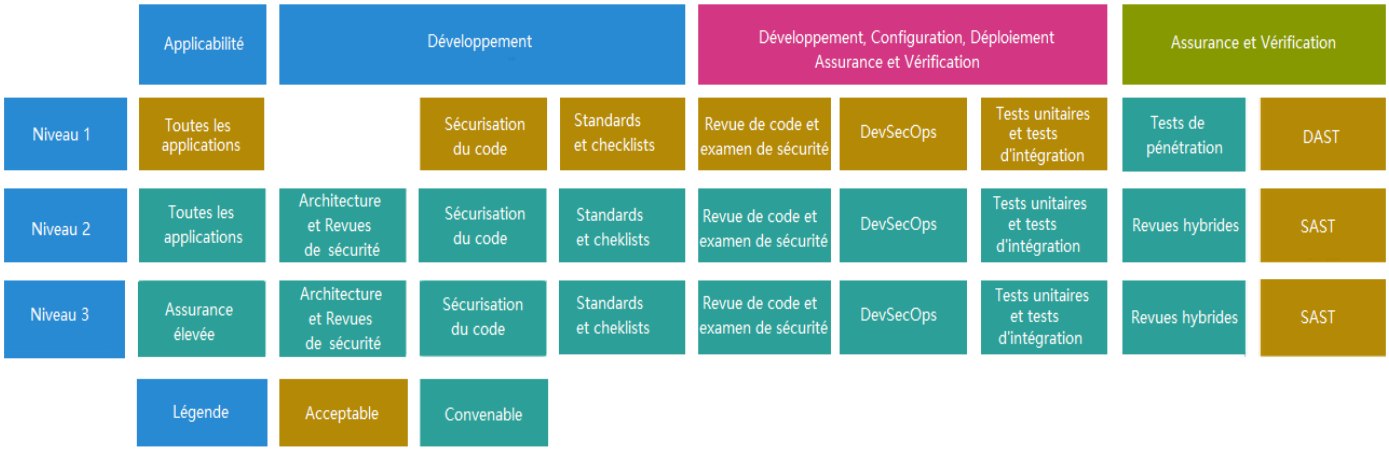


Figure 1 - Niveaux de vérification de la sécurité des applications

Niveaux de vérification de la sécurité des applications

Le référentiel de vérification de la sécurité des applications comprend un total de 286 contrôles et 14 rubriques de vérification (figure 2), il adopte une approche par famille de fonctions et par niveau d'exigences. Selon le niveau de sécurité requis, ce référentiel propose trois niveaux de contrôle.

Figure 2 - Les contrôles par niveau de vérification

Niveau 1 - Niveau basique

Une application de niveau 1 est une solution avec de faibles besoins d'assurance ou celles qui ne traite pas des données sensibles. Elle peut contrer les attaques utilisant les vulnérabilités de sécurité des applications faciles à découvrir, et qui figurent dans le Top 10 de l'OWASP ou toutes autres listes de contrôle similaires. Les tests à ce niveau peuvent être effectués avec une combinaison de méthodes automatiques et manuelles sans accès au code source, à la documentation ou aux développeurs.

Niveau 2 - Niveau standard

Une application de niveau 2 bloque de manière adéquate la plupart des risques associés aux logiciels actuels. Le niveau 2 garantit que les contrôles de sécurité sont en place, efficaces et intégrés dans l'application. Ce niveau est généralement approprié pour les applications qui gèrent des transactions importantes qui mettent en œuvre des fonctions critiques ou traitant des actifs sensibles, ou se rapportant à des secteurs où l'intégrité est primordiale.

Les menaces pesant sur les applications de niveau 2 proviennent généralement du fait que des attaquants compétents et motivés se concentrent sur des cibles spécifiques pendant une longue période avec l'aide d'outils et techniques hautement efficaces. Ceci leur permet de découvrir et d'exploiter les faiblesses persistantes dans ces applications.

Niveau 3 - Niveau avancé

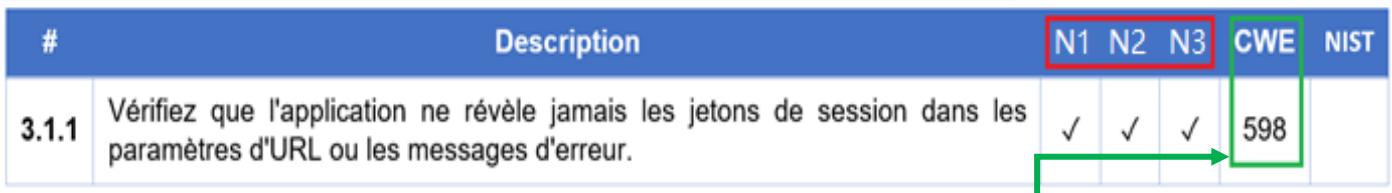
Le niveau 3 du référentiel est le plus haut niveau de vérification. Ce niveau est généralement réservé aux applications qui nécessitent un niveau de vérification de sécurité très important, particulièrement dans les domaines de la défense, de la santé, de la sécurité, des infrastructures critiques.

Les parties prenantes peuvent exiger le niveau 3 pour les applications qui remplissent des fonctions critiques, où une défaillance pourrait avoir un impact significatif sur les opérations de l'organisation ou nuire à sa capacité de survie.

Pour atteindre le niveau 3, il faut mener une analyse très approfondie de l'architecture, du code et des tests à tous les niveaux. Une application sécurisée doit être modulaire afin de faciliter la résilience et l'évolutivité. De même, chaque module doit être séparé par une connexion réseau et/ou hébergé au niveau d'une instance physique propre.

Enfin, une défense en profondeur doit être menée à travers des contrôles qui permettent de garantir la confidentialité (par exemple, le cryptage), l'intégrité (par exemple, les transactions, la validation des entrées), la disponibilité (par exemple, gérer la charge avec élégance), l'authentification (y compris entre les systèmes), la non-répudiation, l'autorisation et l'audit (journalisation).

L'exemple ci-après (figure 3), montre les trois niveaux (N1 à N3) pour la fonction ‘’exigence fondamentale en matière de gestion des sessions’’ relevant de la rebrique ‘’exigence de vérification de la gestion des sessions’’. Il expose une analyse de cette exigence au regard du CWE et fait référence au CWE numéro 598 (figure 4). Ainsi, les équipes qui analysent les exigences ou qui réalisent la mise en œuvre, ou encore celles qui contrôlent la réalisation, peuvent se référer aux explications complémentaires du CWE.



#	Description	N1	N2	N3	CWE	NIST
3.1.1	Vérifiez que l'application ne révèle jamais les jetons de session dans les paramètres d'URL ou les messages d'erreur.	✓	✓	✓	598	

Figure 3 - Exemple de vérification

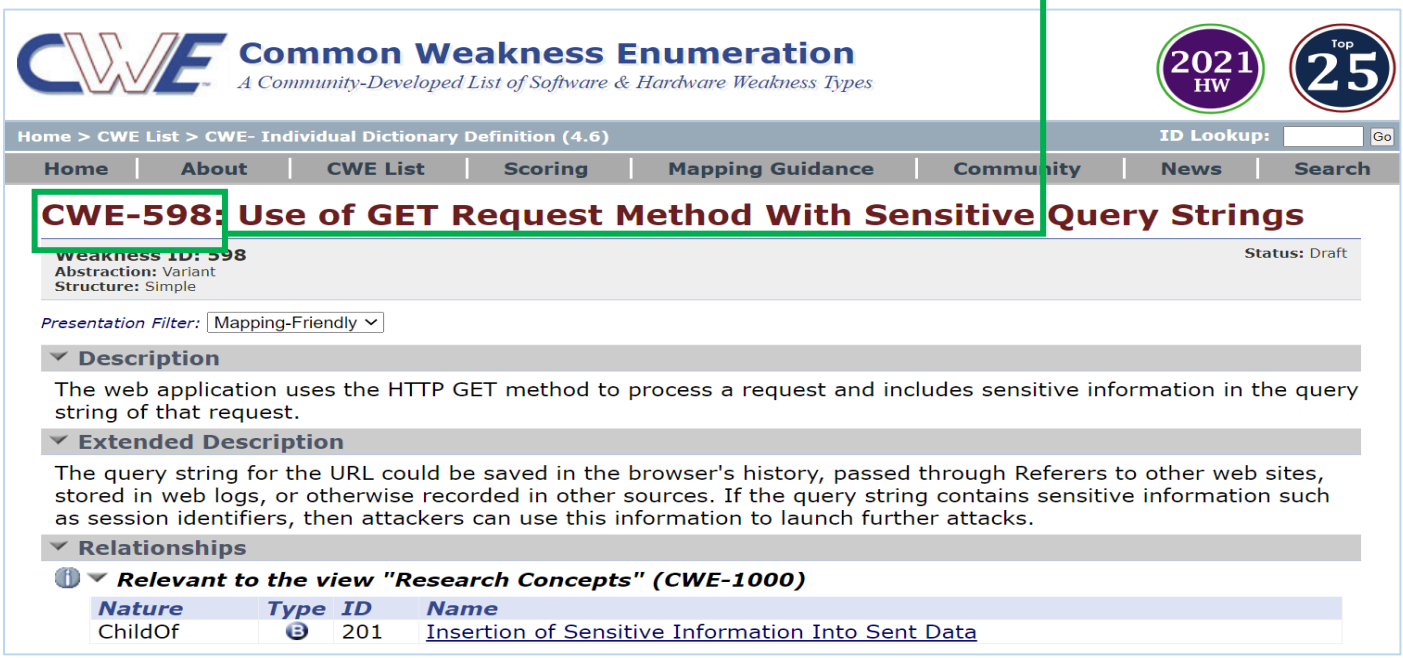


Figure 4 - Exemple de CWE

La figure 5 montre en pourcentage la couverture des contrôles des rubriques de vérification pour les trois niveaux (N1 à N3) du référentiel de vérification de la sécurité des applications et le top 10 de l’OWASP.

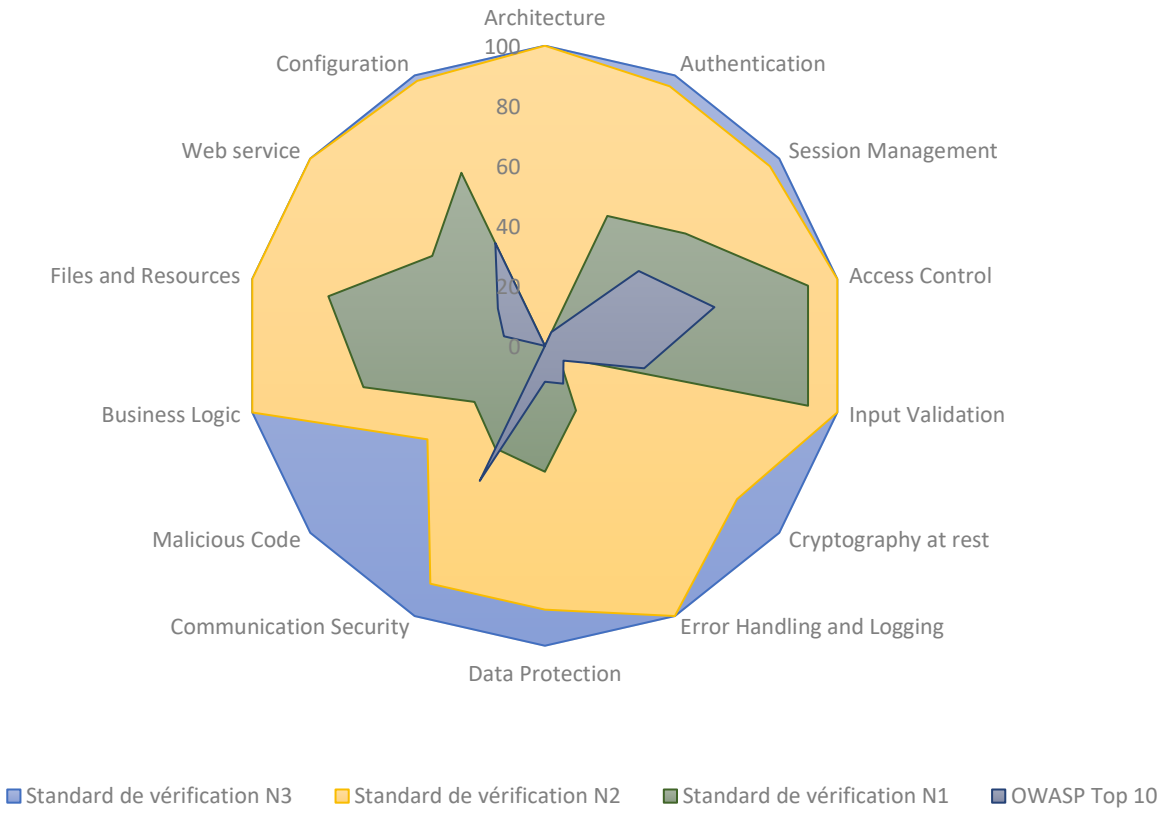


Figure 5 - Comparaison du domaine de sécurité des applications

V1 Architecture, conception et modélisation des menaces

Dans ce chapitre, le référentiel couvre les principaux aspects de toute architecture de sécurité robuste : disponibilité, confidentialité, intégrité du traitement, non-répudiation et respect de la vie privée. Chacun de ces principes de sécurité doit être intégré et appliqué à toutes les applications. Il est essentiel d'adopter l'approche "shift to the left", qui consiste à tester plus tôt dans le cycle de développement logiciel, en commençant par l'habilitation des développeurs avec des listes de contrôle de codage sécurisé, la formation, le codage et les tests, la construction, le déploiement, la configuration et les opérations, et en terminant par des tests indépendants de suivi pour s'assurer que tous les contrôles de sécurité sont présents et fonctionnels.

1.1 Cycle de vie du développement de logiciels sécurisés

#	Description	N1	N2	N3	CWE
1.1.1	Vérifier l'utilisation d'un cycle de développement de logiciel sécurisé qui prend en compte la sécurité à tous les stades du développement. *Verify the use of a secure software development lifecycle that addresses security in all stages of development.*		✓	✓	
1.1.2	Vérifier l'utilisation de la modélisation des menaces pour chaque modification de conception ou planification de sprint afin d'identifier les menaces, de planifier les contre-mesures, de faciliter les réponses appropriées aux risques et d'orienter les tests de sécurité. *Verify the use of threat modelling for every design change or sprint planning to identify threats, plan for countermeasures, facilitate appropriate risk responses, and guide security testing.*		✓	✓	1053
1.1.3	Vérifier que tous les récits utilisateurs et les fonctionnalités contiennent des contraintes de sécurité fonctionnelles, telles que "En tant qu'utilisateur, je devrais pouvoir consulter et modifier mon profil. Je ne devrais pas pouvoir voir ou modifier le profil de quelqu'un d'autre". *Verify that all user stories and features contain functional security constraints, such as "As a user, I should be able to view and edit my profile. I should not be able to view or edit anyone else's profile".*		✓	✓	1110
1.1.4	Vérifier la documentation et la justification de toutes les frontières de confiance de la demande, de ses composantes et des flux de données importants. *Verify documentation and justification of all the application's trust boundaries, components, and significant data flows.*		✓	✓	1059
1.1.5	Vérifier la définition et l'analyse de sécurité de l'architecture de haut niveau de l'application et de tous les services à distance connectés. *Verify definition and security analysis of the application's high-level architecture and all connected remote services.*		✓	✓	1059
1.1.6	Vérifier la mise en œuvre de contrôles de sécurité centralisés, simples (économie de conception), vérifiés, sécurisés et réutilisables pour éviter les contrôles en double, manquants, inefficaces ou peu sûrs. *Verify implementation of centralized, simple (economy of design), vetted, secure, and reusable security controls to avoid duplicate, missing, ineffective, or insecure controls.*		✓	✓	637
1.1.7	Vérifier que tous les développeurs et testeurs disposent d'une liste de contrôle de codage sécurisé, d'exigences de sécurité, de lignes directrices ou de politiques. *Verify availability of a secure coding checklist, security requirements, guideline, or policy to all developers and testers.*		✓	✓	637

1.2 Architecture d'authentification

#	Description	N1	N2	N3	CWE
1.2.1	Vérifier l'utilisation de comptes de système d'exploitation uniques à faible privilège ou spéciaux pour tous les composants, services et serveurs d'application. *Verify the use of unique or special low-privilege operating system accounts for all application components, services, and servers.*		✓	✓	250
1.2.2	Vérifier que les communications entre les composants de l'application, y compris les API, les intergiciels et les couches de données, sont authentifiées. Les composants doivent disposer des privilèges les moins nécessaires. *Verify that communications between application components, including APIs, middleware and data layers, are authenticated. Components should have the least necessary privileges needed.*		✓	✓	306
1.2.3	Vérifier que l'application utilise un mécanisme d'authentification unique approuvé qui est connu pour être sécurisé, peut être étendu pour inclure une authentification forte et dispose d'une journalisation et d'une surveillance suffisantes pour détecter les abus ou les violations de compte. *Verify that the application uses a single vetted authentication mechanism that is known to be secure, can be extended to include strong authentication, and has sufficient logging and monitoring to detect account abuse or breaches.*		✓	✓	306
1.2.4	Vérifier que tous les chemins d'authentification ainsi que les API de gestion des identités implémentent des contrôles de sécurité d'authentification forts et cohérents, de manière qu'il n'y ait pas d'alternatives plus faibles à chaque risque d'application. *Verify that all authentication pathways and identity management APIs implement consistent authentication security control strength, such that there are no weaker alternatives per the risk of the application.*		✓	✓	306

1.3 Architecture de la gestion des sessions

Il s'agit d'un point de repère pour les futures exigences architecturales.

1.4 Architecture de contrôle d'accès

#	Description	N1	N2	N3	CWE
1.4.1	Vérifier que des points d'application de confiance tels que les passerelles de contrôle d'accès, les serveurs et les fonctions sans serveur font respecter les contrôles d'accès. N'imposez jamais de contrôles d'accès au client. *Verify that trusted enforcement points, such as access control gateways, servers, and serverless functions, enforce access controls. Never enforce access controls on the client.*		✓	✓	602
1.4.4	Vérifier que l'application utilise un mécanisme de contrôle d'accès unique et bien étudié pour accéder aux données et ressources protégées. Toutes les demandes doivent passer par ce mécanisme unique pour éviter le copier-coller ou les chemins alternatifs non sécurisés. *Verify the application uses a single and well-vetted access control mechanism for accessing protected data and resources. All requests must pass through this single mechanism to avoid copy and paste or insecure alternative paths.*		✓	✓	284
1.4.5	Vérifier que le contrôle d'accès basé sur les attributs ou les caractéristiques est utilisé, c'est-à-dire que le code vérifie l'autorisation de l'utilisateur pour une caractéristique ou une donnée plutôt que son seul rôle. Les autorisations doivent tout de même être attribuées à l'aide de rôles. *Verify that attribute or feature-based access control is used whereby the code checks the user's authorization for a feature/data item rather than just their role. Permissions should still be allocated using roles.*		✓	✓	275

1.5 Architecturales d'entrée et de sortie

La limite de confiance est toujours un enjeu, il est possible de contourner les décisions prises sur des navigateurs ou des appareils clients non fiables. Cependant, dans les déploiements architecturaux courants d'aujourd'hui, le point d'application de la confiance a considérablement changé. Par conséquent, lorsque le terme "couche de service de confiance" est utilisé dans le référentiel, nous entendons par là tout point d'application de confiance, quel que soit son emplacement, tel qu'un micro service, un API sans serveur, côté serveur, un API de confiance sur un périphérique client qui a un démarrage sécurisé, des API partenaires ou externes, etc.

