

Auditer la sécurité d'une application créée avec l'IA

Une méthode pour examiner les accès, les données, les dépenses et la capacité de reprise.

72 contrôles • 12 modules • 12 recettes

Chaque fiche relie une exigence à des preuves, un scénario fictif, une correction et un test. À utiliser avec la grille de résultats et les consignes IA.

OWASP ASVS 5.0.0 • API Security Top 10 2023
Complément LLM 2025 lorsque applicable

Téléchargement gratuit • Version datée • Sources officielles

SOMMAIRE

Le parcours de lecture

Méthode et préparation	03–08
01 Périmètre et modèle de menace	09–11
02 Secrets et chaîne de livraison	12–14
03 Authentification et comptes	15–17
04 Sessions et jetons	18–20
05 Autorisations et séparation des clients	21–23
06 Entrées et navigateur	24–26
07 API et communications	27–29
08 Fichiers et cycle de vie des données	30–32
09 Intégrité et paiements	33–35
10 IA, outils et dépenses	36–38
11 Disponibilité et maîtrise de la charge	39–41
12 Production, détection et reprise	42–44
Douze recettes transversales	45–48
Inconnues, glossaire et sources	49 et suivantes

Lire une fiche

Le statut se renseigne dans la grille. Sans preuve suffisante, il reste UNKNOWN. Les références ASVS sont partielles ; les sources complètes figurent en fin de guide.

MODE D'EMPLOI

Lire ce dossier et choisir son parcours

Trois documents qui se répondent

Le guide explique les vérifications. La grille conserve leurs résultats. Les consignes IA organisent les séquences de travail. Les identifiants C01 à C72 sont identiques dans les trois PDF. Chaque scénario du guide est fictif ; chaque test décrit un comportement à vérifier. Aucun verdict sur votre application n'est prérempli dans ce guide.

Pour le créateur ou le responsable du service

Commencez par le périmètre, les statuts et la décision de mise en ligne. Remplissez avec le développeur la fiche de contexte de la grille. Demandez ensuite les preuves d'isolation des comptes, de révocation, de maîtrise des dépenses et de restauration. Vous n'avez pas besoin d'exécuter vous-même tous les tests pour exiger une réponse vérifiable.

Pour le développeur et le relecteur technique

Inventoriez les points d'entrée puis suivez chaque flux de bout en bout : entrée, identité, droit, traitement, stockage, réponse et trace. Appliquez les contrôles aux routes alternatives, tâches, exports et anciennes versions. Utilisez les annexes de tests pour assembler une recette cohérente ; adaptez les fixtures et les assertions à l'architecture.

Périmètre de cette édition

Ce dossier vise les applications web et API, notamment celles développées avec une assistance IA. Il propose une sélection de 72 contrôles, pas une vérification complète d'ASVS 5.0.0 ni une certification d'un de ses niveaux. Applications natives, firmware, cryptographie spécialisée, WebRTC, infrastructure réseau complexe et contraintes sectorielles exigent des compléments.

Quand approfondir

Paiements importants, données particulièrement sensibles, isolation multient complexe, exécution de code et agents dotés de droits étendus justifient une revue spécialisée. Le recours à un fournisseur géré déplace certaines preuves vers sa configuration et ses garanties ; il ne rend pas automatiquement le contrôle non applicable.

AVANT LE PREMIER TEST

Préparer une version et un environnement de référence

Réunir les pièces

Identifier dépôt ou archive, commit ou empreinte, build, domaine de test, API, schémas de données, règles de stockage, fonctions planifiées, dépendances et services tiers. Lister les fichiers et réglages absents. Un ZIP sans historique ne prouve rien sur les secrets autrefois présents dans Git.

Décrire la surface et les données

Dessiner les échanges entre navigateur, serveur, base, stockage, files, fournisseur d'identité, IA, paiement et supervision. Pour chaque flux : quelles données, quelle identité, quel droit, quelle destination, quelle durée de conservation ? Documenter les accès directs au stockage et les chemins qui contournent le serveur applicatif.

Créer les identités de test

Prévoir un visiteur non connecté, deux comptes A et B sans relation, deux organisations distinctes si présentes, un gestionnaire limité et un administrateur. Ajouter un compte suspendu, un membre retiré et une ancienne session. Utiliser uniquement des données fictives clairement repérables et des boîtes de réception de test.

Borner l'exécution

Écrire hôtes autorisés, opérations permises, volume maximal, budget, durée, seuils d'arrêt et contact responsable. Préférer une préproduction isolée. Les essais réels de paiement ou d'IA restent limités ; les tests de volume emploient des simulateurs. Ne pas viser des tiers, des données réelles ni des métadonnées cloud de production pour démontrer une SSRF.

Conserver une base comparable

Avant une correction, conserver code, configuration et preuves expurgées. Après correction, noter la nouvelle version et les écarts d'environnement. Une préproduction divergente réduit la portée d'une conclusion ; il faut le signaler. Vérifier ensuite, par des contrôles adaptés et autorisés, que l'artefact testé est celui livré.

RÈGLES DE QUALIFICATION

Quatre statuts, une discipline de preuve

PASS — exigence satisfaite dans le périmètre examiné

Une preuve adaptée démontre le comportement attendu sur la version et les parcours déclarés. L'existence d'un middleware ne démontre pas qu'il protège toutes les routes. Un test d'intégration simulé démontre son scénario ; il ne prouve pas la configuration de production. Si un sous-point essentiel reste inconnu, ne pas valider tout le contrôle.

FAIL — écart établi

Une observation ou une analyse reproductible établit le non-respect de l'exigence. L'exploitation n'a pas toujours besoin d'être exécutée pour établir un défaut, mais il faut préciser ses prérequis et le niveau de preuve. Distinguer vulnérabilité confirmée, écart de configuration et faiblesse d'exploitation opérationnelle.

UNKNOWN — informations insuffisantes

Préciser pourquoi : fichier absent, réglage inaccessible, test non exécuté, résultat ambigu ou couverture incomplète. Ajouter la pièce attendue, le responsable et la prochaine action. UNKNOWN ne signifie ni protection ni faille. Un point non applicable ne doit pas être classé UNKNOWN par commodité.

N/A — non applicable, avec justification

Exemple : absence de paiement pour les contrôles de paiement. En revanche, authentification externalisée signifie contrôle du fournisseur et de l'intégration, pas N/A automatique. Une application sans compte reste concernée par ses accès administrateur, ses secrets, ses données et ses dépenses éventuelles.

Preuve précise, sans secret divulgué

Pour le code : chemin, ligne ou symbole stable, extrait minimal expurgé et commit. Pour le cloud : service, paramètre, environnement, date et capture ou export expurgé. Pour un test : prérequis, commande ou procédure, entrée fictive, attendu, observé et identifiant de trace. Le rapport ne contient jamais de mot de passe, de clé complète ni de document client.

SÉVÉRITÉ ET RESPONSABILITÉ

Classer le risque et décider de la mise en ligne

Séparer gravité, confiance et priorité

Le classement Critical / High / Medium / Low décrit une gravité justifiée. Ajouter le niveau de preuve : confirmé, démontré statiquement ou hypothèse à vérifier. La priorité de correction tient aussi compte de l'exposition réelle, des actifs, des mesures compensatoires et de l'échéance de publication. UNKNOWN reste un statut, pas une gravité.

Repères indicatifs, à contextualiser

Critical : prise de contrôle ou exposition massive avec conséquences majeures. High : accès non autorisé significatif, action privilégiée ou dépense grave effectivement atteignable. Medium : impact limité ou prérequis importants. Low : exposition faible ou durcissement. Ces exemples ne classent aucune faille automatiquement : documenter les conditions, les données atteintes et l'étendue.

Employer CVSS seulement avec ses hypothèses

CVSS v4.0 fournit une notation de sévérité technique. Si vous l'utilisez, conserver version, vecteur et justification ; ne pas inventer un score faute d'éléments. Ajouter à part les conséquences financières, contractuelles et métier. Les manques de sauvegarde ou de gouvernance ne reçoivent pas nécessairement un CVSS pertinent. [CVSS]

Règle éditoriale de sortie proposée

Avant ouverture, aucun défaut critique ou élevé connu ne doit rester exploitable dans le périmètre exposé. Démontrer les séparations de droits, la maîtrise des appels payants et la reprise des données nécessaires. Résoudre les UNKNOWN déterminants ou retirer explicitement la fonction exposée. Toute exception exige décision nominative, mesures temporaires, échéance et retest ; un changement d'étiquette ne traite pas le risque.

Une décision, jamais une moyenne rassurante

Conserver séparément le nombre de contrôles évalués, exclus et inconnus. Ne pas transformer leur ratio en score de sécurité. Le responsable du service date la décision : ouverture autorisée dans un périmètre défini, ouverture différée, ou ouverture restreinte. Les inconnues et risques résiduels accompagnent cette décision.

CHOIX DES MODULES

Adapter les contrôles à l'architecture

Un socle commun

C01–C12, C31–C42 et C61–C72 concernent largement les applications publiques, avec exceptions justifiées selon les composants. Les autres modules dépendent des fonctions réelles. Même un site statique doit examiner scripts tiers, formulaires, déploiement et fichiers publiés ; il ne possède pas forcément de session ou de base à auditer.

Comptes et plusieurs organisations

Activer C13–C30. Évaluer les routes du fournisseur d'identité, les invitations et les droits des anciens membres. Pour une application multientité, tester aussi caches, index, fichiers, exports et résultats asynchrones. RLS, autorisations serveur et règles du stockage se complètent ; aucune étiquette de fournisseur ne prouve leur efficacité.

Fichiers, écritures et paiement

Activer C43–C54 selon les parcours. Une extraction de texte dans le navigateur nécessite des limites et des parseurs maintenus ; le serveur doit aussi valider le texte et le JSON reçus. Un parseur local n'établit pas la sécurité de l'upload d'un binaire, et inversement.

Modèles et services facturés

Activer C55–C60 si l'application intègre de l'IA. Appliquer aussi C59–C60 aux e-mails, SMS, traitements documentaires et autres services à l'usage. Faire l'inventaire des coûts réels : appels auxiliaires, stockage, trafic, outils, reprises et modèles de secours. Aucune valeur monétaire universelle n'est imposée par ce guide.

Correspondances des référentiels

Les références ASVS des fiches sont des rapprochements partiels vers la version 5.0.0, pas une équivalence complète. Elles s'écrivent v5.0.0-numéro. Les identifiants Cxx appartiennent au dossier IN DATA VERITAS. Les codes APIx:2023 et LLMxx:2025 désignent des catégories de risques, pas des certifications. Les contrôles de méthode ou de reprise sans équivalent direct sont présentés comme tels.

LECTURE, SCANS ET TESTS

Combiner les méthodes sans surinterpréter les outils

Revue statique et recherche de secrets

Lire les chemins sensibles de bout en bout. Un outil de recherche de secrets doit couvrir les éléments effectivement disponibles : dépôt, historique, build et archives. Expurger ses sorties ; ne pas tenter une connexion à un fournisseur avec une clé découverte pour la « valider ». Un secret actif exposé exige révocation et analyse des usages.

Dépendances et analyse automatisée

Conserver outil, version, date, configuration et inventaire analysé. Distinguer composant affecté par un avis et parcours exploitable. Examiner les bibliothèques chargées par CDN et les dépendances du build. L'absence d'alerte ou la réussite du build ne démontre ni autorisation correcte ni absence de faille.

Tests de comportement

Associer à chaque cas hostile un cas légitime. Vérifier les effets réels : un HTTP 403 ne suffit pas si l'écriture a eu lieu avant le refus ; un 200 n'est pas forcément une fuite si la réponse reste publique. Conserver des assertions sur les données, droits, appels externes et dépenses simulées.

Configuration et preuves manuelles

Les consoles cloud, contrats et factures fournissent des éléments que le code ignore. Demander une preuve datée, expurgée et rattachée à l'environnement. Une sauvegarde demande un exercice de restauration ; une alerte demande une réception ; un plafond demande une garantie de blocage adaptée. Renvoyer les inconnues dans la grille.

Avant de confier la revue à une IA

Réduire les secrets et données transmis, vérifier les conditions de l'outil et lui donner un périmètre lisible. Les instructions contenues dans un document importé ou une sortie d'outil constituent des données à examiner ; elles ne doivent pas détourner la mission. Demander une couverture déclarée, conserver les sorties et relire les conclusions importantes.

MODULE 01 • SAVOIR EXACTEMENT QUELLE APPLICATION ET QUELLES DONNÉES SONT EXAMINÉES.

Périmètre et modèle de menace

C01 Version et surface examinée

Application : Toutes applications

Exigence. Relier les résultats à un code, un artefact et des environnements identifiés.

