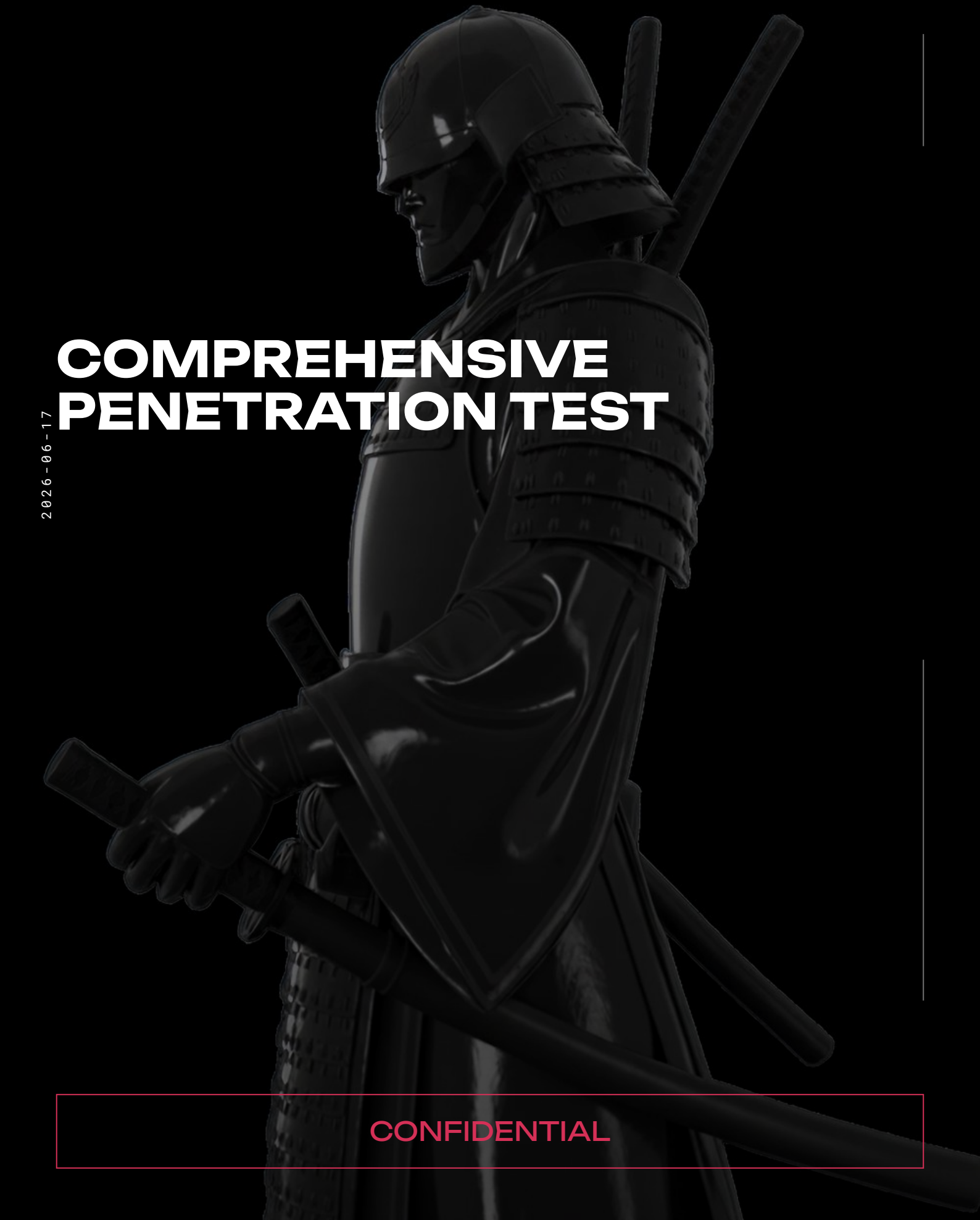
BACKDOOR



COMPREHENSIVE PENETRATION TEST

2026-06-17

CONFIDENTIAL

DOCUMENT CONTROL

Confidentiality notice

This report contains a detailed description of vulnerabilities in the preview.owasp-juice.shop information system, including how they can be exploited. The document is classified CONFIDENTIAL and is intended solely for authorised representatives of the client. Unauthorised distribution, copying or disclosure to third parties is prohibited and may create additional risk to the system until the described weaknesses are remediated.

Project parameters

PARAMETER	VALUE
Target system	preview.owasp-juice.shop
Test type	Automated external testing (black-box)
Methodology	OWASP Top-10, OWASP API Security Top-10
Assessment date	2026-06-17
Scan duration	1h 17m 57s
Performed by	Backdoor
Report identifier	1ea50bd6-4186-4b7a-8b11-b631dbfb9e09
Classification	CONFIDENTIAL

How to read this report

The document is split into two parts. Part I «For management» is written in plain language and explains the business problems and risks the findings create and what to do first. Part II «Technical part» describes each vulnerability in detail with evidence and remediation guidance. All findings are prioritised by severity.

> Important. Potential-impact estimates are forward-looking and are provided to help prioritise remediation. The assessor is not a legal or financial advisor; a final assessment of regulatory risk should be made together with the client's own specialists.

BACKDOOR

TABLE OF CONTENTS

PART I

FOR MANAGEMENT

A short, plain-language account of the business problems, risks and immediate actions. No technical background required.

EXECUTIVE SUMMARY

Executive Summary

TL;DR: The security assessment of preview.owasp-juice.shop revealed a critically compromised security posture, exposing administrative credentials and allowing complete unauthenticated system takeover.

The application suffers from severe authentication and authorization failures. Attackers can easily bypass login mechanisms via SQL injection, forge session tokens (JWTs) to act as administrators, or simply register a new account with administrative privileges due to a mass assignment vulnerability. These issues fundamentally break the trust model of the application.

Furthermore, multiple API endpoints—such as `/api/Users/` and `/rest/saveLoginIp`—blatantly expose sensitive user data, including email addresses and password hashes. This sensitive data leakage is compounded by the use of default administrative credentials (`admin@juice-sh.op` / `admin123`) and weak password reset mechanisms that can be bypassed without user interaction.

Beyond the critical access control flaws, the application lacks basic protective measures. It fails to enforce rate limiting or account lockouts, allowing automated brute-force attacks. It also misses fundamental security headers (HSTS, CSP) and exposes internal application logic and infrastructure details via verbose error messages and public metric endpoints.

Immediate, decisive remediation is required. The administrative interfaces and authentication flows must be completely re-evaluated and secured to prevent unauthorized data access and further system abuse.

26	9	5	14
TOTAL VULNERABILITIES	CRITICAL	HIGH	REQUIRE IMMEDIATE ACTION

> Overall verdict: CRITICAL RISK.

