Vectis
CONSULTING

TEST D'INTRUSION · RAPPORT

Rapport de test d'intrusion

Exemple SA

Test d'intrusion de l'application web et de l'API Exemple

RÉFÉRENCE	VERSION	DATE	CLASSIFICATION
RPT-2026-0001	1.0	17 décembre 2026	CONFIDENTIEL

Vectis Consulting · contact@vectis-consulting.cloud · vectis-consulting.cloud

Contrôle du document

Client	Exemple SA
Mission	ENG-2026-0001 — Test d'intrusion de l'application web et de l'API Exemple
Document	RPT-2026-0001, version 1.0, 17 décembre 2026
Classification	**Confidentiel** — diffusion limitée aux destinataires ci-dessous
Réalisation	Vectis Consulting
Relecture et validation	La Direction, Vectis Consulting
Autorisation	AUT-2026-0001 — périmètre version 2 SHA-256 `6b1e245e1b73353f2266632ead26313f5b4f734df5ba97e41d585874aa615399`

Historique des versions

VERSION	DATE	AUTEUR	MODIFICATIONS
0.9	14 décembre 2026	Vectis Consulting	Projet de rapport
1.0	17 décembre 2026	Direction de Vectis Consulting	Relecture experte, validation et diffusion

Liste de diffusion

DESTINATAIRE	ORGANISATION
Jeanne Exemple Responsable de la sécurité informatique	Exemple SA
La Direction	Vectis Consulting

> **Confidentialité**
>
> Ce rapport décrit des vulnérabilités exploitables des systèmes de Exemple SA. Il est réservé aux destinataires de la liste de diffusion. Ne le transmettez qu'à des personnes qui doivent en connaître, par un canal protégé. Vectis Consulting conserve le rapport final 5 ans après la fin de la mission et supprime les preuves et documents de travail de la mission au plus tard 90 jours après la fin de celle-ci.

Sommaire

1 Synthèse pour la direction 4

2 Périmètre et méthodologie 5

3 Synthèse des vulnérabilités 7

4 Vulnérabilités détaillées 8

VUL-01 Injection SQL dans le paramètre de tri de l'API produits 8

VUL-02 Accès aux factures d'autres clients par modification de l'identifiant 10

VUL-04 Absence de limitation des tentatives de connexion 11

VUL-03 Script intersite (XSS) réfléchi dans la page de recherche 12

VUL-05 En-tête Strict-Transport-Security absent 13

VUL-06 Divulgation des versions des composants serveur 14

5 Plan de remédiation 15

A Annexe — Chronologie 16

B Annexe — Outils 17

C Annexe — Glossaire 18

1 Synthèse pour la direction

Vectis Consulting a réalisé du 8 au 10 décembre 2026 un test d'intrusion en boîte grise de l'application web et de l'API d'Exemple SA. L'objectif était d'identifier les vulnérabilités exploitables par un attaquant externe, avec ou sans compte utilisateur.

Le niveau de risque global est **critique**. Une injection SQL accessible sans authentification permet de lire la base de données de l'application, et un défaut de contrôle d'accès permet à tout client authentifié de consulter les factures des autres clients. Ces deux vulnérabilités exposent des données personnelles et commerciales : elles doivent être corrigées en priorité, et les journaux examinés pour écarter une exploitation antérieure.

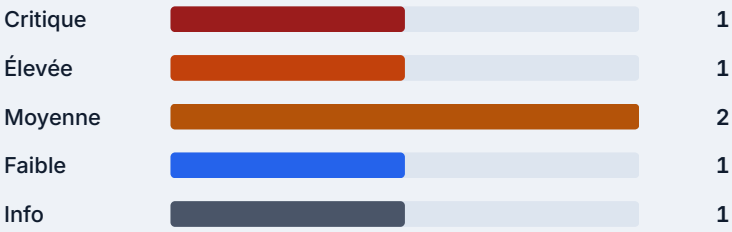
Les autres constats sont de moindre gravité mais facilitent des attaques ciblées contre les utilisateurs. La configuration TLS et la protection des cookies de session sont correctes.