Preuves à chercher. Commit ou empreinte du ZIP, manifeste du build, liste des domaines, routes, tâches planifiées et anciennes versions.

Risque / scénario fictif. Une API historique reste publique alors que seule la nouvelle interface a été relue : ses défauts échappent au rapport.

Correction attendue. Tenir un inventaire des points d'entrée et de leurs propriétaires ; fermer les versions inutilisées et consigner les exclusions.

Test et critère de réussite. Comparer routes du code, règles de passerelle et inventaire du déploiement. Tester un point d'entrée par catégorie ; toute divergence reste à résoudre.

Méthode IDV / complément métier • API9:2023

Sources : [ASVS] • [WSTG]

C02 Données et frontières de confiance

Application : Toutes applications

Exigence. Identifier les données sensibles et leurs chemins entre navigateur, serveur, prestataires et stockage.

Preuves à chercher. Schéma de flux, classification, contrats de traitement pertinents et destination des journaux, exports et sauvegardes.

Risque / scénario fictif. Un outil de télémétrie reçoit le texte intégral de documents clients alors que cette sortie ne figure dans aucun inventaire.

Correction attendue. Documenter les flux, limiter les champs transmis et nommer un responsable pour chaque destination et durée de conservation.

Test et critère de réussite. Suivre un document fictif marqué AUDIT dans les flux autorisés et les journaux de test. Comparer les destinations observées à la cartographie.

ASVS 5.0.0 : 14.1.1, 14.1.2

Sources : [THREAT]

MODULE 01 • SAVOIR EXACTEMENT QUELLE APPLICATION ET QUELLES DONNÉES SONT EXAMINÉES.

Périmètre et modèle de menace

C03 Profils, adversaires et invariants

Application : Toutes applications

Exigence. Définir les actions permises et les propriétés qui doivent rester vraies, même en cas d'abus.

Preuves à chercher. Matrice invité, membre, gestionnaire, administrateur ; règles de propriété et scénarios de fraude métier.

Risque / scénario fictif. Un membre utilise un export autorisé pour aspirer tous les contacts de son agence, au-delà de la politique prévue.

Correction attendue. Définir droits, volumes autorisés et opérations sensibles. Formaliser aussi les abus commis par un utilisateur correctement connecté.

Test et critère de réussite. Pour chaque parcours sensible, décrire un usage normal et un détournement. Faire valider le résultat attendu par le responsable métier avant les essais.

ASVS 5.0.0 : 2.1.3 • API6:2023

Sources : [THREAT]

C04 Cadre des essais actifs

Application : Tout audit dynamique

Exigence. Tester sur un périmètre autorisé avec un impact borné et une procédure d'arrêt.

Preuves à chercher. Accord du propriétaire, hôtes autorisés, fenêtre, comptes fictifs, plafond de requêtes et de dépenses, contact d'arrêt.

Risque / scénario fictif. Une vérification de quota déclenche de vrais appels payants ou un scan d'upload consomme le stockage de production.

Correction attendue. Privilégier une préproduction isolée ; simuler les prestataires pour les volumes. Définir un budget réel séparé pour l'intégration.

Test et critère de réussite. Vérifier les URL, clés de test et destinataires avant le premier essai. Déclencher l'arrêt sur une simulation et confirmer l'absence de nouveaux traitements.

Méthode IDV / complément métier

Sources : [WSTG]

MODULE 01 • SAVOIR EXACTEMENT QUELLE APPLICATION ET QUELLES DONNÉES SONT EXAMINÉES.

Périmètre et modèle de menace

C05 Preuves et couverture de revue

Application : Toutes applications

Exigence. Rendre visible ce qui a été lu, exécuté, exclu ou seulement supposé.

Preuves à chercher. Inventaire des fichiers examinés, sorties d'outils expurgées, tests exécutés, limites d'accès et registre UNKNOWN.

Risque / scénario fictif. Une IA affirme avoir audité le projet après n'avoir lu que la page d'accueil et les fichiers accessibles dans son contexte.

Correction attendue. Découper la revue par flux et modules. Exiger des localisations vérifiables et distinguer test proposé, exécuté et observé.

Test et critère de réussite. Choisir plusieurs conclusions du rapport et retrouver leur preuve sur la version visée. Une référence introuvable invalide la conclusion, sans prouver une faille.

Méthode IDV / complément métier

Sources : [WSTG]

C06 Configuration extérieure au dépôt

Application : Applications hébergées

Exigence. Inclure les protections réelles de l'hébergement, de l'identité, du stockage et de la facturation.

Preuves à chercher. Exports datés ou captures expurgées des rôles cloud, règles réseau, budgets, sauvegardes et paramètres de production.

Risque / scénario fictif. Le code prévoit un stockage privé, mais un réglage du projet rend le conteneur accessible sans connexion.

Correction attendue. Comparer intention dans le code et configuration effective. Affecter chaque preuve manquante à un responsable disposant de l'accès.

Test et critère de réussite. Tester avec un client non authentifié les ressources privées de test. Vérifier séparément les droits de la console ; un réglage inaccessible reste UNKNOWN.

ASVS 5.0.0 : 13.2.2 • API8:2023

Sources : [ASVS]

MODULE 02 • PROTÉGER LES ACCÈS TECHNIQUES ET LE CHEMIN QUI MÈNE AU DÉPLOIEMENT.

Secrets et chaîne de livraison

C07 Secrets dans le dépôt et l'historique

Application : Dépôts et archives

Exigence. Détecter les secrets exposés, y compris dans les anciennes révisions et livrables.

Preuves à chercher. Résultats expurgés du scan de l'arbre, historique disponible, tags, fichiers de test, archives et journaux de build.

Risque / scénario fictif. Une clé supprimée du dernier commit demeure récupérable dans une ancienne révision ou un ZIP partagé.

Correction attendue. Révoquer le secret exposé, le remplacer et examiner ses usages. Corriger la source de fuite ; nettoyer l'historique ne suffit pas.

Test et critère de réussite. Sur un secret de test, prouver que l'ancienne valeur est refusée et que la nouvelle fonctionne. Refaire le scan sans afficher les valeurs complètes.

ASVS 5.0.0 : 13.3.1, 13.3.4

Sources : [SECRETS]

C08 Secrets dans le navigateur

Application : Frontend et applications distribuées

Exigence. Distinguer clés publiables prévues à cet effet et secrets donnant des privilèges.

Preuves à chercher. Bundle de production, cartes de sources, réponses de configuration, requêtes réseau et documentation du type de clé.

Risque / scénario fictif. Une clé privilégiée est injectée dans une variable publique puis distribuée à chaque visiteur.

Correction attendue. Déplacer les usages privilégiés derrière une autorisation serveur. Révoquer toute valeur exposée. Vérifier les permissions des clés publiables.

Test et critère de réussite. Inspecter le build final et les réponses réseau. Avec une clé publiable de test, tenter uniquement des opérations interdites sur des données fictives : elles doivent échouer.

ASVS 5.0.0 : 13.3.1

Sources : [SUPABASE]

MODULE 02 • PROTÉGER LES ACCÈS TECHNIQUES ET LE CHEMIN QUI MÈNE AU DÉPLOIEMENT.

Secrets et chaîne de livraison

C09 Gestion et rotation des secrets

Application : Services avec identifiants techniques

Exigence. Limiter l'accès, la durée et l'étendue des secrets utilisés par chaque composant.

Preuves à chercher. Gestionnaire de secrets, droits de lecture, identités de service et procédure de rotation sans divulgation.

Risque / scénario fictif. Un agent de test compromis détient la même clé que la production et accède aux données réelles.

Correction attendue. Séparer environnements et services ; privilégier les identités temporaires lorsque disponibles. Restreindre les droits et organiser révocation et rotation.

Test et critère de réussite. Faire tourner un secret de préproduction. Vérifier ancienne valeur refusée, continuité attendue et impossibilité pour un service sans droit de lire le nouveau secret.

ASVS 5.0.0 : 13.3.2, 13.3.4

Sources : [SECRETS]

C10 Dépendances et provenance

Application : Application avec composants tiers

Exigence. Connaître les versions réellement livrées et traiter les avis qui les concernent.

Preuves à chercher. Fichier de verrouillage, inventaire des composants, dépendances indirectes, scripts CDN et avis officiels datés.

Risque / scénario fictif. Un paquet au nom proche d'une bibliothèque attendue exécute un script d'installation avec les droits du build.

Correction attendue. Vérifier provenance et scripts, verrouiller les versions, réduire les dépendances et définir un délai de traitement des avis selon le risque.

Test et critère de réussite. Reconstituer le build dans un environnement propre ; comparer les versions installées. Pour chaque avis, documenter version affectée, usage exposé, correction et retest.

ASVS 5.0.0 : 15.1.2, 15.2.1, 15.2.4

Sources : [SUPPLY]

MODULE 02 • PROTÉGER LES ACCÈS TECHNIQUES ET LE CHEMIN QUI MÈNE AU DÉPLOIEMENT.

Secrets et chaîne de livraison

C11 Droits de livraison et environnements

Application : Dépôt, CI/CD et hébergeur

Exigence. Empêcher une contribution non approuvée de lire des secrets ou de déployer en production.

Preuves à chercher. Rôles du dépôt, règles de branche, permissions des workflows, secrets accessibles et séparation production/préproduction.

Risque / scénario fictif. Une branche de contribution exécute un script qui reçoit les secrets de production durant les tests.

Correction attendue. Limiter les droits du pipeline et protéger les étapes privilégiées ; isoler les contributions non fiables. Activer une authentification renforcée des opérateurs.

Test et critère de réussite. Depuis un rôle de contribution fictif, vérifier l'absence de secrets et le refus du déploiement. Confirmer ensuite le parcours autorisé avec un opérateur habilité.

ASVS 5.0.0 : 13.2.2, 15.2.3

Sources : [CICD]

C12 Fichiers et interfaces techniques exposés

Application : Production et prévisualisations

Exigence. Exclure secrets, dépôts, sauvegardes, diagnostics et consoles du périmètre public.

Preuves à chercher. Racine servie, règles du serveur, routes de diagnostic, artefacts publiés et contrôles d'accès des prévisualisations.

Risque / scénario fictif. Un fichier .env ou une archive de sauvegarde reste dans le répertoire publié après une opération de maintenance.

Correction attendue. Publier une liste maîtrisée d'artefacts. Désactiver le debug, protéger les interfaces techniques et supprimer les fichiers inutiles.

Test et critère de réussite. Sur les domaines autorisés, vérifier les chemins sensibles connus du projet. Contrôler le corps reçu : une page d'accueil avec statut 200 ne prouve pas une fuite.

ASVS 5.0.0 : 13.4.1, 13.4.2, 13.4.5, 13.4.7 • API8:2023 API9:2023

Sources : [ASVS]

MODULE 03 • VÉRIFIER UNE IDENTITÉ SANS CRÉER UNE VOIE DE PRISE DE CONTRÔLE.

Authentification et comptes

C13 Stockage des mots de passe

Application : Authentification par mot de passe

Exigence. Rendre une fuite de la base difficile à convertir en mots de passe utilisables.

Preuves à chercher. Fonction de dérivation, paramètres, sels, migration des anciennes empreintes ou preuves du fournisseur d'identité.

Risque / scénario fictif. Une base copiée révèle des mots de passe en clair ou des empreintes rapides faciles à tester hors ligne.

Correction attendue. Employer une bibliothèque reconnue de hachage de mots de passe, par exemple Argon2id avec paramètres documentés ; proscrire chiffrement réversible et hachage rapide seul.

Test et critère de réussite. Créer deux comptes fictifs avec le même mot de passe : empreintes distinctes, connexion valide et mauvais mot de passe refusé. Mesurer le coût de vérification sous charge bornée.

ASVS 5.0.0 : 11.4.2

Sources : [PASSWORD]

C14 Politique de mot de passe et énumération

Application : Comptes avec connexion publique

Exigence. Écarter les secrets triviaux sans révéler les comptes existants ni gêner les gestionnaires.

Preuves à chercher. Règles serveur, liste de mots de passe compromis, formulaires, réponses de connexion et de récupération.

Risque / scénario fictif. Un tiers identifie les adresses inscrites à partir de messages distincts, puis concentre ses tentatives sur elles.

Correction attendue. Accepter les phrases longues, sans troncature ; autoriser le collage. Harmoniser les réponses avant authentification et contrôler les écarts de comportement.

Test et critère de réussite. Comparer compte existant et inexistant sur connexion et récupération, y compris codes et délais. Tester mot de passe long, caractères variés et mot de passe compromis.

ASVS 5.0.0 : 6.2.7, 6.2.8, 6.2.9, 6.3.8 • API2:2023

Sources : [AUTH]

MODULE 03 • VÉRIFIER UNE IDENTITÉ SANS CRÉER UNE VOIE DE PRISE DE CONTRÔLE.