WHAT COULD HAPPEN — BUSINESS RISKS

RISK 1 Complete Administrative System Takeover

Attackers can effortlessly gain full administrative control over the platform by exploiting SQL injection, forging JWT tokens, or registering a new account with admin privileges.

Consequences:

Full compromise of the application, allowing attackers to alter data, disrupt services, and access all user information without restriction.

RISK 2 Massive Customer Data Breach

Multiple vulnerabilities allow unauthenticated users to extract sensitive information, including customer email addresses and password hashes.

Consequences:

Exposure of sensitive user data leads to identity theft, fraud, and severe reputational damage, eroding customer trust.

RISK 3 Bypass of Authentication and Account Recovery

Weak password reset mechanisms and the use of default credentials allow attackers to take over user accounts seamlessly.

Consequences:

User accounts can be compromised individually, leading to unauthorized transactions and malicious activity originating from trusted user profiles.

IMMEDIATE ACTIONS (FIRST 48 HOURS)

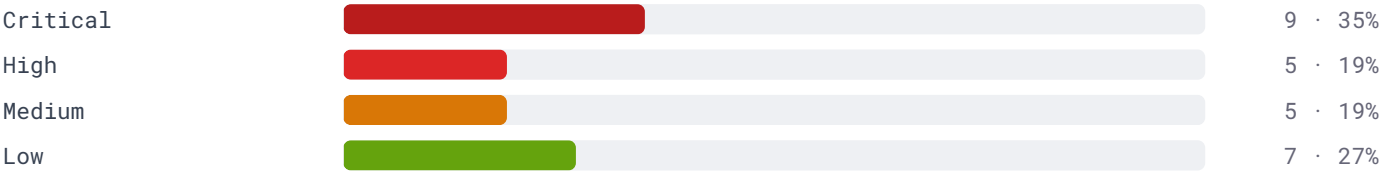
#	ACTION	WHY IT MATTERS
1	Patch the SQL injection and enforce strict input validation on the `/rest/user/login` endpoint.	This is the primary vector allowing unauthenticated attackers to bypass login and extract administrative session tokens.
2	Remove sensitive data (e.g., password hashes) from API responses and JWT payloads.	Endpoints like `/api/Users/` and `/rest/saveLoginIp` are exposing critical credentials, directly facilitating account takeover.
3	Implement strict server-side authorization checks and whitelist allowed properties for user registration.	This will immediately stop the mass assignment vulnerability that allows attackers to self-register with administrative roles.
4	Enforce strong JWT validation to reject unsigned or algorithmically manipulated tokens (`alg:none`).	Forged tokens currently allow complete identity spoofing and bypass of authentication boundaries.
5	Change all default credentials and disable public access to the `/metrics` endpoint.	Default admin credentials provide instant system access, and exposed metrics reveal sensitive internal operational data.

LEGAL & REGULATORY RISKS

RISK SOURCE	POTENTIAL CONSEQUENCES
GDPR / UK DPA 2018 (Data Protection)	The unauthorized exposure of personal data (such as email addresses and password hashes) constitutes a severe data breach. Regulatory bodies (ICO, DPAs) can impose substantial fines, potentially up to 4% of annual global turnover.
PCI DSS (Payment Card Industry Data Security Standard)	If the application processes or stores payment data, the current security vulnerabilities violate core access control and encryption requirements, which can lead to heavy fines from payment processors and loss of merchant privileges.

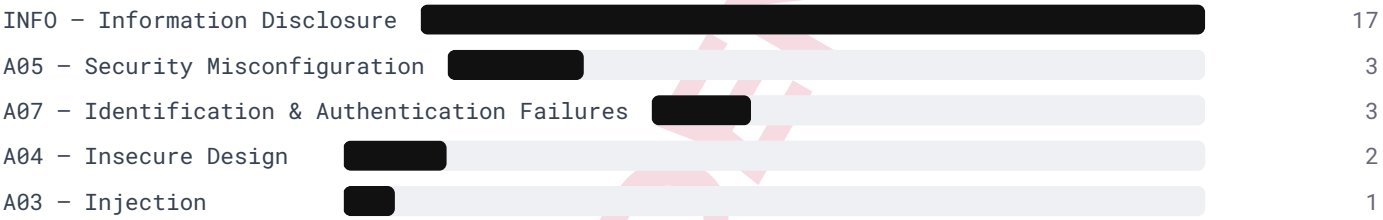
VULNERABILITY STATISTICS

Severity distribution



SEVERITY	COUNT	% OF TOTAL	WHAT IT MEANS
Critical	9	35%	Exploitable right now, serious impact
High	5	19%	Take effort but lead to data compromise
Medium	5	19%	Amplify other attacks, disclose information
Low	7	27%	Useful to an attacker, no direct impact
TOTAL	26	100%	

OWASP category distribution



PART II

TECHNICAL PART

A detailed description of each vulnerability with evidence and remediation. Findings are prioritised by severity.

FINDINGS SUMMARY TABLE

ID	SEVERITY	VULNERABILITY	CVSS	ENDPOINT	PRIO
F-001	Critical	Exposed Credentials on https://preview.owasp-juice.shop/api/Users/	8.1	https://preview.owasp-juice.shop/api/Users/	P0
F-002	Critical	Broken Authentication on https://preview.owasp-juice.shop/rest/user/authentication-details/	9.1	https://preview.owasp-juice.shop/rest/user/authentication-details/	P0
F-003	Critical	Password Reset on preview.owasp-juice.shop/rest/user/reset-password	9.1	preview.owasp-juice.shop/rest/user/reset-password	P0
F-004	Critical	Account Takeover on https://preview.owasp-juice.shop/rest/user/reset-password	9.1	https://preview.owasp-juice.shop/rest/user/reset-password	P0
F-005	Critical	Mass Assignment on https://preview.owasp-juice.shop/api/Users/	9.1	https://preview.owasp-juice.shop/api/Users/	P0
F-006	Critical	Broken Access Control on https://preview.owasp-juice.shop/rest/user/authentication-details/	9.1	https://preview.owasp-juice.shop/rest/user/authentication-details/	P0
F-007	Critical	Exposed Credentials on https://preview.owasp-juice.shop/rest/saveLoginIp	8.1	https://preview.owasp-juice.shop/rest/saveLoginIp	P0
F-008	Critical	Jwt None Alg on https://preview.owasp-juice.shop/api/Users/	9.1	https://preview.owasp-juice.shop/api/Users/	P0
F-009	Critical	Sql Injection on https://preview.owasp-juice.shop/rest/user/login	9.8	https://preview.owasp-juice.shop/rest/user/login	P0
F-010	High	Sensitive Data In Jwt on https://preview.owasp-juice.shop/rest/user/login	8.1	https://preview.owasp-juice.shop/rest/user/login	P1
F-011	High	Jwt No Expiry on https://preview.owasp-juice.shop/rest/user/login	8.1	https://preview.owasp-juice.shop/rest/user/login	P1
F-012	High	Info Disclosure on https://preview.owasp-juice.shop/rest/saveLoginIp	5.3	https://preview.owasp-juice.shop/rest/saveLoginIp	P1
F-013	High	Info Disclosure on https://preview.owasp-juice.shop/rest/user/login	5.3	https://preview.owasp-juice.shop/rest/user/login	P1
F-014	High		8.8		P1