NIVEAU DE RISQUE GLOBAL

Critique

6 vulnérabilités et constats

RÉPARTITION PAR SÉVÉRITÉ



Principales vulnérabilités

CRITIQUE **VUL-01 — Injection SQL dans le paramètre de tri de l'API produits**
Le paramètre `sort` est inséré tel quel dans la requête SQL : la base de données est lisible sans authentification.

ÉLEVÉE **VUL-02 — Accès aux factures d'autres clients par modification de l'identifiant**
L'API renvoie n'importe quelle facture à tout utilisateur authentifié qui en connaît l'identifiant séquentiel.

Recommandations prioritaires

1. Corriger VUL-01 et VUL-02 sans attendre, puis vérifier dans les journaux qu'elles n'ont pas été exploitées.
2. Généraliser les requêtes paramétrées et un contrôle d'autorisation centralisé, côté serveur, pour chaque objet accédé.
3. Limiter les tentatives de connexion et proposer l'authentification multifacteur.
4. Intégrer des tests de sécurité automatisés (analyse statique, tests d'autorisation) dans la chaîne de déploiement.
5. Programmer un contre-audit après correction pour confirmer l'efficacité des mesures.

Points forts observés

- Configuration TLS robuste : seuls TLS 1.2 et 1.3 sont acceptés, avec des suites de chiffrement modernes.
- Les cookies de session portent les attributs HttpOnly, Secure et SameSite=Lax.
- Les messages d'erreur de l'API ne divulguent ni trace d'exécution ni requête SQL.

2 Périmètre et méthodologie

2.1 Périmètre testé

#	CIBLE	TYPE	REMARQUES
1	`https://app.example.com`	URL	Application web (interface utilisateur)
2	`api.example.com`	Nom de domaine	API REST v2, HTTPS uniquement
3	`203.0.113.10`	Adresse IPv4	Serveur frontal, ports 80 et 443

Exclusions

#	EXCLUSION	REMARQUES
1	`https://app.example.com/billing/`	Module de paiement
2	`203.0.113.25`	Serveur de messagerie

2.2 Conditions de réalisation

Type de test	Boîte grise (avec comptes de test fournis par le Client)
Fenêtre autorisée	8 décembre 2026 à 09:00 → 10 décembre 2026 à 18:00
Tests effectifs	8 décembre 2026 à 09:02 → 10 décembre 2026 à 17:41
Adresse IP source	`198.51.100.7`
Comptes de test	client-a (rôle « client »), client-b (rôle « client »)
Éléments créés pendant les tests	Recherche enregistrée « vectis-test » sur le compte client-a : supprimée à la fin des tests.
Autorisation	AUT-2026-0001, périmètre version 2

Heures exprimées en heure de Bruxelles.

2.3 Méthodologie

Les tests suivent les référentiels publics de l'OWASP et du PTES :

- **OWASP Web Security Testing Guide (WSTG)** — catalogue des tests applicatifs, dont les identifiants sont repris dans les références des constats ;
- **OWASP Application Security Verification Standard (ASVS)** — exigences de sécurité applicative ;
- **Penetration Testing Execution Standard (PTES)** — déroulement d'un test d'intrusion.

La mission comprend quatre phases : reconnaissance et cartographie des cibles autorisées ; recherche de vulnérabilités, automatisée et manuelle ; exploitation non destructive, limitée à la démonstration de l'impact ; analyse, validation et cotation des résultats. Les vulnérabilités sont cotées selon la version 4.0 du Common Vulnerability Scoring System (CVSS), publiée par le FIRST.

2.4 Approche de test

> Les tests ont combiné des outils automatisés, dont des outils fondés sur l'intelligence artificielle, et des vérifications manuelles. Le périmètre était imposé au niveau du réseau, indépendamment des outils : seules les cibles autorisées étaient joignables, pendant la fenêtre autorisée, depuis l'adresse IP source déclarée et avec un débit plafonné. Les actions présentant un risque pour les systèmes testés n'ont été réalisées qu'avec l'accord préalable de l'expert, et toutes les actions ont été journalisées.
>
> Chaque vulnérabilité de ce rapport a été relue et validée par un expert avant son émission.

