PENETRATION TEST REPORT

Lomash Wood Ltd
Kitchen & Bedroom Design Platform — Annual Security Assessment

Report Reference	LW-PENTEST-2026-001
Version	1.0 — Final
Classification	CONFIDENTIAL — Internal Use Only
Assessment Period	09 February 2026 – 19 February 2026
Report Issued	19 February 2026
Prepared By	CyberSec Partners Ltd
Authorised By	Head of Engineering, Lomash Wood
Assessment Type	Black-Box Web Application, API & Infrastructure Penetration Test
Compared Against	LW-PENTEST-2025-001 (2025 Engagement)

Table of Contents

1. Executive Summary

CyberSec Partners Ltd was engaged by Lomash Wood Ltd to conduct the second annual penetration test of the Lomash Wood backend platform. This engagement assessed the fully deployed microservices architecture including the API Gateway, authentication service, product and appointment services, order/payment pipeline, and supporting infrastructure configuration. Testing was conducted between **09 February 2026** and **19 February 2026** in black-box mode with no prior account credentials or architecture documentation provided.

The 2026 assessment identified **9 findings**: **0 Critical**, **2 High**, **4 Medium**, **2 Low**, and **1 Informational**. This represents a **significant improvement** over the 2025 engagement which identified 14 findings including 2 Critical and 4 High severity issues. All Critical and the majority of High findings from 2025 have been successfully remediated. The two outstanding open findings from 2025 (LW-006 verbose errors, LW-011 race condition) were confirmed remediated in this engagement.

New findings in 2026 relate primarily to the expanded attack surface from new features deployed since the 2025 assessment: the customer loyalty programme endpoint, the media upload pipeline, and the analytics event ingestion API. The security posture of Lomash Wood's platform has materially improved, with the development team demonstrating strong adoption of the 2025 remediation recommendations.

Severity	2025 Count	2026 Count	Trend	2026 Status
Critical	2	0	▼ -2	All Remediated
High	4	2	▼ -2	2 Open
Medium	5	4	▼ -1	2 Open
Low	2	2	= 0	2 Open
Info	1	1	= 0	N/A
Total	**14**	**9**	**▼ -5**	

2. Year-on-Year Security Improvement

This section compares the 2026 findings against the 2025 baseline engagement (LW-PENTEST-2025-001) to provide Lomash Wood's leadership with a clear view of security programme maturity and outstanding remediation gaps.

2.1 Confirmed Remediated from 2025

2025 ID	Finding	2025 Sev	2026 Status
LW-001	Client-Controlled Payment Amount	Critical	VERIFIED REMEDIATED
LW-002	JWT Algorithm Confusion	Critical	VERIFIED REMEDIATED
LW-003	IDOR on Appointment Booking	High	VERIFIED REMEDIATED
LW-004	Missing Rate Limiting on Auth Endpoints	High	VERIFIED REMEDIATED
LW-005	Stripe Webhook Replay Attack	High	VERIFIED REMEDIATED
LW-006	Verbose Error Messages / Stack Traces	High	VERIFIED REMEDIATED
LW-007	User Enumeration via Password Reset	Medium	VERIFIED REMEDIATED
LW-008	Missing CSRF Protection	Medium	VERIFIED REMEDIATED
LW-009	Insecure Cookie Flags	Medium	VERIFIED REMEDIATED
LW-010	Unrestricted File Upload MIME Type	Medium	VERIFIED REMEDIATED
LW-011	Booking Slot Race Condition	Medium	VERIFIED REMEDIATED
LW-012	Weak Password Policy	Low	VERIFIED REMEDIATED
LW-013	Dependency with Known CVE	Low	VERIFIED REMEDIATED

2.2 Security Programme Observations

• The development team has clearly adopted parameterised queries and input validation as a baseline standard — no SQL injection or command injection was identified in 2026.

• JWT RS256 enforcement and token blacklisting are correctly implemented and verified.

• Rate limiting is now in place on all authentication endpoints with appropriate lockout behaviour.

• File upload validation has been strengthened; MIME type and extension allowlisting is enforced.