Authentification et comptes

C15 Brute force et inscriptions abusives

Application : Connexion et inscription ouvertes

Exigence. Limiter les tentatives et créations automatisées sans blocage permanent facile d'un compte tiers.

Preuves à chercher. Compteurs partagés, politiques par compte, IP et périmètre global ; limites des e-mails et vérifications d'inscription.

Risque / scénario fictif. Un script distribue ses requêtes entre plusieurs instances pour contourner un compte uniquement en mémoire.

Correction attendue. Combiner temporisation, limites adaptées et signaux d'abus ; réserver les mécanismes supplémentaires aux parcours pertinents. Prévoir les IP d'entreprise partagées.

Test et critère de réussite. À faible volume, dépasser un seuil de test sur plusieurs instances. Vérifier refus effectif, absence d'envoi payant indu, retour à la normale et absence de verrouillage permanent.

ASVS 5.0.0 : 6.3.1, 2.4.1 • API2:2023 API6:2023

Sources : [AUTH]

C16 Authentification renforcée et secours

Application : Administration et actions sensibles

Exigence. Protéger les comptes privilégiés et empêcher le contournement par la récupération.

Preuves à chercher. Politique MFA, enrôlement, codes de secours, réauthentification, désactivation des facteurs et comptes de secours.

Risque / scénario fictif. Un administrateur active la MFA, mais un parcours alternatif lui donne une session complète après le seul mot de passe.

Correction attendue. Appliquer la même exigence à toutes les voies. Protéger les changements de facteurs, limiter les essais et journaliser les usages des accès de secours.

Test et critère de réussite. Avec un compte de test, essayer connexion alternative, code de secours rejoué et suppression de facteur sans réauthentification. Vérifier refus et notification attendue.

ASVS 5.0.0 : 6.3.3, 6.3.4, 6.4.4 • API2:2023

Sources : [MFA]

MODULE 03 • VÉRIFIER UNE IDENTITÉ SANS CRÉER UNE VOIE DE PRISE DE CONTRÔLE.

Authentification et comptes

C17 Récupération, invitations et changement d'adresse

Application : Comptes utilisateurs

Exigence. Lier chaque lien sensible au bon compte, au bon usage et à une durée limitée.

Preuves à chercher. Création et validation des jetons, origine des URL, consommation atomique, invalidation et notifications.

Risque / scénario fictif. Un lien d'invitation rejoué rattache un autre compte à une organisation après sa première utilisation.

Correction attendue. Créer des jetons imprévisibles à usage unique ; vérifier identité et destination. Éviter les URL dérivées d'un en-tête Host non fiable et invalider les liens obsolètes.

Test et critère de réussite. Tester jeton expiré, rejoué, présenté pour un autre compte et consommé simultanément. Un seul usage légitime réussit ; les autres n'altèrent aucun droit.

ASVS 5.0.0 : 6.4.3, 6.5.1 • API2:2023

Sources : [\[RESET\]](#)

C18 Connexion déléguée OAuth et OIDC

Application : Connexion via fournisseur externe

Exigence. Accepter seulement une réponse liée au parcours initié et à l'identité attendue.

Preuves à chercher. Bibliothèque utilisée, URI de retour, PKCE, state/nonce selon le protocole, validation issuer/audience et liaison des comptes.

Risque / scénario fictif. Deux fournisseurs emploient le même identifiant utilisateur et l'application fusionne leurs comptes sans vérifier l'émetteur.

Correction attendue. Utiliser une intégration maintenue, des retours stricts et une identité fondée sur émetteur et sujet. Ne pas fusionner les comptes sur une adresse non vérifiée.

Test et critère de réussite. Avec un fournisseur de test, rejouer un retour et modifier émetteur, audience ou contexte de session : refus. Confirmer ensuite une connexion légitime.

ASVS 5.0.0 : 10.1.2, 10.3.3, 10.5.4 • API2:2023

Sources : [\[OAUTH\]](#)

MODULE 04 • MAÎTRISER LA DURÉE, LES USAGES ET LA RÉVOCATION DES ACCÈS.

Sessions et jetons

C19 Cookies et stockage des jetons

Application : Applications avec session

Exigence. Limiter le vol des jetons et leur transmission hors du contexte prévu.

Preuves à chercher. En-têtes Set-Cookie réels, Domain/Path, Secure, HttpOnly, SameSite, stockage navigateur et transit des jetons.

Risque / scénario fictif. Un jeton de session est copié dans une URL puis conservé dans les journaux d'un intermédiaire.

Correction attendue. Employer un transport adapté ; pour les cookies de session, limiter portée et lecture par script. Documenter les risques si des jetons restent accessibles au JavaScript.

Test et critère de réussite. Inspecter le navigateur et le trafic HTTPS de test. Vérifier absence de jeton dans URL, logs et réponse publique ; confirmer les attributs des cookies sur le déploiement.

ASVS 5.0.0 : 3.3.1, 3.3.2, 3.3.4, 14.2.1 • API2:2023

Sources : [SESSION]

C20 Validation des JWT et assertions

Application : Jetons autonomes

Exigence. Ne jamais confondre décodage du contenu et validation cryptographique.

Preuves à chercher. Middleware de vérification, algorithmes autorisés, clés fiables, émetteur, audience, expiration et type de jeton.

Risque / scénario fictif. L'API lit un rôle dans un JWT décodé sans vérifier sa signature et traite le client comme administrateur.

Correction attendue. Vérifier avec une bibliothèque maintenue et des paramètres explicites. Ne pas télécharger une clé depuis une URL arbitraire fournie par le jeton.

Test et critère de réussite. Dans une fixture de test, présenter signature invalide, jeton expiré, mauvaise audience et mauvais type. Tous doivent être rejetés avant l'action protégée.

ASVS 5.0.0 : 9.1.1, 9.1.2, 9.1.3, 9.2.1, 9.2.2, 9.2.3 • API2:2023

Sources : [ASVS]

MODULE 04 • MAÎTRISER LA DURÉE, LES USAGES ET LA RÉVOCATION DES ACCÈS.

Sessions et jetons

C21 Création, rotation et fixation de session

Application : Sessions authentifiées

Exigence. Renouveler l'identité de session lors de l'authentification et des transitions sensibles.

Preuves à chercher. Générateur aléatoire, code de création, renouvellement et comportement des anciens identifiants.

Risque / scénario fictif. Un identifiant pré-authentification reste valable après connexion, ce qui conserve un accès à celui qui le connaissait.

Correction attendue. Utiliser le mécanisme éprouvé du framework et une génération cryptographique ; renouveler le jeton et neutraliser l'ancien selon le cycle prévu.

Test et critère de réussite. Enregistrer l'identifiant avant connexion, se connecter puis le rejouer séparément. L'ancien identifiant ne donne aucun accès authentifié ; le nouveau fonctionne.

ASVS 5.0.0 : 7.2.3, 7.2.4 • API2:2023

Sources : [SESSION]

C22 Expiration et réauthentification

Application : Sessions authentifiées

Exigence. Borner la durée d'accès et vérifier à nouveau l'identité pour les actions à enjeu.

Preuves à chercher. Durées absolue et d'inactivité côté serveur, renouvellement et règles des opérations sensibles.

Risque / scénario fictif. Un ancien onglet conserve indéfiniment l'accès à un compte parce que seule une minuterie du navigateur expire.

Correction attendue. Imposer les délais au serveur ; distinguer prolongation de session et authentification récente. Calibrer les durées au risque et aux usages.

Test et critère de réussite. Avec une horloge de test, dépasser chaque durée puis envoyer une requête directe. Attendre refus ou réauthentification ; vérifier que le brouillon reste récupérable sans fuite.

ASVS 5.0.0 : 7.1.1, 7.3.1, 7.3.2 • API2:2023

Sources : [ASVS]

MODULE 04 • MAÎTRISER LA DURÉE, LES USAGES ET LA RÉVOCATION DES ACCÈS.

Sessions et jetons

C23 Déconnexion et retrait des droits

Application : Sessions, comptes et organisations

Exigence. Faire cesser les accès retirés, y compris avec un jeton précédemment émis.

Preuves à chercher. Révocation serveur, état actif du compte, invalidation des droits en cache et liens entre sessions et organisation.

Risque / scénario fictif. Un salarié retiré d'une agence continue de consulter les documents avec son ancienne session.

Correction attendue. Contrôler l'état actuel et la portée des droits ; invalider les sessions concernées. Pour un jeton autonome, ajouter un mécanisme effectif de révocation.

Test et critère de réussite. Capturer une requête légitime de test ; rejouer après déconnexion, suspension puis retrait d'organisation. Vérifier absence de lecture et d'écriture indues sur toutes les voies.

ASVS 5.0.0 : 7.4.1, 7.4.2 • API2:2023 API1:2023

Sources : [ASVS]

C24 Renouvellement et jeu des refresh tokens

Application : Sessions avec renouvellement

Exigence. Empêcher un jeton de renouvellement ancien de recréer indéfiniment des accès.

Preuves à chercher. Durée absolue, rotation ou liaison cryptographique au client, stockage et détection de réutilisation.

Risque / scénario fictif. Un jeton volé reste utilisable après son remplacement et alimente une deuxième chaîne de sessions.

Correction attendue. Employer la stratégie du fournisseur : rotation avec détection du jeu ou jetons liés au client. Définir la révocation de la famille compromise.

Test et critère de réussite. Renouveler une session fictive puis rejouer l'ancien refresh token. Vérifier rejet et traitement documenté de la famille ; tester aussi les requêtes concurrentes légitimes.

ASVS 5.0.0 : 10.4.5, 10.4.8, 10.4.9 • API2:2023

Sources : [OAUTH]

MODULE 05 • CONTRÔLER CHAQUE ACTION, CHAQUE OBJET ET CHAQUE CHAMP.

Autorisations et séparation des clients

C25 Accès aux objets d'un autre utilisateur

Application : Données rattachées à un compte

Exigence. Vérifier la propriété ou le droit explicite sur chaque ressource, pour chaque opération.

Preuves à chercher. Routes de lecture, écriture, suppression, export et téléchargement ; filtre de propriétaire appliqué côté serveur.

Risque / scénario fictif. Le compte A remplace un identifiant dans une requête et reçoit le document privé de B.

Correction attendue. Centraliser les décisions d'accès et lier la requête au sujet authentifié. Un identifiant aléatoire ne remplace pas cette vérification.

Test et critère de réussite. Créer A et B avec leurs documents. Depuis A, viser celui de B par chaque route : aucune donnée ni modification. Depuis B, vérifier le fonctionnement légitime.

ASVS 5.0.0 : 8.2.2 • API1:2023

Sources : [AUTHZ]

C26 Rôles, administration et champs sensibles

Application : Applications avec plusieurs niveaux de droits

Exigence. Contrôler à la fois les fonctions accessibles et les propriétés modifiables.

Preuves à chercher. Routes administrateur, liste des champs acceptés, contrôles de rôle et réponses contenant des propriétés privées.

Risque / scénario fictif. Un membre ajoute role=admin ou modifie organization_id dans une mise à jour de profil.

Correction attendue. Définir les champs autorisés par action et calculer les attributs sensibles au serveur. Appliquer le contrôle même si le bouton est absent.

Test et critère de réussite. Appeler directement les fonctions privilégiées et envoyer un champ sensible inattendu. Vérifier refus ou ignorance sûre, absence de changement et journal d'audit adapté.

ASVS 5.0.0 : 8.2.1, 8.2.3, 15.3.1, 15.3.3 • API3:2023 API5:2023

Sources : [MASS]

MODULE 05 • CONTRÔLER CHAQUE ACTION, CHAQUE OBJET ET CHAQUE CHAMP.

Autorisations et séparation des clients

C27 Séparation des organisations et des caches

Application : Applications multiclients

Exigence. Conserver l'isolation dans les requêtes, les fichiers, les tâches et les résultats mis en cache.

Preuves à chercher. Propagation du tenant, clés de cache, files de travail, recherches, tableaux de bord et exports.

Risque / scénario fictif. Deux clients interrogent la même URL ; un cache partagé sert au second la réponse privée du premier.

Correction attendue. Inclure l'identité et les droits nécessaires dans la partition des données. Réévaluer l'autorisation à la livraison d'un résultat différé.

Test et critère de réussite. Alternier A et B sur une même ressource logique, à cache froid puis chaud. Inclure tâches asynchrones et changement d'agence ; aucune donnée étrangère ne doit apparaître.

ASVS 5.0.0 : 8.4.1, 14.2.2, 14.2.5 • API1:2023

Sources : [CACHE]

C28 Permissions de base et politiques par ligne

Application : Base ou stockage avec contrôle d'accès

Exigence. Vérifier les politiques avec le rôle réellement employé par chaque parcours.