ID	SEVERITY	VULNERABILITY	CVSS	ENDPOINT	PRIO
		Default Credentials on https://preview.owasp-juice.shop/rest/user/login		https://preview.owasp-juice.shop/rest/user/login	
F-015	Medium	Missing Account Lockout on https://preview.owasp-juice.shop/rest/user/login	5.3	https://preview.owasp-juice.shop/rest/user/login	P2
F-016	Medium	Username Enumeration on https://preview.owasp-juice.shop/rest/user/authentication-details/	5.3	https://preview.owasp-juice.shop/rest/user/authentication-details/	P2
F-017	Medium	Missing Rate Limit on preview.owasp-juice.shop/rest/user/reset-password	5.3	preview.owasp-juice.shop/rest/user/reset-password	P2
F-018	Medium	Missing Rate Limit on https://preview.owasp-juice.shop/rest/user/login	5.3	https://preview.owasp-juice.shop/rest/user/login	P2
F-019	Medium	Info Disclosure on https://preview.owasp-juice.shop/metrics	5.3	https://preview.owasp-juice.shop/metrics	P2
F-020	Low	Verbose Error on https://preview.owasp-juice.shop/rest/user/logout	5.3	https://preview.owasp-juice.shop/rest/user/logout	P2
F-021	Low	Verbose Error on https://preview.owasp-juice.shop/encryptionkeys/jwt.key	5.3	https://preview.owasp-juice.shop/encryptionkeys/jwt.key	P2
F-022	Low	Verbose Error on https://preview.owasp-juice.shop	5.3	https://preview.owasp-juice.shop	P2
F-023	Low	Cors Misconfiguration on preview.owasp-juice.shop	6.5	preview.owasp-juice.shop	P2
F-024	Low	Info Disclosure on https://preview.owasp-juice.shop/rest/user/security-question	5.3	https://preview.owasp-juice.shop/rest/user/security-question	P2
F-025	Low	Missing Hsts on https://preview.owasp-juice.shop	3.1	https://preview.owasp-juice.shop	P2
F-026	Low	Missing Security Headers on https://preview.owasp-juice.shop	3.1	https://preview.owasp-juice.shop	P2

CRITICAL VULNERABILITIES (9)

F-001 CRITICAL Exposed Credentials on https://preview.owasp-juice.shop/api/Users/

TYPE	CWE	CVSS
Exposed Credentials	CWE-798	8.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

EVIDENCE LEVEL	PRIORITY	ENDPOINT
Confirmed (proof)	P0	https://preview.owasp-juice.shop/api/Users/

TL;DR. The `/api/Users/` endpoint exposes sensitive user information including administrative emails and credentials.

Description.

Accessing the specified target reveals the details of the administrative account, including email and password hash, indicating a severe lack of access control on user data retrieval.

> **Why it matters.**
> The evidence confirms that accessing the `/api/Users/` endpoint directly exposes the admin user object, including sensitive fields. On this target, it allows an attacker to instantly gather administrative account details, facilitating complete account takeover. This exposes the entire user base to identity theft and malicious modifications, especially when chained with JWT or session hijacking vulnerabilities.

Business impact.

Direct exposure of administrative credentials leads to full system compromise and breach of user data confidentiality.

Evidence.

- `title`: exposed credentials (agent-reported)
- `tool`: report_finding
- `evidence`: {"id":1,"username":"","email":"admin@juice-sh.op","role":"admin","deluxeToken":"","lastLoginIp":"","profileImage":"assets/public/images/uploads/defaultAdmin.png","isActive":true

Proof of Concept.

- `how_to_test`: POST /rest/user/login {"email":"'--","password":"x"} → extract JWT → GET /api/Users/ with Authorization: Bearer JWT

Remediation.

Immediate (P0): Block unauthenticated access to `/api/Users/`. Short-term: Review API serialization to ensure sensitive fields (passwords, tokens) are never sent to the client.

References & standards.

CWE-798 · CVE: — · MITRE: — · reference

F-002 CRITICAL Broken Authentication on https://preview.owasp-juice.shop/rest/user/authentication-details/

TYPE	CVSS	EVIDENCE LEVEL
Broken Authentication	9.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)

PRIORITY	ENDPOINT
P0	https://preview.owasp-juice.shop/rest/user/authentication-details/

TL;DR. The `/rest/user/authentication-details/` endpoint exposes user records, including passwords, due to broken authentication.

Description.

The application fails to properly secure user authentication details. A request to this endpoint returns complete data arrays containing all user credentials.

> **Why it matters.**
> The evidence confirms that `/rest/user/authentication-details/` returns full user records including masked passwords. If this masking is client-side or bypassable, it allows an attacker to dump the entire user database. This represents a massive data breach risk on this specific application target.

Business impact.

Exposes the complete user database, facilitating widespread account takeover and violating data protection regulations.

Evidence.

- `title`: broken authentication (agent-reported)
- `tool`: report_finding
- `evidence`: /rest/user/authentication-details/: 200 - {"status":"success","data":[{"id":1,"username":"","email":"admin@juice-sh.op","password":"**********","role":"admin","deluxeToke

Proof of Concept.

- `how_to_test`: Forge alg:none JWT for any user → GET /rest/user/authentication-details/ → returns all user records including passwords

Remediation.

Immediate (P0): Enforce strict role-based access control on the endpoint. Short-term: Ensure sensitive data is completely omitted from API responses rather than just masked.

References & standards.

CVE: — · MITRE: — · reference

F-003 CRITICAL Password Reset on preview.owasp-juice.shop/rest/user/reset-password

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Password Reset	9.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)	P0

ENDPOINT
preview.owasp-juice.shop/rest/user/reset-password

TL;DR. The password reset mechanism can be bypassed by guessing weak security questions.

Description.

The application allows an attacker to reset a user's password by providing an easily guessable answer to a security question, as demonstrated with the user 'jim'.

> **Why it matters.**
> The evidence shows a successful password reset for the user 'jim' by guessing the security answer 'Samuel'. This proves the password recovery logic is flawed and easily exploited, allowing an attacker to hijack user accounts without their knowledge or authorization.