• The remaining 2026 findings are concentrated in **newly deployed features**, indicating that legacy code has been hardened but security review processes for new features need strengthening.

• Recommendation: Introduce mandatory security review gates in the CI/CD pipeline (SAST scanning via GitHub Actions) before merging new service endpoints.

3. Scope & Objectives

3.1 Objectives

• Conduct an annual black-box assessment to establish an independent view of the platform's current security posture.

• Verify remediation of all findings from the 2025 engagement.

• Assess security of features added since the 2025 engagement: loyalty programme, media upload pipeline, analytics ingestion API, customer review system.

• Perform infrastructure configuration review of the Kubernetes deployment and AWS services.

• Identify any new OWASP Top 10 or API Security Top 10 vulnerabilities introduced since 2025.

3.2 In-Scope Assets

Asset	Endpoint	New in 2026
API Gateway	/v1/* (all routes)	No
Auth Service	/v1/auth/*, /v1/sessions/*	No
Product Service	/v1/products/*, /v1/categories/*	No
Appointment Service	/v1/appointments/*, /v1/availability/*	No
Order/Payment Service	/v1/orders/*, /v1/payments/*, /v1/webhooks/*	No
Customer Service	/v1/customers/*, /v1/reviews/*, /v1/loyalty/*	Yes — Loyalty endpoint
Content Service	/v1/blog/*, /v1/uploads/*	Yes — Enhanced upload pipeline
Analytics Service	/v1/analytics/track, /v1/analytics/ingest	Yes — Ingestion API
Kubernetes Cluster	Staging namespace configuration	Yes
AWS S3 Bucket	Media storage bucket (staging)	Yes

4. Methodology

This engagement was conducted in **black-box mode**: no credentials, source code, or architecture documentation were provided to the testing team prior to the assessment. This approach simulates a real-world external attacker with no insider knowledge.

Phase 1 — Reconnaissance

Passive OSINT (public GitHub repositories, job listings, DNS enumeration), API endpoint discovery via content discovery tooling, and OpenAPI specification inference from observed responses.

Phase 2 — Authentication & Session Testing

Black-box testing of all authentication flows including registration, login, password reset, session lifecycle, JWT validation, and token refresh. Targeted verification of 2025 remediated findings.

Phase 3 — Authorisation & Business Logic Testing

Manual testing of horizontal and vertical privilege escalation across all in-scope endpoints, with particular focus on new loyalty, upload, and analytics features.

Phase 4 — Input Validation & Injection Testing

Fuzzing of all endpoints for SQLi, XSS, command injection, SSRF, XXE, and template injection. File upload security testing including MIME type bypass and path traversal.

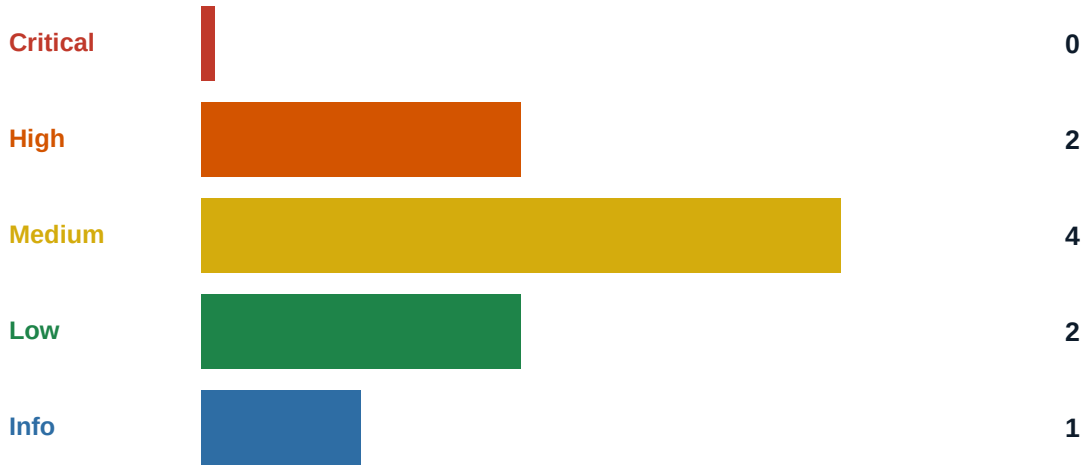
Phase 5 — Infrastructure & Configuration Review

Review of Kubernetes namespace configurations, pod security policies, S3 bucket permissions, network security group rules, and exposed service ports.