2.5 Échelle de sévérité

La sévérité découle du score de base CVSS 4.0, qui mesure l'exploitabilité et l'impact techniques. Elle ne tient pas compte de la criticité métier des données concernées, que le Client est le mieux placé pour apprécier.

SÉVÉRITÉ	SCORE CVSS 4.0	SIGNIFICATION
CRITIQUE	9,0 – 10,0	Exploitation aisée et impact majeur : compromission de données ou de systèmes. À traiter immédiatement.
ÉLEVÉE	7,0 – 8,9	Impact important ou exploitation probable. À traiter à court terme.
MOYENNE	4,0 – 6,9	Impact limité ou exploitation soumise à des conditions. À planifier.
FAIBLE	0,1 – 3,9	Impact faible ou exploitation difficile. À traiter lors des évolutions prévues.
INFO	—	Constat sans impact direct sur la sécurité, ou bonne pratique non appliquée.

2.6 Limites

Un test d'intrusion est limité dans le temps, dans son périmètre et dans ses techniques. Ses résultats décrivent l'état des systèmes pendant la fenêtre de test ; ils ne garantissent pas l'absence d'autres vulnérabilités. Limites propres à cette mission :

- Le module de paiement (https://app.example.com/billing/) et le serveur de messagerie (203.0.113.25) étaient exclus du périmètre.
- Seul le rôle « client » a été testé, au moyen de deux comptes de test ; le rôle administrateur n'était pas couvert.
- Les tests de déni de service et l'ingénierie sociale ne faisaient pas partie de la mission.

3 Synthèse des vulnérabilités

Le tableau ci-dessous présente les vulnérabilités et constats, classés par sévérité décroissante.

ID	INTITULÉ	SÉVÉRITÉ	CVSS	ACTIF CONCERNÉ
VUL-01	Injection SQL dans le paramètre de tri de l'API produits	CRITIQUE	9,3	https://api.example.com/v2/products
VUL-02	Accès aux factures d'autres clients par modification de l'identifiant	ÉLEVÉE	7,1	https://api.example.com/v2/invoices/{id}
VUL-04	Absence de limitation des tentatives de connexion	MOYENNE	6,3	https://api.example.com/v2/session
VUL-03	Script intersite (XSS) réfléchi dans la page de recherche	MOYENNE	5,1	https://app.example.com/search?q=
VUL-05	En-tête Strict-Transport-Security absent	FAIBLE	2,3	https://app.example.com
VUL-06	Divulgation des versions des composants serveur	INFO	—	https://app.example.com, api.example.com

4 Vulnérabilités détaillées

Chaque vulnérabilité a été reproduite et validée avant d'être retenue. Les données personnelles éventuellement présentes dans les preuves ont été masquées.

VUL-01

Injection SQL dans le paramètre de tri de l'API produits

Sévérité	CRITIQUE **9,3** / 10 (CVSS 4.0)
Vecteur CVSS 4.0	`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N`
Actif concerné	`https://api.example.com/v2/products`
Références	CWE-89 · OWASP WSTG-INPV-05

DESCRIPTION

Le paramètre `sort` de l'endpoint `GET /v2/products` est inséré sans validation dans la clause `ORDER BY` de la requête SQL. En y plaçant une expression conditionnelle, un attaquant non authentifié modifie l'ordre des résultats selon la valeur de son choix, ce qui permet d'extraire le contenu de la base de données caractère par caractère (inférence booléenne).

Conformément aux règles d'engagement, l'exploitation s'est limitée à la lecture du nom de l'utilisateur de base de données (`shop_app`) et de la version du serveur (PostgreSQL 15). Aucune donnée n'a été extraite ni modifiée.

IMPACT

Un attaquant peut lire toutes les données accessibles au compte applicatif : comptes clients, adresses, commandes et empreintes de mots de passe. Selon les privilèges de ce compte, il peut aussi modifier ou supprimer des données. La vulnérabilité est exploitable depuis Internet, sans compte, et des outils publics automatisent son exploitation.