#	Description	N1	N2	N3	CWE
1.5.1	Vérifier que les exigences en matière d'entrée et de sortie définissent clairement la manière de traiter et d'exploiter les données en fonction du type, du contenu et de la conformité aux lois, règlements et autres politiques applicables. *Verify that input and output requirements clearly define how to handle and process data based on type, content, and applicable laws, regulations, and other policy compliance.*		✓	✓	1029
1.5.2	Vérifier que la sérialisation n'est pas utilisée lorsque vous communiquez avec des clients non fiables. Si cela n'est pas possible, assurez-vous que des contrôles d'intégrité adéquats (et éventuellement un cryptage si des données sensibles sont envoyées) sont appliqués pour empêcher les attaques de désérialisation[7], y compris l'injection d'objets. *Verify that serialization is not used when communicating with untrusted clients. If this is not possible, ensure that adequate integrity controls (and possibly encryption if sensitive data is sent) are enforced to prevent deserialization attacks including object injection.*		✓	✓	502
1.5.3	Vérifier que la validation des entrées est appliquée sur une couche de service de confiance. *Verify that input validation is enforced on a trusted service layer.*		✓	✓	602
1.5.4	Vérifier que l'encodage de sortie se fait à proximité ou par l'interprète auquel il est destiné. *Verify that output encoding occurs close to or by the interpreter for which it is intended.*		✓	✓	116

1.6 Architecture cryptographique

Les applications doivent être conçues à base d'une architecture cryptographique solide pour pouvoir protéger les données selon leur classification. Tout chiffrer est un gaspillage, ne rien chiffrer est une négligence légale. Un équilibre doit être trouvé, généralement lors de la conception architecturale ou de haut niveau, des sprints de conception ou des pics architecturaux. Concevoir la cryptographie après développement coûtera inévitablement beaucoup plus cher à mettre en œuvre de manière sécurisée que de l'intégrer dès le départ.

Les exigences architecturales sont intrinsèques au code, et donc difficiles à unifier ou à intégrer dans les tests. Les exigences architecturales doivent être prises en compte dans les normes de codage, tout au long de la phase de codage, et doivent être examinées au cours des phases de l'architecture de sécurité, des revues du code par les pairs, ou des rétrospectives.

#	Description	N1	N2	N3	CWE
1.6.1	Vérifier qu'il existe une politique explicite de gestion des clés cryptographiques et que le cycle de vie d'une clé cryptographique suit une norme de gestion des clés telle que NIST SP 800-57. *Verify that there is an explicit policy for management of cryptographic keys and that a cryptographic key lifecycle follows a key management standard such as NIST SP 800-57.*		✓	✓	320

[7] La désérialisation non sécurisée : Est une vulnérabilité qui se produit lorsque des données non fiables sont utilisées pour abuser de la logique d'une application, causer un déni de service (DoS, Deny of Service), ou même exécuter du code arbitraire.

1.6.2	Vérifier que les consommateurs de services cryptographiques protègent les clés et autres secrets en utilisant des coffres-forts de clés ou des alternatives basées sur l'API. *Verify that consumers of cryptographic services protect key material and other secrets by using key vaults or API based alternatives.*		✓	✓	320
1.6.3	Vérifier que toutes les clés et tous les mots de passe sont remplaçables et font partie d'un processus bien défini de recryptage des données sensibles. *Verify that all keys and passwords are replaceable and are part of a well-defined process to re-encrypt sensitive data.*		✓	✓	320
1.6.4	Vérifier que l'architecture traite les clés symétriques, les mots de passe ou les secrets d'API générés par les clients en particulier ceux qui ne sont pas sécurisés, et ne les utilise jamais pour protéger ou accéder à des données sensibles. *Verify that the architecture treats client-side secrets--such as symmetric keys, passwords, or API tokens--as insecure and never uses them to protect or access sensitive data.*		✓	✓	320

1.7 Architecture des erreurs, d'enregistrement et de vérification

#	Description	N1	N2	N3	CWE
1.7.1	Vérifier qu'un format commun de journalisation soit utilisé dans le système. *Verify that a common logging format and approach is used across the system.*		✓	✓	1009
1.7.2	Vérifier que les journaux sont transmis de manière sécurisée à un système de préférence distant pour analyse, détection, alerte et escalade. *Verify that logs are securely transmitted to a preferably remote system for analysis, detection, alerting, and escalation.*		✓	✓	

1.8 Exigences architecturales en matière de protection des données et de la vie privée

#	Description	N1	N2	N3	CWE
1.8.1	Vérifier que toutes les données sensibles sont identifiées et classées en niveaux de protection. *Verify that all sensitive data is identified and classified into protection levels.*		✓	✓	
1.8.2	Vérifier que tous les niveaux de protection sont associés à un ensemble d'exigences de protection, telles que des exigences de cryptage, d'intégrité, de conservation, de respect de la vie privée et d'autres exigences de confidentialité, et que celles-ci sont appliquées dans l'architecture. *Verify that all protection levels have an associated set of protection requirements, such as encryption requirements, integrity requirements, retention, privacy and other confidentiality requirements, and that these are applied in the architecture.*		✓	✓	

1.9 Architecture des communications

#	Description	N1	N2	N3	CWE
1.9.1	Vérifier que l'application chiffre les communications entre les composants, en particulier lorsque ces composants se trouvent dans des conteneurs, systèmes, sites ou fournisseurs de cloud différents. *Verify the application encrypts communications between components, particularly when these components are in different containers, systems, sites, or cloud providers.*		✓	✓	319

1.9.2	Vérifier que les composants de l'application vérifient l'authenticité de chaque partie d'un lien de communication afin de prévenir les attaques de type "man-in-the-middle". Par exemple, les composants d'application doivent valider les certificats et les chaînes TLS. *Verify that application components verify the authenticity of each side in a communication link to prevent person-in-the-middle attacks. For example, application components should validate TLS certificates and chains.*		✓	✓	295

1.10 Architecture des logiciels malveillants

#	Description	N1	N2	N3	CWE
1.10.1	Vérifier qu'un système de contrôle du code source est utilisé, avec des procédures pour s'assurer que les enregistrements sont accompagnés de tickets d'émission ou de modification. Le système de contrôle du code source doit disposer d'un contrôle d'accès et d'utilisateurs identifiables pour permettre la traçabilité de toute modification. *Verify that a source code control system is in use, with procedures to ensure that check-ins are accompanied by issues or change tickets. The source code control system should have access control and identifiable users to allow traceability of any changes.*		✓	✓	284

1.11 Architecture de la logique d'entreprise

#	Description	N1	N2	N3	CWE
1.11.1	Vérifier la définition et la documentation de tous les composants de l'application en ce qui concerne la logique métier ou de sécurité qu'ils fournissent. *Verify the definition and documentation of all application components in terms of the business or security functions they provide.*		✓	✓	1059
1.11.2	Vérifier que tous les flux de logique métier de grande valeur, y compris l'authentification, la gestion de session et le contrôle d'accès, ne partagent pas un état non synchronisé. *Verify that all high-value business logic flows, including authentication, session management and access control, do not share unsynchronized state.*		✓	✓	362
1.11.3	Vérifier que tous les flux de logique métier de grande valeur, y compris l'authentification, la gestion de session et le contrôle d'accès, sont sécurisés et résistants aux conditions de concurrence au temps de contrôle et au temps d'utilisation. *Verify that all high-value business logic flows, including authentication, session management and access control are thread safe and resistant to time-of-check and time-of-use race conditions.*			✓	367

1.12 Architecture sécurisée du téléchargement des fichiers

#	Description	N1	N2	N3	CWE
1.12.2	Vérifier que les fichiers envoyés par l'utilisateur - s'ils doivent être affichés ou téléchargés à partir de l'application - sont servis par des téléchargements en flux d'octets, ou à partir d'un domaine sans rapport, comme un compartiment de stockage de fichiers en cloud. Mettre en œuvre une politique de sécurité du contenu (CSP) appropriée pour réduire le risque de vecteurs XSS ou d'autres attaques provenant du fichier téléchargé. *Verify that user-uploaded files - if required to be displayed or downloaded from the application - are served by either octet stream downloads, or from an unrelated domain, such as a cloud file storage bucket. Implement a suitable Content Security Policy (CSP) to reduce the risk from XSS vectors or other attacks from the uploaded file.*		✓	✓	646

1.13 Architecturales des API

Il s'agit d'un point de repère pour les futures exigences architecturales.

1.14 Architecture de la configuration

#	Description	N1	N2	N3	CWE
1.14.1	Vérifier la séparation des composants de différents niveaux de confiance via des contrôles de sécurité bien définis, des règles de pare-feu, des passerelles API, des proxys inverses, des groupes de sécurité basés sur le cloud ou des mécanismes similaires. *Verify the segregation of components of differing trust levels through well-defined security controls, firewall rules, API gateways, reverse proxies, cloud-based security groups, or similar mechanisms.*		✓	✓	923
1.14.2	Vérifier que les signatures binaires, les connexions de confiance et les nœuds vérifiés sont utilisés pour déployer des binaires sur des dispositifs distants. *Verify that binary signatures, trusted connections, and verified endpoints are used to deploy binaries to remote devices.*		✓	✓	494
1.14.3	Vérifier que le pipeline de construction signale les composants obsolètes ou peu sûrs et prend les mesures appropriées. *Verify that the build pipeline warns of out-of-date or insecure components and takes appropriate actions.*		✓	✓	1104
1.14.4	Vérifier que le pipeline de construction contient une étape de compilation pour créer automatiquement et vérifier le déploiement sécurisé de l'application, en particulier si l'infrastructure de l'application est définie par un logiciel, comme les scripts de construction d'un environnement en nuage. *Verify that the build pipeline contains a build step to automatically build and verify the secure deployment of the application, particularly if the application infrastructure is software defined, such as cloud environment build scripts.*		✓	✓	
1.14.5	Vérifier que les déploiements d'applications sont correctement Sand boxés, conteneurisés et/ou isolés au niveau du réseau pour retarder et dissuader les attaquants d'attaquer d'autres applications, en particulier lorsqu'ils effectuent des actions sensibles ou dangereuses telles que la désérialisation[8]. *Verify that application deployments adequately sandbox, containerize and/or isolate at the network level to delay and deter attackers from attacking other applications, especially when they are performing sensitive or dangerous actions such as deserialization.*		✓	✓	265
1.14.6	Vérifier que l'application n'utilise pas de technologies côté client non supportées, peu sûres ou obsolètes telles que les plugins NSAPI, Flash, Shockwave, ActiveX, Silverlight, NACL ou les applets Java côté client. *Verify the application does not use unsupported, insecure, or deprecated client-side technologies such as NSAPI plugins, Flash, Shockwave, ActiveX, Silverlight, NACL, or client-side Java applets.*		✓	✓	477

[8] Désérialisation : Processus inverse de construction d'une structure de données ou d'un objet à partir d'une série d'octets.

V2 Authentification

L'authentification est un processus permettant au système de s'assurer de la légitimité de la demande d'accès faite par une entité afin d'autoriser l'accès de cette entité à des ressources du système conformément au paramétrage du contrôle d'accès.

2.1 Sécurité des mots de passe

Les mots de passe, appelés "Memorized Secrets" par NIST 800-63, comprennent les mots de passe, les codes PIN, les motifs de déverrouillage, le choix du chaton correct ou d'un autre élément d'image, et les phrases de passe. Ils sont généralement considérés comme "quelque chose que vous savez" et sont souvent utilisés comme des authentificateurs à facteur unique. L'utilisation continue de l'authentification à un facteur unique pose des problèmes importants, notamment les milliards de noms d'utilisateur et de mots de passe valides divulgués sur l'internet, les mots de passe par défaut ou faibles, les tables arc-en-ciel et les dictionnaires ordonnés des mots de passe les plus courants.

Les applications devraient fortement encourager les utilisateurs à adopter l'authentification multi-facteurs, et devraient permettre aux utilisateurs de réutiliser les jetons qu'ils possèdent déjà, tels que les jetons FIDO[9] ou U2F, ou de s'abonner à un fournisseur de services d'accréditation qui fournit une authentification multi-facteurs.

#	Description	N1	N2	N3	CWE	NIST
2.1.1	Vérifier que les mots de passe définis par l'utilisateur comportent au moins 12 caractères, (après la combinaison de plusieurs espaces). *Verify that user set passwords are at least 12 characters in length (after multiple spaces are combined).*	✓	✓	✓	521	5.1.1.2
2.1.2	Vérifier que les mots de passe de 64 caractères sont autorisés et que les mots de passe de plus de 128 caractères sont interdits. *Verify that passwords of at least 64 characters are permitted, and that passwords of more than 128 characters are denied.*	✓	✓	✓	521	5.1.1.2
2.1.3	Vérifier que la troncature du mot de passe n'est pas autorisée. Toutefois, plusieurs espaces consécutifs peuvent être remplacés par un seul espace. *Verify that password truncation is not performed. However, consecutive multiple spaces may be replaced by a single space.*	✓	✓	✓	521	5.1.1.2
2.1.4	Vérifier que tout caractère Unicode imprimable, y compris les caractères neutres en langue tels que les espaces et les emojis, est autorisé dans les mots de passe. *Verify that any printable Unicode character, including language neutral characters such as spaces and Emojis are permitted in passwords.*	✓	✓	✓	521	5.1.1.2
2.1.5	Vérifier que les utilisateurs peuvent changer leur mot de passe. *Verify users can change their password.*	✓	✓	✓	620	5.1.1.2
2.1.6	Vérifier que la fonctionnalité de changement de mot de passe nécessite le mot de passe actuel et le nouveau mot de passe de l'utilisateur. *Verify that password change functionality requires the user's current and new password.*	✓	✓	✓	620	5.1.1.2

[9] Fido : (**F**ast **Id**entity **O**nline) est une spécification technique pour l'authentification de l'identité de l'utilisateur en ligne. elle est utilisée dans des scénarios tels que la connexion par empreinte digitale et la connexion à deux facteurs.

2.1.7	Vérifier que les mots de passe soumis lors de l'enregistrement du compte, de la connexion et de la modification du mot de passe sont vérifiés par rapport à un ensemble de mots de passe non respectés, soit localement (comme les 1 000 ou 10 000 mots de passe les plus courants qui correspondent à la politique du système en matière de mots de passe), soit en utilisant une API externe. En cas d'utilisation d'une API, il convient d'utiliser un mécanisme de vérification de l'absence de connaissance ou un autre mécanisme pour s'assurer que le mot de passe en texte clair n'est pas envoyé ou utilisé pour vérifier l'état de violation du mot de passe. En cas de violation du mot de passe, l'application doit demander à l'utilisateur de définir un nouveau mot de passe non violé. *Verify that passwords submitted during account registration, login, and password change are checked against a set of breached passwords either locally (such as the top 1,000 or 10,000 most common passwords which match the system's password policy) or using an external API. If using an API a zero knowledge proof or other mechanism should be used to ensure that the plain text password is not sent or used in verifying the breach status of the password. If the password is breached, the application must require the user to set a new non-breached password.*	✓	✓	✓	521	5.1.1.2
2.1.8	Vérifier qu'un indicateur de force de mot de passe est implémenté pour aider les utilisateurs à définir un mot de passe plus fort. *Verify that a password strength meter is provided to help users set a stronger password.*	✓	✓	✓	521	5.1.1.2
2.1.9	Vérifier qu'il n'existe pas de règles de composition des mots de passe limitant le type de caractères autorisés. Il ne doit pas y avoir de restrictions sur l'utilisation de majuscules ou de minuscules, de chiffres ou de caractères spéciaux. *Verify that there are no password composition rules limiting the type of characters permitted. There should be no requirement for upper or lower case or numbers or special characters.*	✓	✓	✓	521	5.1.1.2
2.1.10	Vérifier qu'il n'y a pas d'exigences en matière de rotation périodique du mot de passe ou d'historique des mots de passe. *Verify that there are no periodic credential rotation or password history requirements.*	✓	✓	✓	263	5.1.1.2
2.1.11	Vérifier que la fonction "coller", les aides de mot de passe du navigateur et les gestionnaires de mots de passe externes sont autorisés. *Verify that "paste" functionality, browser password helpers, and external password managers are permitted.*	✓	✓	✓	521	5.1.1.2
2.1.12	Vérifier que l'utilisateur peut choisir d'afficher temporairement l'intégralité du mot de passe masqué ou d'afficher temporairement le dernier caractère tapé du mot de passe sur les plates-formes qui ne disposent pas de cette fonctionnalité intégrée. *Verify that the user can choose to either temporarily view the entire masked password, or temporarily view the last typed character of the password on platforms that do not have this as built-in functionality.*	✓	✓	✓	521	5.1.1.2

2.2 Authentificateurs

L'agilité des authentificateurs est essentielle pour que les applications puissent résister à l'épreuve du temps. Les vérificateurs d'applications Refactor[10] permettent d'ajouter des authentificateurs supplémentaires en fonction des préférences de l'utilisateur, et de résilier de manière ordonnée les authentificateurs obsolètes ou dangereux.

#	Description	N1	N2	N3	CWE	NIST
2.2.1	Vérifier que les contrôles anti-automatisation sont efficaces pour atténuer les attaques par violation des tests d'accréditation, par énumération exhaustive (brute-force) et par verrouillage de compte. Ces contrôles comprennent le blocage des mots de passe les plus courants, les verrouillages progressifs, la limitation de débit, les CAPTCHA, les délais toujours plus longs entre les tentatives, les restrictions d'adresse IP ou les restrictions basées sur le risque telles que l'emplacement, la première connexion sur un appareil, les tentatives récentes de déverrouillage du compte, ou autres. Vérifier qu'il n'y a pas plus de 100 tentatives infructueuses par heure sur un seul compte. *Verify that anti-automation controls are effective at mitigating breached credential testing, brute force, and account lockout attacks. Such controls include blocking the most common breached passwords, soft lockouts, rate limiting, CAPTCHA, ever increasing delays between attempts, IP address restrictions, or risk-based restrictions such as location, first login on a device, recent attempts to unlock the account, or similar. Verify that no more than 100 failed attempts per hour is possible on a single account.*	✓	✓	✓	307	5.2.2 / 5.1.1.2 / 5.1.4.2 / 5.1.5.2
2.2.2	Vérifier que l'utilisation d'authentificateurs faibles (tels que les SMS et les e-mails) se limite à la vérification secondaire et à l'approbation des transactions et ne remplace pas les méthodes d'authentification plus sûres. Vérifier que les méthodes plus fortes sont proposées avant les méthodes faibles, que les utilisateurs sont conscients des risques, ou que des mesures appropriées sont en place pour limiter les risques de compromission du compte. *Verify that the use of weak authenticators (such as SMS and email) is limited to secondary verification and transaction approval and not as a replacement for more secure authentication methods. Verify that stronger methods are offered before weak methods, users are aware of the risks, or that proper measures are in place to limit the risks of account compromise.*	✓	✓	✓	304	5.2.10
2.2.3	Vérifier que des notifications sécurisées sont envoyées aux utilisateurs après la mise à jour des détails d'authentification, tels que la réinitialisation de l'identifiant, le changement d'adresse électronique ou d'adresse, la connexion à partir d'un lieu inconnu ou risqué. L'utilisation de notifications "push" - plutôt que de SMS ou d'e-mail - est préférable, mais en l'absence de notifications "push", les SMS ou les e-mails sont acceptables tant qu'aucune information sensible n'est divulguée dans la notification. *Verify that secure notifications are sent to users after updates to authentication details, such as credential resets, email or address changes, logging in from unknown or risky locations. The use of push notifications - rather than SMS or email - is preferred, but in the absence of push notifications, SMS or email is acceptable as long as no sensitive information is disclosed in the notification.*	✓	✓	✓	620	

[10] Refactor : Technique d'amélioration d'un code source, on peut soit écrire un code source entièrement nouveau, soit restructurer le code par petites étapes.