Preuves à chercher. Droits SQL, politiques RLS lorsqu'applicables, identité du backend et droits des conteneurs de fichiers.

Risque / scénario fictif. Des politiques RLS existent, mais les tests utilisent seulement un rôle qui les contourne ; l'isolation annoncée n'est pas démontrée.

Correction attendue. Appliquer le moindre privilège. Tester rôles anon, authentifié et service ; avec PostgreSQL, considérer propriétaires, superutilisateurs et BYPASSRLS.

Test et critère de réussite. Avec chaque rôle réel de test, lire et écrire des lignes A/B, y compris insertion et changement de propriétaire. Vérifier les privilèges indépendamment des politiques.

ASVS 5.0.0 : 13.2.2 • API1:2023 API8:2023

Sources : [RLS] • [DATABASE]

MODULE 05 • CONTRÔLER CHAQUE ACTION, CHAQUE OBJET ET CHAQUE CHAMP.

Autorisations et séparation des clients

C29 Services, tâches et accès internes

Application : Workers, cron et fonctions techniques

Exigence. Authentifier les appels techniques et restreindre leurs droits au besoin réel.

Preuves à chercher. Identités de service, validation des messages, accès aux queues et routes de maintenance.

Risque / scénario fictif. Une fonction de nettoyage est accessible sans contrôle parce que son URL était supposée connue seulement du planificateur.

Correction attendue. Authentifier la source, limiter le réseau si adapté et contrôler les objets visés. Éviter un compte technique universel disposant de tous les droits.

Test et critère de réussite. Appeler une fonction de test sans identité, avec une autre identité puis avec la bonne. Seul le dernier cas réalise l'opération autorisée dans son périmètre.

ASVS 5.0.0 : 13.2.1, 13.2.2 • API5:2023

Sources : [AUTHZ]

C30 Défaillance des contrôles et accès obsolètes

Application : Toute fonction protégée

Exigence. Refuser une opération sensible lorsque son autorisation ne peut pas être établie.

Preuves à chercher. Gestion des erreurs du service d'identité, replis, cache de droits et comportement après changement de rôle.

Risque / scénario fictif. Une panne de vérification déclenche une valeur par défaut qui accorde l'accès au lieu de le refuser.

Correction attendue. Définir explicitement le comportement sûr en panne. Réserver les modes dégradés aux données et actions qui ne nécessitent pas le droit indisponible.

Test et critère de réussite. Simuler une erreur de lecture des droits puis une réponse invalide. Aucune opération protégée ne réussit ; les fonctions publiques et la reprise normale restent utilisables.

ASVS 5.0.0 : 16.5.3 • API5:2023

Sources : [ERROR]

MODULE 06 • EMPÊCHER LES DONNÉES REÇUES DE DEVENIR DES INSTRUCTIONS EXÉCUTABLES.

Entrées et navigateur

C31 Schémas et limites des entrées

Application : Toutes entrées non fiables

Exigence. Valider au serveur types, tailles, formats et cohérence métier.

Preuves à chercher. Schémas des corps, paramètres et en-têtes ; limites avant lecture complète et comportement sur champs dupliqués.

Risque / scénario fictif. Une valeur de quota négative ou une chaîne à la place d'un nombre contourne une vérification trop permissive.

Correction attendue. Utiliser des schémas explicites et une validation positive. Refuser les ambiguïtés sensibles, borner profondeur et taille ; ne pas se fier au formulaire.

Test et critère de réussite. Envoyer types incorrects, champs inconnus sensibles, corps trop grand et paramètres contradictoires. Vérifier refus stable avant effet métier et acceptation du cas valide.

ASVS 5.0.0 : 2.2.1, 2.2.2, 15.3.5, 15.3.7 • API3:2023

Sources : [\[INPUT\]](#)

C32 Injections SQL et requêtes de données

Application : Accès à une base ou un moteur de requêtes

Exigence. Séparer les données utilisateur de la structure exécutée par le moteur.

Preuves à chercher. Requêtes construites, SQL brut, fonctions de l'ORM, filtres NoSQL et champs de tri.

Risque / scénario fictif. Un filtre de recherche devient une condition de requête et élargit les résultats à des données privées.

Correction attendue. Paramétrer les valeurs et limiter explicitement noms de colonnes, opérateurs et directions de tri. Examiner aussi les échappatoires SQL brut de l'ORM.

Test et critère de réussite. Dans une base fictive, soumettre des caractères de contrôle et des filtres structurés inattendus. Aucune modification de structure, fuite de lignes ou erreur SQL publique.

ASVS 5.0.0 : 1.2.4

Sources : [\[SQL\]](#)

MODULE 06 • EMPÊCHER LES DONNÉES REÇUES DE DEVENIR DES INSTRUCTIONS EXÉCUTABLES.

Entrées et navigateur

C33 XSS et rendu HTML ou Markdown

Application : Contenu affiché dans le navigateur

Exigence. Traiter textes, imports et sorties IA comme des données non fiables.

Preuves à chercher. Points de rendu HTML, Markdown, attributs, URL et JavaScript ; configuration du nettoyage et de l'encodage.

Risque / scénario fictif. Un commentaire enregistré contient du contenu actif ensuite exécuté dans le navigateur d'un administrateur.

Correction attendue. Utiliser les rendus texte sûrs et un assainissement maintenu si HTML nécessaire. Adapter l'encodage au contexte ; interdire les protocoles d'URL dangereux.

Test et critère de réussite. Utiliser un marqueur de test local inoffensif dans chaque surface de rendu. Vérifier absence d'exécution et de requête externe, y compris après sauvegarde et export.

ASVS 5.0.0 : 1.2.1, 1.2.3, 1.3.1, 1.3.5

Sources : [XSS]

C34 CSRF et actions déclenchées par le navigateur

Application : Authentification envoyée automatiquement

Exigence. Empêcher un site tiers de déclencher une action protégée à l'insu du visiteur.

Preuves à chercher. Méthodes HTTP, mécanisme anti-CSRF, contrôle d'origine et types de corps acceptés ; paramètres SameSite.

Risque / scénario fictif. Une page tierce soumet une modification de compte en utilisant les cookies du visiteur connecté.

Correction attendue. Activer la protection adaptée du framework et vérifier les origines. Ne pas modifier d'état via GET ; ne pas confondre absence de lecture CORS et refus de l'action.

Test et critère de réussite. Depuis une origine de test distincte, soumettre l'action sans protection puis avec jeton erroné. Aucun effet ne doit survenir ; le parcours normal reste valide.

ASVS 5.0.0 : 3.5.1, 3.5.2, 3.5.3

Sources : [CSRF]

MODULE 06 • EMPÊCHER LES DONNÉES REÇUES DE DEVENIR DES INSTRUCTIONS EXÉCUTABLES.

Entrées et navigateur

C35 Commandes, modèles, XML et désérialisation

Application : Parseurs, conversions ou exécution dynamique

Exigence. Éviter qu'un contenu reçu commande le système, un moteur de modèle ou un parseur.

Preuves à chercher. Appels système, eval, moteurs de gabarit, options XML et formats de sérialisation.

Risque / scénario fictif. Un convertisseur reçoit un nom de fichier interprété comme une option ou un document XML demande une ressource extérieure.

Correction attendue. Éviter le shell et les modèles construits par l'utilisateur ; paramétrer les appels. Désactiver les entités XML externes et les formats de désérialisation dangereux.

Test et critère de réussite. Sur un service isolé, envoyer des fixtures inoffensives contenant métacaractères et référence externe vers un témoin local. Attendre refus ou données littérales, sans action externe.

ASVS 5.0.0 : 1.2.5, 1.3.7, 1.5.1, 1.5.2

Sources : [INJECTION] · [XML]

C36 Protections du navigateur et ressources tierces

Application : Frontend web

Exigence. Réduire les conséquences d'un contenu injecté ou d'une dépendance chargée à distance.

Preuves à chercher. En-têtes effectifs CSP, cadrage, type MIME et politique de référent ; scripts externes et intégrité.

Risque / scénario fictif. Une page privée est embarquée dans une interface trompeuse ou un script tiers modifié s'exécute avec ses privilèges.

Correction attendue. Définir une CSP adaptée, restreindre le cadrage et les ressources. Versionner les scripts ; utiliser SRI lorsque compatible et justifier les exceptions.

Test et critère de réussite. Contrôler les réponses du déploiement et essayer cadrage non autorisé et script non approuvé. Vérifier aussi connexion et paiement après changement de politique.

ASVS 5.0.0 : 3.4.3, 3.4.4, 3.4.5, 3.4.6, 3.6.1

Sources : [HEADERS]

MODULE 07 • LIMITER LES DESTINATIONS, LES PROTOCOLES ET LES ACCÈS EXPOSÉS.

API et communications

C37 CORS et origines autorisées

Application : API consommée depuis un navigateur

Exigence. N'autoriser les lectures interorigines privées que pour les clients prévus.

Preuves à chercher. Liste des origines, traitement des credentials et en-têtes des réponses normales et d'erreur.

Risque / scénario fictif. L'API recopie toute origine reçue avec les cookies autorisés et expose une réponse privée à un site tiers.

Correction attendue. Définir une liste exacte pour les ressources privées ; distinguer API publiques. CORS ne fournit aucune autorisation aux clients hors navigateur.

Test et critère de réussite. Tester origine admise, refusée et null avec navigateur. Vérifier que les réponses privées restent inaccessibles et que l'authentification serveur reste exigée en appel direct.

ASVS 5.0.0 : 3.4.2 • API8:2023

Sources : [REST]

C38 SSRF et récupération d'URL

Application : Import d'URL, aperçu, webhook sortant

Exigence. Restreindre les destinations réellement jointes, après résolution et redirection.

Preuves à chercher. Validation des schémas, DNS, IP IPv4/IPv6, ports et redirections ; règles de sortie réseau.

Risque / scénario fictif. Un lien public accepté redirige ensuite vers un service interne accessible depuis le serveur.

Correction attendue. Limiter destinations et protocoles selon l'usage ; bloquer les plages interdites, contrôler chaque saut et borner temps et volume. Préférer une sortie réseau contrôlée.

Test et critère de réussite. Employer des services témoins isolés pour simuler destination interdite et redirection. Vérifier qu'aucune connexion n'atteint le témoin interne ; ne pas viser de métadonnées réelles.

ASVS 5.0.0 : 1.3.6, 13.2.4, 15.3.2 • API7:2023

Sources : [SSRF]

MODULE 07 • LIMITER LES DESTINATIONS, LES PROTOCOLES ET LES ACCÈS EXPOSÉS.

API et communications

C39 TLS, certificats et en-têtes de proxy

Application : Services exposés et communications internes

Exigence. Protéger le transport et ne croire que les informations ajoutées par les intermédiaires approuvés.

Preuves à chercher. Terminaison TLS, validation des certificats sortants, chaîne de proxy et source de l'adresse IP.

Risque / scénario fictif. Un client forge un en-tête d'adresse pour contourner une limite IP ou une application ignore une erreur de certificat.

Correction attendue. Exiger TLS adapté, conserver la validation des certificats et configurer les proxys de confiance. Ne pas utiliser directement un en-tête fourni par le client.

Test et critère de réussite. Sur une destination de test à certificat invalide, attendre un refus. Falsifier les en-têtes de proxy et vérifier qu'ils ne changent ni identité ni droits ni compte.

ASVS 5.0.0 : 12.2.1, 12.3.2, 4.1.3 • API8:2023

Sources : [TLS]

C40 Routes, méthodes et versions d'API

Application : API REST ou équivalent

Exigence. Appliquer les mêmes exigences aux routes alternatives et aux opérations par lots.

Preuves à chercher. Inventaire des méthodes, anciennes versions, routes debug, fonctions cachées et limites des lots.

Risque / scénario fictif. La route principale est protégée, mais une ancienne route d'export renvoie encore les mêmes données sans contrôle.

Correction attendue. Retirer les versions inutiles, limiter les méthodes et appliquer les vérifications à chaque élément d'un lot. Documenter l'exposition volontaire.

Test et critère de réussite. Comparer l'inventaire aux routes réelles. Tester méthodes non prévues et lot mêlant objets autorisés/interdits : aucun objet interdit ne doit être traité.

ASVS 5.0.0 : 4.1.4, 13.4.5 • API9:2023 API1:2023

Sources : [REST]

MODULE 07 • LIMITER LES DESTINATIONS, LES PROTOCOLES ET LES ACCÈS EXPOSÉS.

API et communications

C41 GraphQL, WebSocket et flux longs

Application : Interfaces présentes dans le projet

Exigence. Contrôler complexité, messages et autorisations tout au long de la connexion.

