Know the weakness. Build the strength

Web Application Security Assessment Report

Gray Box Assessment — ACME Marketplace

Client: ACME Corporation (Sample)
Application: ACME Marketplace (Web Application)
Report ID: AIS-WEB-2026-09-001
Date: 3 September 2026
Version: 1.0
Classification: STRICTLY CONFIDENTIAL

Illustrative fictional sample: created by AishiSec to demonstrate report format. It does not represent a real client engagement.
AishiSec — Know the Weakness. Build the Strength.

Document Control

Document History

Version	Author	Reviewed By	Date	Change Description
1.0	AishiSec Security Team	AishiSec Security Team	3 September 2026	Final sample report

Distribution

Name	Email
[Client Security Team]	[security@client.example]

Notice

This document contains information which is the proprietary property of AishiSec. It is provided to the client for the sole purpose of remediating the findings described herein and must not be reproduced, distributed or used for any other purpose without prior written consent. Nothing in this document constitutes a guarantee, warranty or licence, express or implied. The findings represent a point-in-time assessment and do not guarantee that no other weaknesses exist.

1. Introduction

AishiSec was engaged to perform a gray box Web Application Security Assessment of the ACME Marketplace application. The assessment covered the customer-facing web application, its supporting REST API, authentication and session management, business logic flows (browse, cart, checkout, orders, invoices), the password reset flow, and the underlying security configuration. Automated scanning was used for coverage; every finding was manually verified and reproduced.

1.1 Scope

Item	Detail
Application Name	ACME Marketplace (Web Application)
URL(s)	https://marketplace.example-acme.com https://api.example-acme.com (supporting API)
User Roles	Customer, Guest, Seller, Administrator
Environment	UAT (User Acceptance Testing)

1.2 Test Attributes and Details

Attribute	Detail
Starting Vector	External — over the internet
Target Criticality	High — application processes payments and personal data
Test Aggressiveness	Cautious and calculated
Test Conspicuousness	Clear — coordinated with the client team
Proof of Concepts	Wherever possible and applicable
Assessment Window	24 August 2026 — 2 September 2026
Effort	8 working days
Team	AishiSec Security Team · security@aishisec.com

1.3 Constraints and Limitations

The assessment was performed against the UAT environment over the internet, during agreed testing windows. Testing was non-destructive: no denial-of-service techniques, no destructive data changes and no social engineering were performed. Third-party components (payment gateway, CDN) were tested only for integration behaviour and treated as out of scope for their internals.

1.4 Risk Rating Classification

Findings are rated per the OWASP risk rating methodology using CVSS 3.1. Each finding carries both a technical severity and a business impact statement.

Rating	CVSS Score	Description
CRITICAL	9.0 – 10.0	Total loss of confidentiality, integrity or availability; complete compromise of the application or its data.
HIGH	7.0 – 8.9	Major loss of confidentiality, integrity or availability; significant exposure of sensitive data or functionality.
MEDIUM	4.0 – 6.9	Significant weakness that requires specific conditions to exploit; moderate business impact.
LOW	0.1 – 3.9	Minor weakness with limited direct impact; a hardening opportunity.
INFORMATIONAL	0.0	No direct exploitability; observation or improvement opportunity.

2. Executive Summary

Illustrative fictional sample: created by AishiSec to demonstrate report format. It does not represent a real client engagement.

AishiSec was engaged by ACME Corporation (Sample) to perform a gray box Web Application Security Assessment of the ACME Marketplace application. The objective was to identify exploitable weaknesses, verify their real-world impact and provide practical remediation guidance.

During the assessment, 10 findings were identified: 2 Critical, 2 High, 3 Medium, 2 Low and 1 Informational. The most significant findings were a SQL injection in the login flow that allows complete authentication bypass, and a privilege escalation weakness that allows an ordinary customer to obtain administrative access.

The application handles payments and personal data, so the Critical findings should be treated as release blockers. The remediation plan in Section 4 orders the fixes by business risk.