2.2.4	Vérifier la résistance à l'usurpation d'identité contre le phishing, comme l'utilisation de l'authentification multi-facteurs, les dispositifs cryptographiques avec intention (comme les clés connectées avec un "push to authenticate"), ou à des niveaux AAL supérieurs, les certificats côté client. *Verify impersonation resistance against phishing, such as the use of multi-factor authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side certificates.*			✓	308	5.2.5
2.2.5	Vérifier que lorsqu'un fournisseur de services d'accréditation (CSP) et l'application vérifiant l'authentification sont séparés, un TLS mutuellement authentifié est en place entre les deux points terminaux. *Verify that where a Credential Service Provider (CSP) and the application verifying authentication are separated, mutually authenticated TLS is in place between the two endpoints.*			✓	319	5.2.6
2.2.6	Vérifier la résistance aux attaques de rejeu par l'utilisation obligatoire de dispositifs OTP, d'authentificateurs cryptographiques ou de codes de consultation. *Verify replay resistance through the mandated use of One-time Passwords (OTP) devices, cryptographic authenticators, or lookup codes.*			✓	308	5.2.8
2.2.7	Vérifier l'intention d'authentification en exigeant l'entrée d'un jeton OTP ou une action initiée par l'utilisateur telle qu'une pression sur un bouton d'une clé matérielle FIDO. *Verify intent to authenticate by requiring the entry of an OTP token or user-initiated action such as a button press on a FIDO hardware key.*			✓	308	5.2.9

2.3 Cycle de vie des authentificateurs

Les authentificateurs sont les mots de passe, les jetons logiciels, les jetons matériels et les dispositifs biométriques. Le cycle de vie des authentificateurs est essentiel pour la sécurité d'une application - si quelqu'un peut s'enregistrer lui-même sur un compte sans preuve d'identité, il ne peut guère faire confiance à l'affirmation de son identité. Pour les systèmes sensibles, comme les systèmes bancaires, il est essentiel de mettre davantage l'accent sur l'enregistrement et la délivrance de justificatifs d'identité et de dispositifs pour assurer la sécurité de l'application.

Remarque : les mots de passe ne doivent pas avoir une durée de vie maximale ni être soumis à une rotation. Les mots de passe doivent être vérifiés et non remplacés régulièrement.

#	Description	N1	N2	N3	CWE	NIST
2.3.1	Vérifier que les mots de passe ou codes d'activation initiaux générés par le système DOIVENT être générés de manière aléatoire et sécurisée, DOIVENT comporter au moins 6 caractères, PEUVENT contenir des lettres et des chiffres, et expirent après une courte période de temps. Ces secrets initiaux ne doivent pas être autorisés à devenir le mot de passe à long terme. *Verify system generated initial passwords or activation codes SHOULD be securely randomly generated, SHOULD be at least 6 characters long, and MAY contain letters and numbers, and expire after a short period of time. These initial secrets must not be permitted to become the long term password*	✓	✓	✓	330	5.1.1.2 / A.3
2.3.2	Vérifier que l'inscription et l'utilisation de dispositifs d'authentification fournis par l'abonné sont prises en charge, comme les jetons U2F ou FIDO. *Verify that enrolment and use of user-provided authentication devices are supported, such as a U2F or FIDO tokens.*		✓	✓	308	6.1.3
2.3.3	Vérifier que les instructions de renouvellement sont envoyées suffisamment tôt pour renouveler les authentificateurs à durée déterminée. *Verify that renewal instructions are sent with sufficient time to renew time bound authenticators.*		✓	✓	287	6.1.4

2.4 Stockage des identifiants

Les architectes et les développeurs doivent se conformer à cette section lorsqu'ils construisent ou remanient du code. Cette section ne peut être entièrement vérifiée qu'en utilisant la révision du code source ou par des tests unitaires ou d'intégration sécurisés. Les tests d'intrusions ne peuvent pas identifier l'un de ces problèmes.

Cette section ne peut pas être soumise à un test de pénétration, les contrôles ne sont donc pas marqués comme N1. Cependant, cette section est d'une importance vitale pour la sécurité des données d'identification en cas de vol.

#	Description	N1	N2	N3	CWE	NIST
2.4.1	Vérifier que les mots de passe sont stockés sous une forme qui résiste aux attaques hors ligne. Les mots de passe DOIVENT être salés[11] et hachés en utilisant une fonction approuvée à sens unique ou de hachage de mot de passe. Les fonctions de dérivation de clé et de hachage de mot de passe prennent un mot de passe, un sel et un facteur de coût (ex. : nombre d'itération algorithmique) comme intrants lors de la génération d'un hachage de mot de passe. *Verify that passwords are stored in a form that is resistant to offline attacks. Passwords SHALL be salted and hashed using an approved one-way key derivation or password hashing function. Key derivation and password hashing functions take a password, a salt, and a cost factor as inputs when generating a password hash.*		✓	✓	916	5.1.1.2
2.4.2	Vérifier que le sel a une longueur d'au moins 32 bits et qu'il est choisi arbitrairement pour minimiser les collisions de la valeur du sel parmi les hashs stockés. Pour chaque authentifiant, une valeur de sel unique et le hachage qui en résulte DOIVENT être stockés. *Verify that the salt is at least 32 bits in length and be chosen arbitrarily to minimize salt value collisions among stored hashes. For each credential, a unique salt value and the resulting hash SHALL be stored.*		✓	✓	916	5.1.1.2
2.4.3	Vérifier que si le PBKDF2 est utilisé, le nombre d'itérations DOIT être aussi important que les performances du serveur de vérification le permettent, généralement au moins 100 000 itérations. *Verify that if PBKDF2 is used, the iteration count SHOULD be as large as verification server performance will allow, typically at least 100,000 iterations.*		✓	✓	916	5.1.1.2
2.4.4	Vérifier que si bcrypt est utilisé, le facteur de travail DOIT être aussi important que les performances du serveur de vérification le permettent, généralement au moins 10. *Verify that if bcrypt is used, the work factor SHOULD be as large as verification server performance will allow, with a minimum of 10.*		✓	✓	916	5.1.1.2
2.4.5	Vérifier qu'une itération supplémentaire d'une fonction de dérivation clé est effectuée, en utilisant une valeur de sel (salt value) qui est secrète et connue uniquement du vérificateur. Générer la valeur de sel en utilisant un générateur de bits aléatoires approuvé [SP 800-90Ar1] et fournir au moins la sécurité minimale spécifiée dans la dernière révision de la norme SP 800-131A. La valeur de sel secrète DOIT être stockée séparément des mots de passe hachés (par exemple, dans un dispositif spécialisé comme un module de sécurité matériel). *Verify that an additional iteration of a key derivation function is performed, using a salt value that is secret and known only to the verifier. Generate the salt value using an approved random bit generator [SP 800-90Ar1] and provide at least the minimum security strength specified in the latest revision of SP 800-131A. The secret salt value SHALL be stored separately from the hashed passwords (e.g., in a specialized device like a hardware security module).*		✓	✓	916	5.1.1.2

[11] Le salage de mot de passe : Une méthode pour rendre l'empreinte des mots de passe plus sûre en ajoutant aux mots de passe une chaîne de caractères aléatoires avant de calculer leur empreinte md5.

2.5 Récupération des identifiants

#	Description	N1	N2	N3	CWE	NIST
2.5.1	Vérifier que si un secret d'activation initiale ou de récupération du système est envoyé à l'utilisateur, il est à usage unique, limité dans le temps et aléatoire. *Verify that a system generated initial activation or recovery secret is not sent in clear text to the user.*	✓	✓	✓	640	5.1.1.2
2.5.2	Vérifier que les indices de mot de passe ou l'authentification basée sur la connaissance (dites "questions secrètes") ne sont pas présents. *Verify password hints or knowledge-based authentication (so-called "secret questions") are not present.*	✓	✓	✓	640	5.1.1.2
2.5.3	Vérifier que la récupération du mot de passe ne révèle en aucune façon le mot de passe actuel. *Verify password credential recovery does not reveal the current password in any way.*	✓	✓	✓	640	5.1.1.2
2.5.4	Vérifier que les comptes partagés ou par défaut ne sont pas présents (par exemple "root", "admin" ou "sa"). *Verify shared or default accounts are not present (e.g. "root", "admin", or "sa").*	✓	✓	✓	16	5.1.1.2 / A.3
2.5.5	Vérifier que si un facteur d'authentification est modifié ou remplacé, l'utilisateur est informé de cet événement. *Verify that if an authentication factor is changed or replaced, that the user is notified of this event.*	✓	✓	✓	304	6.1.2.3
2.5.6	Vérifier les mots de passe oubliés, et les autres chemins de récupération utilisent un mécanisme de récupération sécurisé, tel que TOTP ou autre soft token, mobile push, ou un autre mécanisme de récupération hors ligne. *Verify forgotten password, and other recovery paths use a secure recovery mechanism, such as time-based OTP (TOTP) or other soft token, mobile push, or another offline recovery mechanism.*	✓	✓	✓	640	5.1.1.2
2.5.7	Vérifier qu'en cas de perte des facteurs d'authentification OTP ou multi-facteurs, la preuve d'identité est effectuée au même niveau que lors de l'inscription. *Verify that if OTP or multi-factor authentication factors are lost, that evidence of identity proofing is performed at the same level as during enrolment.*		✓	✓	308	6.1.2.3

2.6 Vérificateurs des secrets

Les tables d'authentifications secrètes sont des listes pré-générées de codes secrets, similaires aux numéros d'autorisation de transaction TAN[12], aux codes de récupération des médias sociaux ou à une grille contenant un ensemble de valeurs aléatoires. Ils sont distribués aux utilisateurs en toute sécurité. Ces codes de ''recherche'' sont utilisés une fois, et une fois qu'ils sont tous utilisés, la liste secrète de ''recherche'' est abandonnée. Ce type d'authentificateur est considéré comme "quelque chose que vous avez".

#	Description	N1	N2	N3	CWE	NIST
2.6.1	Vérifier que les secrets de la table d'authentification ne peuvent être utilisés qu'une seule fois. *Verify that lookup secrets can be used only once.*		✓	✓	308	5.1.2.2

[12] TAN : Transaction authorization number.

2.6.2	Vérifier que les secrets de la table d'authentification ont un caractère aléatoire suffisant (112 bits d'entropie) ou, s'ils ont moins de 112 bits d'entropie, qu'ils sont salés avec un sel unique et aléatoire de 32 bits et hachés avec un hachage unidirectionnel approuvé. *Verify that lookup secrets have sufficient randomness (112 bits of entropy), or if less than 112 bits of entropy, salted with a unique and random 32-bit salt and hashed with an approved one-way hash.*		✓	✓	330	5.1.2.2
2.6.3	Vérifier que les secrets de la table d'authentification résistent aux attaques hors ligne, comme les valeurs prévisibles. *Verify that lookup secrets are resistant to offline attacks, such as predictable values.*		✓	✓	310	5.1.2.2

2.7 Vérificateurs hors bande

Dans le passé, un vérificateur hors bande aurait été un courriel ou un SMS contenant un lien de réinitialisation du mot de passe. Les attaquants utilisent ce faible mécanisme pour réinitialiser des comptes qu'ils ne contrôlent pas encore, par exemple en prenant le compte de courrier électronique d'une personne et en réutilisant tout lien de réinitialisation mis à leur disposition. Il existe de meilleurs moyens de gérer la vérification hors bande.

Les authentificateurs hors bande sécurisés sont des dispositifs physiques qui peuvent communiquer avec le vérificateur par un canal secondaire sécurisé. Les notifications "push notifications" vers les appareils mobiles en sont des exemples. Ce type d'authentificateur est considéré comme "quelque chose que vous avez". Lorsqu'un utilisateur souhaite s'authentifier, l'application de vérification envoie un message à l'authentificateur hors bande via une connexion à l'authentificateur directement ou indirectement par le biais d'un service tiers. Le message contient un code d'authentification (généralement un nombre aléatoire de six chiffres ou un dialogue d'approbation modale). L'application de vérification attend de recevoir le code d'authentification par le canal principal et compare le hachage de la valeur reçue au hachage du code d'authentification original. En cas de correspondance, le vérificateur hors bande peut supposer que l'utilisateur est authentique.

Le référentiel de vérification de la sécurité des applications suppose qu'une minorité de développeurs recouvrent aux nouveaux authentificateurs hors bande, tels que les notifications "push notifications". Les contrôles suivants du standard s'appliquent ainsi aux vérificateurs, tels que l'API d'authentification, les applications et les mises en œuvre de l'authentification unique. Si vous développez un nouveau authentificateur hors bande, il faut se référer à NIST 800-63B 5.1.3.1.

Les authentificateurs hors bande dangereux tels que le courrier électronique et la voix sur IP ne sont pas autorisés. Les authentifications PSTN et SMS sont actuellement "restreintes" par NIST et devraient être interdites au profit des notifications "push" ou similaires. Si vous devez utiliser l'authentification hors bande par téléphone ou SMS, consulter le paragraphe 5.1.3.3.

#	Description	N1	N2	N3	CWE	NIST
2.7.1	Vérifier que les authentificateurs de texte en clair hors bande (NIST "restreint"), tels que les SMS ou le PSTN, ne sont pas proposés par défaut, et que des alternatives plus fortes, telles que les notifications "push", sont proposées en premier lieu. *Verify that clear text out of band (NIST "restricted") authenticators, such as SMS or PSTN, are not offered by default, and stronger alternatives such as push notifications are offered first.*	✓	✓	✓	287	5.1.3.2
2.7.2	Vérifier que le vérificateur hors bande expire les demandes d'authentification, les codes ou les jetons hors bande après 10 minutes. *Verify that the out of band verifier expires out of band authentication requests, codes, or tokens after 10 minutes.*	✓	✓	✓	287	5.1.3.2
2.7.3	Vérifier que les demandes d'authentification, codes ou jetons hors bande du vérificateur ne sont utilisables qu'une seule fois, et uniquement pour la demande d'authentification originale. *Verify that the out of band verifier authentication requests, codes, or tokens are only usable once, and only for the original authentication request.*	✓	✓	✓	287	5.1.3.2

2.7.4	Vérifier que l'authentificateur et le vérificateur hors bande communiquent sur un canal indépendant sécurisé. *Verify that the out of band authenticator and verifier communicates over a secure independent channel.*	✓	✓	✓	523	5.1.3.2
2.7.5	Vérifier que le vérificateur hors bande ne conserve qu'une version hachée du code d'authentification. *Verify that the out of band verifier retains only a hashed version of the authentication code.*		✓	✓	256	5.1.3.2
2.7.6	Vérifier que le code d'authentification initial est généré par un générateur de nombres aléatoires sécurisé, contenant au moins 20 bits d'entropie (en général, un nombre aléatoire de six chiffres est suffisant). *Verify that the initial authentication code is generated by a secure random number generator, containing at least 20 bits of entropy (typically a six digital random number is sufficient).*		✓	✓	310	5.1.3.2

2.8 Vérificateurs uniques

Les mots de passe à usage unique (OTP) sont des jetons physiques ou logiciels qui affichent un défi pseudo-aléatoire à usage unique en constante évolution. Ces dispositifs rendent le phishing (usurpation d'identité) difficile, mais pas impossible. Ce type d'authentifiant est considéré comme "quelque chose que vous avez". Les jetons multi-facteurs sont similaires aux OTP à facteur unique, mais nécessitent un code PIN valide, un déverrouillage biométrique, une insertion USB ou un appariement NFC ou une valeur supplémentaire (comme les calculatrices de signature de transaction) à saisir pour créer l'OTP final.

#	Description	N1	N2	N3	CWE	NIST
2.8.1	Vérifier que les OTP basées sur le temps ont une durée de vie définie avant d'expirer. *Verify that time-based OTPs have a defined lifetime before expiring.*	✓	✓	✓	613	5.1.4.2 / 5.1.5.2
2.8.2	Vérifier que les clés symétriques utilisées pour vérifier les OTP soumises sont hautement protégées, par exemple en utilisant un module de sécurité matériel ou un stockage de clés basé sur un système d'exploitation sécurisé. *Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage.*		✓	✓	320	5.1.4.2 / 5.1.5.2
2.8.3	Vérifier que des algorithmes cryptographiques approuvés sont utilisés dans la génération, dans la préparation et dans la vérification des OTP. *Verify that approved cryptographic algorithms are used in the generation, seeding, and verification of OTPs.*		✓	✓	326	5.1.4.2 / 5.1.5.2
2.8.4	Vérifier que l'OTP basé sur le temps ne peut être utilisé qu'une seule fois pendant la période de validité. *Verify that time-based OTP can be used only once within the validity period.*		✓	✓	287	5.1.4.2 / 5.1.5.2
2.8.5	Vérifier que si un jeton OTP multi-facteur basé sur le temps est réutilisé pendant la période de validité, il est enregistré et rejeté avec des notifications sécurisées envoyées au détenteur du dispositif. *Verify that if a time-based multi-factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device.*		✓	✓	287	5.1.5.2
2.8.6	Vérifier que le générateur OTP physique à facteur unique peut être révoqué en cas de vol ou autre perte. S'assurer que la révocation est immédiatement effective pour toutes les sessions connectées, quel que soit le lieu. *Verify physical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location.*		✓	✓	613	5.2.1

2.8.7	Vérifier que les authentificateurs biométriques sont limités à une utilisation en tant que facteurs secondaires en conjonction avec quelque chose que vous avez et quelque chose que vous connaissez. *Verify that biometric authenticators are limited to use only as secondary factors in conjunction with either something you have and something you know.*		o[13]	✓	308	5.2.3

2.9 Vérificateurs cryptographiques

Les clés de sécurité cryptographiques sont des cartes à puce ou des clés FIDO, où l'utilisateur doit brancher ou apparier le dispositif cryptographique à l'ordinateur pour compléter l'authentification. Les vérificateurs envoient un défi aux dispositifs ou logiciels cryptographiques, et le dispositif ou le logiciel calcule une réponse basée sur une clé cryptographique stockée en toute sécurité.

Les exigences relatives aux dispositifs et logiciels cryptographiques à facteur unique et aux dispositifs et logiciels cryptographiques à facteurs multiples sont les mêmes, car la vérification de l'authentificateur cryptographique prouve la possession du facteur d'authentification.

#	Description	N1	N2	N3	CWE	NIST
2.9.1	Vérifier que les clés cryptographiques utilisées dans la vérification sont stockées de manière sûre et protégées contre la divulgation, par exemple en utilisant un TPM ou un HSM, ou un service OS qui peut utiliser ce stockage sécurisé. *Verify that cryptographic keys used in verification are stored securely and protected against disclosure, such as using a Trusted Platform Module (TPM) or Hardware Security Module (HSM), or an OS service that can use this secure storage.*		✓	✓	320	5.1.7.2
2.9.2	Vérifier que le (nonce) de défi est d'une longueur d'au moins 64 bits, et qu'il est statistiquement unique ou unique pendant la durée de vie du dispositif cryptographique. *Verify that the challenge nonce is at least 64 bits in length, and statistically unique or unique over the lifetime of the cryptographic device.*		✓	✓	330	5.1.7.2
2.9.3	Vérifier que des algorithmes cryptographiques approuvés sont utilisés dans la génération, la préparation et la vérification. *Verify that approved cryptographic algorithms are used in the generation, seeding, and verification.*		✓	✓	327	5.1.7.2

2.10 Authentification des services

Cette section n'est pas testable par tests d'intrusions, et n'a donc aucune exigence de N1. Toutefois, si elle est utilisée dans une architecture, un codage ou une révision de code sécurisé, veuillez supposer que le logiciel (tout comme le Java Key Store) est l'exigence minimale de la N1. Le stockage de secrets en texte clair n'est en aucun cas acceptable.

#	Description	N1	N2	N3	CWE	NIST
2.10.1	Vérifier que les secrets d'intégration (intra-service) ne reposent pas sur des informations d'identification permanentes telles que des mots de passe, des clés API ou des comptes privilégiés partagés. *Verify that intra-service secrets do not rely on unchanging credentials such as passwords, API keys or shared accounts with privileged access.*		OS assisté	HSM	287	5.1.1.1

[13] O : Recommandée mais pas exigée.