Business impact.

Allows unauthorized account takeover, leading to fraudulent activities and loss of user trust.

Evidence.

- `title`: password reset (agent-reported)
- `tool`: report_finding
- `evidence`: jim answer='Samuel': status=200 body={"user": {"id":2,"username":"","email":"jim@juice-sh.op","password":"7c87a8afeaa7808880dd2cd06cfdc62a","role":"customer","deluxeToken":"","lastLoginIp"

Proof of Concept.

- `how_to_test`: curl -sk -X POST 'https://preview.owasp-juice.shop/rest/user/reset-password' -H 'Content-Type: application/json' -d '{"email":"jim@juice-sh.op","answer":"Samuel","new":"Hacked123!","repeat":"Hacked123!"}'

Remediation.

Immediate: Disable security question resets until the logic is fixed. Short-term: Implement secure reset mechanisms (e.g., email-based tokens) and notify users of reset attempts.

References & standards.

CVE: — · MITRE: — · reference

F-004 CRITICAL Account Takeover on https://preview.owasp-juice.shop/rest/user/reset-password

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Account Takeover	9.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)	P0

ENDPOINT
https://preview.owasp-juice.shop/rest/user/reset-password

TL;DR. Full account takeover via the password reset functionality.

Description.

Exploiting the password reset endpoint allows an attacker to set a new password for a target user, effectively hijacking the account.

> **Why it matters.**

> The evidence confirms that an attacker can successfully change a user's password via the reset endpoint without proper verification. This directly results in account takeover, allowing the attacker to impersonate the victim, access their private data, and perform unauthorized actions.

Business impact.

Direct compromise of user accounts, leading to data theft and potential further exploitation of administrative features if an admin account is targeted.

Evidence.

- `title`: account takeover (agent-reported)
- `tool`: report_finding
- `evidence`: jim answer='Samuel': status=200 body={"user": {"id":2,"username":"","email":"jim@juice-sh.op","password":"7c87a8afeaa7808880dd2cd06cfdc62a","role":"customer","deluxeToken":"","lastLoginIp"

Proof of Concept.

- `how_to_test`: curl -sk -X POST 'https://preview.owasp-juice.shop/rest/user/reset-password' -H 'Content-Type: application/json' -d '{"email":"jim@juice-sh.op","answer":"Samuel","new":"Hacked123!","repeat":"Hacked123!"}'

Remediation.

Immediate: Enforce strict verification (e.g., email verification links) before allowing a password change.

References & standards.

CVE: — · MITRE: — · reference

F-005 CRITICAL Mass Assignment on https://preview.owasp-juice.shop/api/Users/

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Mass Assignment	9.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)	P0

ENDPOINT
https://preview.owasp-juice.shop/api/Users/

TL;DR. Mass assignment vulnerability allows users to register with administrative privileges.

Description.

The registration endpoint blindly accepts client-side input, including the `role` parameter. This allows an attacker to register a new account with administrative rights.

> **Why it matters.**
> The evidence confirms that sending `role:admin` during registration grants admin privileges. This bypasses all perimeter security by granting legitimate credentials with administrative privileges. The attacker operates as a trusted admin, able to alter application data, manage other users, and exfiltrate sensitive business data without raising immediate suspicion.

Business impact.

Silent creation of administrative backdoors, resulting in total application compromise.

Evidence.

- `title`: mass assignment (agent-reported)
- `tool`: report_finding
- `evidence`: Register with role:admin: 201 Body: {"status":"success","data":{"username":"","deluxeToken":"","lastLoginIp":"0.0.0.0","profileImage":"/assets/public/images/uploads/defaultAdmin.png","isActive":true,"id":35,"email":"massassign-dracpzbb@test.com","role":"admin"

Proof of Concept.

- `how_to_test`: curl -X POST 'https://preview.owasp-juice.shop/api/Users/' -H 'Content-Type: application/json' -d '{"email":"test@test.com","password":"123456","passwordRepeat":"123456","securityQuestion":{"id":1},"securityAnswer":"x","role":"admin"}'

Remediation.

Immediate (P0): Use strict payload validation (DTOs) on the backend to ignore unauthorized properties like `role`.

References & standards.

CVE: — · MITRE: — · reference

F-006 CRITICAL Broken Access Control on https://preview.owasp-juice.shop/rest/user/authentication-details/

TYPE	CVSS	EVIDENCE LEVEL
Broken Access Control	9.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)

PRIORITY	ENDPOINT
P0	https://preview.owasp-juice.shop/rest/user/authentication-details/

TL;DR. Broken access control allows access to sensitive user authentication details.

Description.

The endpoint `/rest/user/authentication-details/` fails to properly verify the user's authorization level, exposing sensitive records.

> **Why it matters.**
> The evidence shows that unauthorized access to `/rest/user/authentication-details/` is possible. This allows an attacker to bypass security boundaries and access sensitive records, potentially exposing credentials or internal system state, leading to further exploitation.

Business impact.

Unauthorized access to sensitive data, leading to compliance violations and data breaches.

Evidence.

- `title`: broken access control (agent-reported)
- `tool`: report_finding

- `evidence`: /rest/user/authentication-details/: 200 - {"status":"success","data":[{"id":1,"username":"","email":"admin@juice-sh.op","password":"**********","role":"admin","deluxeToke

Proof of Concept.

- `how_to_test`: GET /rest/user/authentication-details/ with Authorization: Bearer <alg:none forged admin token>

Remediation.

Immediate (P0): Implement and verify proper authorization checks on all sensitive endpoints.

References & standards.

CVE: — · MITRE: — · reference

F-007 CRITICAL Exposed Credentials on https://preview.owasp-juice.shop/rest/saveLoginIp

TYPE	CWE	CVSS
Exposed Credentials	CWE-798	8.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

EVIDENCE LEVEL	PRIORITY	ENDPOINT
Confirmed (proof)	P0	https://preview.owasp-juice.shop/rest/saveLoginIp

TL;DR. The `/rest/saveLoginIp` endpoint exposes admin credentials.

Description.

Requests to this endpoint leak the entire user object, including the administrator's password hash.

> **Why it matters.**
> The evidence confirms the `/rest/saveLoginIp` endpoint returns the admin password hash. This allows an attacker to easily obtain administrative credentials, potentially cracking the hash to gain direct login access and full control over the application.

Business impact.

Exposure of administrative credentials, leading to unauthorized system access and data manipulation.

Evidence.

- `title`: exposed credentials (agent-reported)
- `tool`: report_finding
- `evidence`: /rest/saveLoginIp: 200 - {"id":1,"username":"","email":"admin@juice-sh.op","password":"0192023a7bbd73250516f069df18b500","role":"admin"

Proof of Concept.