2.1 Findings Overview

Severity	Count	Reference IDs
CRITICAL	2	AIS-WEB-C-01, AIS-WEB-C-02
HIGH	2	AIS-WEB-H-01, AIS-WEB-H-02
MEDIUM	3	AIS-WEB-M-01, AIS-WEB-M-02, AIS-WEB-M-03
LOW	2	AIS-WEB-L-01, AIS-WEB-L-02
INFORMATIONAL	1	AIS-WEB-I-01

2.2 OWASP Categorisation

OWASP Category	Status	Reference IDs
A01:2021 — Broken Access Control	Fail	AIS-WEB-C-02, AIS-WEB-H-01
A02:2021 — Cryptographic Failures	Fail	AIS-WEB-M-02
A03:2021 — Injection	Fail	AIS-WEB-C-01
A04:2021 — Insecure Design	Fail	AIS-WEB-H-02
A05:2021 — Security Misconfiguration	Fail	AIS-WEB-M-03, AIS-WEB-L-01, AIS-WEB-L-02, AIS-WEB-I-01
A07:2021 — Identification and Authentication Failures	Fail	AIS-WEB-M-01

2.3 Risk Summary

Reference	Severity	CVSS	Title
AIS-WEB-C-01	CRITICAL	9.8	SQL Injection — Authentication Bypass
AIS-WEB-C-02	CRITICAL	9.8	Broken Access Control — Privilege Escalation to Administrator
AIS-WEB-H-01	HIGH	7.5	Broken Object-Level Access Control — Invoice Data Disclosure
AIS-WEB-H-02	HIGH	7.5	Business Logic Flaw — Order Total Manipulation
AIS-WEB-M-01	MEDIUM	5.3	Missing Rate Limiting on the Login Endpoint
AIS-WEB-M-02	MEDIUM	5.3	Sensitive Information in API Responses
AIS-WEB-M-03	MEDIUM	6.1	Host Header Poisoning in the Password Reset Flow

Reference	Severity	CVSS	Title
AIS-WEB-L-01	LOW	3.7	Missing Security Headers
AIS-WEB-L-02	LOW	3.1	Verbose Error Messages
AIS-WEB-I-01	INFORMATIONAL	0.0	Software Version Disclosure

3. Vulnerability Descriptions & Recommendations

Findings are grouped by severity. Each finding follows the AishiSec report format: what is wrong, what an attacker could do, why it matters, how it was verified and how to fix it.

Critical Risk Vulnerabilities

AIS-WEB-C-01 — SQL Injection — Authentication Bypass

Severity	CRITICAL
CVSS 3.1	9.8 — CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Weakness	CWE-89: Improper Neutralization of Special Elements used in an SQL Command
OWASP Category	A03:2021 — Injection
MITRE ATT&CK	T1190 — Exploit Public-Facing Application
Affected	POST /api/v1/auth/login — parameter: username

Summary

The login endpoint concatenates the supplied username into an SQL query without parameterisation. An attacker can submit a crafted username that changes the meaning of the query, logging in as any account — including administrator accounts — without knowing the password.

Technical Description

- The username parameter is passed directly into an SQL WHERE clause.
- The application does not use parameterised queries or an ORM for this query.
- Database errors are reflected in responses, confirming the injection point.

Business Impact

Complete authentication bypass. An attacker obtains an authenticated session as any user, including administrators, and can then read, modify or delete any data in the application — orders, invoices, customer records and payment references. Combined with other findings, this is a release blocker.

Reproduction Steps

1. Open the login page at https://marketplace.example-acme.com/login.
2. Submit the following request, replacing the normal username with the payload shown below.
3. Observe that the application returns a valid session token for the first administrator account.

Evidence

```
POST /api/v1/auth/login HTTP/2
Host: marketplace.example-acme.com
Content-Type: application/json

{"username":"admin' OR '1'='1'--","password":"anything"}

HTTP/2 200 OK
{"status":"success","token":"eyJhbGciOi…","role":"ADMIN"}
```

Remediation

- Replace the concatenated query with parameterised queries or the ORM query builder for every database call.
- Apply least-privilege database accounts for the application.
- Remove database error details from HTTP responses.

References

- https://owasp.org/www-community/attacks/SQL_Injection
- https://cwe.mitre.org/data/definitions/89.html

Retest status: Fixed and verified in retest (2 September 2026).