Preuves à chercher. Coût et profondeur GraphQL, authentification WebSocket, origine, taille des messages et révocation des abonnements.

Risque / scénario fictif. Un abonnement temps réel continue de diffuser les données d'une agence après le retrait du membre.

Correction attendue. Borner la complexité et les messages ; vérifier droits par opération et à leur changement. Fermer ou restreindre les flux dont le droit a expiré.

Test et critère de réussite. Tester une requête au-delà de la limite et un abonnement A/B fictif. Retirer le droit pendant le flux : aucune nouvelle donnée protégée ne doit être transmise.

ASVS 5.0.0 : 4.3.1, 4.4.1, 4.4.2 • API4:2023 API1:2023

Sources : [WEBSOCKET]

C42 Réponses et intégrations de services tiers

Application : Tout appel à un prestataire

Exigence. Valider les données reçues sans leur accorder une confiance implicite.

Preuves à chercher. Schémas de réponse, destinations, délais, taille maximale, gestion des statuts et comptes techniques.

Risque / scénario fictif. Une réponse tierce invalide est traitée comme une réussite et active un droit ou injecte du contenu dans la page.

Correction attendue. Valider format et sens métier, borner lecture et redirections et isoler les traitements risqués. Associer la réponse à l'opération attendue.

Test et critère de réussite. Simuler statut 200 avec JSON invalide, champs manquants, corps trop grand et réponse tardive. Aucun faux succès, droit indu ou lecture sans limite.

ASVS 5.0.0 : 2.2.1, 13.2.6 • API10:2023

Sources : [API10]

MODULE 08 • RÉDUIRE LES RISQUES LORS DE LA RÉCEPTION, DU STOCKAGE ET DE LA SUPPRESSION.

Fichiers et cycle de vie des données

C43 Validation des fichiers reçus

Application : Uploads ou imports locaux

Exigence. Contrôler formats et volumes avant les étapes coûteuses.

Preuves à chercher. Limites avant lecture, validation réelle du type, bibliothèques de parsing et comportement sur contenu discordant.

Risque / scénario fictif. Un document annoncé comme image est envoyé à un parseur qui accepte un autre format plus dangereux.

Correction attendue. Définir types admis et tailles ; vérifier le contenu sans croire extension ou MIME déclaré. Isoler le parseur et maintenir ses dépendances.

Test et critère de réussite. Tester fichier admis, format discordant, fichier vide et taille excessive. Le refus survient avant stockage définitif ou traitement lourd ; le cas normal reste lisible.

ASVS 5.0.0 : 5.1.1.1, 5.2.1, 5.2.2 • API4:2023

Sources : [\[UPLOAD\]](#)

C44 Archives, chemins et traitement des fichiers

Application : Décompression et conversions

Exigence. Borner le volume décompressé et empêcher l'écriture hors de la zone prévue.

Preuves à chercher. Limites sur archives, dimensions d'images, durée CPU, chemins normalisés, liens symboliques et droits du processus.

Risque / scénario fictif. Une archive contient un chemin sortant du dossier temporaire ou exige une décompression disproportionnée.

Correction attendue. Limiter nombre d'entrées, volume et ratio ; ignorer les chemins utilisateurs, refuser les liens non nécessaires et isoler la conversion sans droits superflus.

Test et critère de réussite. Utiliser de petites fixtures de traversée et des limites abaissées pour simuler l'expansion excessive. Vérifier arrêt borné et absence de fichier en dehors du répertoire de test.

ASVS 5.0.0 : 5.2.3, 5.2.5, 5.2.6, 5.3.2, 5.3.3 • API4:2023

Sources : [\[UPLOAD\]](#)

MODULE 08 • RÉDUIRE LES RISQUES LORS DE LA RÉCEPTION, DU STOCKAGE ET DE LA SUPPRESSION.

Fichiers et cycle de vie des données

C45 Stockage privé, partage et téléchargement

Application : Fichiers conservés ou distribués

Exigence. Conserver les droits jusqu'au téléchargement et neutraliser le contenu actif.

Preuves à chercher. Politiques du stockage, autorisation de téléchargement, durée et révocation des liens, en-têtes et quarantaine.

Risque / scénario fictif. Un export privé reste accessible par son URL après retrait des droits ou un fichier reçu s'exécute comme une page de l'application.

Correction attendue. Servir les fichiers avec le contexte approprié, sans exécution serveur ; limiter les liens signés. Scanner ou neutraliser les contenus selon les risques et formats.

Test et critère de réussite. Tenter accès anonyme, accès B sur fichier A et lien expiré. Pour la révocation, mesurer le délai réellement garanti. Vérifier téléchargement sûr d'un fichier de test actif.

ASVS 5.0.0 : 5.3.1, 5.4.1, 5.4.3 • API1:2023

Sources : [UPLOAD]

C46 Chiffrement et protection des clés

Application : Données sensibles stockées

Exigence. Adapter la protection aux données, y compris les sauvegardes et les métadonnées.

Preuves à chercher. Options de chiffrement effectives, accès aux clés, bibliothèque, gestion des nonces et séparation des permissions.

Risque / scénario fictif. Une sauvegarde chiffrée est copiée avec sa clé au même endroit et l'accès compromis révèle les deux.

Correction attendue. Utiliser les mécanismes éprouvés du fournisseur ou des bibliothèques reconnues ; limiter les accès aux clés et documenter leur récupération et leur rotation.

Test et critère de réussite. Vérifier la configuration de stockage et un échantillon fictif restauré. Avec une identité dépourvue du droit de clé, vérifier le refus de déchiffrement.

ASVS 5.0.0 : 11.2.1, 11.3.3, 14.2.4

Sources : [CRYPTO]

MODULE 08 • RÉDUIRE LES RISQUES LORS DE LA RÉCEPTION, DU STOCKAGE ET DE LA SUPPRESSION.

Fichiers et cycle de vie des données

C47 Rétenion, effacement et prestataires

Application : Données personnelles ou confidentielles

Exigence. Définir les copies conservées, leurs destinataires et leur suppression vérifiable.

Preuves à chercher. Durées, flux tiers, index, caches, sauvegardes et conditions de traitement applicables aux comptes réellement utilisés.

Risque / scénario fictif. Un document supprimé reste disponible dans un index de recherche ou est renvoyé par un cache.

Correction attendue. Propager la suppression aux copies actives ; gérer les sauvegardes par expiration et procédure de restauration. Réduire les données envoyées et vérifier les options des fournisseurs.

Test et critère de réussite. Supprimer un document fictif puis rechercher son marqueur dans les systèmes actifs. Documenter les délais résiduels et vérifier que la restauration réapplique les suppressions prévues.

ASVS 5.0.0 : 14.2.3, 14.2.7

Sources : [PRIVACY]

C48 Brouillons, exports et poste partagé

Application : Interface avec données privées

Exigence. Éviter les fuites après déconnexion et les interprétations actives lors des exports.

Preuves à chercher. Cache navigateur, stockage local, nettoyage du DOM, CSV/tableurs et données présentes dans les fichiers exportés.

Risque / scénario fictif. Sur un ordinateur partagé, le compte suivant retrouve le brouillon privé précédent ; une cellule exportée est interprétée comme formule.

Correction attendue. Définir la conservation locale, purger au changement de compte et exporter des valeurs sûres. Ne pas exposer de colonnes sensibles inutiles.

Test et critère de réussite. Se déconnecter puis ouvrir une autre session et revenir en arrière. Exporter un marqueur commençant comme une formule et vérifier son affichage comme texte dans le tableur cible.

ASVS 5.0.0 : 14.3.1, 14.3.2, 1.2.10

Sources : [EXPORT]

MODULE 09 • PRÉSERVER LA COHÉRENCE MALGRÉ LES DOUBLES ACTIONS ET LA CONCURRENCE.

Intégrité et paiements

C49 Transitions et règles métier

Application : Parcours à étapes ou à droits variables

Exigence. Imposer l'ordre et les conditions des opérations au serveur.

Preuves à chercher. États autorisés, vérifications d'abonnement, activation, résiliation et droits calculés.

Risque / scénario fictif. Le navigateur appelle directement l'étape finale et active une fonction réservée sans remplir les conditions précédentes.

Correction attendue. Définir les transitions admises et vérifier chaque précondition au serveur. Ne pas accorder un droit à partir d'un simple écran de succès.

Test et critère de réussite. Sur données fictives, sauter une étape, revenir en arrière et rejouer une transition. L'état final reste cohérent et aucune fonction non acquise n'est ouverte.

ASVS 5.0.0 : 2.3.1, 2.3.2 • API6:2023

Sources : [BUSINESS]

C50 Écritures concurrentes et cohérence

Application : Données modifiables ou ressources limitées

Exigence. Éviter pertes de mises à jour et dépassements lors d'opérations simultanées.

Preuves à chercher. Transactions, contraintes d'unicité, versions, écritures conditionnelles et traitement des conflits.

Risque / scénario fictif. Deux requêtes lisent le même solde, dépensent chacune le dernier crédit et écrasent leurs résultats.

Correction attendue. Rendre contrôle et mutation atomiques ; utiliser transactions ou comparaison de version adaptée au stockage. Résoudre les conflits depuis un état frais.

Test et critère de réussite. Avec un seul crédit fictif, lancer un petit groupe d'opérations concurrentes. Une seule consommation réussit ; les écritures acquittées restent présentes après relecture.

ASVS 5.0.0 : 2.3.3, 2.3.4, 15.4.2 • API6:2023

Sources : [BUSINESS]

MODULE 09 • PRÉSERVER LA COHÉRENCE MALGRÉ LES DOUBLES ACTIONS ET LA CONCURRENCE.

Intégrité et paiements

C51 Idempotence et réponse perdue

Application : Écritures, tâches et appels facturés

Exigence. Reconnaître une répétition de la même opération sans en doubler les effets.

Preuves à chercher. Clé d'opération, liaison au compte et au contenu, réservation atomique et consultation d'état.

Risque / scénario fictif. La connexion coupe après une écriture réussie ; le client réessaie et crée une deuxième commande.

Correction attendue. Associer une clé stable à l'opération logique. Retrouver son état ou son résultat ; refuser une même clé avec un contenu différent.

Test et critère de réussite. Répéter en parallèle une opération fictive avec la même clé, puis modifier son contenu. Attendre un seul effet et un conflit explicite pour le second contenu.

ASVS 5.0.0 : 2.3.3, 15.4.2 • API6:2023

Sources : [IDEMP]

C52 Prix, paiement et activation

Application : Application avec paiement

Exigence. Calculer le montant au serveur et activer le droit seulement après vérification fiable.

Preuves à chercher. Catalogue serveur, commande et propriétaire, identifiant fournisseur, devise, statut et conditions d'activation.

Risque / scénario fictif. Un client modifie le prix envoyé ou visite directement une URL de retour pour obtenir un abonnement.

Correction attendue. Créer une commande contrôlée et vérifier côté serveur montant, devise, propriétaire et statut attendu. Prévoir refus, attente, remboursement et renouvellement.

Test et critère de réussite. Dans le mode test du fournisseur, altérer le montant client, visiter le retour sans paiement et reprendre un paiement interrompu. Aucune activation induite ni double commande.

ASVS 5.0.0 : 2.2.3, 2.3.1 • API6:2023

Sources : [PAYMENT]

MODULE 09 • PRÉSERVER LA COHÉRENCE MALGRÉ LES DOUBLES ACTIONS ET LA CONCURRENCE.

Intégrité et paiements

C53 Webhooks signés, rejoués ou désordonnés

Application : Notifications entrantes

Exigence. Authentifier la notification et appliquer une seule transition métier cohérente.

Preuves à chercher. Vérification sur corps brut, fraîcheur selon le fournisseur, unicité d'événement et gestion de l'ordre.

Risque / scénario fictif. Un événement est reçu deux fois ou un ancien événement réactive un abonnement déjà résilié.

Correction attendue. Vérifier signature et contexte, accepter durablement puis traiter. Dédupliquer par événement et protéger aussi l'unicité de l'effet métier.

Test et critère de réussite. Avec des événements de test, vérifier signature erronée, corps altéré, doublon et ordre inversé. Attendre refus des faux et état final conforme à la source fiable.

ASVS 5.0.0 : 2.3.3 • API10:2023

Sources : [WEBHOOK]

C54 Enregistrement, conflits et états d'interface

Application : Parcours asynchrones

Exigence. Ne confirmer que les résultats connus et préserver la cohérence des reprises.

Preuves à chercher. Contrat de réponse, état persistant de l'opération, gestion de conflit, reprise du brouillon et messages accessibles.

Risque / scénario fictif. L'écran annonce un enregistrement après un timeout alors que l'écriture a échoué ; l'utilisateur ferme sa seule copie.