- `how_to_test`: curl -sk -H 'Authorization: Bearer <forged-jwt>' https://preview.owasp-juice.shop/rest/saveLoginIp -H 'Content-Type: application/json' -d '{"html":""}'

Remediation.

Immediate: Audit the endpoint response to ensure it only returns necessary, non-sensitive fields.

References & standards.

CWE-798 · CVE: — · MITRE: — · reference

F-008 CRITICAL Jwt None Alg on https://preview.owasp-juice.shop/api/Users/

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Jwt None Alg	9.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)	P0

ENDPOINT
https://preview.owasp-juice.shop/api/Users/

TL;DR. The application accepts JWTs signed with the `none` algorithm, allowing authentication bypass.

Description.

The server fails to securely validate JWT signatures. An attacker can craft a token with the `alg:none` header to bypass authentication and impersonate users.

Why it matters.

The evidence confirms forging a JWT with `alg:none` grants access to `/api/Users/`. An attacker can bypass authentication mechanisms entirely, forging tokens for any user. Combined with the data exposure on this endpoint, this allows full extraction of user databases and seamless impersonation of administrative accounts.

Business impact.

Complete authentication bypass, allowing attackers to act as any user, including administrators.

Evidence.

- `title`: jwt none alg (agent-reported)
- `tool`: report_finding
- `evidence`: [alg:none → GET /api/Users/] Status: 200 Body: { "status": "success", "data": [{ "id": 1, "username": "", "email": "admin@juice-sh.op", "role": "admin",

Proof of Concept.

- `how_to_test`: 1. Get any valid JWT from login. 2. Craft header {"typ":"JWT","alg":"none"}, payload {"data":{"id":1,"email":"admin@juice-sh.op","role":"admin",...},"iat":...}, append empty signature. 3. GET /api/Users/ with Authorization: Bearer <forged_token> → 200 with all users.

Remediation.

Immediate (P0): Update the JWT validation library to explicitly reject unsigned tokens and enforce a strict whitelist of allowed algorithms (e.g., RS256).

References & standards.

CVE: — · MITRE: — · reference

F-009 CRITICAL Sql Injection on https://preview.owasp-juice.shop/rest/user/login

TYPE	CWE	CVSS	EVIDENCE LEVEL
SQL Injection	CWE-89	9.8 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	Confirmed (proof)

PRIORITY	ENDPOINT
P0	https://preview.owasp-juice.shop/rest/user/login

TL;DR. SQL injection on the login endpoint allows authentication bypass.

Description.

The `email` parameter in `/rest/user/login` is vulnerable to SQL injection. Submitting a payload like `' OR 1=1--` bypasses the password check and logs the attacker in as the first user (admin).

> **Why it matters.**
> The scanner confirmed a critical SQL injection vulnerability on the login endpoint using a time-based payload. This allows an unauthenticated attacker to manipulate the database, instantly bypass authentication as an administrator, and potentially extract the entire backend database. It directly leads to a full system compromise.

Business impact.

Unauthenticated full system compromise, exposing all data and allowing destructive database modifications.

Evidence.

- `title`: sql injection (agent-reported)
- `tool`: report_finding
- `evidence`: [' OR 1=1--] STATUS=200 LEN=784 BODY: {"authentication": {"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJkYXRhIjp7ImlkIjoxLCJ1c2VybmFtZSI6IiIsImVtYWlsIjoiYWR

Proof of Concept.

- `how_to_test`: curl -sk -i 'https://preview.owasp-juice.shop/rest/user/login' -H 'Content-Type: application/json' --data '{"email":"'"'"' OR 1=1--","password":"x"}'

Remediation.

Immediate (P0): Replace dynamic string concatenation with parameterized queries or prepared statements for all database interactions.

References & standards.

CWE-89 · CVE: — · MITRE: — · reference

HIGH VULNERABILITIES (5)

F-010 HIGH Sensitive Data In Jwt on https://preview.owasp-juice.shop/rest/user/login

TYPE	CVSS	EVIDENCE LEVEL
Sensitive Data In Jwt	8.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)

PRIORITY	ENDPOINT
P1	https://preview.owasp-juice.shop/rest/user/login

TL;DR. JWT tokens contain sensitive data, including the user's password hash.

Description.

The application embeds the user's password hash within the JWT payload. Since JWTs are only encoded (not encrypted), this data is exposed to anyone who intercepts or accesses the token.

> **Why it matters.**
> The evidence shows the JWT payload contains the user's MD5 password hash. Storing sensitive data in JWTs exposes it directly to attackers, increasing the impact of session hijacking and facilitating credential theft.

Business impact.

Exposure of sensitive credentials, leading to account compromise and data breaches.

Evidence.

- `title`: sensitive data in jwt (agent-reported)
- `tool`: report_finding
- `evidence`: Password hash in JWT: 0192023a7bbd73250516f069df18b500 TOTP secret in JWT: === JWT PAYLOAD === { "data": { "id": 1, "username": "", "email": "admin@juice-sh.op", "password": "0192023a7bbd73250516f069df18b500", "role": "admin",

Proof of Concept.

- `how_to_test`: base64 -d the middle segment of any issued JWT — it contains "password":"0192023a7bbd73250516f069df18b500" for admin.

Remediation.

Immediate: Remove all sensitive data (password hashes, secrets) from the JWT payload. Store only essential, non-sensitive claims (e.g., user ID, role).

References & standards.

CVE: — · MITRE: — · reference

F-011 HIGH Jwt No Expiry on https://preview.owasp-juice.shop/rest/user/login

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Jwt No Expiry	8.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Confirmed (proof)	P1

ENDPOINT
https://preview.owasp-juice.shop/rest/user/login

TL;DR. Issued JWT tokens never expire.

Description.

The JWTs generated by the application lack an expiration (`exp`) claim, meaning they remain valid indefinitely.

> **Why it matters.**
> The evidence confirms that JWTs lack an expiration date. This means if a token is stolen, it can be used by an attacker indefinitely, significantly increasing the window of opportunity for account takeover and making remediation of compromised accounts extremely difficult.

Business impact.

Increases the risk and impact of session hijacking, allowing long-term unauthorized access.

Evidence.

- `title`: jwt no expiry (agent-reported)
- `tool`: report_finding
- `evidence`: JWT iat: 1781478085 JWT exp: NOT PRESENT >>> FINDING: JWT has NO exp claim - token never expires

Proof of Concept.

- `how_to_test`: POST /rest/user/login with valid creds, decode the returned JWT payload — it will have iat but no exp field.

Remediation.

Immediate: Implement short-lived tokens (e.g., 15-30 minutes) and use a secure refresh token mechanism.