PREUVES

Preuve 1. Condition vraie : les produits sont triés par nom

```
GET /v2/products?sort=(CASE+WHEN+(SELECT+substr(current_user,1,1))='s'+THEN+name+ELSE+price+END) HTTP/1.1
Host: api.example.com
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

[{"id":12,"name":"Adaptateur USB-C","price":19.9},{"id":4,"name":"Câble HDMI 2 m","price":9.5}, …]
```

Preuve 2. Condition fausse : les mêmes produits sont triés par prix

```
GET /v2/products?sort=(CASE+WHEN+(SELECT+substr(current_user,1,1))='x'+THEN+name+ELSE+price+END) HTTP/1.1
Host: api.example.com

HTTP/1.1 200 OK

[{"id":4,"name":"Câble HDMI 2 m","price":9.5},{"id":12,"name":"Adaptateur USB-C","price":19.9}, …]
```

ÉTAPES DE REPRODUCTION

1. Envoyer `GET /v2/products?sort=name` et noter l'ordre des résultats.
2. Envoyer `GET /v2/products?sort=(CASE WHEN (1=1) THEN name ELSE price END)` : l'ordre est identique.
3. Remplacer `(1=1)` par `(1=2)` : les résultats sont triés par prix, ce qui confirme que l'expression injectée est évaluée par la base de données.
4. Remplacer la condition par `(SELECT substr(current_user,1,1))='s'` pour lire une donnée caractère par caractère.

RECOMMANDATION

Ne jamais insérer de valeur fournie par l'utilisateur dans une requête SQL. Pour le tri, accepter uniquement une liste fermée de valeurs (`name`, `price`, `created_at`) associées côté serveur à des noms de colonnes fixes, et utiliser des requêtes paramétrées pour toutes les autres valeurs.

Réduire les privilèges du compte `shop_app` au strict nécessaire, puis examiner les journaux d'accès de l'API pour détecter une exploitation antérieure.

POUR ALLER PLUS LOIN

- OWASP — SQL Injection Prevention Cheat Sheet
  https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
- MITRE — CWE-89: Improper Neutralization of Special Elements used in an SQL Command
  https://cwe.mitre.org/data/definitions/89.html

VUL-02

Accès aux factures d'autres clients par modification de l'identifiant

Sévérité	ÉLEVÉE **7,1** / 10 (CVSS 4.0)
Vecteur CVSS 4.0	`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N`
Actif concerné	`https://api.example.com/v2/invoices/{id}`
Références	CWE-639 · OWASP WSTG-ATHZ-04 · OWASP API1:2023

DESCRIPTION

L'endpoint `GET /v2/invoices/{id}` renvoie la facture demandée sans vérifier qu'elle appartient au compte de l'utilisateur authentifié. Les identifiants étant séquentiels, toutes les factures peuvent être parcourues.

Pour ne pas accéder aux données de clients réels, la preuve a été établie entre les deux comptes de test fournis : le compte `client-a` a pu lire une facture du compte `client-b`.

IMPACT

Tout client disposant d'un compte peut consulter les factures de tous les autres clients : noms, adresses de facturation, montants et détail des commandes. Il s'agit de données personnelles : une exploitation constituerait une violation de données au sens du RGPD.

PREUVES

Preuve 1. Le jeton du compte client-a donne accès à une facture du compte client-b (extrait masqué)

```
GET /v2/invoices/10482 HTTP/1.1
Host: api.example.com
Authorization: Bearer eyJhbGciOi…(client-a)

HTTP/1.1 200 OK
Content-Type: application/json

{"id":10482,"account":"client-b","customer":"T*** C***","total":"249.00","lines":[…]}
```

ÉTAPES DE REPRODUCTION

1. Se connecter avec le compte `client-a` et récupérer son jeton d'accès.
2. Se connecter avec le compte `client-b` et noter l'identifiant d'une de ses factures (ici 10482).
3. Avec le jeton de `client-a`, envoyer `GET /v2/invoices/10482` : la facture de `client-b` est renvoyée.