AIS-WEB-C-02 — Broken Access Control — Privilege Escalation to Administrator

Severity	CRITICAL
CVSS 3.1	9.8 — CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Weakness	CWE-269: Improper Privilege Management
OWASP Category	A01:2021 — Broken Access Control
MITRE ATT&CK	T1078 — Valid Accounts
Affected	PUT /api/v1/users/me — parameter: role

Summary

The profile update endpoint accepts a role field that is not supposed to be client-controlled. A newly registered user can change their own role from CUSTOMER to ADMIN, after which the server-side authorisation checks treat them as an administrator.

Technical Description

- The API performs mass assignment from the request body without an allow-list.
- Authorisation decisions are based on the role value stored in the user's own record, which the user can edit.
- No server-side validation rejects or ignores the role field.

Business Impact

Any internet user can register an account, promote themselves to administrator and reach every administrative function: user management, order management, refunds and configuration. This turns a signup form into full application compromise.

Reproduction Steps

1. Register a new customer account.
2. Send the profile update request below, changing the role field to ADMIN.
3. Log out and log in again; the account now sees the administrative dashboard and its API endpoints.

Evidence

```
PUT /api/v1/users/me HTTP/2
Host: api.example-acme.com
Authorization: Bearer eyJhbGciOi…
Content-Type: application/json

{"name":"Test User","role":"ADMIN"}

HTTP/2 200 OK
{"id":1042,"name":"Test User","role":"ADMIN"}
```

Remediation

- Define an explicit allow-list of client-editable fields; never accept the role field from the client.
- Store and check the role server-side on every privileged request.
- Audit the admin endpoints for the same missing checks.

References

- https://owasp.org/Top10/A01_2021-Broken_Access_Control/
- https://cwe.mitre.org/data/definitions/269.html

Retest status: Fixed and verified in retest (2 September 2026).

High Risk Vulnerabilities

AIS-WEB-H-01 — Broken Object-Level Access Control — Invoice Data Disclosure

Severity	HIGH
CVSS 3.1	7.5 — CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Weakness	CWE-639: Authorization Bypass Through User-Controlled Key
OWASP Category	A01:2021 — Broken Access Control
MITRE ATT&CK	T1213 — Data from Information Repositories
Affected	GET /api/v1/invoices/{invoiceId}

Summary

Invoice identifiers are sequential and the invoice endpoint does not verify that the caller owns the invoice. Anyone who knows (or guesses) an invoice number can download any customer's invoice — name, address, phone number, order contents and last four digits of the payment card.

Technical Description

- Invoice IDs are sequential integers, predictable by enumeration.
- No ownership check is performed before returning the invoice.
- The invoice PDF includes the customer's full billing address and partial card data.

Business Impact

Customer personal data of every buyer on the platform can be harvested at scale. This is a privacy breach under data-protection law, a customer-trust incident and a compliance problem — with one request per invoice.

Reproduction Steps

1. Open an invoice while logged out using only its numeric ID: GET /api/v1/invoices/10023.
2. Increment the ID and repeat.
3. Observe that every valid ID returns a full invoice PDF with personal data.

Evidence

```
GET /api/v1/invoices/10024 HTTP/2
Host: api.example-acme.com

HTTP/2 200 OK
Content-Type: application/pdf

… invoice PDF containing customer name, address, phone and card last-four …
```

Remediation

- Enforce object-level ownership checks on every record endpoint, server-side.
- Replace sequential invoice IDs with unguessable identifiers (UUIDs) where displayed to customers.
- Remove card digits and unnecessary personal data from stored invoices.

References

- https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/
- https://cwe.mitre.org/data/definitions/639.html

Retest status: Fixed and verified in retest (2 September 2026).

AIS-WEB-H-02 — Business Logic Flaw — Order Total Manipulation

Severity	HIGH
CVSS 3.1	7.5 — CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N
Weakness	CWE-840: Business Logic Errors
OWASP Category	A04:2021 — Insecure Design
MITRE ATT&CK	T1499 — Endpoint Denial of Service (related pattern: abuse of business workflow)
Affected	POST /api/v1/checkout — parameters: items[].price, items[].quantity

Summary

The checkout endpoint trusts the price and quantity values sent by the client. A guest user can change an item's price to a nominal amount — or a negative quantity — and the order total is recalculated from those client-supplied values before payment.