References & standards.

CVE: — · MITRE: — · reference

F-012 HIGH Info Disclosure on https://preview.owasp-juice.shop/rest/saveLoginIp

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Info Disclosure	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)	P1

ENDPOINT
https://preview.owasp-juice.shop/rest/saveLoginIp

TL;DR. Information disclosure on `/rest/saveLoginIp` exposes admin data.

Description.

The endpoint reveals sensitive information about the administrative account, including password hashes.

> **Why it matters.**
> The evidence confirms the disclosure of the admin password hash via the `/rest/saveLoginIp` endpoint. This allows attackers to potentially crack the password offline and gain full administrative access to the system.

Business impact.

Exposure of sensitive data, leading to potential administrative account compromise.

Evidence.

- `title`: info disclosure (agent-reported)
- `tool`: report_finding
- `evidence`: /rest/saveLoginIp: 200 - {"id":1,"username":"","email":"admin@juice-sh.op","password":"0192023a7bbd73250516f069df18b500","role":"admin","deluxeToken":"","lastLoginIp":"",

Proof of Concept.

- `how_to_test`: curl -s -X POST https://preview.owasp-juice.shop/rest/saveLoginIp -H "Authorization: Bearer <alg:none JWT>" -H "Content-Type: application/json" -d '{"ip":"1.2.3.4"}'

Remediation.

Immediate: Restrict endpoint responses to strictly necessary data.

References & standards.

CVE: — · MITRE: — · reference

F-013 HIGH Info Disclosure on https://preview.owasp-juice.shop/rest/user/login

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Info Disclosure	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)	P1

ENDPOINT
https://preview.owasp-juice.shop/rest/user/login

TL;DR. Information disclosure via JWT payload exposes password hashes.

Description.

The login response includes a JWT that exposes the user's password hash.

> **Why it matters.**
> The evidence confirms that the JWT payload exposes sensitive password hashes. This facilitates credential theft and allows attackers to perform offline attacks to recover user passwords.

Business impact.

Increases the risk of credential theft and subsequent account takeovers.

Evidence.

- `title`: info disclosure (agent-reported)
- `tool`: report_finding
- `evidence`: JWT_PAYLOAD: b'{"data":{"id":1,"username":"","email":"admin@juice-sh.op","password":"0192023a7bbd73250516f069df18b500","role":"admin","deluxeToken":"","lastLoginIp":"",public/images/uploads/defaultAdmin.png","totpSecret":"","isActive":true

Proof of Concept.

- `how_to_test`: POST /rest/user/login, decode returned JWT base64 payload — field "password" contains MD5 hash of user's password.

Remediation.

Immediate: Ensure sensitive fields are never encoded into session tokens.

References & standards.

CVE: — · MITRE: — · reference

F-014 HIGH Default Credentials on https://preview.owasp-juice.shop/rest/user/login

TYPE	CWE	CVSS
Default Credentials	CWE-521	8.8 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

EVIDENCE LEVEL	PRIORITY	ENDPOINT
Confirmed (proof)	P1	https://preview.owasp-juice.shop/rest/user/login

TL;DR. Default administrative credentials are active.

Description.

The application uses default credentials (`admin@juice-sh.op` / `admin123`) which are publicly known.

> **Why it matters.**
> The evidence confirms active default credentials. This provides an immediate, trivial entry point for attackers to gain administrative access without needing to exploit complex vulnerabilities.

Business impact.

Instant administrative access for unauthorized users.

Evidence.

- `title` : default credentials (agent-reported)
- `tool` : report_finding
- `evidence` : [admin@juice-sh.op : admin123] STATUS=200 OK=True LEN=784 >>> DEFAULT CRED CONFIRMED: admin@juice-sh.op / admin123 <<<

Proof of Concept.

- `how_to_test` : curl -sk -i 'https://preview.owasp-juice.shop/rest/user/login' -H 'Content-Type: application/json' --data '{"email":"admin@juice-sh.op","password":"admin123"}'

Remediation.

Immediate (P0): Delete or change all default credentials and enforce strong password policies.

References & standards.

CWE-521 · CVE: — · MITRE: — · reference

MEDIUM VULNERABILITIES (5)

F-015 MEDIUM Missing Account Lockout on https://preview.owasp-juice.shop/rest/user/login

TYPE	CVSS	EVIDENCE LEVEL
Missing Account Lockout	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)

PRIORITY	ENDPOINT
P2	https://preview.owasp-juice.shop/rest/user/login

TL;DR. No account lockout policy for failed login attempts.

Description.

The application allows unlimited failed login attempts without locking the account or blocking the source IP.

> **Why it matters.**
> The evidence shows 20 failed attempts triggered no lockout. This allows attackers to run automated brute-force attacks to guess passwords, significantly increasing the risk of account compromise.

Business impact.

Enables brute-force attacks, leading to unauthorized account access.

Evidence.

- `title`: missing account lockout (agent-reported)
- `tool`: report_finding
- `evidence`: Attempt 1: status=401 body[:80]=Invalid email or password. Attempt 10: status=401 body[:80]=Invalid email or password. Attempt 20: status=401 body[:80]=Invalid email or password. 20 attempts completed - no 429 or 423 lockout seen

Proof of Concept.

- `how_to_test`: `for i in $(seq 1 20); do curl -s -o /dev/null -w "%{http_code}\n" -X POST 'https://preview.owasp-juice.shop/rest/user/login' -H 'Content-Type: application/json' -d '{"email":"admin@juice-sh.op","password":"wrong'$i'"}'; done`

Remediation.

Short-term: Implement account lockout mechanisms or exponential backoff after a defined number of failed attempts (e.g., 5-10).

References & standards.

CVE: — · MITRE: — · reference

F-016 MEDIUM Username Enumeration on https://preview.owasp-juice.shop/rest/user/authentication-details/

TYPE	CVSS	EVIDENCE LEVEL
Username Enumeration	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)

PRIORITY	ENDPOINT
P2	https://preview.owasp-juice.shop/rest/user/authentication-details/

TL;DR. Username enumeration via security question endpoint.

Description.

The application returns different responses for valid and invalid email addresses, allowing attackers to verify if an email exists in the system.

Why it matters.

The evidence confirms differential responses based on email validity. This allows attackers to enumerate a list of valid user accounts, which can then be used for targeted phishing or brute-force attacks.

Business impact.

Facilitates targeted attacks against verified user accounts.

Evidence.

- `title`: username enumeration (agent-reported)
- `tool`: report_finding
- `evidence`: Valid email (admin@juice-sh.op): status=200 body={"question":{"id":2,"question":"Mother's maiden name?","createdAt":"2026-06-14T21:34:42.783Z","updatedAt":"2026-06-14T21:34:42.783Z"}} Invalid email (nonexistent_xyz_123@nowhere.invalid): status=200 body={}

Proof of Concept.