RECOMMANDATION

Vérifier côté serveur, à chaque requête, que l'objet demandé appartient au compte de l'utilisateur, de préférence dans un composant d'autorisation commun à tous les endpoints. Des identifiants non prévisibles (UUID) compliquent l'énumération mais ne remplacent pas ce contrôle. Ajouter des tests automatisés couvrant l'accès croisé entre comptes.

POUR ALLER PLUS LOIN

- OWASP API Security Top 10 2023 — API1: Broken Object Level Authorization
  https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/
- MITRE — CWE-639: Authorization Bypass Through User-Controlled Key
  https://cwe.mitre.org/data/definitions/639.html

VUL-04

Absence de limitation des tentatives de connexion

Sévérité	MOYENNE **6,3** / 10 (CVSS 4.0)
Vecteur CVSS 4.0	`CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N`
Actif concerné	`https://api.example.com/v2/session`
Références	CWE-307 · OWASP WSTG-ATHN-03

DESCRIPTION

L'endpoint d'authentification `POST /v2/session` accepte un nombre illimité de tentatives, sans délai, sans verrouillage temporaire et sans CAPTCHA. Le test s'est limité à 50 tentatives successives sur le compte de test `client-a`, toutes traitées en moins de 15 secondes.

IMPACT

Un attaquant peut essayer massivement des mots de passe courants ou issus de fuites sur l'ensemble des comptes (pulvérisation de mots de passe, réutilisation d'identifiants). Les comptes protégés par un mot de passe faible ou réutilisé peuvent être compromis.

PREUVES

Preuve 1. 50 tentatives en 14 secondes, aucune limitation

```
#   heure         statut  réponse
1   10:12:03.114  401     {"error":"invalid_credentials"}
2   10:12:03.402  401     {"error":"invalid_credentials"}
…
50  10:12:17.236  401     {"error":"invalid_credentials"}
```

ÉTAPES DE REPRODUCTION

1. Envoyer 50 requêtes `POST /v2/session` avec l'identifiant `client-a` et des mots de passe erronés.
2. Constater que toutes les réponses arrivent sans délai, sans verrouillage du compte ni défi supplémentaire.

RECOMMANDATION

Limiter les tentatives par compte et par adresse IP, avec un délai croissant après plusieurs échecs, sans verrouillage définitif exploitable pour bloquer des comptes. Proposer l'authentification multifacteur, refuser les mots de passe connus pour avoir fuité, et surveiller les pics d'échecs.

POUR ALLER PLUS LOIN

- OWASP — Authentication Cheat Sheet
  https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
- MITRE — CWE-307: Improper Restriction of Excessive Authentication Attempts
  https://cwe.mitre.org/data/definitions/307.html

VUL-03

Script intersite (XSS) réfléchi dans la page de recherche

Sévérité	MOYENNE **5,1** / 10 (CVSS 4.0)
Vecteur CVSS 4.0	`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:A/VC:N/VI:N/VA:N/SC:L/SI:L/SA:N`
Actif concerné	`https://app.example.com/search?q=`
Références	CWE-79 · OWASP WSTG-INPV-01

DESCRIPTION

La valeur du paramètre `q` est réinsérée dans l'attribut `value` du champ de recherche sans encodage. Un attaquant peut fermer l'attribut et injecter du code JavaScript, exécuté dans le navigateur de toute personne qui ouvre un lien piégé. La preuve a utilisé une charge inoffensive affichant le nom de domaine. Aucune politique de sécurité du contenu (CSP) ne limite l'exécution de scripts.

IMPACT

Le code injecté agit avec la session de la victime : lecture des informations affichées, actions en son nom, fausse page de connexion. Le cookie de session porte l'attribut HttpOnly et ne peut donc pas être lu directement par le script.

PREUVES

Preuve 1. La charge est réinsérée telle quelle dans la page

```
GET /search?q=%22%3E%3Csvg%20onload%3Dalert(document.domain)%3E HTTP/1.1
Host: app.example.com

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8

<input type="search" name="q" value="><svg onload=alert(document.domain)>">
```

ÉTAPES DE REPRODUCTION

1. Ouvrir `https://app.example.com/search?q=%22%3E%3Csvg%20onload%3Dalert(document.domain)%3E` dans un navigateur.
2. Constater l'affichage d'une boîte de dialogue contenant `app.example.com`.