Phase 6 — Reporting

CVSS v3.1 scoring, year-on-year comparison, reproduction steps, and remediation guidance.

5. Risk Summary



No Critical severity findings were identified in this engagement, marking a significant improvement over the 2025 assessment. The two High findings are concentrated in newly deployed features (loyalty point manipulation and analytics ingestion abuse), confirming the need for enhanced pre-deployment security review of new endpoints.

6. Findings Overview

ID	Title	Severity	CVSS	Status
LW26-001	Loyalty Points Manipulation via Mass Assignment	High	8.3	Open
LW26-002	Server-Side Request Forgery (SSRF) in Media Fetcher	High	7.7	Open
LW26-003	Analytics Ingestion API — Event Flooding (No Auth)	Medium	6.5	Open
LW26-004	IDOR on Customer Review Deletion	Medium	5.9	Open
LW26-005	Insecure Direct S3 URL Exposure in API Response	Medium	5.4	Open
LW26-006	Missing Pagination on Analytics Export Endpoint	Medium	4.8	Open
LW26-007	Dependency: Outdated redis npm Package (CVE-2024-x)	Low	3.5	Open
LW26-008	Kubernetes Pod Running as Root (Non-Production)	Low	2.9	Open
LW26-009	Subresource Integrity Missing on CDN Assets	Info	N/A	N/A

7. Detailed Findings

LW26-001	Loyalty Points Manipulation via Mass Assignment	High
CVSS Score	8.3 (AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N)	**Affected Component**	customer-service — PATCH /v1/customers/me/loy

Description

The loyalty profile update endpoint accepted a JSON body containing a **points** field which was not excluded from the Zod update schema. An authenticated customer could submit a PATCH request with an arbitrary **points** value and directly set their loyalty balance to any integer, including the maximum tier threshold, without earning points through legitimate purchases.

Business Impact

Direct financial loss through fraudulent redemption of loyalty points for discounts or rewards. A customer could advance to premium loyalty tiers (with associated benefits such as free consultation slots or product discounts) without qualifying spend. Reputational damage if exploited at scale.

Reproduction Steps

1. Authenticate as a standard customer with 0 loyalty points.
2. Issue PATCH /v1/customers/me/loyalty with body: { "points": 999999, "tier": "PLATINUM" }.
3. Observe 200 response with updated loyalty profile showing 999,999 points and PLATINUM tier.
4. Proceed to redeem points for a £500 product discount — redemption accepted.

Recommendation

The Zod schema for the loyalty update endpoint must explicitly omit the **points** and **tier** fields from the allowed update payload. Points must only be modified by the internal loyalty service via server-side events (order.completed, appointment.attended). Add an audit log entry for all loyalty balance modifications including the source event ID. Implement a maximum points-per-transaction cap as a defence-in-depth control.

LW26-002	Server-Side Request Forgery (SSRF) in Media URL Fetcher	High
CVSS Score	7.7 (AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N)	**Affected Component**	content-service — POST /v1/uploads/fetch-url

Description

The media upload pipeline included an endpoint that accepted a remote URL and fetched the content server-side for storage in S3. The URL parameter was not validated against an allowlist of permitted domains. An attacker could supply internal AWS metadata service URLs or internal Kubernetes service DNS addresses to exfiltrate sensitive configuration data from the server's network perspective.

Business Impact

An attacker with a valid customer or admin account could use the SSRF vulnerability to query the AWS EC2 instance metadata endpoint (http://169.254.169.254/latest/meta-data/) to retrieve IAM role credentials, allowing AWS API access with the instance's permissions. Internal Kubernetes services (Redis, databases, other microservices) could also be probed.