Technical Description

- The cart contents are recomputed from client-supplied line items at checkout.
- Prices are not re-verified against the product catalogue server-side.
- Negative quantities produce credit balances that can be applied to later orders.

Business Impact

Direct financial loss: goods can be purchased at arbitrary prices, and credit balances can be accumulated. Because the flow works for guest users, this can be repeated at scale without an account.

Reproduction Steps

1. Add an item to the cart as a guest.
2. Submit the checkout request below with the item price changed to 1.00 and quantity set to 10.
3. Observe the payment page shows the manipulated total.

Evidence

```
POST /api/v1/checkout HTTP/2
Host: api.example-acme.com
Content-Type: application/json

{"items":[{"sku":"SKU-8841","price":1.00,"quantity":10}],"payment":"cod"}

HTTP/2 200 OK
{"orderId":"ORD-7712","total":10.00}
```

Remediation

- Never accept prices from the client: resolve them from the product catalogue server-side.
- Validate quantities (positive integers, stock limits) on the server.
- Add server-side assertions that the final total matches the catalogue calculation.

References

- https://owasp.org/Top10/A04_2021-Insecure_Design/
- https://cwe.mitre.org/data/definitions/840.html

Retest status: Fixed and verified in retest (2 September 2026).

Medium Risk Vulnerabilities

AIS-WEB-M-01 — Missing Rate Limiting on the Login Endpoint

Severity	MEDIUM
CVSS 3.1	5.3 — CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Weakness	CWE-799: Improper Control of Interaction Frequency
OWASP Category	A07:2021 — Identification and Authentication Failures
MITRE ATT&CK	T1110 — Brute Force
Affected	POST /api/v1/auth/login

Summary

The login endpoint applies no throttling or lockout. Automated tools can try tens of thousands of passwords per hour against a single account without being slowed or blocked.

Technical Description

- No rate limit, CAPTCHA or account lockout is enforced.
- Responses distinguish 'user not found' from 'wrong password', enabling account enumeration.
- No monitoring alerts on repeated failures.

Business Impact

Credential-stuffing and password-guessing campaigns become practical, leading to account takeover — the starting point for fraud and further attacks against the platform.

Reproduction Steps

1. Submit 1,000 incorrect login attempts for a known account in 10 minutes.
2. Observe that all requests are processed normally with no throttling or lockout.

Evidence

```
10,000 attempts per hour accepted with HTTP 200/401 responses; no Retry-After header, no lockout observed.
```

Remediation

- Enforce per-account and per-IP throttling on login attempts.
- Introduce exponential backoff or CAPTCHA after repeated failures.
- Use identical error messages for unknown user and wrong password.

References

- https://cwe.mitre.org/data/definitions/799.html
- https://attack.mitre.org/techniques/T1110/

Retest status: Fixed and verified in retest (2 September 2026).

AIS-WEB-M-02 — Sensitive Information in API Responses

Severity	MEDIUM
CVSS 3.1	5.3 — CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Weakness	CWE-200: Exposure of Sensitive Information to an Unauthorized Actor
OWASP Category	A02:2021 — Cryptographic Failures
MITRE ATT&CK	T1083 — File and Directory Discovery
Affected	Multiple endpoints (user profile, order history, settings)

Summary

API responses include internal fields the application never displays: database record IDs, internal server paths, an internal service token and the user's full date of birth.

Technical Description

- Responses serialise the full database row rather than a defined DTO.
- An internal service token is exposed in the settings response.
- Error responses include absolute server file paths.

Business Impact

Leaked internals make follow-on attacks easier — attackers learn the data model, the server layout and gain a token that may be useful against other internal services.

Reproduction Steps

1. Call GET /api/v1/users/me with a valid session.
2. Observe fields such as internalId, dbRowId, serviceToken and serverPath in the response.

Evidence

```
{"id":1042,"internalId":8871,"serviceToken":"svc_internal_7f3a…","serverPath":"/var/www/app/…","dob":"1990-01-0
```

Remediation

- Define explicit response DTOs containing only the fields the client needs.
- Rotate the exposed internal service token immediately.
- Return generic error messages without server paths.

References

- https://cwe.mitre.org/data/definitions/200.html
- https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

Retest status: Fixed and verified in retest (2 September 2026).