Correction attendue. Distinguer succès, erreur et résultat inconnu. Confirmer après écriture durable et proposer une vérification d'état avant toute répétition.

Test et critère de réussite. Simuler réponse perdue, JSON illisible, conflit entre onglets et session expirée. Vérifier messages exacts, absence de perte silencieuse et navigation clavier utilisable.

ASVS 5.0.0 : 2.3.3

Sources : [STATUS]

MODULE 10 • ENCADRER LES ACCÈS DU MODÈLE ET LE COÛT DE CHAQUE TRAITEMENT.

IA, outils et dépenses

C55 Injection de prompt directe et indirecte

Application : Application utilisant un modèle

Exigence. Empêcher les contenus non fiables de décider des droits ou des actions de l'application.

Preuves à chercher. Frontières entre instructions, documents, sorties d'outils et règles d'autorisation ; tests adversariaux conservés.

Risque / scénario fictif. Un document importé ordonne au modèle d'envoyer un autre document privé vers une destination extérieure.

Correction attendue. Traiter les documents comme données hostiles, limiter les destinations et faire appliquer les droits hors du modèle. Un filtre de prompt seul ne garantit pas la protection.

Test et critère de réussite. Dans un environnement isolé, placer une consigne contradictoire dans un document fictif. Vérifier absence d'accès étranger, d'envoi externe ou d'action interdite malgré la réponse du modèle.

Méthode IDV / complément métier • LLM01:2025

Sources : [PROMPT]

C56 Outils et actions des agents

Application : IA appelant des outils ou connecteurs

Exigence. Limiter les actions à des capacités explicitement autorisées pour le compte courant.

Preuves à chercher. Catalogue d'outils, paramètres validés, identités, scopes, limites de boucle et étapes d'approbation métier.

Risque / scénario fictif. Un assistant disposant d'un accès global au stockage exécute une suppression demandée dans un contenu piégé.

Correction attendue. Restreindre outils et données au besoin, vérifier chaque appel au serveur et demander une validation explicite pour les actions sensibles définies. Isoler l'exécution de code.

Test et critère de réussite. Faire proposer une action hors périmètre par un modèle simulé. Le serveur la refuse même si l'IA l'affirme autorisée ; une action normale de test reste possible.

Méthode IDV / complément métier • LLM06:2025

Sources : [AGENT]

MODULE 10 • ENCADRER LES ACCÈS DU MODÈLE ET LE COÛT DE CHAQUE TRAITEMENT.

IA, outils et dépenses

C57 RAG, index et séparation documentaire

Application : Recherche ou contexte documentaire IA

Exigence. Filtrer les documents selon les droits avant leur entrée dans le contexte du modèle.

Preuves à chercher. Ingestion, métadonnées de propriétaire, filtres de recherche, caches, citations et suppression des index.

Risque / scénario fictif. Une recherche retrouve le document de B dans un index partagé et l'injecte dans la réponse destinée à A.

Correction attendue. Appliquer les droits à la recherche et à la récupération. Protéger la provenance, les mises à jour et l'effacement ; ne pas confier l'isolation à une consigne du modèle.

Test et critère de réussite. Indexer deux documents fictifs avec marqueurs distincts. Avec A, inspecter aussi le contexte récupéré, pas seulement la réponse : aucune donnée de B. Refaire après retrait des droits.

Méthode IDV / complément métier • LLM08:2025

Sources : [RAG]

C58 Sorties IA, confidentialité et décisions

Application : Toute sortie d'un modèle

Exigence. Valider le résultat avant affichage, exécution, stockage ou décision sensible.

Preuves à chercher. Schéma des sorties, nettoyage du rendu, contrôle des destinations et données envoyées au fournisseur.

Risque / scénario fictif. Une sortie IA ressemble à une commande valide et l'application l'exécute sans vérifier son objet ni ses arguments.

Correction attendue. Considérer le résultat comme non fiable. Valider structure, droits et règles métier ; ne pas utiliser le prompt système pour stocker des secrets ou remplacer un contrôle.

Test et critère de réussite. Simuler HTML actif, commande interdite et données hors schéma. Attendre refus ou rendu neutre. Contrôler les journaux et requêtes fournisseur avec des données marquées fictives.

Méthode IDV / complément métier • LLM02:2025 LLM05:2025 LLM07:2025

Sources : [OUTPUT]

MODULE 10 • ENCADRER LES ACCÈS DU MODÈLE ET LE COÛT DE CHAQUE TRAITEMENT.

IA, outils et dépenses

C59 Quotas, concurrence et budget atomique

Application : Appels payants IA ou autres services

Exigence. Vérifier l'abonnement et réserver un coût borné avant de déclencher la dépense.

Preuves à chercher. Registre partagé, réservation atomique, plafonds utilisateur/organisation/global et inventaire des coûts auxiliaires.

Risque / scénario fictif. Plusieurs requêtes passent ensemble un contrôle de solde non atomique et lancent davantage d'appels que le budget disponible.

Correction attendue. Borner entrée, sortie, étapes et outils. Réserver une enveloppe prudente incluant reprises et modèles de secours ; réconcilier ensuite, sans libérer une consommation inconnue.

Test et critère de réussite. Avec fournisseur simulé et budget d'un seul traitement, lancer des demandes simultanées sur plusieurs instances. Compter les appels : seule la dépense autorisée doit partir.

ASVS 5.0.0 : 2.3.4, 2.4.1 • API4:2023 LLM10:2025

Sources : [CONSUMPTION]

C60 Coupe-circuit financier et plafonds fournisseur

Application : Services facturés à l'usage

Exigence. Arrêter les nouvelles dépenses avant une dérive incontrôlée.

Preuves à chercher. Réglages réels des comptes, portée des plafonds, délais d'application, alertes et commande d'arrêt testée.

Risque / scénario fictif. Une alerte de budget envoie un e-mail mais les générations, e-mails ou traitements continuent d'être facturés.

Correction attendue. Distinguer alertes et limites bloquantes, vérifier leur comportement contractuel et technique. Garder une marge pour les opérations en vol et protéger tous les postes facturés.

Test et critère de réussite. Abaisser un plafond applicatif de test et simuler le dépassement : aucun nouvel appel payant. Vérifier séparément le plafond fournisseur sans dépenser le budget réel ; sinon UNKNOWN.

ASVS 5.0.0 : 2.4.1 • API4:2023 LLM10:2025

Sources : [CONSUMPTION]

MODULE 11 • DONNER UNE LIMITE AUX RESSOURCES, AUX ATTENTES ET AUX REPRISES.

Disponibilité et maîtrise de la charge

C61 Débit et automatisation des parcours

Application : Routes publiques et authentifiées

Exigence. Limiter les abus par opération sans se contenter d'un compteur IP global.

Preuves à chercher. Règles par route, compte, organisation et application ; stockage des compteurs et réponse de dépassement.

Risque / scénario fictif. Un compte appelle massivement une fonction d'export ou de recherche qui reste sous la limite générale de connexion.

Correction attendue. Différencier coûts et usages ; combiner débit, volume et signaux d'abus. Faire appliquer les limites sur toutes les instances et routes équivalentes.

Test et critère de réussite. Avec seuils de test faibles, dépasser chaque limite via deux instances et routes alternatives. Vérifier refus avant traitement, compteurs cohérents et récupération après la fenêtre.

ASVS 5.0.0 : 2.4.1 • API4:2023 API6:2023

Sources : [API4]

C62 Files, traitements et surcharge

Application : Opérations longues ou coûteuses

Exigence. Limiter les tâches en attente et en cours ; préserver les parcours essentiels.

Preuves à chercher. Capacité des files, limites de workers, admission, priorités, expiration des tâches et politique de rejet.

Risque / scénario fictif. Une hausse de trafic crée une file illimitée qui absorbe mémoire et délai, puis empêche les utilisateurs de se connecter.

Correction attendue. Définir des capacités bornées, une réponse de saturation et une répartition équitable. Éviter de démarrer plus de travail que les dépendances n'en absorbent.

Test et critère de réussite. Sur une préproduction avec prestataires simulés, dépasser progressivement la capacité. Attendre rejet maîtrisé, mémoire stable et reprise sans avalanche lorsque la charge baisse.

ASVS 5.0.0 : 15.1.3, 15.2.2 • API4:2023

Sources : [OVERLOAD]

MODULE 11 • DONNER UNE LIMITE AUX RESSOURCES, AUX ATTENTES ET AUX REPRISES.

Disponibilité et maîtrise de la charge

C63 Délais, flux et annulation

Application : Réseau, IA et traitements asynchrones

Exigence. Donner une fin à chaque attente et libérer les ressources devenues inutiles.

Preuves à chercher. Délais de connexion, premier contenu, inactivité et durée totale ; propagation de l'annulation et nettoyage.

Risque / scénario fictif. Un fournisseur garde une connexion ouverte sans résultat, immobilisant un worker et une réservation de budget.

Correction attendue. Définir des délais par phase et une durée globale. Propager l'annulation lorsque supportée ; distinguer arrêt demandé, confirmé et consommation inconnue.

Test et critère de réussite. Simuler fournisseur silencieux, flux bloqué et fermeture client. Vérifier délai borné, libération des ressources et conservation prudente du budget si la facturation reste inconnue.

ASVS 5.0.0 : 13.1.3, 13.2.6 • API4:2023

Sources : [RETRY]

C64 Reprises et propagation des pannes

Application : Appels distants avec retries

Exigence. Éviter de multiplier erreurs, effets métier et dépenses en cas de panne.

Preuves à chercher. Nombre maximal d'essais, budget cumulé, délais croissants, dispersion et coupe-circuit.

Risque / scénario fictif. Chaque couche réessaie plusieurs fois ; une panne unique engendre une multiplication d'appels payants et de connexions.

Correction attendue. Centraliser la stratégie, retenter seulement les erreurs compatibles et tenir compte de Retry-After. Associer idempotence et coupe-circuit ; ne pas retenter un budget épuisé.

Test et critère de réussite. Simuler erreurs temporaires, refus permanent et panne durable. Mesurer le nombre total d'appels et le temps final ; aucun doublon métier ni boucle sans fin.

ASVS 5.0.0 : 13.1.3, 16.5.2 • API4:2023

Sources : [RETRY]

MODULE 11 • DONNER UNE LIMITE AUX RESSOURCES, AUX ATTENTES ET AUX REPRISES.

Disponibilité et maîtrise de la charge

C65 Saturation de la base et du stockage

Application : Base de données ou service persistant

Exigence. Borner connexions, requêtes et volumes pour éviter l'épuisement partagé.

Preuves à chercher. Pool par instance, capacité globale, index, pagination, requêtes lentes, délais et libération des connexions.

Risque / scénario fictif. L'augmentation des instances multiplie leurs pools et dépasse la capacité de la base malgré une charge applicative modérée.

Correction attendue. Dimensionner globalement, paginer et indexer les parcours observés. Fixer limites de requête, stockage et concurrence ; corriger les fuites de connexions.

Test et critère de réussite. Avec un jeu synthétique représentatif, mesurer requêtes longues, connexions et stockage pendant une montée bornée. Vérifier retour au niveau initial et absence de perte d'écriture.

ASVS 5.0.0 : 13.1.2, 13.2.6 • API4:2023

Sources : [DATABASE] • [DOS]

C66 Charge, pics et récupération

Application : Avant exposition ou croissance sensible

Exigence. Mesurer un niveau de service et ses limites sur une charge explicitement définie.

Preuves à chercher. Profil d'usage, données synthétiques, objectifs p95/p99, seuils d'arrêt et rapport de récupération.

Risque / scénario fictif. Un essai compte seulement les utilisateurs inscrits ; un pic de générations simultanées dépasse ensuite la capacité réelle.

Correction attendue. Définir activités, tailles et concurrence ; tester montée, pic et endurance dans le périmètre autorisé. Simuler les fournisseurs pour la masse, borner les vrais appels.

Test et critère de réussite. Exécuter le profil documenté puis réduire la charge. Vérifier objectifs convenus, absence de perte d'écriture acquittée et de fuite entre comptes, puis récupération dans le délai fixé.

ASVS 5.0.0 : 15.2.2 • API4:2023

Sources : [OVERLOAD]

MODULE 12 • DÉTECTER LES INCIDENTS, PROTÉGER LES TRACES ET RESTAURER LE SERVICE.

Production, détection et reprise

C67 Journaux utiles et non divulguants

Application : Tous composants

Exigence. Tracer les événements nécessaires sans fabriquer un nouveau dépôt de secrets.

Preuves à chercher. Schéma des logs, accès, rétention, horodatage, corrélation et filtrage avant export.

