STRIKE

OFFENSIVE
SECURITY 24/7

SAMPLE THREAT EMULATION REPORT

Web Application Security Assessment — Sample Report

This is an illustrative sample of a Strike Threat Emulation report, based on a demo environment. Target details, credentials, and identifiers shown are fictional and used for demonstration purposes only.

Table of Contents

Executive Summary

Acme Commerce — Demo Environment

https://demo.acme-commerce.example

CRITICAL | WEB | STAGING

1 THREAT EMULATIONS	**11** FINDINGS	CRITICAL	2	CLOSED: 0/2
		HIGH	4	CLOSED: 0/4
		MEDIUM	3	CLOSED: 0/3
		LOW	2	CLOSED: 0/2

→ THREAT EMULATION OBJECTIVES

Comprehensive Security Assessment - OWASP Top 10 + Business Logic Validation

> This is a condensed sample: a full Strike report typically includes every finding discovered, with severity-ranked triage, business-impact scoring, and guided remediation tracking inside the platform.

Methodology

Risk Evaluation

Vulnerability severity is calculated using the impact the vulnerability would have on the business if successfully exploited, combined with the likelihood that an attacker could exploit it.

Impact

Technical impact is broken down across traditional security domains: confidentiality, integrity, availability, and accountability. Business impact requires understanding what matters to the specific organization running the application — financial damage, reputational harm, regulatory non-compliance, and privacy violations are common factors.

Severity Ratings

CRITICAL — Successful exploitation can cause multiple catastrophic effects on the organization or other organizations.

HIGH — Severe service degradation. The organization cannot perform primary functions, or organizational assets are damaged.

MEDIUM — The organization can perform primary functions, but capabilities are reduced with a negative impact on organizational assets.

LOW — Limited service degradation; effectiveness is noticeably reduced with minor impact on organizational assets.

CVSS 3.1 Scoring

Vulnerability severity is scored using the Common Vulnerability Scoring System (CVSS) version 3.1, an industry-standard framework for consistent, objective vulnerability assessment. Strike primarily uses the Base metric group — Attack Vector, Attack Complexity, Privileges Required, User Interaction, Scope, and impact on Confidentiality, Integrity, and Availability — to calculate the overall severity score used for prioritization.

Findings Summary

Each finding includes its score and current status, grouped by severity.

◆ Critical		
SQL Injection — Authentication Bypass	9.8	PENDING FIX
Mass Assignment — Privilege Escalation on Registration	9.8	PENDING FIX

▲ High		
IDOR — Cross-Account Data Access	7.5	PENDING FIX
Unauthenticated Price Manipulation	8.6	PENDING FIX
Directory Listing — Sensitive Files Exposed	7.5	PENDING FIX
DOM-Based Cross-Site Scripting (XSS)	7.1	PENDING FIX

△ Medium		
Verbose Error Messages	5.3	PENDING FIX
Weak / Guessable User Passwords	6.5	PENDING FIX
Password Change Without Current Password Check	6.5	PENDING FIX

▽ Low		
Missing CAPTCHA on Sensitive Forms	3.7	PENDING FIX
Client-Side Validation Bypass	3.7	PENDING FIX

Critical Findings

SQL Injection — Authentication Bypass 9.8

DESCRIPTION

A SQL injection vulnerability in the login endpoint allows complete authentication bypass. By injecting a SQL payload into the email parameter, an attacker can bypass password verification entirely and obtain a valid session for any account, including administrative ones.

STEPS TO REPRODUCE

```
Request:  POST /rest/user/login
          {"email":"' OR 1=1--","password":"x"}

Response: HTTP/1.1 200 OK
          {"authentication":{"token":"eyJ...","umail":"admin@demo.acme-commer
ce.example"}}
```

IMPACT

Complete authentication bypass allowing attackers to gain administrative access to any account without valid credentials. This enables full system compromise and access to all user data.

SUGGESTED REMEDIATION

1. Use parameterized queries / prepared statements.
2. Implement input validation and sanitization on all user-supplied fields.
3. Use an ORM with automatic parameterization.
4. Add WAF rules to detect common SQL injection patterns as defense in depth.

Mass Assignment — Privilege Escalation on Registration 9.8

DESCRIPTION

The user registration endpoint accepts a "role" parameter, allowing any unauthenticated visitor to register a new account directly with administrator privileges.

STEPS TO REPRODUCE

```
Request:  POST /api/Users/
          {"email":"attacker@test.com","password":"Test123!","role":"admin"}

Response: HTTP/1.1 201 Created
          {"status":"success","data":{"id":24,"role":"admin"}}
```

IMPACT

Any attacker can create administrative accounts from scratch, completely bypassing access controls, without needing any existing credentials.

SUGGESTED REMEDIATION

1. Implement an allowlist of fields accepted during registration.
2. Never accept role or privilege parameters directly from the client.
3. Use explicit DTOs that define exactly which fields are writable.
4. Assign roles exclusively through server-side logic.

High Findings

IDOR — Cross-Account Data Access 7.5

DESCRIPTION

Any authenticated user can access other users' shopping baskets by simply changing the basket ID parameter in the request. No ownership check is performed server-side.

STEPS TO REPRODUCE

```
Request:  GET /rest/basket/2  (authenticated as user ID 1)
Response: HTTP/1.1 200 OK
          {"data":{"id":2,"UserId":2,"Products":[...]}}
```

IMPACT

Privacy breach exposing other users' shopping preferences and purchase intent. Could be leveraged for competitive intelligence or targeted social engineering.

SUGGESTED REMEDIATION

1. Verify basket ownership on every request.
2. Confirm the authenticated user ID matches the resource owner.
3. Return 403 Forbidden for unauthorized access attempts.
4. Add a centralized authorization middleware layer.

> The remaining High, Medium, and Low findings from this assessment follow the same structure — full reproduction steps, business impact, and remediation guidance — and are available in the complete report delivered through the Strike platform.

STRIKE

Want to see what a full Threat Emulation report looks like for your own environment?

BOOK A DEMO – strike.sh/contact