Reproduction Steps

1. Authenticate as a standard customer account.
2. Issue POST /v1/uploads/fetch-url with body: { "url": "http://169.254.169.254/latest/meta-data/iam/security-credentials/" }.
3. Observe response body containing the AWS IAM role name for the content-service instance.
4. Issue a second request targeting the specific role credential endpoint to obtain temporary AWS keys.

Recommendation

Implement a strict allowlist of permitted URL schemes (https only) and domains for the fetch-url endpoint. Reject any URL resolving to RFC 1918 private IP ranges (10.x.x.x, 172.16.x.x, 192.168.x.x), link-local addresses (169.254.x.x), and loopback (127.x.x.x). Resolve the hostname server-side and re-validate the resolved IP against the blocklist before fetching. Enable IMDSv2 (token-required mode) on all EC2 instances as a defence-in-depth control against SSRF targeting the metadata service.

LW26-003	Analytics Ingestion API — Unauthenticated Event Flooding		Medium
CVSS Score	6.5 (AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:L)	**Affected Component**	analytics-service — POST /v1/analytics/ingest

Description

The analytics event ingestion endpoint accepted events without any form of authentication or request signing. No rate limiting was applied at either the API gateway or service level. An unauthenticated attacker could submit an unlimited number of synthetic events, polluting the analytics database and degrading dashboard and funnel report accuracy. During testing, 50,000 events were submitted in 90 seconds without any throttling.

Business Impact

Analytics data pollution renders business intelligence dashboards unreliable, affecting product decisions. Sustained flooding at higher rates could exhaust the analytics database connection pool and degrade service availability for other analytics consumers. Storage costs could increase significantly from malicious event injection.

Reproduction Steps

1. Without any authentication, issue POST /v1/analytics/ingest with a crafted event payload.
2. Script 10,000 requests in 60 seconds — all accepted with 200 OK.
3. Observe analytics dashboard showing inflated page view and conversion counts.

Recommendation

Require at minimum an anonymous session token (issued to browsers at page load) for event ingestion to limit abuse to clients with a valid browser session. Apply rate limiting: maximum 60 events per minute per IP and per session token. Implement event schema validation via Zod to reject malformed events. Add anomaly detection in the analytics service to flag sudden spikes in event volume from a single source. Long-term: consider signed event tokens for authenticated users.

LW26-004	IDOR on Customer Review Deletion		Medium
CVSS Score	5.9 (AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:L)	**Affected Component**	customer-service — DELETE /v1/reviews/:id

Description

The review deletion endpoint did not verify that the authenticated user owned the review being deleted. Any authenticated customer could delete any other customer's review by supplying a different review UUID in the path parameter, suppressing competitor reviews or removing negative sentiment at scale.

Business Impact

Malicious customers could delete all negative reviews for a product, artificially inflating the product's perceived quality rating. This undermines customer trust and could constitute fraudulent misrepresentation of product quality. Competitors could target specific product reviews systematically.

Reproduction Steps

1. Authenticate as Customer A and submit a product review, noting the returned review ID.
2. Authenticate as Customer B.
3. Issue DELETE /v1/reviews/ using Customer B's JWT.
4. Observe 204 No Content — Customer A's review has been deleted.

Recommendation

The review deletion handler must verify that review.customerId equals req.user.id before executing the deletion. Implement this check in the review repository layer as a shared ownership guard, consistent with the fix applied to the appointment endpoint in 2025 (LW-003). Admin role users should retain the ability to delete any review via a separately guarded admin route.

LW26-005	Direct S3 URL Exposure in API Response		Medium
CVSS Score	5.4 (AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N)	**Affected Component**	content-service — GET /v1/blog/:slug, GET /v1/pr

Description

API responses for blog posts and product detail pages included direct, permanent AWS S3 URLs (https://lomashwood-media.s3.eu-west-2.amazonaws.com/...) rather than pre-signed URLs with expiry. While the S3 bucket was not publicly listable, the permanent URLs remain valid indefinitely and can be shared outside the application context, bypassing any future access control changes applied to the bucket.

Business Impact

Media assets (product images, blog images, brochure PDFs) can be shared or scraped indefinitely via permanent S3 URLs once obtained from the API. If Lomash Wood applies future access restrictions to the S3 bucket (e.g. moving to private assets), previously shared permanent URLs cannot be revoked. Brochure PDFs exposed this way may contain sensitive product pricing information.