RECOMMANDATION

Encoder toute donnée selon le contexte où elle est insérée (ici, encodage d'attribut HTML), idéalement grâce à l'échappement automatique du moteur de gabarits. Déployer ensuite une politique de sécurité du contenu stricte, en défense en profondeur.

POUR ALLER PLUS LOIN

- OWASP — Cross Site Scripting Prevention Cheat Sheet
  https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- MITRE — CWE-79: Improper Neutralization of Input During Web Page Generation
  https://cwe.mitre.org/data/definitions/79.html

VUL-05

En-tête Strict-Transport-Security absent

Sévérité	FAIBLE **2,3** / 10 (CVSS 4.0)
Vecteur CVSS 4.0	`CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:P/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N`
Actif concerné	`https://app.example.com`
Références	CWE-319 · OWASP WSTG-CONF-07

DESCRIPTION

Les réponses HTTPS de l'application ne contiennent pas l'en-tête `Strict-Transport-Security`. La version HTTP (port 80) redirige vers HTTPS, mais le navigateur n'est pas invité à utiliser HTTPS d'emblée lors des visites suivantes.

IMPACT

Un attaquant placé sur le même réseau que la victime (Wi-Fi public, par exemple) peut intercepter la première requête HTTP et maintenir la victime sur une version non chiffrée du site.

PREUVES

Preuve 1. En-têtes de la réponse HTTPS

```
HTTP/2 200
content-type: text/html; charset=utf-8
set-cookie: sid=…; Path=/; Secure; HttpOnly; SameSite=Lax
x-content-type-options: nosniff
```

ÉTAPES DE REPRODUCTION

1. Envoyer `curl -sI https://app.example.com` et constater l'absence de l'en-tête `Strict-Transport-Security`.

RECOMMANDATION

Ajouter `Strict-Transport-Security: max-age=31536000; includeSubDomains` après avoir vérifié que tous les sous-domaines sont servis en HTTPS, puis envisager l'inscription à la liste de préchargement HSTS.

POUR ALLER PLUS LOIN

- OWASP — HTTP Strict Transport Security Cheat Sheet
  https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
- RFC 6797 — HTTP Strict Transport Security (HSTS)
  https://www.rfc-editor.org/rfc/rfc6797

VUL-06

Divulgation des versions des composants serveur

Sévérité	INFO
Vecteur CVSS 4.0	Sans objet (constat informatif)
Actif concerné	`https://app.example.com, api.example.com`
Références	CWE-497 · OWASP WSTG-INFO-02

DESCRIPTION

Les réponses contiennent les en-têtes `Server: nginx/1.24.0` et `X-Powered-By: Express`. Aucune vulnérabilité exploitable liée à ces versions n'a été identifiée pendant les tests.

IMPACT

Ces informations aident un attaquant à cibler des vulnérabilités connues lorsque de nouvelles failles sont publiées.

PREUVES

Preuve 1. En-têtes de réponse

```
HTTP/1.1 200 OK
Server: nginx/1.24.0
X-Powered-By: Express
```

ÉTAPES DE REPRODUCTION

1. Envoyer `curl -sI https://api.example.com/v2/products` et lire les en-têtes `Server` et `X-Powered-By`.

RECOMMANDATION

Masquer les versions : `server_tokens off;` dans la configuration nginx, et `app.disable('x-powered-by')` dans l'application Express.

POUR ALLER PLUS LOIN

- OWASP WSTG — Fingerprint Web Server
  https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server

5 Plan de remédiation

Le plan ci-dessous ordonne les corrections selon la sévérité des vulnérabilités. Les délais sont indicatifs ; ils sont à ajuster selon l'exposition et la criticité métier des systèmes concernés.