2.10.2	Vérifier que si des mots de passe sont requis pour l'authentification de service, le compte de service utilisé n'est pas un compte par défaut. (par exemple, root/root ou admin/admin sont par défaut dans certains services lors de l'installation). *Verify that if passwords are required for service authentication, the service account used is not a default credential. (e.g. root/root or admin/admin are default in some services during installation).*		OS assisté	HSM	255	5.1.1.1
2.10.3	Vérifier que les mots de passe sont stockés avec une protection suffisante pour empêcher les attaques de récupération hors ligne, y compris l'accès au système local. *Verify that passwords are stored with sufficient protection to prevent offline recovery attacks, including local system access.*		Assistance du système d'exploitation	HSM	522	5.1.1.1
2.10.4	Vérifier que les mots de passe, les intégrations avec les bases de données et les systèmes tiers, les seeds et les secrets internes, ainsi que les clés API sont gérés de manière sécurisée et ne sont pas inclus dans le code source ou stockés dans des dépôts de code source. Ce type de stockage DEVRAIT résister aux attaques hors ligne. L'utilisation d'un stockage de clés logiciel sécurisé (N1), d'un module de plate-forme de confiance (TPM) ou d'un module de sécurité matériel (N3) est recommandée pour le stockage des mots de passe. *Verify passwords, integrations with databases and third-party systems, seeds and internal secrets, and API keys are managed securely and not included in the source code or stored within source code repositories. Such storage SHOULD resist offline attacks. The use of a secure software key store (L1), hardware TPM, or an HSM (L3) is recommended for password storage.*		OS assisté	HSM	798	

V3 Gestion des sessions

L'une des composantes essentielles de toute application web ou API est le mécanisme par lequel elle contrôle et maintient l'état pour un utilisateur ou un dispositif qui interagit avec elle. La gestion de session transforme un protocole sans état en protocole avec état, ce qui est essentiel pour différencier les différents utilisateurs ou appareils.

Il faut s'assurer qu'une application vérifiée satisfait les exigences de gestion de session de haut niveau suivantes :

- Les sessions sont uniques à chaque individu et ne peuvent être devinées ou partagées ;
- Les sessions sont invalidées lorsqu'elles ne sont plus nécessaires et sont interrompues pendant les périodes d'inactivité.

3.1 Sécurité fondamentale en matière de gestion des sessions

#	Description	N1	N2	N3	CWE	NIST
3.1.1	Vérifier que l'application ne révèle jamais les jetons de session dans les paramètres d'URL. *Verify the application never reveals session tokens in URL parameters.*	✓	✓	✓	598	

3.2 Session contraignante

#	Description	N1	N2	N3	CWE	NIST
3.2.1	Vérifier que l'application génère un nouveau jeton de session sur l'authentification de l'utilisateur. *Verify the application generates a new session token on user authentication.*	✓	✓	✓	384	7.1
3.2.2	Vérifier que les jetons de session possèdent au moins 128 bits d'entropie. *Verify that session tokens possess at least 64 bits of entropy.*	✓	✓	✓	331	7.1
3.2.3	Vérifier que l'application ne stocke que des jetons de session dans le navigateur en utilisant des méthodes sûres telles que les cookies correctement sécurisés (voir section 3.4) ou le stockage de session HTML 5. *Verify the application only stores session tokens in the browser using secure methods such as appropriately secured cookies (see section 3.4) or HTML 5 session storage.*	✓	✓	✓	539	7.1
3.2.4	Vérifier que les jetons de session sont générés à l'aide d'algorithmes cryptographiques approuvés. *Verify that session tokens are generated using approved cryptographic algorithms.*		✓	✓	331	7.1

Le TLS ou un autre canal de transport sécurisé est obligatoire pour la gestion des sessions. Cette question est traitée dans le chapitre sur la sécurité des communications.

3.3 Fin de session

Les durées de session ont été alignées sur la norme NIST 800-63, qui autorise des durées de session beaucoup plus longues que celles traditionnellement autorisées par les normes de sécurité. Les organisations doivent examiner le tableau ci-dessous, et si un délai plus long est souhaitable, en fonction du risque de l'application, la valeur NIST doit être la limite supérieure des délais d'inactivité de la session.

Dans ce contexte, la valeur N1 est IAN1/AAN1, la valeur N2 est IAN2/AAN3, la valeur N3 est IAN3/AAN3. Pour IAN2/AAN2 et IAN3/AAN3, le délai d'inactivité le plus court est la limite inférieure des délais d'inactivité pour être déconnecté ou réauthentifié pour reprendre la session.

#	Description	N1	N2	N3	CWE	NIST
3.3.1	Vérifier que la déconnexion et l'expiration invalident le jeton de session. Vérifier que la déconnexion et l'expiration invalident le jeton de session, de sorte que le bouton Précédent ne reprenne pas une session authentifiée, y compris entre les parties de confiance. *Verify that logout and expiration invalidate the session token, such that the back button or a downstream relying party does not resume an authenticated session, including across relying parties.*	✓	✓	✓	613	7.1

3.3.2	Si les authentificateurs permettent aux utilisateurs de rester connectés, vérifier que la réauthentification a lieu périodiquement, que ce soit en cas d'utilisation active ou après une période d'inactivité. *If authenticators permit users to remain logged in, verify that re-authentication occurs periodically both when actively used or after an idle period.*	30 jours	12 heures ou 30 minutes d'inactivité	12 heures ou 15 minutes d'inactivité	613	7.2
3.3.3	Vérifier que l'application offre la possibilité de mettre fin à toutes les autres sessions actives après une modification réussie du mot de passe (y compris la modification via la réinitialisation/récupération du mot de passe), et que cette option est effective dans toute l'application, la connexion fédérée (le cas échéant) et toute partie qui se fie à elle. *Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties.*		✓	✓	613	
3.3.4	Vérifier que les utilisateurs sont en mesure de consulter et (après avoir saisi à nouveau leurs identifiants de connexion) de se déconnecter d'une ou de toutes les sessions et de tous les dispositifs actuellement actifs. *Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices.*		✓	✓	613	7.1

3.4 Gestion de sessions basées sur les cookies

#	Description	N1	N2	N3	CWE	NIST
3.4.1	Vérifier que les jetons de sessions basés sur des cookies ont l'attribut "Secure". *Verify that cookie-based session tokens have the 'Secure' attribute set.*	✓	✓	✓	614	7.1.1
3.4.2	Vérifier que les jetons de sessions basés sur des cookies ont l'attribut "HttpOnly". *Verify that cookie-based session tokens have the 'HttpOnly' attribute set.*	✓	✓	✓	1004	7.1.1
3.4.3	Vérifier que les jetons de sessions basés sur des cookies utilisent l'attribut "SameSite" pour limiter l'exposition aux attaques de contrefaçon par requête intersite. *Verify that cookie-based session tokens utilize the 'SameSite' attribute to limit exposure to cross-site request forgery attacks.*	✓	✓	✓	16	7.1.1
3.4.4	Vérifier que les jetons de session basés sur les cookies utilisent le préfixe "__Host" (voir références) pour assurer la confidentialité des cookies de session. *Verify that cookie-based session tokens use the "__Host-" prefix so cookies are only sent to the host that initially set the cookie.*	✓	✓	✓	16	7.1.1
3.4.5	Vérifier que si l'application est publiée sous un nom de domaine avec d'autres applications qui définissent ou utilisent des cookies de session susceptibles de les remplacer ou de les divulguer, définissez l'attribut de chemin dans les jetons de session basés sur les cookies en utilisant le chemin le plus précis possible. *Verify that if the application is published under a domain name with other applications that set or use session cookies that might disclose the session cookies, set the path attribute in cookie-based session tokens using the most precise path possible.*	✓	✓	✓	16	7.1.1

3.5 Gestion de sessions à jetons

La gestion des sessions basées sur des jetons comprend les clés JWT, OAuth, SAML et API. Parmi celles-ci, les clés API sont connues pour être faibles et ne doivent pas être utilisées dans un nouveau code.

#	Description	N1	N2	N3	CWE	NIST
3.5.1	Vérifier que l'application permet aux utilisateurs de révoquer les jetons OAuth qui forment des relations d'approbation avec les applications liées. *Verify the application allows users to revoke OAuth tokens that form trust relationships with linked applications.*		✓	✓	290	7.1.2
3.5.2	Vérifier que l'application utilise des jetons de sessions plutôt que des secrets et des clés d'API statiques, sauf dans le cas d'anciennes implémentations (legacy). *Verify the application uses session tokens rather than static API secrets and keys, except with legacy implementations.*		✓	✓	798	
3.5.3	Vérifier que les jetons de sessions sans état utilisent les signatures numériques, le cryptage et d'autres contre-mesures pour se protéger contre les attaques par altération, mise sous enveloppe, rediffusion, chiffrement nul et substitution de clés. *Verify that stateless session tokens use digital signatures, encryption, and other countermeasures to protect against tampering, enveloping, replay, null cipher, and key substitution attacks.*		✓	✓	345	

3.6 Réauthentification fédérée

Cette section concerne les personnes qui écrivent le code de la partie de relais (RP) ou du fournisseur de services d'accréditation (CSP). Si vous comptez sur un code mettant en œuvre ces caractéristiques, assurez-vous que ces questions sont traitées correctement.

#	Description	N1	N2	N3	CWE	NIST
3.6.1	Vérifier que les parties qui se fient à la procédure précisent le délai maximal d'authentification aux fournisseurs de services d'authentification (CSP) et que ces derniers réauthentifient l'abonné s'ils n'ont pas utilisé de session pendant cette période. *Verify that Relying Parties (RPs) specify the maximum authentication time to Credential Service Providers (CSPs) and that CSPs re-authenticate the user if they haven't used a session within that period.*			✓	613	7.2.1
3.6.2	Vérifier que les fournisseurs de services d'accréditation (CSP) informent les parties ayant fait confiance au dernier événement d'authentification, afin de permettre aux RP de déterminer s'ils doivent réauthentifier l'utilisateur. *Verify that Credential Service Providers (CSPs) inform Relying Parties (RPs) of the last authentication event, to allow RPs to determine if they need to re-authenticate the user.*			✓	613	7.2.1

3.7 Défenses contre l'exploitation de la gestion des sessions

Il existe un petit nombre d'attaques de gestion de session, dont certaines sont liées à l'expérience utilisateur des sessions. Auparavant, sur la base des exigences de la norme ISO 27002, on exigeait le blocage de plusieurs sessions simultanées. Le blocage de sessions simultanées n'est plus approprié, non seulement parce que les utilisateurs aujourd'hui disposent de nombreux appareils ou que l'application est une API sans session de navigateur, mais parce que dans la plupart de ces implémentations, le dernier authentificateur réussis appartient souvent à l'attaquant. Cette section fournit des conseils de premier plan sur la dissuasion, le retard et la détection des attaques de gestion de session à l'aide de code.

#	Description	N1	N2	N3	CWE	NIST
3.7.1	Vérifier que l'application garantit une session de connexion complète et valide ou exige une nouvelle authentification ou une vérification secondaire avant d'autoriser toute transaction sensible ou modification de compte. *Verify the application ensures a full, valid login session or requires re-authentication or secondary verification before allowing any sensitive transactions or account modifications.*	✓	✓	✓	306	

V4 Contrôle d'accès

L'autorisation est le concept qui consiste à ne permettre l'accès aux ressources qu'à ceux qui sont autorisés à les utiliser. Il faut s'assurer qu'une application vérifiée satisfait aux exigences de haut niveau suivantes :

- Les personnes qui accèdent aux ressources possèdent une autorisation valide pour le faire ;
- Les utilisateurs sont associés à un ensemble bien défini de rôles et de privilèges ;
- Les métadonnées relatives aux rôles et aux autorisations sont protégées contre toute rediffusion ou altération.

4.1 Conception générale du contrôle d'accès

#	Description	N1	N2	N3	CWE
4.1.1	Vérifier que l'application applique les règles de contrôle d'accès sur une couche de service de confiance, en particulier si le contrôle d'accès côté client est présent et pourrait être contourné. *Verify that the application enforces access control rules on a trusted service layer, especially if client-side access control is present and could be bypassed.*	✓	✓	✓	602
4.1.2	Vérifier que tous les attributs des utilisateurs et des données et les informations sur les politiques utilisées par les contrôles d'accès ne peuvent être manipulés par les utilisateurs finaux, sauf autorisation spécifique. *Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.*	✓	✓	✓	639
4.1.3	Vérifier que le principe du moindre privilège existe. Les utilisateurs ne doivent pouvoir accéder qu'aux fonctions, fichiers de données, URL, contrôleurs, services et autres ressources pour lesquels ils possèdent une autorisation spécifique. Cela implique une protection contre l'usurpation et l'élévation des privilèges. *Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege.*	✓	✓	✓	285
4.1.5	Vérifier que les contrôles d'accès échouent de manière sûre, y compris lorsqu'une exception se produit. *Verify that access controls fail securely including when an exception occurs.*	✓	✓	✓	285

4.2 Contrôle d'accès au niveau des opérations

#	Description	N1	N2	N3	CWE
4.2.1	Vérifier que les données sensibles et les API sont protégées contre les attaques par référence directe à un objet (IDOR) non sécurisées visant la création, la lecture, la mise à jour et la suppression d'enregistrements, telles que la création ou la mise à jour de l'enregistrement de quelqu'un d'autre, la consultation de tous les enregistrements ou la suppression de tous les enregistrements. *Verify that sensitive data and APIs are protected against Insecure Direct Object Reference (IDOR) attacks targeting creation, reading, updating and deletion of records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records.*	✓	✓	✓	639
4.2.2	Vérifier que l'application ou le cadriciel applique un mécanisme anti-CSRF fort pour protéger les fonctionnalités authentifiées, et qu'une anti-automation ou un anti-CSRF efficace protège les fonctionnalités non authentifiées. *Verify that the application or framework enforces a strong anti-CSRF mechanism to protect authenticated functionality, and effective anti-automation or anti-CSRF protects unauthenticated functionality.*	✓	✓	✓	352

4.3 Autres considérations relatives au contrôle d'accès

#	Description	N1	N2	N3	CWE
4.3.1	Vérifier que les interfaces administratives utilisent une authentification multifactorielle appropriée pour empêcher toute utilisation non autorisée. *Verify administrative interfaces use appropriate multi-factor authentication to prevent unauthorized use.*	✓	✓	✓	419
4.3.2	Vérifier que la navigation dans les répertoires est désactivée, sauf si vous le souhaitez délibérément. En outre, les applications ne doivent pas permettre la découverte ou la divulgation de métadonnées de fichiers ou de répertoires, tels que les dossiers Thumbs.db, .DS_Store, .git ou .svn. *Verify that directory browsing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of file or directory metadata, such as Thumbs.db, .DS_Store, .git or .svn folders.*	✓	✓	✓	548
4.3.3	Vérifier que la demande dispose d'une autorisation supplémentaire (telle qu'une authentification renforcée ou adaptative) pour les systèmes à faible valeur, et/ou d'une séparation des tâches pour les demandes à valeur élevée afin de faire appliquer les contrôles anti-fraude en fonction du risque de fraude de la demande et de la fraude passée. *Verify that directory browsing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of file or directory metadata, such as Thumbs.db, .DS_Store, .git or .svn folders.*		✓	✓	732

V5 Validation, assainissement et de vérification de l'encodage

La faiblesse la plus courante en matière de sécurité des applications web est l'incapacité à valider correctement les données provenant du client ou de l'environnement avant de les utiliser directement sans aucun encodage de sortie. Cette faiblesse est à l'origine de presque toutes les vulnérabilités importantes des applications web, telles que le Cross-Site Scripting (XSS), l'injection SQL, l'injection d'interpréteur, les attaques locales/Unicode, les attaques de système de fichiers et les débordements de mémoire tampon.

Assurez-vous qu'une application vérifiée satisfait aux exigences de haut niveau suivantes :

- La validation des entrées et l'architecture de codage des sorties ont un pipeline convenu pour prévenir les attaques par injection ;
- Les données d'entrée sont fortement typées, validées, vérifiées en plage ou en longueur ou, au pire, aseptisées ou filtrées ;
- Les données de sortie sont codées selon le contexte des données, aussi près que possible de l'interpréteur.

5.1 Validation des entrées

Des contrôles de validation des entrées correctement mis en œuvre, utilisant une liste d'autorisation positive et un fort typage des données, peuvent éliminer plus de 90 % de toutes les attaques par injection. Les contrôles de longueur et de portée peuvent encore réduire ce phénomène. Il est nécessaire de mettre en place une validation d'entrée sécurisée pendant l'architecture de l'application, les sprints de conception, le codage et les tests unitaires et d'intégration. Bien que nombre de ces éléments ne puissent être trouvés dans les tests de pénétration, les résultats de leur non-implantation se trouvent généralement dans la section 5.3 - Exigences en matière de codage de sortie et de prévention des injections. Il est recommandé aux développeurs et aux réviseurs de codes sécurisés de traiter cette section comme si la N1 était requise pour tous les éléments afin de prévenir les injections.