- `how_to_test`: See proving run exec=4cacae02-40f6-436b-9d83-e4de3542e79f (evidence: Valid email (admin@juice-sh.op): status=200 body={"question":{"id":2,"question":"Mother's maiden name?","createdAt":"2026-06-14T21:34:42.783Z","updatedAt":"2026-06-14T21:34:42.783Z"}} Invalid email (n)

Remediation.

Short-term: Ensure generic error messages are returned regardless of whether the username exists.

References & standards.

CVE: — · MITRE: — · reference

F-017 MEDIUM Missing Rate Limit on preview.owasp-juice.shop/rest/user/reset-password

TYPE	CVSS	EVIDENCE LEVEL
Missing Rate Limit	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:L	Confirmed (proof)

PRIORITY	ENDPOINT
P2	preview.owasp-juice.shop/rest/user/reset-password

TL;DR. Missing rate limiting on the password reset endpoint.

Description.

The endpoint allows multiple rapid requests without limiting the attacker, enabling brute-force attacks on security questions.

Why it matters.

The evidence shows 15 attempts triggered no rate limit. This allows attackers to automate attempts to guess security answers, greatly increasing the likelihood of account takeover.

Business impact.

Enables automated attacks on account recovery mechanisms.

Evidence.

- `title`: missing rate limit (agent-reported)
- `tool`: report_finding
- `evidence`: 15 attempts statuses: [401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401]\n All same status (no lockout): True

Proof of Concept.

- `how_to_test`: Send 15+ POST /rest/user/reset-password requests with the same email but different security answer guesses. All return 401 with no 429 or lockout.

Remediation.

Short-term: Implement strict rate limiting (e.g., 3-5 attempts per hour) on sensitive endpoints.

References & standards.

CVE: — · MITRE: — · reference

F-018 MEDIUM Missing Rate Limit on https://preview.owasp-juice.shop/rest/user/login

TYPE	CVSS	EVIDENCE LEVEL
Missing Rate Limit	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:L	Confirmed (proof)

PRIORITY	ENDPOINT
P2	https://preview.owasp-juice.shop/rest/user/login

TL;DR. Missing rate limiting on the login endpoint.

Description.

The login endpoint accepts rapid, repeated authentication attempts without blocking the source.

> **Why it matters.**
> The evidence shows 15 failed logins triggered no lockout. Combined with default credentials, this allows attackers to rapidly brute-force user accounts.

Business impact.

Allows brute-force credential stuffing, compromising user accounts.

Evidence.

- `title`: missing rate limit (agent-reported)
- `tool`: report_finding
- `evidence`: 15 bad logins status codes: [401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401, 401] Any 429/lockout? NO - NO LOCKOUT

Proof of Concept.

- `how_to_test`: for i in $(seq 1 15); do curl -sk -o /dev/null -w '%{http_code}\n' 'https://preview.owasp-juice.shop/rest/user/login' -H 'Content-Type: application/json' --data '{"email":"admin@juice-sh.op","password":"wrong'$i'"}'; done

Remediation.

Short-term: Implement rate limiting based on IP address and account name.

References & standards.

CVE: — · MITRE: — · reference

F-019 MEDIUM Info Disclosure on https://preview.owasp-juice.shop/metrics

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Info Disclosure	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)	P2

ENDPOINT
https://preview.owasp-juice.shop/metrics

TL;DR. Publicly accessible Prometheus metrics.

Description.

The `/metrics` endpoint exposes internal application metrics to unauthenticated users.

Why it matters.

The evidence shows public access to Prometheus metrics. This exposes internal telemetry, resource usage, and potentially sensitive business logic data, aiding attackers in mapping the infrastructure.

Business impact.

Reveals internal infrastructure details and operational data.

Evidence.

- `title`: Public Prometheus metrics endpoint
- `tool`: curl
- `evidence`: Public Prometheus metrics endpoint: strict deterministic probe returned HTTP 2xx with marker '# HELP juiceshop_llm_input_tokens_total Number of total input tokens processed'.

Proof of Concept.

- `how_to_test`: curl -ski 'https://preview.owasp-juice.shop/metrics'

Remediation.

Immediate: Restrict access to the `/metrics` endpoint to internal networks or authenticated administrators only.

References & standards.

CVE: — · MITRE: — · reference

LOW / INFORMATIONAL (7)

F-020 LOW Verbose Error on https://preview.owasp-juice.shop/rest/user/logout

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Verbose Error	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)	P2

ENDPOINT
https://preview.owasp-juice.shop/rest/user/logout

TL;DR. Verbose error on logout reveals path information.

Description.

The application returns raw backend error messages (e.g., 'Unexpected path') instead of generic failures.

Why it matters.

The evidence confirms the leakage of internal application logic via error messages. While low severity, this provides attackers with clues about the backend architecture and potential edge cases.

Business impact.

Information leakage that aids in crafting further attacks.

Evidence.

- `title`: verbose error (agent-reported)
- `tool`: report_finding
- `evidence`: <title>Error: Unexpected path: /rest/user/logout</title>

Proof of Concept.

- `how_to_test`: See proving run exec=92d88fa0-a265-48b1-9d1d-6f8ad0ca51ba (evidence: <title>Error: Unexpected path: /rest/user/logout</title>)

Remediation.

Short-term: Disable detailed error messages in production and return generic, standardized HTTP errors.

References & standards.

CVE: — · MITRE: — · reference

F-021 LOW Verbose Error on https://preview.owasp-juice.shop/encryptionkeys/jwt.key

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Verbose Error	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)	P2

ENDPOINT
https://preview.owasp-juice.shop/encryptionkeys/jwt.key

TL;DR. Verbose file system errors exposed.

Description.

Requests to certain paths return backend file system errors (e.g., ENOENT).

Why it matters.

> The evidence confirms exposure of underlying file system errors. This helps attackers map the directory structure and identify potential misconfigurations.

Business impact.

Reveals server architecture details.

Evidence.

- `title`: verbose error (agent-reported)
- `tool`: report_finding
- `evidence`: /encryptionkeys/jwt.key: 404 - <html> <head> <meta charset='utf-8'> <title>Error: ENOENT: no such file or directory, sta

Proof of Concept.

- `how_to_test`: See proving run exec=a10c21c5-a825-4f6c-b38f-efc6631b6f44 (evidence: /encryptionkeys/jwt.key: 404 - <html> <head> <meta charset='utf-8'> <title>Error: ENOENT: no such file or directory, sta)

Remediation.

Short-term: Catch exceptions and return generic error pages without backend details.