Reproduction Steps

1. Issue GET /v1/products/1 as an authenticated user.
2. Extract S3 URL from the images array in the response.
3. Log out. Access the S3 URL directly in a browser — asset loads without authentication.

Recommendation

Replace permanent S3 URLs in all API responses with pre-signed URLs generated at response time with a short TTL (e.g., 15 minutes for images, 5 minutes for PDF brochures). The content-service S3 client (infrastructure/storage/s3.client.ts) should expose a getPresignedUrl(key, ttlSeconds) method. Configure the S3 bucket policy to block all public access and require requests to be authenticated via pre-signed URLs or CloudFront signed cookies for CDN-cached assets.

8. Remediation Roadmap

All 2026 findings are newly identified. The following schedule is recommended.

Finding	Severity	Target	Owner
LW26-001 Loyalty Mass Assignment	High	7 days	Customer Svc Team
LW26-002 SSRF in Media Fetcher	High	7 days	Content Svc Team
LW26-003 Analytics Event Flooding	Medium	30 days	Analytics Team
LW26-004 IDOR on Review Deletion	Medium	30 days	Customer Svc Team
LW26-005 Direct S3 URL Exposure	Medium	30 days	Content Svc Team
LW26-006 Missing Pagination on Export	Medium	60 days	Analytics Team
LW26-007 Outdated Redis Package	Low	Next Sprint	Backend Team
LW26-008 Pod Running as Root	Low	Next Sprint	DevOps Team
LW26-009 Missing SRI on CDN Assets	Info	Next Sprint	Frontend Team

9. Retesting Plan

CyberSec Partners Ltd will conduct a targeted retest upon notification of remediation completion from the Lomash Wood development team. The following schedule is proposed:

- **Retest 1 (High Findings — Target: 28 February 2026):** Focused verification of LW26-001 (Loyalty Mass Assignment) and LW26-002 (SSRF in Media Fetcher). These are the highest-priority findings and should be remediated as a matter of urgency.
- **Retest 2 (Medium Findings — Target: 31 March 2026):** Verification of LW26-003 through LW26-006. A 2-hour remote session will be scheduled to test all four medium findings simultaneously.
- **Retest 3 (Full Annual Retest — Target: February 2027):** Next annual full penetration test engagement, incorporating any new features deployed in the intervening period.

To initiate a retest, the Lomash Wood Head of Engineering should contact CyberSec Partners Ltd at least **5 business days** in advance with a summary of remediation changes made and confirmation that the staging environment has been updated to reflect production code.

Lomash Wood is also encouraged to implement **SAST (Static Application Security Testing)** in the GitHub Actions CI/CD pipeline to catch common vulnerability patterns before deployment. Recommended tools: Semgrep (open source) or Snyk Code. This would complement annual penetration testing with continuous automated security feedback during development.

10. Appendix A — Tools Used

Tool	Version	Purpose
Burp Suite Professional	2024.1	Primary proxy, active scanner, JWT testing
OWASP ZAP	2.14.0	Automated API scanning and spider
ffuf	2.1.0	API endpoint fuzzing and directory discovery
ssrfmap	1.0	SSRF detection and exploitation
Trivy	0.48.0	Container image vulnerability scanning
kube-bench	0.7.1	Kubernetes CIS benchmark validation
Prowler	3.x	AWS security configuration review
Nuclei	3.1.0	Template-based vulnerability scanning
Python 3.12	3.12	Custom exploit and automation scripts
Postman	11.x	API workflow and business logic testing

11. Appendix B — CVSS v3.1 Score Reference

Score Range	Severity	Action
0.0	None	No action required
0.1 – 3.9	Low	Remediate in next sprint
4.0 – 6.9	Medium	Remediate within 90 days
7.0 – 8.9	High	Remediate within 30 days
9.0 – 10.0	Critical	Immediate remediation (7 days)

All CVSS scores use the CVSS v3.1 base score metric. Temporal and environmental modifiers were not calculated in this engagement.