#	Description	N1	N2	N3	CWE
5.1.1	Vérifier que l'application dispose de défenses contre les attaques de pollution des paramètres HTTP, en particulier si le cadre de l'application ne fait aucune distinction quant à la source des paramètres de la requête (GET, POST, cookies, en-têtes ou variables d'environnement). *Verify that the application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, or environment variables).*	✓	✓	✓	235
5.1.2	Vérifier que les cadriciels protègent contre les attaques par assignation massive de paramètres, ou que l'application dispose de contre-mesures pour protéger contre l'assignation dangereuse de paramètres, comme le marquage des champs privés ou similaires. *Verify that frameworks protect against mass parameter assignment attacks, or that the application has countermeasures to protect against unsafe parameter assignment, such as marking fields private or similar.*	✓	✓	✓	915
5.1.3	Vérifier que toutes les entrées (champs de formulaire HTML, demandes REST, paramètres URL, en-têtes HTTP, cookies, fichiers batch, flux RSS, etc) sont validées par une validation positive (liste d'autorisation). *Verify that all input (HTML form fields, REST requests, URL parameters, HTTP headers, cookies, batch files, RSS feeds, etc) is validated using positive validation (allow lists).*	✓	✓	✓	20
5.1.4	Vérifier que les données structurées sont fortement typées et validées par rapport à un schéma défini comprenant les caractères, la longueur et le modèle autorisés (par exemple, numéros de carte de crédit ou de téléphone, ou valider que deux champs connexes sont raisonnables, comme vérifier la correspondance entre la banlieue et le code postal). *Verify that structured data is strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers, e-mail addresses, telephone numbers, or validating that two related fields are reasonable, such as checking that suburb and zip/postcode match).*	✓	✓	✓	20

5.1.5	Vérifier que les redirections et les transferts d'URL n'autorisent que les destinations prévues (visibles sur une liste d'autorisation), ou affichez un avertissement lors d'une redirection vers un contenu potentiellement non fiable. *Verify that URL redirects and forwards only allow destinations which appear on an allow list, or show a warning when redirecting to potentially untrusted content.*	✓	✓	✓	601

5.2 Exigences en matière d'assainissement et de « bac à sable »

#	Description	N1	N2	N3	CWE
5.2.1	Vérifier que toutes les entrées HTML non fiables provenant d'éditeurs WYSIWYG ou similaires sont correctement assainit avec une bibliothèque ou une fonction de Framework de nettoyage HTML. *Verify that all untrusted HTML input from WYSIWYG editors or similar is properly sanitized with an HTML sanitizer library or framework feature.*	✓	✓	✓	116
5.2.2	Vérifier que les données non structurées sont assainit afin d'appliquer les mesures de sécurité telles que les caractères et la longueur autorisés. *Verify that unstructured data is sanitized to enforce safety measures such as allowed characters and length.*	✓	✓	✓	138
5.2.3	Vérifier que l'application assainit les entrées de l'utilisateur avant de passer aux systèmes de messagerie pour protéger contre l'injection SMTP ou IMAP. *Verify that the application sanitizes user input before passing to mail systems to protect against SMTP or IMAP injection.*	✓	✓	✓	147
5.2.4	Vérifier que l'application évite l'utilisation de eval() ou d'autres fonctions d'exécution de code dynamique. Lorsqu'il n'y a pas d'alternative, toute entrée utilisateur incluse doit être assainit ou mise en sandbox avant d'être exécutée. *Verify that the application avoids the use of eval() or other dynamic code execution features. Where there is no alternative, any user input being included must be sanitized or sandboxed before being executed.*	✓	✓	✓	95
5.2.5	Vérifier que l'application protège contre les attaques par injection de modèles en veillant à ce que toute entrée de l'utilisateur incluse soit aseptisée ou mise en bac à sable. *Verify that the application protects against template injection attacks by ensuring that any user input being included is sanitized or sandboxed.*	✓	✓	✓	94
5.2.6	Vérifier que l'application protège contre les attaques SSRF, en validant ou en assainissant les données non fiables ou les métadonnées de fichiers HTTP, comme les noms de fichiers et les champs de saisie d'URL, utiliser la liste d'autorisation des protocoles, domaines, chemins et ports. *Verify that the application protects against SSRF attacks, by validating or sanitizing untrusted data or HTTP file metadata, such as filenames and URL input fields, and uses allow lists of protocols, domains, paths and ports.*	✓	✓	✓	918
5.2.7	Vérifier que l'application assainit, désactive ou met en place une isolation (sandbox) pour le contenu scriptable fourni par l'utilisateur (Scalable Vector Graphics - SVG), en particulier en ce qui concerne les XSS résultant de scripts natif au code présent et foreignObject. *Verify that the application sanitizes, disables, or sandboxes user-supplied Scalable Vector Graphics (SVG) scriptable content, especially as they relate to XSS resulting from inline scripts, and foreignObject.*	✓	✓	✓	159
5.2.8	Vérifier que l'application assainit, désactive ou met en sandbox le contenu des scripts ou des modèles d'expression fournis par l'utilisateur, tels que les feuilles de style Markdown, CSS ou XSL, le BBCode ou autres. *Verify that the application sanitizes, disables, or sandboxes user-supplied scriptable or expression template language content, such as Markdown, CSS or XSL stylesheets, BBCode, or similar.*	✓	✓	✓	94

5.3 Encodage de sortie/output et prévention des injections

L'encodage de la sortie est essentiel pour la sécurité de toute application. Généralement, l'encodage de sortie n'est pas persistant, mais utilisé pour rendre la sortie sûre dans le contexte de sortie approprié pour une utilisation immédiate. Le défaut d'encodage de la sortie entraînera une application non sécurisée, injectable et dangereuse.

#	Description	N1	N2	N3	CWE
5.3.1	Vérifier que l'encodage de sortie est pertinent pour l'interprète et le contexte requis. Par exemple, utilisez des encodeurs spécifiques pour les valeurs HTML, les attributs HTML, JavaScript, les paramètres URL, les en-têtes HTTP, SMTP et autres selon le contexte, en particulier à partir d'entrées non fiables (par exemple les noms avec Unicode ou apostrophes, comme ねこ ou O'Hara). *Verify that output encoding is relevant for the interpreter and context required. For example, use encoders specifically for HTML values, HTML attributes, JavaScript, URL parameters, HTTP headers, SMTP, and others as the context requires, especially from untrusted inputs (e.g. names with Unicode or apostrophes, such as ねこ or O'Hara).*	✓	✓	✓	116
5.3.2	Vérifier que l'encodage de sortie préserve le jeu de caractères et la langue choisis par l'utilisateur, de sorte que tout point de caractère Unicode soit valide et traité en toute sécurité. *Verify that output encoding preserves the user's chosen character set and locale, such that any Unicode character point is valid and safely handled.*	✓	✓	✓	176
5.3.3	Vérifier que l'encodage des sorties en fonction du contexte, de préférence automatisé - ou au pire, manuel - protège contre le XSS réfléchi, stocké et basé sur le DOM. *Verify that context-aware, preferably automated - or at worst, manual - output escaping protects against reflected, stored, and DOM based XSS.*	✓	✓	✓	79
5.3.4	Vérifier que la sélection de données ou les requêtes de base de données (par exemple SQL, HQL, ORM, NoSQL) utilisent des requêtes paramétrées, des ORM, des cadres d'entités, ou sont autrement protégées contre les attaques par injection de base de données. *Verify that data selection or database queries (e.g. SQL, HQL, ORM, NoSQL) use parameterized queries, ORMs, entity frameworks, or are otherwise protected from database injection attacks.*	✓	✓	✓	89
5.3.5	Vérifier que, lorsque des mécanismes paramétrés ou plus sûrs ne sont pas présents, un encodage de sortie spécifique au contexte est utilisé pour se protéger contre les attaques par injection, comme l'utilisation de l'échappement SQL pour se protéger contre l'injection SQL. *Verify that where parameterized or safer mechanisms are not present, context-specific output encoding is used to protect against injection attacks, such as the use of SQL escaping to protect against SQL injection.*	✓	✓	✓	89
5.3.6	Vérifier que l'application protège contre les attaques par injection JSON, les attaques Eval JSON et l'évaluation des expressions JavaScript. *Verify that the application protects against JSON injection attacks, JSON eval attacks, and JavaScript expression evaluation.*	✓	✓	✓	830
5.3.7	Vérifier que l'application protège contre les vulnérabilités de LDAP Injection, ou que des contrôles de sécurité spécifiques pour empêcher des injections LDAP ont été mis en place. *Verify that the application protects against LDAP injection vulnerabilities, or that specific security controls to prevent LDAP injection have been implemented.*	✓	✓	✓	90

5.3.8	Vérifier que l'application protège contre l'injection de commandes du système d'exploitation et que les appels du système d'exploitation utilisent des requêtes paramétrées du système d'exploitation ou utilisent l'encodage contextuel de la sortie de la ligne de commande. *Verify that the application protects against OS command injection and that operating system calls use parameterized OS queries or use contextual command line output encoding.*	✓	✓	✓	78
5.3.9	Vérifier que l'application protège contre les attaques par inclusion de fichier local (LFI) ou par inclusion de fichier distant (RFI). *Verify that the application protects against Local File Inclusion (LFI) or Remote File Inclusion (RFI) attacks.*	✓	✓	✓	829
5.3.10	Vérifier que l'application protège contre les attaques par injection XPath ou par injection XML. *Verify that the application protects against XPath injection or XML injection attacks.*	✓	✓	✓	643

Note : L'utilisation de requêtes paramétrées ou l'échappement du SQL n'est pas toujours suffisant ; les noms de tables et de colonnes, ORDER BY, etc. ne peuvent pas être évités. L'inclusion de données déjouées fournies par l'utilisateur dans ces champs entraîne l'échec des requêtes ou de l'injection SQL.

Note : Le format SVG autorise explicitement le script ECMA dans presque tous les contextes, de sorte qu'il peut ne pas être possible de bloquer complètement tous les vecteurs XSS SVG. Si un téléchargement SVG est nécessaire, nous recommandons fortement de traiter ces fichiers téléchargés en tant que texte/plain, ou d'utiliser un domaine de contenu distinct fourni par l'utilisateur pour empêcher que les attaques XSS prenne le relais de l'application.

5.4 Mémoire, chaînes de caractères et de code non géré

Les exigences suivantes ne s'appliquent que lorsque l'application utilise un langage système ou un code non géré.

#	Description	N1	N2	N3	CWE
5.4.1	Vérifier que l'application utilise une chaîne de caractères à mémoire sécurisée, une copie mémoire sécurisée et l'arithmétique des pointeurs pour détecter ou empêcher les débordements de pile, de mémoire tampon ou de tas. *Verify that the application uses memory-safe string, safer memory copy and pointer arithmetic to detect or prevent stack, buffer, or heap overflows.*		✓	✓	120
5.4.2	Vérifier que les chaînes de format ne prennent pas d'entrée potentiellement hostile, et sont constantes. *Verify that format strings do not take potentially hostile input, and are constant.*		✓	✓	134
5.4.3	Vérifier que les techniques de validation des signes, des plages et des entrées sont utilisées pour éviter les débordements d'entiers. *Verify that sign, range, and input validation techniques are used to prevent integer overflows.*		✓	✓	190

5.5 Prévention de la désérialisation

#	Description	N1	N2	N3	CWE
5.5.1	Vérifier que les objets sérialisés utilisent des contrôles d'intégrité ou sont cryptés pour empêcher la création d'objets hostiles ou la falsification de données. *Verify that serialized objects use integrity checks or are encrypted to prevent hostile object creation or data tampering.*	✓	✓	✓	502
5.5.2	Vérifier que l'application restreint correctement les analyseurs XML pour n'utiliser que la configuration la plus restrictive possible et pour s'assurer que les fonctions dangereuses telles que la résolution d'entités externes sont désactivées pour empêcher les XXE. *Verify that the application correctly restricts XML parsers to only use the most restrictive configuration possible and to ensure that unsafe features such as resolving external entities are disabled to prevent XML external Entity (XXE) attacks.*	✓	✓	✓	611

5.5.3	Vérifier que la désérialisation des données non fiables est évitée ou protégée à la fois dans le code personnalisé et les bibliothèques tierces (comme les analyseurs JSON, XML et YAML). *Verify that deserialization of untrusted data is avoided or is protected in both custom code and third-party libraries (such as JSON, XML and YAML parsers).*	✓	✓	✓	502
5.5.4	Vérifier que lors de l'analyse de JSON dans les navigateurs ou les backends basés sur JavaScript, JSON.parse est utilisé pour analyser le document JSON. N'utilisez pas eval() pour analyser JSON. *Verify that when parsing JSON in browsers or JavaScript-based backends, JSON.parse is used to parse the JSON document. Do not use eval() to parse JSON.*	✓	✓	✓	95

V6 Cryptographie stockée

Il faut s'assurer qu'une application vérifiée satisfait aux exigences de haut niveau suivantes :

- Tous les modules cryptographiques échouent de manière sécurisée et que les erreurs sont traitées correctement ;
- Un générateur de nombres aléatoires approprié est utilisé ;
- L'accès aux clés est géré de manière sécurisée.

6.1 Classification des données

L'actif le plus important est constitué par les données traitées, stockées ou transmises par une application. Il faut toujours procéder à une évaluation de l'impact sur la vie privée afin de classer correctement les besoins en matière de protection des données de toute donnée stockée.

#	Description	N1	N2	N3	CWE
6.1.1	Vérifier que les données privées réglementées sont stockées sous forme cryptée pendant le repos, comme les informations d'identification personnelle (IIP), les informations personnelles sensibles ou les données considérées comme susceptibles d'être soumises à la GDPR de l'UE. *Verify that regulated private data is stored encrypted while at rest, such as Personally Identifiable Information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR.*		✓	✓	311
6.1.2	Vérifier que les données de santé réglementées sont stockées de manière cryptée pendant le repos, comme les dossiers médicaux, les détails des dispositifs médicaux ou les dossiers de recherche désanonymisés. *Verify that regulated health data is stored encrypted while at rest, such as medical records, medical device details, or de-anonymized research records.*		✓	✓	311
6.1.3	Vérifier que les données financières réglementées sont stockées de manière cryptée lorsqu'elles sont au repos, telles que les comptes financiers, les défauts ou les antécédents de crédit, les dossiers fiscaux, l'historique des salaires, les bénéficiaires ou les dossiers de marché ou de recherche désanonymisés. *Verify that regulated financial data is stored encrypted while at rest, such as financial accounts, defaults or credit history, tax records, pay history, beneficiaries, or de-anonymized market or research records.*		✓	✓	311

6.2 Algorithmes

Les récents progrès de la cryptanalyse font que quelques algorithmes ainsi que les longueurs de clé, jugés robustes au paravent ne le sont plus. Il est donc important d'avoir la possibilité de modifier ces algorithmes.

Bien que cette section ne soit pas facilement testée lors des tests d'intrusion, les développeurs devraient considérer toute cette section comme obligatoire même si le N1 est absent de la plupart des éléments.

#	Description	N1	N2	N3	CWE
6.2.1	Vérifier que tous les modules cryptographiques échouent en toute sécurité, et que les erreurs sont traitées de manière à ne pas permettre les attaques de type "Padding Oracle". *Verify that all cryptographic modules fail securely, and errors are handled in a way that does not enable Padding Oracle attacks.*	✓	✓	✓	310
6.2.2	Vérifier que des algorithmes, des bibliothèques cryptographiques et des modes éprouvés par l'industrie ou approuvés par le gouvernement sont utilisés, au lieu de la cryptographie codée sur mesure. *Verify that industry proven or government approved cryptographic algorithms, modes, and libraries are used, instead of custom coded cryptography.*		✓	✓	327
6.2.3	Vérifier que le vecteur d'initialisation du chiffrement, la configuration du chiffrement et les modes de blocage sont configurés de manière sécurisée en utilisant les derniers conseils. *Verify that encryption initialization vector, cipher configuration, and block modes are configured securely using the latest advice.*		✓	✓	326

6.2.4	Vérifier que les algorithmes de chiffrement ou de hachage, les longueurs de clé, le nombre de rondes, les chiffrements ou les modes, peuvent être reconfigurés, mis à niveau ou échangés à tout moment, pour se protéger contre les failles cryptographiques. *Verify that random number, encryption or hashing algorithms, key lengths, rounds, ciphers or modes, can be reconfigured, upgraded, or swapped at any time, to protect against cryptographic breaks.*		✓	✓	326
6.2.5	Vérifier que les modes de blocs non sécurisés connus (c'est-à-dire ECB, etc.), les modes de remplissage (c'est-à-dire PKCS#1 v1.5, etc.), les chiffrements avec des blocs de petites tailles (c'est-à-dire Triple-DES, Blowfish, etc.) et les algorithmes de hachage faibles (c'est-à-dire MD5, SHA1, etc.) ne sont pas utilisés, sauf si cela est nécessaire pour la rétrocompatibilité. *Verify that known insecure block modes (i.e. ECB, etc.), padding modes (i.e. PKCS#1 v1.5, etc.), ciphers with small block sizes (i.e. Triple-DES, Blowfish, etc.), and weak hashing algorithms (i.e. MD5, SHA1, etc.) are not used unless required for backwards compatibility.*		✓	✓	326
6.2.6	Vérifier que les nonces, vecteurs d'initialisation et autres numéros à usage unique ne doivent pas être utilisés plus d'une fois avec une clé de cryptage donnée. La méthode de génération doit être appropriée à l'algorithme utilisé. *Verify that nonces, initialization vectors, and other single use numbers must not be used more than once with a given encryption key. The method of generation must be appropriate for the algorithm being used.*		✓	✓	326
6.2.7	Vérifier que les données cryptées sont authentifiées par des signatures, des modes de chiffrement authentifiés pour s'assurer que le texte chiffré n'est pas altéré par une partie non autorisée. *Verify that encrypted data is authenticated via signatures, authenticated cipher modes, or HMAC to ensure that ciphertext is not altered by an unauthorized party.*			✓	326
6.2.8	Vérifier que toutes les opérations cryptographiques sont à temps constant, sans opérations de "court-circuit" dans les comparaisons, les calculs ou les retours, afin d'éviter les fuites d'informations. *Verify that all cryptographic operations are constant-time, with no 'short-circuit' operations in comparisons, calculations, or returns, to avoid leaking information.*			✓	385

6.3 Valeurs aléatoires

La véritable génération de nombres pseudo-aléatoires (PRNG) est très difficile à réaliser. En général, les bonnes sources d'entropie au sein d'un système sont rapidement épuisées si les générateurs sont trop longtemps utilisés. Les sources moins aléatoires peuvent conduire à des clés et des secrets prévisibles.

#	Description	N1	N2	N3	CWE
6.3.1	Vérifier que tous les nombres aléatoires, noms de fichiers aléatoires, GUIDs aléatoires et chaînes aléatoires sont générés en utilisant le générateur de nombres aléatoires sécurisé cryptographiquement approuvé par le module cryptographique lorsque ces valeurs aléatoires sont destinées à ne pas être devinées par un attaquant. *Verify that all random numbers, random file names, random GUIDs, and random strings are generated using the cryptographic module's approved cryptographically secure random number generator when these random values are intended to be not guessable by an attacker.*		✓	✓	338
6.3.2	Vérifier que les GUID aléatoires sont créés en utilisant l'algorithme GUID v4, et un générateur de nombres pseudo-aléatoires sécurisé cryptographiquement (CSPRNG). Les GUID créés à l'aide d'autres générateurs de nombres pseudo-aléatoires peuvent être prévisibles. *Verify that random GUIDs are created using the GUID v4 algorithm, and a Cryptographically-secure Pseudo-random Number Generator (CSPRNG). GUIDs created using other pseudo-random number generators may be predictable.*		✓	✓	338

6.3.3	Vérifier que les nombres aléatoires sont créés avec une entropie correcte même lorsque l'application est soumise à une forte charge, ou que l'application se dégrade gracieusement dans de telles circonstances. *Verify that random numbers are created with proper entropy even when the application is under heavy load, or that the application degrades gracefully in such circumstances.*			✓	338

6.4 Gestion du secret

Bien que cette section ne soit pas facilement testée, les développeurs devraient considérer toute cette section comme obligatoire même si la N1 est absente de la plupart des éléments.

#	Description	N1	N2	N3	CWE
6.4.1	Vérifier qu'une solution de gestion des secrets, telle qu'un coffre-fort de clés, est utilisé pour créer, stocker, contrôler l'accès aux secrets et les détruire en toute sécurité. *Verify that a secrets management solution such as a key vault is used to securely create, store, control access to and destroy secrets.*		✓	✓	798
6.4.2	Vérifier que le matériel clé ne soit pas exposé à l'application mais utilise plutôt un module de sécurité isolé comme un coffre-fort pour les opérations cryptographiques. *Verify that key material is not exposed to the application but instead uses an isolated security module like a vault for cryptographic operations.*		✓	✓	320

V7 Traitement des erreurs et exigences de vérification de l'enregistrement

L'objectif premier du traitement et de la journalisation des erreurs est de fournir des informations utiles à l'utilisateur, aux administrateurs et aux équipes de réponse aux incidents. L'objectif n'est pas de créer des quantités importantes de journaux, mais des journaux de haute qualité, avec plus de signal que de bruit rejeté.

Les journaux de haute qualité contiennent souvent des données sensibles et doivent être protégés conformément aux directives en matière de confidentialité des données. Cela devrait inclure :

- Ne pas collecter ou enregistrer des informations sensibles, sauf si cela est spécifiquement requis ;
- Veiller à ce que toutes les informations enregistrées soient traitées de manière sûre et protégées conformément à leur classification ;
- Veiller à ce que les journaux ne soient pas conservés éternellement, mais qu'ils aient une durée de vie absolue aussi courte que possible.

Si les journaux contiennent des données privées ou sensibles, ces journaux deviennent des informations sensibles et donc particulièrement attrayantes pour les attaquants.

Il est également important de s'assurer que l'application échoue en toute sécurité et que les erreurs ne divulguent pas d'informations inutiles.

7.1 Contenu des journaux

L'enregistrement d'informations sensibles est dangereux et les journaux deviennent ainsi eux-mêmes classifiés. Ceci signifie qu'ils doivent être cryptés, faire l'objet de politiques de conservation et être divulgués lors d'audits de sécurité. A cet effet, seules les informations nécessaires devraient être conservées dans les journaux, et certainement pas les paiements, les justificatifs d'identité (y compris les jetons de session), les informations sensibles ou personnelles.