Risque / scénario fictif. Une erreur écrit cookies, clés ou documents complets dans un service de logs accessible à trop de personnes.

Correction attendue. Journaliser actions, identifiants internes, statut et durée ; masquer les données sensibles. Protéger intégrité, accès et transfert des traces ; prévenir les injections de logs.

Test et critère de réussite. Déclencher connexion refusée, modification de droit et erreur avec marqueurs fictifs. Vérifier corrélation, absence des valeurs sensibles et refus de lecture par un compte non habilité.

ASVS 5.0.0 : 16.2.1, 16.2.5, 16.4.1, 16.4.2

Sources : [LOGGING]

C68 Erreurs publiques et comportement sûr

Application : API et interface

Exigence. Fournir un message utile sans exposer les détails internes ni transformer une panne en succès.

Preuves à chercher. Gestionnaires d'erreur, codes métier, identifiant de requête et capture des exceptions non prévues.

Risque / scénario fictif. Une erreur renvoie requête SQL, chemin serveur ou jeton ; un échec de validation laisse quand même la transaction se poursuivre.

Correction attendue. Séparer message public et diagnostic interne expurgé. Conserver un identifiant de corrélation, une issue explicite et une gestion sûre des exceptions.

Test et critère de réussite. Simuler données invalides, panne de base et erreur inattendue. Vérifier absence de détails sensibles, absence d'effet indu et présence d'une trace interne exploitable.

ASVS 5.0.0 : 16.5.1, 16.5.3, 16.5.4 • API8:2023

Sources : [ERROR]

MODULE 12 • DÉTECTER LES INCIDENTS, PROTÉGER LES TRACES ET RESTAURER LE SERVICE.

Production, détection et reprise

C69 Monitoring et alertes reçues

Application : Service en production

Exigence. Détecter panne, abus et dérive financière avec une personne en mesure d'agir.

Preuves à chercher. Sondes externes, métriques de latence et d'erreur, seuils de dépenses, destinataires et procédures.

Risque / scénario fictif. L'accueil renvoie 200 alors que toutes les écritures échouent ; aucune alerte ne vérifie un parcours métier.

Correction attendue. Surveiller aussi enregistrement, files, dépendances et coût. Utiliser un espace synthétique dédié et un canal de secours ; éviter les sondes destructrices.

Test et critère de réussite. En préproduction, déclencher une panne de parcours et un seuil de dépenses simulé. Vérifier réception par le titulaire et le suppléant, délai mesuré et procédure suivie.

ASVS 5.0.0 : 16.4.3

Sources : [MONITOR]

C70 Sauvegarde et restauration démontrée

Application : Données ou configuration non reproductibles

Exigence. Récupérer un ensemble cohérent dans les délais de perte et de reprise acceptés.

Preuves à chercher. Périmètre des copies, chiffrement, séparation des accès, alertes d'échec, RPO/RTO et compte rendu de restauration.

Risque / scénario fictif. Les sauvegardes existent, mais les clés manquent ou les droits des documents ne sont pas restaurés avec les fichiers.

Correction attendue. Sauvegarder données, droits et paramètres indispensables ; protéger une copie contre la suppression courante. Programmer des restaurations isolées et mesurées.

Test et critère de réussite. Restaurer une copie autorisée dans un environnement isolé. Vérifier comptes, rattachements et fichiers ; mesurer RPO/RTO et empêcher la réactivation de sessions révoquées.

Méthode IDV / complément métier

Sources : [BACKUP]

MODULE 12 • DÉTECTER LES INCIDENTS, PROTÉGER LES TRACES ET RESTAURER LE SERVICE.

Production, détection et reprise

C71 Artefact, migrations et retour arrière

Application : Chaque livraison sensible

Exigence. Livrer la version testée et savoir retrouver un état exploitable après échec.

Preuves à chercher. Empreinte du build, manifeste déployé, migrations, compatibilité des schémas et procédure de retour.

Risque / scénario fictif. Un retour au code précédent échoue parce qu'une migration a supprimé une colonne dont il dépend.

Correction attendue. Tester les migrations sur copie isolée ; prévoir compatibilité, restauration ou procédure inverse validée. Séparer retour du code et retour des données.

Test et critère de réussite. Déployer l'artefact identifié en préproduction, appliquer la migration puis exercer le plan de retour. Vérifier intégrité, droits et compatibilité sans annoncer un rollback non testé.

Méthode IDV / complément métier • API8:2023

Sources : [CICD]

C72 Incident, clôture et réévaluation

Application : Toute ouverture publique

Exigence. Désigner qui décide, intervient et confirme les corrections sur la version cible.

Preuves à chercher. Contacts, procédure de révocation, arrêt des dépenses, conservation des traces, registre des risques et retests.

Risque / scénario fictif. Une clé fuit, mais personne ne sait arrêter les appels ni identifier la version compromise ; le rapport ancien reste présenté comme valable.

Correction attendue. Préparer une procédure courte d'incident et une décision de mise en ligne signée. Retester les correctifs et les UNKNOWN déterminants ; fixer une prochaine revue.

Test et critère de réussite. Faire un exercice fictif : alerte, qualification, arrêt, révocation, vérification et reprise. Mesurer les délais et consigner les écarts sans supprimer les traces utiles.

Méthode IDV / complément métier

Sources : [INCIDENT]

RECETTES 01 À 03

Des essais qui traversent plusieurs contrôles

T01 Isolation de deux comptes

Contrôles : C25-C28, C45, C57

Préparer. Deux utilisateurs A/B, chacun avec un document marqué ; deux organisations si applicables.

Exécuter. Rejouer avec A les lectures, modifications, suppressions, exports et téléchargements visant B. Répéter sur cache chaud et sur résultat asynchrone.

Prouver. A ne reçoit aucune donnée protégée et ne modifie rien. B conserve son accès légitime. Examiner aussi contexte RAG et objets de stockage.

T02 Révocation réellement appliquée

Contrôles : C23-C24, C41

Préparer. Session valide, requête enregistrée et éventuel flux de données ouvert.

Exécuter. Déconnecter, suspendre, retirer l'organisation ou réduire le rôle dans des essais distincts ; rejouer la requête et renouveler le jeton.

Prouver. Les droits retirés ne sont plus utilisables. Mesurer tout délai résiduel et le qualifier ; aucun nouveau message privé ne passe dans le flux.

T03 Dernier crédit et concurrence

Contrôles : C50, C59-C61

Préparer. Budget fictif d'une opération, deux instances, fournisseur simulé avec compteur.

Exécuter. Lancer un petit lot simultané avec des clés d'opération distinctes, puis vérifier solde, réservations et appels.

Prouver. Un seul traitement autorisé, aucune dépense supplémentaire, état cohérent après réconciliation et après erreur simulée du registre.

RECETTES 04 À 06

Des essais qui traversent plusieurs contrôles

T04 Réponse perdue et idempotence

Contrôles : C51-C54

Préparer. Opération d'écriture fictive avec identifiant stable ; réponse interceptable en test.

Exécuter. Laisser réussir l'écriture puis perdre sa réponse. Relancer avec même clé et même contenu, ensuite avec un contenu différent.

Prouver. Un seul effet métier ; résultat existant ou état consultable. Contenu différent refusé ; l'interface ne confirme pas un état inconnu.

T05 Notification de paiement

Contrôles : C52-C53

Préparer. Mode test du fournisseur ; commande et événement de paiement fictifs.

Exécuter. Présenter faux événement, corps modifié, doublon puis événements hors ordre. Visiter aussi la page de retour sans paiement.

Prouver. Aucun droit indu ni double crédit. État final rapproché de la commande et du statut fiable du fournisseur.

T06 Contenu actif dans les rendus

Contrôles : C33, C35, C48, C58

Préparer. Marqueurs inoffensifs, navigateur isolé et stockage fictif.

Exécuter. Parcourir création, édition, rendu Markdown, aperçu, export et réponse IA simulée avec des entrées hors format.

Prouver. Pas d'exécution ni d'appel extérieur non prévu ; contenu rejeté ou rendu comme donnée. Le cas normal reste utilisable.

RECETTES 07 À 09

Des essais qui traversent plusieurs contrôles

T07 Fichier ou URL trompeurs

Contrôles : C38, C43-C45

Préparer. Fichiers synthétiques petits, services témoins isolés et limites abaissées.

Exécuter. Tester format discordant, traversée de chemin, expansion excessive et redirection vers une destination témoin interdite.

Prouver. Refus avant effet dangereux, aucun fichier hors zone autorisée, aucune connexion vers le témoin interdit et ressources bornées.

T08 Fournisseur lent ou défaillant

Contrôles : C42, C63-C64

Préparer. Fournisseur simulé avec modes silence, flux interrompu, 429 et réponse invalide.

Exécuter. Exécuter chaque mode, fermer le client et attendre l'issue ; compter tous les retries et vérifier la réservation de coût.

Prouver. Temps total et essais bornés, nettoyage correct, message explicite, pas de faux succès ni restitution aveugle d'un coût inconnu.

T09 Instructions hostiles dans un document

Contrôles : C55-C58

Préparer. Documents fictifs A/B et outil simulé qui enregistre les propositions d'action.

Exécuter. Inclure une instruction demandant accès au document de B ou appel hors périmètre ; examiner contexte, propositions et appels réels.

Prouver. Les barrières applicatives refusent tout accès ou action non autorisé, indépendamment du texte produit par le modèle.

RECETTES 10 À 12

Des essais qui traversent plusieurs contrôles

T10 Pic, limites et retour au calme

Contrôles : C61-C66

Préparer. Préproduction isolée, profil représentatif et seuils d'arrêt validés.

Exécuter. Monter progressivement puis appliquer un pic borné. Suivre latence, erreurs, queue, connexions et coût simulé ; redescendre la charge.

Prouver. Rejet maîtrisé au-delà de la capacité, absence de perte d'écriture acquittée ou fuite de droits, récupération mesurée.

T11 Alerte et absence de secrets dans les logs

Contrôles : C67-C69

Préparer. Identifiants fictifs et canal de test ; liste des données interdites de journalisation.

Exécuter. Déclencher échecs d'authentification, panne métier et seuil de coût simulé ; rechercher les marqueurs dans les traces autorisées.

Prouver. Alerte reçue par le bon destinataire, corrélation possible et valeurs sensibles absentes ; aucune trace accessible au rôle non habilité.

T12 Restauration et retour de version

Contrôles : C70-C72

Préparer. Copie autorisée, environnement isolé, objectifs RPO/RTO et plan de retour.

Exécuter. Restaurer données et droits ; vérifier fichiers, suppressions à réappliquer et sessions retirées. Exercer le retour prévu après migration.

Prouver. Service cohérent, pas de résurrection d'accès compromis, délais mesurés et écarts consignés. Un retour du code seul n'est pas présumé suffisant.

CLÔTURE DE L'AUDIT

Les UNKNOWN qui demandent une preuve manuelle

Hébergement et identités

Qui accède au projet, au dépôt, aux secrets et à la production ? Les rôles, la MFA des opérateurs, les environnements de prévisualisation et l'artefact réel exigent des preuves datées. Références : C06, C11, C12, C71.

Données, stockage et prestataires

Les permissions effectives, les lieux et durées de conservation, les copies chez les sous-traitants et les paramètres de confidentialité ne se déduisent pas tous du code. Références : C28, C45-C47, C58.

Dépenses et capacité

Quelles limites bloquent réellement ? Dans quel projet, sur quelle période et avec quel délai ? Quels plafonds agrégés s'appliquent aux services annexes ? Les capacités annoncées sans mesure restent à vérifier. Références : C59-C66.

Détection et reprise

Qui reçoit les alertes ? Quel compte rendu démontre une restauration et un retour après migration ? Où sont conservées les traces d'incident ? Références : C67-C72.

Traiter chaque inconnue

Pour chacune : contrôle, information manquante, conséquence possible, preuve demandée, responsable, date cible et résultat réel. Ne pas fermer le point sur une simple déclaration. Conserver les preuves hors du PDF si elles sont sensibles, avec un identifiant interne et un accès restreint.

La clôture ne vaut que pour un périmètre

Une nouvelle fonction, un changement de fournisseur, un ajout de droits, une migration ou un incident déclenche une revue ciblée. Conserver l'historique des versions du rapport et les résultats précédents.

GLOSSAIRE

Le vocabulaire utile

Authentification / autorisation

Vérifier une identité / décider de ses actions et données accessibles.

Tenant

Organisation ou client dont les données doivent rester séparées de celles des autres.

RLS

Règles d'accès par ligne dans une base ; leur effet dépend des rôles et privilèges utilisés.

Idempotence

Répéter la même opération logique sans multiplier son effet métier.

CORS / CSRF

Règles de lecture entre origines dans le navigateur / action déclenchée à l'insu d'un utilisateur authentifié.