AIS-WEB-M-03 — Host Header Poisoning in the Password Reset Flow

Severity	MEDIUM
CVSS 3.1	6.1 — CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Weakness	CWE-601: URL Redirection to Untrusted Site
OWASP Category	A05:2021 — Security Misconfiguration
MITRE ATT&CK	T1566 — Phishing (delivery via poisoned link)
Affected	POST /api/v1/password-reset/request

Summary

The password reset email is built using the Host header supplied by the client. An attacker can request a reset for a victim's account while sending an attacker-controlled Host, causing the reset link to point to the attacker's domain.

Technical Description

- The application constructs reset URLs from the request Host header.
- No allow-list of permitted hostnames is enforced.
- The reset token itself is embedded in the emailed link.

Business Impact

An attacker who tricks a victim into clicking the poisoned link — or who controls the reset request — can intercept the reset token and take over the account.

Reproduction Steps

1. Send the reset request below for the victim's email with a modified Host header.
2. Observe the victim receives a reset link pointing to the attacker's domain.

Evidence

```
POST /api/v1/password-reset/request HTTP/2
Host: attacker-controlled.example
Content-Type: application/json

{"email":"victim@example.com"}

Reset email received by victim contains:
https://attacker-controlled.example/reset?token=7f3a9c…
```

Remediation

- Never build URLs from the client-supplied Host header — use the configured canonical domain.
- Enforce an allow-list of valid Host values at the edge and application layers.
- Bind reset tokens to the account and expire them after a short window.

References

- https://cwe.mitre.org/data/definitions/601.html
- https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

Retest status: Fixed and verified in retest (2 September 2026).

Low Risk Vulnerabilities

AIS-WEB-L-01 — Missing Security Headers

Severity	LOW
CVSS 3.1	3.7 — CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Weakness	CWE-693: Protection Mechanism Failure
OWASP Category	A05:2021 — Security Misconfiguration
MITRE ATT&CK	T1189 — Drive-by Compromise (reduced by missing headers)
Affected	All application responses

Summary

Several security headers are absent, including Strict-Transport-Security, a strict Content-Security-Policy and X-Content-Type-Options. The application relies on HTTPS but never tells browsers to enforce it.

Technical Description

- No HSTS header is returned.
- Content-Security-Policy is missing on several pages.
- X-Content-Type-Options is absent.

Business Impact

Browsers may fall back to plain HTTP in some flows, and MIME-sniffing or injected content faces fewer browser-side protections. These headers are cheap, defence-in-depth controls.

Reproduction Steps

1. Inspect the response headers of any page.
2. Observe the missing headers listed above.

Evidence

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
(Strict-Transport-Security, Content-Security-Policy and X-Content-Type-Options headers are absent)
```

Remediation

- Enable Strict-Transport-Security with includeSubDomains and a long max-age.
- Deploy a strict Content-Security-Policy and test it in report-only mode first.
- Add X-Content-Type-Options: nosniff, Referrer-Policy and Permissions-Policy.

References

- https://cwe.mitre.org/data/definitions/693.html
- https://owasp.org/www-project-secure-headers/

Retest status: Remediation in progress — scheduled for the next retest.

AIS-WEB-L-02 — Verbose Error Messages

Severity	LOW
CVSS 3.1	3.1 — CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Weakness	CWE-209: Generation of Error Message Containing Sensitive Information
OWASP Category	A05:2021 — Security Misconfiguration
MITRE ATT&CK	T1083 — File and Directory Discovery
Affected	Invalid input on login and search endpoints

Summary

Application errors return internal detail to the client: SQL fragments, framework versions and absolute server paths.

Technical Description

- Database errors are reflected in responses (see AIS-WEB-C-01).
- Framework stack traces appear on malformed input.
- Server file paths are included in error bodies.

Business Impact

Error detail is reconnaissance gold: it confirms injection points and reveals the technology stack, making every later attack cheaper.

Reproduction Steps

1. Submit a single quote in the search box.
2. Observe the SQL error and server path in the response.

Evidence