La section 7.1 couvre le Top 10 de l'OWASP 2017:A10. Ces règles ne sont pas vérifiables par des tests d'intrusions, il est important pour :

- Les développeurs de s'assurer de la conformité totale avec cette section, comme si tous les éléments étaient marqués comme N1 ;
- Les pentesters de vérifier la conformité totale avec tous les éléments de la 7.1 par entretien, captures d'écran ou confirmation.

#	Description	N1	N2	N3	CWE
7.1.1	Vérifier que la demande n'enregistre pas les références ou les détails de paiement. Les jetons de session ne doivent être stockés dans les journaux que sous une forme hachée et irréversible. *Verify that the application does not log credentials or payment details. Session tokens should only be stored in logs in an irreversible, hashed form.*	✓	✓	✓	532
7.1.2	Vérifier que l'application n'enregistre pas d'autres données sensibles telles que définies par les lois locales sur la protection de la vie privée ou la politique de sécurité pertinente. *Verify that the application does not log other sensitive data as defined under local privacy laws or relevant security policy.*	✓	✓	✓	532
7.1.3	Vérifier que l'application enregistre les événements pertinents pour la sécurité, y compris les événements d'authentification réussis et échoués, les échecs de contrôle d'accès, les échecs de désérialisation et les échecs de validation des entrées. *Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures.*		✓	✓	778
7.1.4	Vérifier que chaque événement consigné dans le journal contient les informations nécessaires pour permettre une enquête détaillée sur la chronologie de l'événement. *Verify that each log event includes necessary information that would allow for a detailed investigation of the timeline when an event happens.*		✓	✓	778

7.2 Traitement des journaux

Il est essentiel d'enregistrer en temps utile les événements de vérification, le triage et l'escalade. Les journaux de l'application sont clairs et peuvent être facilement surveillés et analysés, soit localement, soit envoyés à un système de surveillance à distance.

La section 7.2 couvre le Top 10 de l'OWASP 2017:A10. Comme 2017:A10 et cette section ne sont pas testables, il est important pour :

- Les développeurs de s'assurer de la conformité totale avec cette section, comme si tous les éléments étaient marqués comme N1 ;
- Les pentesters de vérifier la conformité totale avec tous les éléments de la 7.2 par entretien, captures d'écran ou confirmation.

#	Description	N1	N2	N3	CWE
7.2.1	Vérifier que toutes les décisions d'authentification sont consignées, sans stocker d'identifiants de session ou de mots de passe sensibles. Cela devrait inclure les demandes avec les métadonnées pertinentes nécessaires aux enquêtes de sécurité. *Verify that all authentication decisions are logged, without storing sensitive session tokens or passwords. This should include requests with relevant metadata needed for security investigations.*		✓	✓	778
7.2.2	Vérifier que toutes les décisions de contrôle d'accès peuvent être enregistrées et que toutes les décisions qui ont échoué sont enregistrées. Cela devrait inclure les demandes avec les métadonnées pertinentes nécessaires aux enquêtes de sécurité. *Verify that all access control decisions can be logged and all failed decisions are logged. This should include requests with relevant metadata needed for security investigations.*		✓	✓	285

7.3 Protection des journaux

Les journaux qui peuvent être trivialement modifiés ou supprimés sont inutiles pour les enquêtes et les poursuites. La divulgation des journaux peut révéler des détails internes sur l'application ou les données qu'elle contient. Il convient de prendre des précautions pour protéger les journaux contre toute divulgation, modification ou suppression non autorisée.

#	Description	N1	N2	N3	CWE
7.3.1	Vérifier que tous les composants de journalisation encodent correctement les données pour empêcher l'injection de journal. *Verify that all logging components appropriately encode data to prevent log injection.*		✓	✓	117
7.3.3	Vérifier que les journaux de sécurité sont protégés contre tout accès et toute modification non autorisés. *Verify that security logs are protected from unauthorized access and modification.*		✓	✓	200
7.3.4	Vérifier que les sources de temps sont synchronisées avec l'heure et le fuseau horaire corrects. Envisager sérieusement de n'enregistrer les données qu'en UTC si les systèmes sont globaux pour faciliter l'analyse criminalistique post-incident. *Verify that time sources are synchronized to the correct time and time zone. Strongly consider logging only in UTC if systems are global to assist with post-incident forensic analysis.*		✓	✓	

L'encodage des journaux, (section 7.3.1), est difficile à tester et à examiner à l'aide d'outils dynamiques automatisés et de tests de pénétration, mais les architectes, les développeurs et les réviseurs de code source devraient le considérer comme une exigence de niveau 1.

7.4 Traitement des erreurs

L'objectif du traitement des erreurs est de permettre à l'application de fournir des événements pertinents pour la sécurité en vue de la surveillance, du triage et de l'escalade. Lorsque vous enregistrez des événements liés à la sécurité, assurez-vous que le journal a un but et qu'il peut être distingué par le SIEM ou un logiciel d'analyse.

#	Description	N1	N2	N3	CWE
7.4.1	Vérifier qu'un message générique s'affiche lorsqu'une erreur inattendue ou sensible à la sécurité se produit, éventuellement avec un identifiant unique que le personnel de soutien peut utiliser pour enquêter. *Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate.*	✓	✓	✓	210
7.4.2	Vérifier que le traitement des exceptions est utilisé dans tout le code source pour tenir compte des conditions d'erreur prévues et imprévues. *Verify that exception handling (or a functional equivalent) is used across the codebase to account for expected and unexpected error conditions.*		✓	✓	544
7.4.3	Vérifier qu'un gestionnaire d'erreurs de "dernier recours" est défini, qui prendra en compte toutes les exceptions non traitées. *Verify that a "last resort" error handler is defined which will catch all unhandled exceptions.*		✓	✓	431

Certains langages, tels que Swift et Go et selon la pratique courante de conception de nombreux langages fonctionnels, ne prennent pas en charge les exceptions ou les gestionnaires d'événements de dernier recours. Dans ce cas, les architectes et les développeurs doivent utiliser un modèle, un langage ou un cadre convivial pour s'assurer que les applications peuvent gérer en toute sécurité des événements exceptionnels, inattendus ou liés à la sécurité.

V8 Protection des données

Il y a trois éléments clés pour une bonne protection des données : Confidentialité, intégrité et disponibilité (CIA). Cette norme suppose que la protection des données est appliquée sur un système fiable, tel qu'un serveur, qui a été renforcé et dispose de protections suffisantes.

Les applications doivent supposer que tous les dispositifs des utilisateurs sont compromis d'une manière ou d'une autre. Lorsqu'une application transmet ou stocke des informations sensibles sur des dispositifs non sécurisés, tels que des ordinateurs, des téléphones et des tablettes partagés, l'application est chargée de s'assurer que les données stockées sur ces dispositifs sont cryptées et ne peuvent pas être facilement obtenues, modifiées ou divulguées de manière illicite.

En matière de protection des données, une application vérifiée satisfait aux exigences de haut niveau suivantes :

- Confidentialité : Les données doivent être protégées contre toute consultation ou divulgation non autorisées, tant pendant leur transit que lors de leur stockage ;
- Intégrité : Les données doivent être protégées contre toutes tentatives de création, modification ou suppression malveillantes par des attaquants non autorisés ;
- Disponibilité : Les données doivent être accessibles aux utilisateurs autorisés, selon les besoins.

8.1 Protection générale des données

#	Description	N1	N2	N3	CWE
8.1.1	Vérifier que l'application protège les données sensibles contre la mise en cache dans des composants du serveur tels que les équilibreurs de charge et les caches d'applications. *Verify the application protects sensitive data from being cached in server components such as load balancers and application caches.*		✓	✓	524
8.1.2	Vérifier que toutes les copies en cache ou temporaires de données sensibles stockées sur le serveur sont protégées contre tout accès non autorisé ou purgées/invalidées après que l'utilisateur autorisé a accédé aux données sensibles. *Verify that all cached or temporary copies of sensitive data stored on the server are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data.*		✓	✓	524
8.1.3	Vérifier que l'application minimise le nombre de paramètres dans une requête, tels que les champs cachés, les variables Ajax, les cookies et les valeurs d'en-tête. *Verify the application minimizes the number of parameters in a request, such as hidden fields, Ajax variables, cookies and header values.*		✓	✓	233
8.1.4	Vérifier que l'application peut détecter et alerter sur un nombre anormal de demandes, par exemple par IP, par utilisateur, par total, par heure ou par jour, ou tout ce qui a un sens pour l'application. *Verify the application can detect and alert on abnormal numbers of requests, such as by IP, user, total per hour or day, or whatever makes sense for the application.*		✓	✓	770
8.1.5	Vérifier que des sauvegardes régulières des données importantes sont effectuées et que des tests de restauration des données sont effectués. *Verify that regular backups of important data are performed and that test restoration of data is performed.*			✓	19
8.1.6	Vérifier que les sauvegardes sont stockées en toute sécurité pour éviter que les données ne soient volées ou corrompues. *Verify that backups are stored securely to prevent data from being stolen or corrupted.*			✓	19

8.2 Protection des données côté client

#	Description	N1	N2	N3	CWE
8.2.1	Vérifier que l'application définit suffisamment d'en-têtes anticaching pour que les données sensibles ne soient pas mises en cache dans les navigateurs modernes. *Verify the application sets sufficient anti-caching headers so that sensitive data is not cached in modern browsers.*	✓	✓	✓	525

8.2.2	Vérifier que les données stockées dans le stockage du navigateur (telles que localStorage, sessionStorage, IndexedDB ou les cookies) ne contiennent pas de données sensibles. *Verify that data stored in browser storage (such as localStorage, sessionStorage, IndexedDB, or cookies) does not contain sensitive data.*	✓	✓	✓	922
8.2.3	Vérifier que les données authentifiées sont effacées du stockage du client, tel que le DOM du navigateur, après la fin du client ou de la session. *Verify that authenticated data is cleared from client storage, such as the browser DOM, after the client or session is terminated.*	✓	✓	✓	922

8.3 Données privées sensibles

Cette section permet de protéger les données sensibles contre la création, la lecture, la mise à jour ou la suppression sans autorisation.

Le respect de cette section implique le respect du contrôle d'accès V4, et en particulier la section 4.2. Par exemple, la protection contre les mises à jour ou la divulgation non autorisées d'informations personnelles sensibles nécessite le respect de la section 4.2.1.

Les réglementations et les lois relatives à la protection de la vie privée, ont une incidence directe sur la manière dont les applications doivent aborder la mise en œuvre du stockage, de l'utilisation et de la transmission des informations personnelles sensibles. Cela va des sanctions sévères à de simples conseils.

#	Description	N1	N2	N3	CWE
8.3.1	Vérifier que les données sensibles sont envoyées au serveur dans le corps ou les en-têtes du message HTTP, et que les paramètres de la chaîne de requête de tout verbe HTTP ne contiennent pas de données sensibles. *Verify that sensitive data is sent to the server in the HTTP message body or headers, and that query string parameters from any HTTP verb do not contain sensitive data.*	✓	✓	✓	319
8.3.2	Vérifier que les utilisateurs disposent d'une méthode pour supprimer ou exporter leurs données sur demande. *Verify that users have a method to remove or export their data on demand.*	✓	✓	✓	212
8.3.3	Vérifier que les utilisateurs disposent d'un langage clair concernant la collecte et l'utilisation des informations personnelles fournies et que les utilisateurs ont donné leur consentement pour l'utilisation de ces données avant qu'elles ne soient utilisées de quelque manière que ce soit. *Verify that users are provided clear language regarding collection and use of supplied personal information and that users have provided opt-in consent for the use of that data before it is used in any way.*	✓	✓	✓	285
8.3.4	Vérifier que toutes les données sensibles créées et traitées par l'application ont été identifiées, et s'assurer qu'une politique est en place sur la manière de traiter les données sensibles. *Verify that all sensitive data created and processed by the application has been identified, and ensure that a policy is in place on how to deal with sensitive data.*	✓	✓	✓	200
8.3.5	Vérifier que l'accès aux données sensibles est contrôlé (sans enregistrer les données sensibles elles-mêmes), si les données sont collectées en vertu des directives pertinentes sur la protection des données ou si l'enregistrement de l'accès est nécessaire. *Verify accessing sensitive data is audited (without logging the sensitive data itself), if the data is collected under relevant data protection directives or where logging of access is required.*		✓	✓	532
8.3.6	Vérifier que les informations sensibles contenues dans la mémoire sont écrasées dès qu'elles ne sont plus nécessaires pour atténuer les attaques de vidage de la mémoire, en utilisant des zéros ou des données aléatoires. *Verify that sensitive information contained in memory is overwritten as soon as it is no longer required to mitigate memory dumping attacks, using zeroes or random data.*		✓	✓	226

8.3.7	Vérifier que les informations sensibles ou privées qui doivent être cryptées, le sont à l'aide d'algorithmes approuvés qui assurent à la fois la confidentialité et l'intégrité. *Verify that sensitive or private information that is required to be encrypted, is encrypted using approved algorithms that provide both confidentiality and integrity.*		✓	✓	327
8.3.8	Vérifier que les informations personnelles sensibles font l'objet d'une classification de conservation des données, de sorte que les données anciennes ou périmées soient supprimées automatiquement, selon un calendrier ou selon la situation. *Verify that sensitive personal information is subject to data retention classification, such that old or out of date data is deleted automatically, on a schedule, or as the situation requires.*		✓	✓	285

Lorsqu'on envisage la protection des données, il faut avant tout tenir compte de l'extraction ou de la modification de masse ou de l'utilisation excessive. Les exigences de chaque système sont susceptibles d'être très différentes, de sorte que la décision d'être "anormal" doit tenir compte du modèle de menace et du risque commercial. Les critères importants sont la capacité de détecter, de dissuader ou, de préférence, de bloquer ces actions anormales de masse.

V9 Communications

Toute application vérifiée doit satisfaire les exigences de haut niveau suivantes :

- Le TLS ou un cryptage fort est toujours utilisé, quelle que soit la sensibilité des données transmises ;
- Les conseils de configuration actualisés sont utilisés pour activer ou ordonner les algorithmes et les clés de chiffrement préférés ;
- Les algorithmes et les clés de chiffrement faibles ou bientôt obsolètes ne sont utilisés qu'en dernier recours ;
- Les algorithmes et les clés de chiffrement non sécurisés, ou non recommandés doivent être désactivés.

Les principaux conseils concernant la configuration sécurisée du TLS changent fréquemment, souvent en raison d'une défaillance dans les algorithmes ou les clés de chiffrements existants. Les versions les plus récentes des outils utilisés pour évaluer et configurer le TLS et l'ordre et la sélection d'algorithme préférés doivent être privilégiés. La configuration doit être vérifiée périodiquement pour s'assurer que la bonne configuration des communications sécurisées est toujours activée.

9.1 Sécurité des communications des clients

Les communications avec les clients ne sont pas toujours cryptées. En effet, l'utilisation de TLS 1.2 ou d'une version ultérieure est pratiquement obligatoire pour les navigateurs et les moteurs de recherche modernes. La configuration doit être régulièrement revue à l'aide d'outils en ligne afin de s'assurer que les dernières recommandations sont implémentées.

#	Description	N1	N2	N3	CWE
9.1.1	Vérifier que le TLS est utilisé pour toutes les connexions des clients et ne revient pas à des protocoles non sécurisés ou non chiffrés. *Verify that TLS is used for all client connectivity, and does not fall back to insecure or unencrypted communications.*	✓	✓	✓	319
9.1.2	Vérifier à l'aide d'outils de test (up to date TLS) que seules les suites de chiffrement fortes sont activées, les suites de chiffrement les plus fortes étant définies comme préférées. *Verify using up to date TLS testing tools that only strong cipher suites are enabled, with the strongest cipher suites set as preferred.*	✓	✓	✓	326
9.1.3	Vérifier que seules les dernières versions recommandées du protocole TLS sont activées, telles que TLS 1.2 et TLS 1.3. La dernière version du protocole TLS devrait être l'option préférée. *Verify that only the latest recommended versions of the TLS protocol are enabled, such as TLS 1.2 and TLS 1.3. The latest version of the TLS protocol should be the preferred option.*	✓	✓	✓	326

9.2 Sécurité des communications du serveur

Les communications entre serveurs ne se limitent pas à HTTP. Des connexions sécurisées vers et depuis d'autres systèmes, tels que les systèmes de surveillance, les outils de gestion, l'accès à distance et SSH, les intergiciels, les bases de données, les ordinateurs centraux, les systèmes partenaires ou sources externes doivent être mis en place. Toutes ces connexions doivent être cryptées pour éviter toute interception depuis l'intérieur".

#	Description	N1	N2	N3	CWE
9.2.1	Vérifier que les connexions vers et depuis le serveur utilisent des certificats TLS de confiance. Lorsque des certificats générés en interne ou auto-signés sont utilisés, le serveur doit être configuré pour ne faire confiance qu'à des AC internes spécifiques et à des certificats auto-signés spécifiques. Tous les autres doivent être rejetés. *Verify that connections to and from the server use trusted TLS certificates. Where internally generated or self-signed certificates are used, the server must be configured to only trust specific internal CAs and specific self-signed certificates. All others should be rejected.*		✓	✓	295

9.2.2	Vérifier que les communications cryptées telles que TLS sont utilisées pour toutes les connexions entrantes et sortantes, y compris pour les ports de gestion, la surveillance, l'authentification, les appels d'API ou de service web, les connexions de base de données, de cloud, de serverless, d'ordinateur central, externes et de partenaires. Le serveur ne doit pas se rabattre sur des protocoles non sécurisés ou non chiffrés. *Verify that encrypted communications such as TLS is used for all inbound and outbound connections, including for management ports, monitoring, authentication, API, or web service calls, database, cloud, serverless, mainframe, external, and partner connections. The server must not fall back to insecure or unencrypted protocols.*		✓	✓	319
9.2.3	Vérifier que toutes les connexions cryptées à des systèmes externes qui impliquent des informations ou des fonctions sensibles sont authentifiées. *Verify that all encrypted connections to external systems that involve sensitive information or functions are authenticated.*		✓	✓	287
9.2.4	Vérifier que la révocation de certification appropriée, telle que le protocole OCSP[14] est activée et configurée. *Verify that proper certification revocation, such as Online Certificate Status Protocol (OCSP) Stapling, is enabled and configured.*		✓	✓	299
9.2.5	Vérifier que les échecs de connexion TLS en arrière-plan sont enregistrés. *Verify that backend TLS connection failures are logged.*			✓	544

[14] OCSP : Online Certificate Status Protocol.

V10 Codes malveillants

Le code source doit satisfaire les exigences de haut niveau suivantes :

- L'activité malveillante est traitée de manière sûre et appropriée pour ne pas affecter le reste de l'application ;
- Il n'y a pas de bombes logiques (à retardement) ou d'autres attaques préprogrammées pour s'activer à un instant ''t'';
- Il n'y a pas de portes dérobées, Easter Eggs[15], d'attaques au salami, de rootkits ou de code non autorisé pouvant être contrôlé par un attaquant.

Il convient ainsi de tout mettre en œuvre pour s'assurer que le code ne comporte pas de code malveillant inhérent ou de fonctionnalité indésirable.

10.1 Intégrité du code

La meilleure défense contre les codes malveillants est de "trust, but verify". L'introduction d'un code non autorisé ou malveillant dans un code est souvent une infraction pénale. Les développeurs doivent régulièrement examiner le code, en particulier le temps d'accès, les E/S ou les fonctions réseau.