ID	ACTION RECOMMANDÉE	SÉVÉRITÉ
Immédiat — sous 7 jours		
VUL-01	Liste fermée de colonnes de tri, requêtes paramétrées, revue des journaux Injection SQL dans le paramètre de tri de l'API produits	CRITIQUE
Court terme — sous 30 jours		
VUL-02	Contrôle d'appartenance côté serveur sur chaque objet Accès aux factures d'autres clients par modification de l'identifiant	ÉLEVÉE
Moyen terme — sous 90 jours		
VUL-04	Limitation progressive des tentatives, MFA Absence de limitation des tentatives de connexion	MOYENNE
VUL-03	Encodage contextuel des sorties, puis CSP Script intersite (XSS) réfléchi dans la page de recherche	MOYENNE
Selon le cycle de maintenance		
VUL-05	Ajouter l'en-tête HSTS En-tête Strict-Transport-Security absent	FAIBLE
VUL-06	Masquer les en-têtes de version Divulgation des versions des composants serveur	INFO

Une fois les corrections déployées, un contre-audit ciblé permet de confirmer leur efficacité.

A Annexe — Chronologie

DATE ET HEURE	ÉVÉNEMENT
27/11/2026 11:15	Périmètre v2 figé, lettre d'autorisation AUT-2026-0001 émise
30/11/2026 10:40	Lettre d'autorisation signée par le Client et vérifiée
08/12/2026 09:02	Début des tests depuis 198.51.100.7, après confirmation de l'expert
08/12/2026 14:37	VUL-01 identifiée ; exploitation limitée à la preuve
08/12/2026 14:58	Alerte transmise au Client (vulnérabilité critique)
10/12/2026 17:41	Fin des tests, fermeture du filtrage réseau
14/12/2026 16:00	Projet de rapport
17/12/2026 11:00	Relecture experte terminée, rapport livré

Heures exprimées en heure de Bruxelles.

B Annexe — Outils

Catégories d'outils utilisées pendant la mission. Tout le trafic de test est passé par le filtrage du périmètre décrit au point 2.4.

CATÉGORIE D'OUTILS	USAGE
Proxy d'interception	Capture, rejeu et journalisation des requêtes HTTP(S)
Scanners automatisés	Découverte des services et contenus exposés, détection de faiblesses connues, à débit limité
Scripts spécifiques	Vérifications ciblées et preuves de concept non destructives
Outils d'analyse fondés sur l'IA	Analyse des réponses et aide au tri des résultats, sous la revue de l'expert
Navigateur web et outils de développement	Vérification manuelle et reproduction des vulnérabilités
Calculateur CVSS 4.0	Cotation de la sévérité selon la spécification du FIRST

C Annexe — Glossaire

ASVS
Application Security Verification Standard : référentiel d'exigences de sécurité applicative publié par l'OWASP.

Boîte noire, boîte grise
Test réalisé sans compte (boîte noire) ou avec des comptes de test fournis par le Client (boîte grise).

CVSS
Common Vulnerability Scoring System : système de cotation de la sévérité des vulnérabilités, publié par le FIRST (version 4.0).

CWE
Common Weakness Enumeration : catalogue des types de faiblesses logicielles, tenu par MITRE.

Exploitation non destructive
Démonstration d'une vulnérabilité sans altérer les données ni la disponibilité du système.

Fenêtre de test
Période, fixée dans la lettre d'autorisation, pendant laquelle les tests sont autorisés.

HSTS
HTTP Strict Transport Security : en-tête qui impose au navigateur l'usage de HTTPS.

IDOR
Insecure Direct Object Reference : accès à un objet d'un autre utilisateur en modifiant son identifiant, faute de contrôle d'autorisation.

Injection SQL
Insertion de code SQL dans une requête à la base de données au moyen d'une donnée fournie par l'utilisateur.

IP source
Adresse IP d'où provient le trafic de test, déclarée dans la lettre d'autorisation.

OWASP
Open Worldwide Application Security Project : fondation qui publie des référentiels de sécurité applicative.

Périmètre
Liste des cibles autorisées, fixée dans la lettre d'autorisation et identifiée par son empreinte SHA-256.

PTES
Penetration Testing Execution Standard : description de référence du déroulement d'un test d'intrusion.

WSTG
Web Security Testing Guide : guide de tests de sécurité des applications web publié par l'OWASP.

XSS
Cross-Site Scripting (script intersite) : exécution de code JavaScript injecté dans le navigateur d'un utilisateur.