```
HTTP/2 500 Internal Server Error
{"error":"SQLSTATE[42000]: Syntax error … near \"''\" at /var/www/app/src/Db.php:412"}
```

Remediation

- Return generic error messages to clients.
- Log detailed errors server-side for the development team.
- Disable debug mode in UAT and production.

References

- https://cwe.mitre.org/data/definitions/209.html

Retest status: Remediation in progress — scheduled for the next retest.

Informational Risk Vulnerabilities

AIS-WEB-I-01 — Software Version Disclosure

Severity	INFORMATIONAL
CVSS 3.1	0.0 — —
Weakness	CWE-200: Exposure of Sensitive Information to an Unauthorized Actor
OWASP Category	A05:2021 — Security Misconfiguration
MITRE ATT&CK	T1083 — File and Directory Discovery
Affected	HTTP response headers — Server and X-Powered-By

Summary

The application discloses its web server and framework versions in response headers, allowing an attacker to match the exact version against public vulnerability databases.

Technical Description

- Server: nginx/1.18.0 is returned on every response.
- X-Powered-By: PHP/7.4.19 is returned on API responses.

Business Impact

No direct exploitability — but it shortens an attacker's reconnaissance. PHP 7.4 reached end-of-life in November 2022 and no longer receives security updates.

Reproduction Steps

1. Inspect the response headers of any request.

Evidence

```
HTTP/2 200 OK
Server: nginx/1.18.0
X-Powered-By: PHP/7.4.19
```

Remediation

- Remove or genericise the Server and X-Powered-By headers.
- Plan the upgrade to a supported PHP version as a matter of urgency.

References

- https://cwe.mitre.org/data/definitions/200.html

Retest status: Remediation in progress — scheduled for the next retest.

4. Remediation Roadmap

Immediate (0–14 days)

- Fix AIS-WEB-C-01 (SQL injection) with parameterised queries.
- Fix AIS-WEB-C-02 (privilege escalation) with an allow-list and server-side role checks.
- Fix AIS-WEB-H-02 (order total manipulation) with server-side price resolution.
- Rotate the exposed internal service token (AIS-WEB-M-02).

Short term (30 days)

- Fix AIS-WEB-H-01 (invoice IDOR) with ownership checks and unguessable identifiers.
- Add login throttling and account lockout (AIS-WEB-M-01).
- Rebuild password reset links from the canonical domain (AIS-WEB-M-03).
- Remove sensitive fields from API responses (AIS-WEB-M-02).

Long term (60 days)

- Deploy the security headers (AIS-WEB-L-01).
- Genericise error messages and disable debug mode (AIS-WEB-L-02).
- Hide version banners and plan the PHP upgrade (AIS-WEB-I-01).
- Schedule a retest to verify all fixes (Section 5).

5. Retesting Policy

After remediation, AishiSec retests the affected areas to verify that each fix closes the issue and that nothing else regressed. Retest results are documented against each finding. Findings fixed and verified in the first retest are marked in Section 3.

6. Methodology

The assessment combined automated scanning for coverage with manual testing for depth, structured around the OWASP Web Security Testing Guide. Findings are mapped to OWASP categories and MITRE ATT&CK; techniques so the client can align them with existing security operations.

Phase	Description
Reconnaissance & mapping	Enumerate the application surface: endpoints, parameters, roles and technology stack (OWASP WSTG —
Authentication & session testing	Login, logout, password reset, OTP flows, session fixation and token handling (OWASP WSTG — Authen
Authorisation testing	Horizontal and vertical access control, IDOR, mass assignment and function-level checks (OWASP WSTG
Input & business logic testing	Injection classes, validation gaps and manipulation of business workflows such as checkout and pricing (C
Configuration & exposure testing	Security headers, error handling, version disclosure and API data exposure (OWASP WSTG — Configura
Verification & reporting	Every tool-reported signal is manually verified, evidence is captured, and each finding is rated for both tech

7. Limitations and Disclaimer

This assessment is a point-in-time evaluation of the UAT environment as it existed during the testing window. It does not guarantee that no other weaknesses exist, nor that weaknesses cannot be introduced by later changes. Automated scanning was used for coverage, and every reported finding was manually verified. No denial-of-service, destructive or social engineering techniques were used. Findings are classified per the risk

rating table in Section 1.4. This report is provided on a best-effort basis and does not constitute a warranty of security.