#	Description	N1	N2	N3	CWE
10.1.1	Vérifier qu'un outil d'analyse de code est utilisé pour détecter les codes potentiellement malveillants, tels que les fonctions temporelles, les opérations de fichiers et les connexions réseau non sécurisées. *Verify that a code analysis tool is in use that can detect potentially malicious code, such as time functions, unsafe file operations and network connections.*			✓	749

10.2 Recherche de code malveillant

Les codes malveillants sont extrêmement rares et difficiles à détecter. L'examen manuel ''ligne par ligne'' du code peut aider à rechercher des bombes logiques, mais même le plus expérimenté des examinateurs de code aura du mal à trouver un code malveillant même s'il sait qu'il existe.

Il n'est pas possible de se conformer à cette section sans un accès complet au code source, y compris aux bibliothèques des tiers.

#	Description	N1	N2	N3	CWE
10.2.1	Vérifier que le code source de l'application et les bibliothèques tierces ne contiennent pas de "numéro de téléphone domestique" ou de capacités de collecte de données non autorisées. Lorsque de telles fonctionnalités existent, obtenez l'autorisation de l'utilisateur pour leur fonctionnement avant de collecter des données. *Verify that the application source code and third party libraries do not contain unauthorized phone home or data collection capabilities. Where such functionality exists, obtain the user's permission for it to operate before collecting any data.*		✓	✓	359
10.2.2	Vérifier que l'application ne demande pas d'autorisations inutiles ou excessives pour les caractéristiques ou capteurs liés à la vie privée, tels que les contacts, les caméras, les microphones ou l'emplacement. *Verify that the application does not ask for unnecessary or excessive permissions to privacy related features or sensors, such as contacts, cameras, microphones, or location.*		✓	✓	272

[15] Easter Eggs : Une fonction cachée au sein d'un programme (image animée, jeu, message électronique, etc.) accessible grâce à un mot-clé ou à une combinaison de touches ou de clics.

10.2.3	Vérifier que le code source de l'application et les bibliothèques tierces ne contiennent pas de portes dérobées, telles que des comptes ou des clés codées en dur ou supplémentaires non documentées, des obscurcissements de code, des blobs binaires non documentés, des rootkits, ou des fonctions de débogage anti-débogage, non sécurisées, ou encore des fonctionnalités obsolètes, non sécurisées ou cachées qui pourraient être utilisées de manière malveillante si elles étaient découvertes. *Verify that the application source code and third party libraries do not contain back doors, such as hard-coded or additional undocumented accounts or keys, code obfuscation, undocumented binary blobs, rootkits, or anti-debugging, insecure debugging features, or otherwise out of date, insecure, or hidden functionality that could be used maliciously if discovered.*			✓	507
10.2.4	Vérifier que le code source de l'application et les bibliothèques tierces ne contiennent pas de bombes à retardement en recherchant les fonctions liées à la date et à l'heure. *Verify that the application source code and third party libraries do not contain time bombs by searching for date and time related functions.*			✓	511
10.2.5	Vérifier que le code source de l'application et les bibliothèques tierces ne contiennent pas de code malveillant, tel que des attaques de type salami, des contournements logiques ou des bombes logiques. *Verify that the application source code and third party libraries do not contain malicious code, such as salami attacks, logic bypasses, or logic bombs.*			✓	511
10.2.6	Vérifier que le code source de l'application et les bibliothèques tierces ne contiennent pas d'œufs de Pâques ou toute autre fonctionnalité potentiellement indésirable. *Verify that the application source code and third party libraries do not contain Easter eggs or any other potentially unwanted functionality.*			✓	507

10.3 Intégrité des applications

Une fois qu'une application est déployée, un code malveillant peut encore être inséré. Les applications doivent se protéger contre les attaques courantes, telles que l'exécution de code non signé provenant de sources non fiables et les rachats de sous-domaines.

#	Description	N1	N2	N3	CWE
10.3.1	Vérifier que si l'application dispose d'une fonction de mise à jour automatique du client ou du serveur, les mises à jour doivent être obtenues par des canaux sécurisés et signées numériquement. Le code de mise à jour doit valider la signature numérique de la mise à jour avant l'installation ou l'exécution de la mise à jour. *Verify that if the application has a client or server auto-update feature, updates should be obtained over secure channels and digitally signed. The update code must validate the digital signature of the update before installing or executing the update.*	✓	✓	✓	16
10.3.2	Vérifier que l'application utilise des protections d'intégrité, telles que la signature de code ou l'intégrité des sous-ressources. L'application ne doit pas charger ou exécuter du code provenant de sources non fiables, comme des includes de chargement, des modules, des plugins, du code ou des bibliothèques provenant de sources non fiables ou de l'Internet. *Verify that the application employs integrity protections, such as code signing or sub resource integrity. The application must not load or execute code from untrusted sources, such as loading includes, modules, plugins, code, or libraries from untrusted sources or the Internet.*	✓	✓	✓	353

10.3.3	Vérifier que l'application est protégée contre les reprises de sous-domaines si elle repose sur des entrées DNS ou des sous-domaines DNS, tels que des noms de domaine expirés, des pointeurs DNS ou CNAME obsolètes, des projets expirés dans des dépôts de code source publics, ou des API de nuages transitoires, des fonctions sans serveur, ou des espaces de stockage (*autogen-bucket-id*.cloud.example.com) ou similaires. Les protections peuvent consister à s'assurer que les noms DNS utilisés par les applications sont régulièrement vérifiés pour détecter toute expiration ou modification. *Verify that the application has protection from subdomain takeovers if the application relies upon DNS entries or DNS subdomains, such as expired domain names, out of date DNS pointers or CNAMEs, expired projects at public source code repos, or transient cloud APIs, serverless functions, or storage buckets (autogen-bucket-id.cloud.example.com) or similar. Protections can include ensuring that DNS names used by applications are regularly checked for expiry or change.*	✓	✓	✓	350

V11 Logique métier

Toute demande vérifiée satisfait aux exigences de haut niveau suivantes :

- Le flux logique de l'entreprise est séquentiel, traité dans l'ordre, et ne peut être contourné ;
- La logique métier comprend des limites pour détecter et prévenir les attaques automatisées ;
- Les flux de logique commerciale de grande valeur ont pris en compte les cas d'abus et les acteurs malveillants, et disposent de protections contre l'usurpation, l'altération, la répudiation, la divulgation d'informations et les attaques par élévation de privilèges.

11.1 Sécurité de la logique métier

La sécurité de la logique métier est tellement spécifique à chaque utilisation qu'aucune liste de contrôle n'est universelle. La sécurité de la logique des parties prenantes doit être conçue pour protéger contre les menaces externes probables, elle ne peut pas être ajoutée en utilisant des pares-feux d'applications web ou des communications sécurisées. L'utilisation de la modélisation des menaces lors des sprints de conception est recommandée (par exemple OWASP Cornucopia ou des outils similaires).

#	Description	N1	N2	N3	CWE
11.1.1	Vérifier que l'application traitera seulement les flux de logique métier pour un utilisateur dans l'ordre séquentiel des étapes et sans sauter d'étapes. *Verify that the application will only process business logic flows for the same user in sequential step order and without skipping steps.*	✓	✓	✓	841
11.1.2	Vérifier que l'application traitera seulement les flux de logiques métier, toutes les étapes étant traitées en temps humain réaliste, c'est-à-dire que les transactions ne sont pas soumises trop rapidement (effectuer par un robot). *Verify that the application will only process business logic flows with all steps being processed in realistic human time, i.e. transactions are not submitted too quickly.*	✓	✓	✓	799
11.1.3	Vérifier que l'application comporte des limites appropriées pour des actions ou des transactions commerciales spécifiques qui sont correctement exécutées par utilisateur. *Verify the application has appropriate limits for specific business actions or transactions which are correctly enforced on a per user basis.*	✓	✓	✓	770
11.1.4	Vérifier que l'application dispose de contrôles anti-automatisation suffisants pour détecter et protéger contre l'exfiltration de données, les demandes excessives de logique métiers, les téléchargements excessifs de fichiers ou les attaques par déni de service. *Verify that the application has anti-automation controls to protect against excessive calls such as mass data exfiltration, business logic requests, file uploads or denial of service attacks.*	✓	✓	✓	770
11.1.5	Vérifier que l'application a des limites ou une validation de la logique métier pour se protéger contre les risques ou les menaces commerciales probables, identifiés à l'aide de la modélisation des menaces ou de méthodologies similaires. *Verify the application has business logic limits or validation to protect against likely business risks or threats, identified using threat modeling or similar methodologies.*	✓	✓	✓	841
11.1.6	Vérifier que la demande ne souffre pas de problèmes de "temps de contrôle au moment de l'utilisation" (TOCTOU) ou d'autres situations de compétition (race condition) pour les opérations sensibles. *Verify that the application does not suffer from "Time Of Check to Time Of Use" (TOCTOU) issues or other race conditions for sensitive operations.*		✓	✓	367

11.1.7	Vérifier que les moniteurs de demande ne présentent pas d'événements ou d'activités inhabituels du point de vue de la logique métier. Par exemple, des tentatives d'effectuer des actions hors service ou des actions qu'un utilisateur normal ne tenterait jamais. *Verify that the application monitors for unusual events or activity from a business logic perspective. For example, attempts to perform actions out of order or actions which a normal user would never attempt.*		✓	✓	754
11.1.8	Vérifier que l'application dispose d'alertes configurables lorsque des attaques automatisées ou une activité inhabituelle sont détectées. *Verify that the application has configurable alerting when automated attacks or unusual activity is detected.*		✓	✓	390

V12 Dossiers et ressources

Toute application vérifiée doit satisfaire les exigences de haut niveau suivantes :

- Les données des fichiers non fiables doivent être traitées en conséquence et de manière sécurisée ;
- Les données de fichiers non fiables obtenues à partir de sources non fiables sont stockées en dehors de la racine web et avec des permissions limitées.

12.1 Téléchargement de fichiers

Bien que les bombes zip soient facilement testables à l'aide de techniques de test de pénétration, elles sont considérées au minimum comme N2 ceci permet d'encourager la prise en compte de la conception et du développement avec des tests manuels minutieux, et pour éviter que les tests de pénétration manuels ou automatisés engendrent une condition de déni de service.

#	Description	N1	N2	N3	CWE
12.1.1	Vérifier que la demande n'accepte pas de fichiers volumineux qui pourraient remplir l'espace de stockage ou provoquer un déni de service. *Verify that the application will not accept large files that could fill up storage or cause a denial of service.*	✓	✓	✓	400
12.1.2	Vérifier que l'application vérifie les fichiers compressés (par exemple, .zip, gz, docx, odt) par rapport à la taille maximale autorisée non compressée et par rapport au nombre maximal de fichiers avant de les décompresser. *Verify that the application checks compressed files (e.g. zip, gz, docx, odt) against maximum allowed uncompressed size and against maximum number of files before uncompressing the file.*		✓	✓	409
12.1.3	Vérifier qu'un quota de taille de fichier et un nombre maximum de fichiers par utilisateur sont appliqués pour s'assurer qu'un seul utilisateur ne peut pas remplir le stockage avec trop de fichiers, ou des fichiers excessivement volumineux. *Verify that a file size quota and maximum number of files per user is enforced to ensure that a single user cannot fill up the storage with too many files, or excessively large files.*		✓	✓	770

12.2 Intégrité des fichiers

#	Description	N1	N2	N3	CWE
12.2.1	Vérifier que les fichiers obtenus de sources non fiables sont validés comme étant du type attendu en fonction du contenu du fichier. *Verify that files obtained from untrusted sources are validated to be of expected type based on the file's content.*		✓	✓	434

12.3 Exécution des fichiers

#	Description	N1	N2	N3	CWE
12.3.1	Vérifier que les métadonnées de nom de fichier soumises par l'utilisateur ne sont pas utilisées directement par les systèmes de fichiers du système ou du cadre et qu'une API URL est utilisée pour protéger contre la traversée du chemin (path traversal). *Verify that user-submitted filename metadata is not used directly by system or framework filesystems and that a URL API is used to protect against path traversal.*	✓	✓	✓	22
12.3.2	Vérifier que les métadonnées de nom de fichier soumises par l'utilisateur sont validées ou ignorées pour empêcher la divulgation, la création, la mise à jour ou la suppression de fichiers locaux (LFI). *Verify that user-submitted filename metadata is validated or ignored to prevent the disclosure, creation, updating or removal of local files (LFI).*	✓	✓	✓	73

12.3.3	Vérifier que les métadonnées de nom de fichier soumises par l'utilisateur sont validées ou ignorées pour empêcher la divulgation ou l'exécution de fichiers distants (RFI), qui peuvent également conduire à des SSRF. *Verify that user-submitted filename metadata is validated or ignored to prevent the disclosure or execution of remote files via Remote File Inclusion (RFI) or Server-side Request Forgery (SSRF) attacks.*	✓	✓	✓	98
12.3.4	Vérifier que l'application protège contre le téléchargement de fichiers réfléchis (RFD) en validant ou en ignorant les noms de fichiers soumis par les utilisateurs dans un paramètre JSON, JSONP ou URL, l'en-tête Content-Type de la réponse doit être défini sur text/plain, et l'en-tête Content-Disposition doit avoir un nom de fichier fixe. *Verify that the application protects against Reflective File Download (RFD) by validating or ignoring user-submitted filenames in a JSON, JSONP, or URL parameter, the response Content-Type header should be set to text/plain, and the Content-Disposition header should have a fixed filename.*	✓	✓	✓	641
12.3.5	Vérifier que les métadonnées de fichiers non fiables ne sont pas utilisées directement avec l'API système ou les bibliothèques, pour se protéger contre l'injection de commandes du système d'exploitation. Verify that untrusted file metadata is not used directly with system API or libraries, to protect against OS command injection.	✓	✓	✓	78
12.3.6	Vérifier que l'application n'inclut pas et n'exécute pas de fonctionnalités provenant de sources non fiables, telles que des réseaux de distribution de contenu non vérifiés, des bibliothèques JavaScript, des bibliothèques node npm ou des DLL côté serveur. *Verify that the application does not include and execute functionality from untrusted sources, such as unverified content distribution networks, JavaScript libraries, node npm libraries, or server-side DLLs.*		✓	✓	829

12.4 Stockage des fichiers

#	Description	N1	N2	N3	CWE
12.4.1	Vérifier que les fichiers obtenus de sources non fiables sont stockés en dehors de la racine web, avec des permissions limitées. *Verify that files obtained from untrusted sources are stored outside the web root, with limited permissions.*	✓	✓	✓	552
12.4.2	Vérifier que les fichiers obtenus de sources non fiables sont analysés par des scanners antivirus pour empêcher le téléchargement de contenus malveillants connus. *Verify that files obtained from untrusted sources are scanned by antivirus scanners to prevent upload and serving of known malicious content.*	✓	✓	✓	509

12.5 Téléchargement des fichiers

#	Description	N1	N2	N3	CWE
12.5.1	Vérifier que l'application web est configuré pour ne servir que les fichiers ayant des extensions de fichier spécifiques afin d'éviter les informations involontaires et les fuites de code source. Par exemple, les fichiers de sauvegarde (par exemple .bak), les fichiers de travail temporaires (par exemple .swp), les fichiers compressés (.zip, .tar.gz, etc) et les autres extensions couramment utilisées par les éditeurs doivent être bloqués, sauf si cela est nécessaire. *Verify that the web tier is configured to serve only files with specific file extensions to prevent unintentional information and source code leakage. For example, backup files (e.g. .bak), temporary working files (e.g. .swp), compressed files (.zip, .tar.gz, etc) and other extensions commonly used by editors should be blocked unless required.*	✓	✓	✓	552
12.5.2	Vérifier que les demandes directes aux fichiers téléchargés ne seront jamais exécutées en tant que contenu HTML/JavaScript. *Verify that direct requests to uploaded files will never be executed as HTML/JavaScript content.*	✓	✓	✓	434

12.6 Protection des SSRF

#	Description	N1	N2	N3	CWE
12.6.1	Vérifier que le serveur web ou d'application est configuré avec une liste d'autorisation de ressources ou de systèmes à partir desquels le serveur peut envoyer des requêtes ou charger des données/fichiers. *Verify that the web or application server is configured with an allow list of resources or systems to which the server can send requests or load data/files from.*	✓	✓	✓	918

V13 API et services Web

Une application vérifiée doit utiliser des API de service de confiance (utilisant généralement JSON ou XML ou GraphQL) tel :

- Une authentification, une gestion de session et une autorisation adéquates de tous les services web ;
- Une validation d'entrée de tous les paramètres qui passent d'un niveau de confiance inférieur à un niveau supérieur ;
- Des contrôles de sécurité efficaces pour tous les types d'API, y compris les API en cloud et les API sans serveur.

13.1 Sécurité générique des services web

#	Description	N1	N2	N3	CWE
13.1.1	Vérifier que tous les composants de l'application utilisent les mêmes encodages et analyseurs pour éviter les attaques par analyse qui exploitent des comportements différents d'URI ou d'analyse de fichiers qui pourraient être utilisés dans les attaques SSRF et RFI. *Verify that all application components use the same encodings and parsers to avoid parsing attacks that exploit different URI or file parsing behavior that could be used in SSRF and RFI attacks.*	✓	✓	✓	116
13.1.3	Vérifier que les URL des API n'exposent pas d'informations sensibles, telles que la clé API, les jetons de session, etc. *Verify API URLs do not expose sensitive information, such as the API key, session tokens etc.*	✓	✓	✓	598
13.1.4	Vérifier que les décisions d'autorisation sont prises à la fois à l'URI, appliquées par la sécurité programmatique ou déclarative au niveau du contrôleur ou du routeur, et au niveau des ressources, appliquées par des autorisations basées sur des modèles de permission. *Verify that authorization decisions are made at both the URI, enforced by programmatic or declarative security at the controller or router, and at the resource level, enforced by model-based permissions.*		✓	✓	285
13.1.5	Vérifier que les demandes contenant des types de contenu inattendus ou manquants sont rejetées avec les en-têtes appropriés (statut de réponse HTTP 406 Inacceptable ou 415 Type de support non pris en charge). *Verify that requests containing unexpected or missing content types are rejected with appropriate headers (HTTP response status 406 Unacceptable or 415 Unsupported Media Type).*		✓	✓	434

13.2 Services web RESTful

La validation du schéma JSON en est à un stade préliminaire de normalisation. Il est la meilleure pratique pour les services web RESTful. De ce fait, il faut utiliser des stratégies de validation de données supplémentaires en combinaison avec la validation de schéma JSON :

- Validation de l'objet JSON, par exemple s'il y a des éléments manquants ou en trop.
- Validation des valeurs de l'objet JSON en utilisant des méthodes de validation d'entrée standard, telles que le type de données, le format de données, la longueur, etc.
- et validation formelle du schéma JSON.

En attendant que le schéma JSON soit mature, il faut surveiller attentivement toutes les bibliothèques de validation de schémas JSON utilisées, car elles devront être mises à jour régulièrement.