XSS / SSRF

Exécution de contenu actif dans le navigateur / requête du serveur vers une destination détournée.

SAST / DAST / SCA

Analyse du code / tests d'une application exécutée / analyse de ses composants et dépendances.

RPO / RTO

Perte de données maximale tolérée / délai de rétablissement visé après incident.

p95 / p99

Temps sous lequel se terminent 95 % / 99 % des opérations mesurées, dans le profil testé.

RAG

Recherche de documents ajoutés au contexte d'un modèle avant la génération de sa réponse.

Budget / quota / débit / concurrence

Coût maximal / volume d'usage / demandes par période / opérations simultanées.

Fail closed

Refuser l'action protégée lorsque le droit nécessaire ne peut pas être vérifié.

CONSULTÉES LE 6 OCTOBRE 2026

Sources officielles et références

Les recommandations et scénarios sont adaptés et rédigés pour ce dossier. Les liens mènent aux sources primaires. Les documentations évolutives doivent être rapprochées des versions installées.

[ASVS] OWASP ASVS 5.0.0 — version figée de mai 2025

Exigences techniques ; les correspondances du guide sont partielles et ne valent pas conformité au standard complet.

https://raw.githubusercontent.com/OWASP/ASVS/v5.0.0/5.0.0/OWASP_Application_Security_Verification_Standard_5.0.0_en.pdf

[API] OWASP API Security Top 10 — 2023

Repères API1 à API10 pour les familles de risques des interfaces.

<https://api-security.owasp.org/editions/2023/en/0x11-t10/>

[WSTG] OWASP Web Security Testing Guide — 4.2

Méthode de test ; distinguer cette version publiée de la version de développement.

<https://owasp.org/www-project-web-security-testing-guide/>

[LLM] OWASP Top 10 for LLM Applications — 2025

Complément pour les applications qui intègrent effectivement des modèles.

<https://genai.owasp.org/llm-top-10/>

[THREAT] OWASP — Threat Modeling

Cartographie, frontières de confiance et cas d'abus.

https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html

[AUTHZ] OWASP — Authorization

Contrôles côté serveur, accès aux objets et refus par défaut.

https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html

[AUTH] OWASP — Authentication

Parcours de connexion, énumération et protection contre les tentatives.

https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

[PASSWORD] OWASP — Password Storage

Choix et paramétrage du hachage des mots de passe.

https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

CONSULTÉES LE 6 OCTOBRE 2026

Sources officielles et références

[MFA] OWASP — Multifactor Authentication

Enrôlement, récupération et protection des facteurs.

https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html

[RESET] OWASP — Forgot Password

Jetons de récupération, réponses publiques et absence de contournement.

https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html

[SESSION] OWASP — Session Management

Cookies, création, rotation, expiration et retrait des sessions.

https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

[OAUTH] OWASP — OAuth2

Flux délégués, contexte de transaction et protection des jetons.

https://cheatsheetseries.owasp.org/cheatsheets/OAuth2_Cheat_Sheet.html

[SECRETS] OWASP — Secrets Management

Inventaire, permissions, révocation et rotation des secrets.

https://cheatsheetseries.owasp.org/cheatsheets/Secrets_Management_Cheat_Sheet.html

[INPUT] OWASP — Input Validation

Validation positive des données et des contraintes attendues.

https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

[MASS] OWASP — Mass Assignment

Restriction des champs modifiables par le client.

https://cheatsheetseries.owasp.org/cheatsheets/Mass_Assignment_Cheat_Sheet.html

[SQL] OWASP — SQL Injection Prevention

Séparation entre requêtes et données ; attention aux requêtes brutes.

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

CONSULTÉES LE 6 OCTOBRE 2026

Sources officielles et références

[XSS] OWASP — Cross Site Scripting Prevention

Encodage adapté au contexte et assainissement du contenu.

https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

[CSRF] OWASP — Cross-Site Request Forgery Prevention

Protection des actions déclenchées dans un navigateur authentifié.

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

[INJECTION] OWASP — OS Command Injection Defense

Éviter les appels shell et maîtriser les arguments.

https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html

[XML] OWASP — XML External Entity Prevention

Configuration restrictive des parseurs XML.

https://cheatsheetseries.owasp.org/cheatsheets/XML_External_Entity_Prevention_Cheat_Sheet.html

[HEADERS] OWASP — HTTP Headers

En-têtes du navigateur à vérifier dans les réponses effectives.

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html

[REST] OWASP — REST Security

Méthodes, types de contenu, CORS et protection des API.

https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html

[SSRF] OWASP — Server Side Request Forgery Prevention

Destinations distantes, résolutions, redirections et réseaux interdits.

https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html

[TLS] OWASP — Transport Layer Security

Chiffrement des transports et validation des certificats.

https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Security_Cheat_Sheet.html

CONSULTÉES LE 6 OCTOBRE 2026

Sources officielles et références

[WEBSOCKET] OWASP — WebSocket Security

Origine, authentification, messages et maintien des droits.

https://cheatsheetseries.owasp.org/cheatsheets/WebSocket_Security_Cheat_Sheet.html

[UPLOAD] OWASP — File Upload

Formats, tailles, chemins et traitement des fichiers non fiables.

https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html

[CRYPTO] OWASP — Cryptographic Storage

Protection des données et gestion des clés de chiffrement.

https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html

[DATABASE] OWASP — Database Security

Accès aux données, moindre privilège et configuration.

https://cheatsheetseries.owasp.org/cheatsheets/Database_Security_Cheat_Sheet.html

[CACHE] OWASP — Web Cache Security

Isolation et prévention de l'exposition de contenus privés par les caches.

https://cheatsheetseries.owasp.org/cheatsheets/Web_Cache_Security_Cheat_Sheet.html

[PRIVACY] OWASP — User Privacy Protection

Réduction de l'exposition des données ; ne constitue pas un audit juridique.

https://cheatsheetseries.owasp.org/cheatsheets/User_Privacy_Protection_Cheat_Sheet.html

[EXPORT] OWASP — CSV Injection

Risques de formules lors de l'ouverture d'un export dans un tableur.

https://community.owasp.org/attacks/CSV_Injection

[SUPPLY] OWASP — Software Supply Chain Security

Provenance, dépendances et composants indirects.

https://cheatsheetseries.owasp.org/cheatsheets/Software_Supply_Chain_Security_Cheat_Sheet.html

CONSULTÉES LE 6 OCTOBRE 2026

Sources officielles et références

[CICD] OWASP — CI/CD Security

Permissions, secrets et isolation de la chaîne de livraison.

https://cheatsheetseries.owasp.org/cheatsheets/CI_CD_Security_Cheat_Sheet.html

[BUSINESS] OWASP — Business Logic Security

Invariants métier, enchaînements, concurrence et abus.

https://cheatsheetseries.owasp.org/cheatsheets/Business_Logic_Security_Cheat_Sheet.html

[DOS] OWASP — Denial of Service

Limites de ressources et traitement des entrées coûteuses.

https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html

[LOGGING] OWASP — Logging

Traçabilité, confidentialité et protection des journaux.

https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html

[ERROR] OWASP — Error Handling

Messages publics, diagnostics internes et gestion des exceptions.

https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html

[API4] OWASP API4:2023 — Unrestricted Resource Consumption

Épuisement de ressources et coûts des services facturés par requête.

<https://api-security.owasp.org/editions/2023/en/0xa4-unrestricted-resource-consumption/>

[API10] OWASP API10:2023 — Unsafe Consumption of APIs

Vérification des réponses et comportements des services tiers.

<https://api-security.owasp.org/editions/2023/en/0xaa-unsafe-consumption-of-apis/>

[PROMPT] OWASP LLM01:2025 — Prompt Injection

Instructions hostiles directes ou contenues dans des sources importées.

<https://genai.owasp.org/llmrisk/llm01-prompt-injection/>

CONSULTÉES LE 6 OCTOBRE 2026

Sources officielles et références

[AGENT] OWASP LLM06:2025 — Excessive Agency

Droits, fonctionnalités et autonomie excessive des outils du modèle.

<https://genai.owasp.org/llmrisk/llm062025-excessive-agency/>

[RAG] OWASP LLM08:2025 — Vector and Embedding Weaknesses

Isolation et intégrité des sources retrouvées par un système documentaire.

<https://genai.owasp.org/llmrisk/llm082025-vector-and-embedding-weaknesses/>

[OUTPUT] OWASP LLM05:2025 — Improper Output Handling

Validation des sorties avant leur consommation par un autre système.

<https://genai.owasp.org/llmrisk/llm052025-improper-output-handling/>

[CONSUMPTION] OWASP LLM10:2025 — Unbounded Consumption

Boucles, volumes, coûts et limites d'usage des modèles.

<https://genai.owasp.org/llmrisk/llm102025-unbounded-consumption/>

[SUPABASE] Supabase — API keys

Exemple de distinction entre clé publiable et clé privilégiée ; ne pas généraliser à tous les fournisseurs.

<https://supabase.com/docs/guides/getting-started/api-keys>

[RLS] PostgreSQL — Row Security Policies

Vérifier les droits et les exceptions selon le rôle et la version effectivement utilisés.

<https://www.postgresql.org/docs/current/ddl-rowsecurity.html>

[IDEMP] Stripe — Idempotent requests

Exemple de garanties fournisseur, à compléter par une unicité métier interne.

https://docs.stripe.com/api/idempotent_requests

[PAYMENT] Stripe — Payment Intents

Cycle de vie du paiement et association à une commande.

<https://docs.stripe.com/payments/payment-intents>

CONSULTÉES LE 6 OCTOBRE 2026

Sources officielles et références

[WEBHOOK] Stripe — Webhooks

Signature, corps brut, doublons, ordre et traitement des événements.

<https://docs.stripe.com/webhooks>

[STATUS] W3C — Understanding WCAG 2.2, Status Messages

Rendre les informations de progression et de résultat perceptibles.

<https://www.w3.org/WAI/WCAG22/Understanding/status-messages.html>

[OVERLOAD] Google SRE — Handling Overload

Admission, saturation et gestion de la surcharge.

<https://sre.google/sre-book/handling-overload/>

[RETRY] Microsoft Azure Architecture — Retry pattern

Reprises bornées et prise en compte de l'idempotence.

<https://learn.microsoft.com/en-us/azure/architecture/patterns/retry>

[MONITOR] Google SRE — Monitoring Distributed Systems

Surveillance et alertes liées au service effectivement rendu.

<https://sre.google/sre-book/monitoring-distributed-systems/>

[BACKUP] Cybermalveillance.gouv.fr — Les sauvegardes

Protection des copies et vérification de la restauration.

<https://www.cybermalveillance.gouv.fr/tous-nos-contenus/bonnes-pratiques/sauvegardes>

[INCIDENT] NIST SP 800-61 Rev. 3 — Incident Response

Organisation de la préparation, de la réponse et du rétablissement.

<https://csrc.nist.gov/pubs/sp/800/61/r3/final>

[CVSS] FIRST — CVSS v4.0, User Guide

Séparer sévérité technique et décision de traitement dans un contexte donné.

<https://www.first.org/cvss/v4.0/user-guide>

NOTE DE PUBLICATION

Version, attribution et limites

Édition 1.0 — 6 octobre 2026

Dossier générique IN DATA VERITAS. Les constats, chiffres et noms propres à un projet particulier ont été retirés. Aucune application réelle n'a été auditée dans ces documents. Les scénarios et recettes servent à construire une vérification autorisée, pas à déclarer une vulnérabilité existante.

Attribution et réutilisation

Ce dossier propose une adaptation pédagogique originale fondée notamment sur les ressources de la fondation OWASP, dont ASVS 5.0.0 et les Cheat Sheets. Publication sous licence Creative Commons Attribution – Partage dans les mêmes conditions 4.0 International (CC BY-SA 4.0). Citer IN DATA VERITAS, la version et les sources OWASP pertinentes ; signaler les modifications. Les marques et logos OWASP restent ceux de leurs titulaires.

Portée

Les contrôles de méthode, d'intégrité, d'expérience utilisateur, de dépenses et de reprise complètent la sélection de contrôles techniques. Leur présence ne vaut ni certification OWASP, ni audit juridique ou RGPD, ni garantie d'absence de faille. Une sélection de correspondances ASVS ne démontre pas la conformité à toutes les exigences d'un niveau.

Maintenance du dossier

Revoir les liens, avis de vulnérabilité, versions et garanties fournisseur avant chaque réutilisation. Les seuils de débit, durées, budgets, RPO et RTO se décident selon les actifs et usages. Aucune valeur d'exemple ne représente une capacité mesurée de votre application.

Documents complémentaires

02-grille-audit-securite-remplissable.pdf

03-consignes-audit-ia.pdf