#	Description	N1	N2	N3	CWE
13.2.1	Vérifier que les méthodes HTTP RESTful activées sont un choix valable pour l'utilisateur ou une action, comme par exemple empêcher les utilisateurs normaux d'utiliser le verbe DELETE ou PUT sur des API ou des ressources protégées. *Verify that enabled RESTful HTTP methods are a valid choice for the user or action, such as preventing normal users using DELETE or PUT on protected API or resources.*	✓	✓	✓	650

13.2.2	Vérifier que la validation du schéma JSON est en place et vérifiée avant d'accepter la saisie. *Verify that JSON schema validation is in place and verified before accepting input.*	✓	✓	✓	20
13.2.3	Vérifier que les services Web RESTful qui utilisent des cookies sont protégés contre la falsification de requêtes intersites via l'utilisation d'au moins un ou plusieurs des éléments (double submit cookie pattern, CSRF nonces, or Origin request header checks). *Verify that RESTful web services that utilize cookies are protected from Cross-Site Request Forgery via the use of at least one or more of the following: double submit cookie pattern, CSRF nonces, or Origin request header checks.*	✓	✓	✓	352
13.2.5	Vérifier que les services REST inspectent explicitement le type de contenu entrant est celui attendu, tel que application/xml ou application/json. *Verify that REST services explicitly check the incoming Content-Type to be the expected one, such as application/xml or application/json.*		✓	✓	436
13.2.6	Vérifier que les en-têtes et les données utiles sont dignes de confiance et intègre. Exiger un cryptage fort pour le transport (TLS uniquement) peut être suffisant dans de nombreux cas, car il assure à la fois la protection de la confidentialité et de l'intégrité. Les messages signés peuvent fournir une assurance supplémentaire en plus des protections de transport pour les applications de haute sécurité, mais elles entraînent une complexité et des risques supplémentaires à comparer avec les avantages. *Verify that the message headers and payload are trustworthy and not modified in transit. Requiring strong encryption for transport (TLS only) may be sufficient in many cases as it provides both confidentiality and integrity protection. Per-message digital signatures can provide additional assurance on top of the transport protections for high-security applications but bring with them additional complexity and risks to weigh against the benefits.*		✓	✓	345

13.3 Service web SOAP

#	Description	N1	N2	N3	CWE
13.3.1	Vérifier que la validation du schéma XSD a lieu pour garantir un document XML correctement formé, suivie de la validation de chaque champ de saisie avant tout traitement de ces données. *Verify that XSD schema validation takes place to ensure a properly formed XML document, followed by validation of each input field before any processing of that data takes place.*	✓	✓	✓	20
13.3.2	Vérifier que la charge utile du message est signée en utilisant WS-Security pour assurer un transport fiable entre le client et le service. *Verify that the message payload is signed using WS-Security to ensure reliable transport between client and service.*		✓	✓	345

Note : En raison de problèmes liés aux attaques XXE contre les DTD, la validation des DTD ne doit pas être utilisée, et l'évaluation des DTD cadre doit être désactivée conformément aux exigences définies dans la configuration V14.

13.4 GraphQL

#	Description	N1	N2	N3	CWE
13.4.1	Vérifier qu'un mécanisme de limitation d'allocation de ressource ou de limitation de complexité soit en place pour prévenir les dénis de services. *Verify that a query allow list or a combination of depth limiting and amount limiting is used to prevent GraphQL or data layer expression Denial of Service (DoS) as a result of expensive, nested queries. For more advanced scenarios, query cost analysis should be used.*		✓	✓	770

13.4.2	Vérifier que la logique d'autorisation de GraphQL ou d'une autre couche de données doit être mise en œuvre au niveau de la couche de logique d'entreprise au lieu de la couche GraphQL. *Verify that GraphQL or other data layer authorization logic should be implemented at the business logic layer instead of the GraphQL layer.*		✓	✓	285

V14 Configuration

Toute application vérifiée doit satisfaire les conditions suivantes :

- Un environnement de construction sécurisé, reproductible et automatisable ;
- Une gestion des dépendances étroite et une configuration renforcée, de sorte que les composants obsolètes ou non sécurisés ne soient pas inclus dans l'application.

La configuration de l'application "out of the box" doit être sûre pour être sur Internet, ce qui signifie une configuration "out of the box".

14.1 Build et déploiement

Les pipelines de construction sont la base d'une sécurité reproductible : chaque fois qu'un élément non sécurisé est découvert, il peut être corrigé dans le code source, les scripts de construction ou de déploiement, et testé automatiquement. Nous encourageons fortement l'utilisation de pipelines de compilation avec des contrôles de sécurité et de dépendance automatiques qui avertissent ou interrompent la compilation afin d'éviter que des problèmes de sécurité connus ne soient déployés en production. Les étapes manuelles effectuées de manière irrégulière conduisent directement à des erreurs de sécurité évitables.

Alors que l'industrie se dirige vers un modèle DevSecOps, il est important de garantir la disponibilité et l'intégrité continues du déploiement et de la configuration pour atteindre un état "known good". Dans le passé, si un système était piraté, il fallait des jours, voire des mois, pour prouver qu'aucune autre intrusion n'avait eu lieu. Aujourd'hui, avec l'avènement des infrastructures définies par logiciel, des déploiements A/B rapides sans aucun temps d'arrêt, et des constructions automatisées conteneurisées, il est possible de construire, de durcir et de déployer automatiquement et en continu un "known good" en remplacement de tout système compromis.

Si les modèles traditionnels sont toujours en place, des mesures manuelles doivent être prises pour renforcer et sauvegarder cette configuration afin de permettre le remplacement rapide des systèmes compromis par des systèmes à haute intégrité et sans compromis, et ce dans les meilleurs délais.

La conformité à cette section nécessite un système de construction automatisé et l'accès à des scripts de construction et de déploiement.

#	Description	N1	N2	N3	CWE
14.1.1	Vérifier que les processus de construction et de déploiement des applications sont effectués de manière sûre et répétable, comme l'automatisation des CI / CD, la gestion automatisée de la configuration et les scripts de déploiement automatisés. *Verify that the application build and deployment processes are performed in a secure and repeatable way, such as CI / CD automation, automated configuration management, and automated deployment scripts.*		✓	✓	
14.1.2	Vérifier que les drapeaux du compilateur sont configurés pour activer toutes les protections et les avertissements disponibles contre les débordements de mémoire tampon, y compris la randomisation de la pile, la prévention de l'exécution des données, et pour casser la compilation si un pointeur, une mémoire, une chaîne de format, un entier ou une chaîne de caractères dangereux sont trouvés. *Verify that compiler flags are configured to enable all available buffer overflow protections and warnings, including stack randomization, data execution prevention, and to break the build if an unsafe pointer, memory, format string, integer, or string operations are found.*		✓	✓	120
14.1.3	Vérifier que la configuration du serveur est durcie conformément aux recommandations du serveur d'application et des cadres utilisés. *Verify that server configuration is hardened as per the recommendations of the application server and frameworks in use.*		✓	✓	16

14.1.4	Vérifier que l'application, la configuration et toutes les dépendances peuvent être redéployées à l'aide de scripts de déploiement automatisés, construites à partir d'un runbook documenté et testé dans un délai raisonnable, ou restaurées à partir de sauvegardes en temps utile. *Verify that the application, configuration, and all dependencies can be re-deployed using automated deployment scripts, built from a documented and tested runbook in a reasonable time, or restored from backups in a timely fashion.*		✓	✓	
14.1.5	Vérifier que les administrateurs autorisés peuvent vérifier l'intégrité de toutes les configurations pertinentes pour la sécurité afin de détecter les altérations. *Verify that authorized administrators can verify the integrity of all security-relevant configurations to detect tampering.*			✓	

14.2 Dépendances

La gestion des dépendances est essentielle au bon fonctionnement de toute application, quel que soit son type. L'incapacité à se tenir à jour avec des dépendances obsolètes ou peu sûres est la cause première des attaques les plus importantes et les plus coûteuses à ce jour.

Remarque : au niveau 1, la conformité à la norme 14.2.1 concerne les observations ou les détections de bibliothèques et de composants côté client et autres, plutôt que l'analyse statique du code de construction ou l'analyse des dépendances, plus précise. Ces techniques plus précises pourraient être découvertes par des entretiens, le cas échéant.

#	Description	N1	N2	N3	CWE
14.2.1	Vérifier que tous les composants sont à jour, de préférence en utilisant un vérificateur de dépendances pendant le temps de construction ou de compilation. *Verify that all components are up to date, preferably using a dependency checker during build or compile time*	✓	✓	✓	1026
14.2.2	Vérifier que toutes les fonctionnalités, la documentation, les exemples d'applications et les configurations inutiles sont supprimés. Verify that all unneeded features, documentation, sample applications and configurations are removed.	✓	✓	✓	1002
14.2.3	Vérifier que si les actifs d'application, tels que les bibliothèques JavaScript, les feuilles de style CSS ou les polices web, sont hébergés en externe sur un réseau de diffusion de contenu (CDN) ou un fournisseur externe, l'intégrité des sous-ressources (SRI) est utilisée pour valider l'intégrité de l'actif. *Verify that if application assets, such as JavaScript libraries, CSS or web fonts, are hosted externally on a Content Delivery Network (CDN) or external provider, Subresource Integrity (SRI) is used to validate the integrity of the asset.*	✓	✓	✓	829
14.2.4	Vérifier que les composants tiers proviennent de dépôts prédéfinis, fiables et continuellement entretenus. *Verify that third party components come from pre-defined, trusted and continually maintained repositories.*		✓	✓	829
14.2.5	Vérifier qu'un catalogue d'inventaire de toutes les bibliothèques tierces en service est tenu à jour. *Verify that a Software Bill of Materials (SBOM) is maintained of all third party libraries in use.*		✓	✓	
14.2.6	Vérifier que la surface d'attaque est réduite en mettant en bac à sable ou en encapsulant des bibliothèques tierces pour n'exposer que le comportement requis dans l'application. *Verify that the attack surface is reduced by sandboxing or encapsulating third party libraries to expose only the required behaviour into the application.*		✓	✓	265

14.3 Divulgation involontaire de la sécurité

Les configurations de production devraient être renforcées pour se protéger contre les attaques courantes, telles que les consoles de débogage, relever la barre pour les attaques de type "cross-site scripting" (XSS) et "remote file inclusion" (RFI), et pour éliminer les "vulnérabilités" triviales de découverte d'informations qui sont la marque indésirable

de nombreux rapports de tests de pénétration. Nombre de ces problèmes sont rarement considérés comme un risque important, mais ils sont liés à d'autres vulnérabilités. Si ces problèmes ne sont pas présents par défaut, elle place la barre plus haute avant que la plupart des attaques puissent réussir.

#	Description	N1	N2	N3	CWE
14.3.2	Vérifier que les modes de débogage du serveur web ou d'application et du cadre d'application sont désactivés en production afin d'éliminer les fonctionnalités de débogage, les consoles de développement et les divulgations de sécurité non intentionnelles. *Verify that web or application server and application framework debug modes are disabled in production to eliminate debug features, developer consoles, and unintended security disclosures.*	✓	✓	✓	497
14.3.3	Vérifier que les en-têtes HTTP ou toute partie de la réponse HTTP n'exposent pas d'informations détaillées sur la version des composants du système. *Verify that the HTTP headers or any part of the HTTP response do not expose detailed version information of system components.*	✓	✓	✓	200

14.4 Sécurité des en-têtes HTTP

#	Description	N1	N2	N3	CWE
14.4.1	Vérifier que chaque réponse HTTP contient un en-tête Content-Type. Les types de contenu text/*, /+xml et application/xml doivent également spécifier un jeu de caractères sûr (par exemple, UTF-8, ISO-8859-1). *Verify that every HTTP response contains a Content-Type header. Also specify a safe character set (e.g., UTF-8, ISO-8859-1) if the content types are text/*, /+xml and application/xml. Content must match with the provided Content-Type header.*	✓	✓	✓	173
14.4.2	Vérifier que toutes les réponses de l'API contiennent Content-Disposition : attachment ; filename="api.json" (ou tout autre nom de fichier approprié pour le type de contenu). *Verify that all API responses contain a Content-Disposition: attachment; filename="api.json" header (or other appropriate filename for the content type).*	✓	✓	✓	116
14.4.3	Vérifier qu'une politique de sécurité du contenu (CSP) est en place pour aider à atténuer l'impact des attaques XSS comme les vulnérabilités d'injection HTML, DOM, JSON et JavaScript. *Verify that a Content Security Policy (CSP) response header is in place that helps mitigate impact for XSS attacks like HTML, DOM, JSON, and JavaScript injection vulnerabilities.*	✓	✓	✓	1021
14.4.4	Vérifier que toutes les réponses contiennent X-Content-Type-Options: nosniff. *Verify that all responses contain a X-Content-Type-Options: nosniff header*	✓	✓	✓	116
14.4.5	Vérifier que l'en-tête Strict-Transport-Security est inclus dans toutes les réponses et pour tous les sous-domaines, comme Strict-Transport-Security : max-age=15724800 ; includeSubdomains. *Verify that a Strict-Transport-Security header is included on all responses and for all subdomains, such as Strict-Transport-Security: max-age=15724800; includeSubdomains.*	✓	✓	✓	523
14.4.6	Vérifier qu'un en-tête "Referrer-Policy" approprié est inclus, tel que "no-referrer" ou "same-origin". *Verify that a suitable Referrer-Policy header is included to avoid exposing sensitive information in the URL through the Referer header to untrusted parties.*	✓	✓	✓	116

14.4.7	Vérifier que le contenu d'une application web ne peut pas être intégré par défaut dans un site tiers et que l'intégration des ressources exactes n'est autorisée que si nécessaire en utilisant un en-tête approprié tel "Content-Security-Policy: frame-ancestors" ou "X-Frame-Options". *Verify that the content of a web application cannot be embedded in a third-party site by default and that embedding of the exact resources is only allowed where necessary by using suitable Content-Security-Policy: frame-ancestors and X-Frame-Options response headers*	✓	✓	✓	1021

14.5 Validation des en-têtes de requête http

#	Description	N1	N2	N3	CWE
14.5.1	Vérifier que le serveur d'application accepte uniquement les méthodes HTTP utilisées par l'application / API (incluant les requêtes de type OPTIONS), et les alertes sur toutes les demandes qui sont invalides pour le contexte de l'application. *Verify that the application server only accepts the HTTP methods in use by the application/API, including pre-flight OPTIONS, and logs/alerts on any requests that are not valid for the application context.*	✓	✓	✓	749
14.5.2	Vérifier que l'en-tête Origin fourni n'est pas utilisé pour les décisions d'authentification ou de contrôle d'accès, car l'en-tête Origin peut facilement être modifié par un attaquant. *Verify that the supplied Origin header is not used for authentication or access control decisions, as the Origin header can easily be changed by an attacker.*	✓	✓	✓	346
14.5.3	Vérifier que l'en-tête "Cross-Origin Resource Sharing" (CORS) Access-Control-Allow-Origin utilise une liste d'autorisation stricte de domaines et sous-domaines de confiance pour la comparaison avec l'origine "null" et ne la prend pas en charge. *Verify that the Cross-Origin Resource Sharing (CORS) Access-Control-Allow-Origin header uses a strict allow list of trusted domains and subdomains to match against and does not support the "null" origin.*	✓	✓	✓	346
14.5.4	Vérifier que les en-têtes HTTP ajoutés par un proxy de confiance ou des dispositifs SSO, tels qu'un jeton au porteur, sont authentifiés par l'application. *Verify that HTTP headers added by a trusted proxy or SSO devices, such as a bearer token, are authenticated by the application.*		✓	✓	306

Conclusion

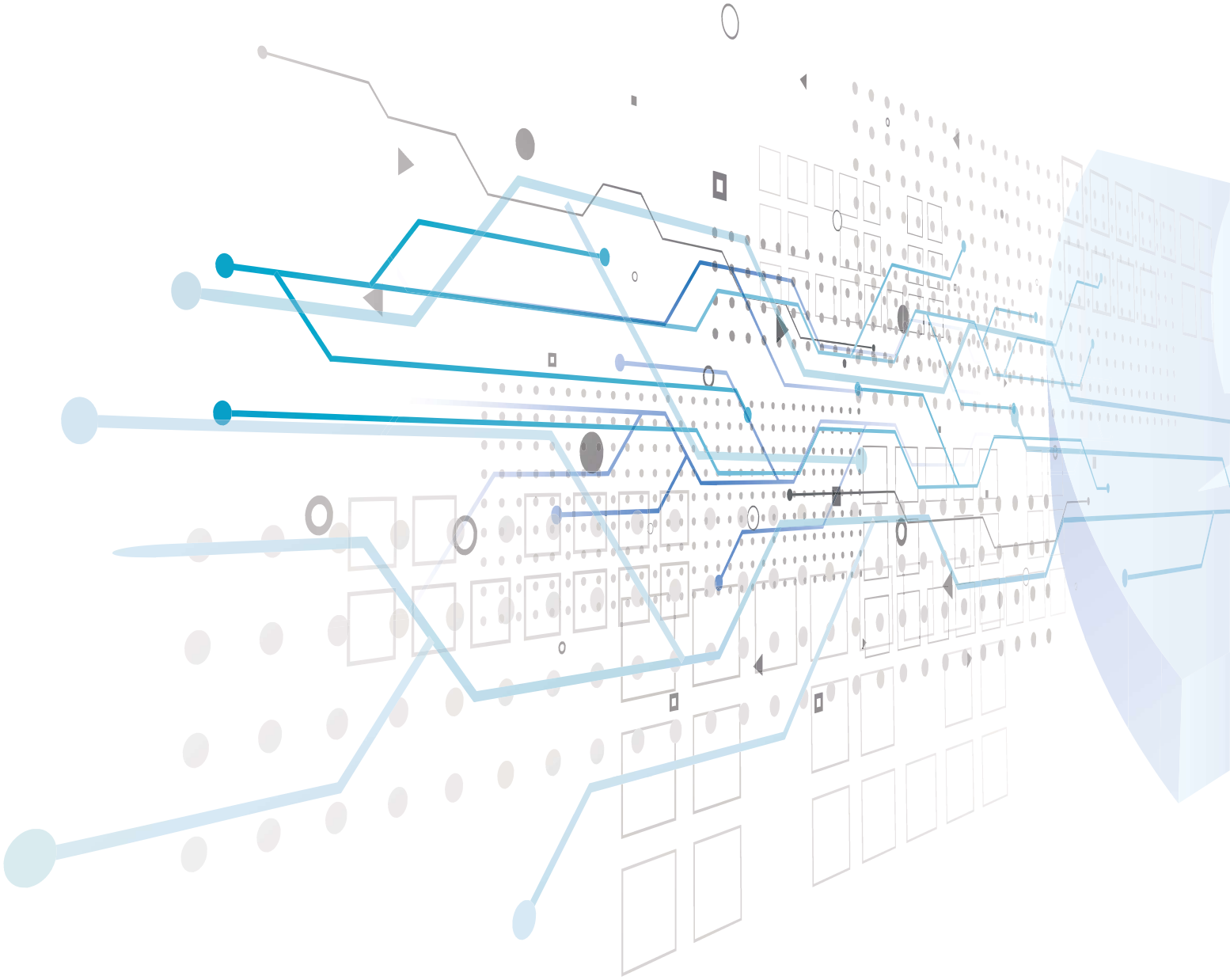
L'objectif de la sécurité logicielle est d’assurer la confidentialité, l'intégrité et la disponibilité des informations traitées par un service applicatif. Il est impératif d'intégrer la sécurité dès la phase de conception, et de la respecter le long du cycle de vie du logiciel.

En adoptant cette démarche, les développeurs peuvent affirmer que les logiciels développés respectent les meilleures pratiques de sécurité.

La mise en œuvre de mesures de sécurité après le déploiement d’un logiciel est nettement plus coûteuse et n’offre généralement qu’une protection limitée par rapport à la sécurité intégrée dès le début du processus suivant ce référentiel.

Pour gérer efficacement les problèmes de sécurité, il est nécessaire d'intégrer une réflexion axée sur la sécurité tout au long du processus de développement. Cela réduit le risque d’ignorer des exigences de sécurité qui peuvent être importantes d’une part et d’autre part, pour éviter de commettre des erreurs critiques dans la conception du logiciel.

Adopter ce référentiel de vérification de la sécurité des applications constitue un gage de sécurité, notamment si au moins le niveau 2 est privilégié.



RÉFÉRENTIEL DE VÉRIFICATION DE LA SÉCURITÉ DES APPLICATIONS

DSR
Tél : +212 5 37 54 03 95
Fax: +212 5 37 71 39 73
Email: dsr@dgssi.gov.ma

maCERT
Tél : +212 5 37 57 21 47
Fax: +212 5 37 57 20 53
Email: contact@macert.gov.ma

DAFCE
Tél : +212 5 37 54 03 87
Fax: +212 5 37 54 01 81
Email: dafce@dgssi.gov.ma

www dgssi.gov.ma