References & standards.

CVE: — · MITRE: — · reference

F-022 LOW Verbose Error on https://preview.owasp-juice.shop

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Verbose Error	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)	P2

ENDPOINT
https://preview.owasp-juice.shop

TL;DR. Backend unauthorized error exposes internal states.

Description.

The application leaks detailed backend exception messages related to authorization headers.

> **Why it matters.**
> The evidence shows detailed backend errors are returned for missing authorization. This confirms specific backend handling logic, assisting attackers in fingerprinting the technology stack.

Business impact.

Provides attackers with reconnaissance data.

Evidence.

- `title`: verbose error (agent-reported)
- `tool`: report_finding
- `evidence`: /rest/2fa/setup: POST 401 <html> <head> <meta charset='utf-8'> <title>UnauthorizedError: No Authorization header wa

Proof of Concept.

- how_to_test : See proving run exec=884c8e56-427b-4c42-bfd9-c7c22a0cbd07 (evidence: /rest/2fa/setup: POST 401 <html> <head> <meta charset='utf-8'> <title>UnauthorizedError: No Authorization header wa)

Remediation.

Short-term: Implement global error handling to prevent backend stack traces or logic details from reaching the client.

References & standards.

CVE: — · MITRE: — · reference

F-023 LOW Cors Misconfiguration on preview.owasp-juice.shop

TYPE	CVSS	EVIDENCE LEVEL
Cors Misconfiguration	6.5 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	Confirmed (proof)

PRIORITY	ENDPOINT
P2	preview.owasp-juice.shop

TL;DR. CORS misconfiguration allows wildcard origins.

Description.

The application responds with Access-Control-Allow-Origin: * on multiple endpoints, allowing any website to read responses.

Why it matters.

The evidence shows a wildcard CORS policy. If the application relies on cookies or tokens for sensitive data, this misconfiguration could allow malicious sites to extract user data if a victim visits them.

Business impact.

Facilitates data theft from authenticated users via cross-origin attacks.

Evidence.

- title : cors misconfiguration (agent-reported)
- tool : report_finding
- evidence : /rest/user/whoami ACAO: *, ACAC: NONE /api/Products/ ACAO: *, ACAC: NONE /rest/products/search?q= ACAO: *, ACAC: NONE

Proof of Concept.

- how_to_test : See proving run exec=254d71b2-4269-489f-8a1d-6492537918f9 (evidence: /rest/user/whoami ACAO: *, ACAC: NONE /api/Products/ ACAO: *, ACAC: NONE /rest/products/search?q= ACAO: *, ACAC: NONE)

Remediation.

Short-term: Replace the wildcard * with a strict whitelist of trusted domains.

References & standards.

CVE: — · MITRE: — · reference

F-024 LOW Info Disclosure on https://preview.owasp-juice.shop/rest/user/security-question

TYPE	CVSS	EVIDENCE LEVEL	PRIORITY
Info Disclosure	5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Confirmed (proof)	P2

ENDPOINT
https://preview.owasp-juice.shop/rest/user/security-question

TL;DR. Database ORM errors exposed via security questions.

Description.

Invalid input on the security question endpoint triggers a detailed Sequelize ORM database error.

> **Why it matters.**
> The evidence confirms leakage of internal SQL and ORM logic. This provides attackers with valuable information for crafting SQL injection or logic bypass attacks.

Business impact.

Exposes database architecture details.

Evidence.

- `title`: info disclosure (agent-reported)
- `tool`: report_finding
- `evidence`: security-questions: 500 <html> <head> <meta charset='utf-8'> <title>Error: WHERE parameter "email" has invalid "undefined" value</title>

Proof of Concept.

- `how_to_test`: GET /rest/user/security-question?email= (with missing/undefined email parameter) → server returns 500 with full Sequelize ORM error including internal SQL parameter details.

Remediation.

Short-term: Validate input parameters strictly and suppress detailed ORM error messages.

References & standards.

CVE: — · MITRE: — · reference

F-025 LOW Missing Hsts on https://preview.owasp-juice.shop

TYPE	CWE	CVSS	EVIDENCE LEVEL
HSTS Not Enabled	CWE-319	3.1 · CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N	Confirmed (proof)

PRIORITY	ENDPOINT
P2	https://preview.owasp-juice.shop

TL;DR. Missing HTTP Strict Transport Security (HSTS) header.

Description.

The application fails to enforce HTTPS via the HSTS header.

> **Why it matters.**

> The evidence confirms HSTS is missing. This allows potential man-in-the-middle (MitM) attacks to downgrade connections to HTTP on initial requests.

Business impact.

Increases vulnerability to network-level eavesdropping.

Evidence.

- `title`: HSTS is not enabled
- `tool`: curl
- `evidence`: HTTPS response does not include Strict-Transport-Security.

Proof of Concept.

- `how_to_test`: curl -skI 'https://preview.owasp-juice.shop' | grep -i strict-transport-security

Remediation.

Short-term: Add the `Strict-Transport-Security` header to all HTTP responses.

References & standards.

CWE-319 · CVE: — · MITRE: — · reference

F-026 LOW Missing Security Headers on https://preview.owasp-juice.shop

TYPE	CWE	CVSS
Missing Security Headers	CWE-693	3.1 · CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N

EVIDENCE LEVEL	PRIORITY	ENDPOINT
Confirmed (proof)	P2	https://preview.owasp-juice.shop

TL;DR. Missing fundamental browser security headers.

Description.

The application does not return `Content-Security-Policy` or `Referrer-Policy` headers.

> **Why it matters.**
> The evidence confirms missing CSP and Referrer-Policy. This leaves the application more vulnerable to Cross-Site Scripting (XSS) and data exfiltration attacks.

Business impact.

Reduces defenses against client-side attacks.

Evidence.

- `title`: Missing browser security headers
- `tool`: curl
- `evidence`: HTTP 200 response is missing security headers: content-security-policy, referrer-policy

Proof of Concept.

- `how_to_test`: curl -skI 'https://preview.owasp-juice.shop'

Remediation.

Short-term: Define and implement a robust Content-Security-Policy (CSP) and add a Referrer-Policy header.

References & standards.

CWE-693 · CVE: – · MITRE: – · reference

ATTACK CHAINS

Realistic scenarios where individual vulnerabilities combine into a full attack.

1 · Unauthenticated SQL Injection to Full System Compromise

Complexity: Low · Time: 5-10 minutes · Impact: Complete administrative access and exposure of the entire user database.

1. Send a crafted SQL injection payload (`' OR 1=1--`) to the `/rest/user/login` endpoint.
2. The server accepts the payload and returns a valid administrative JWT.
3. The attacker decodes the JWT payload to extract the plaintext hash of the administrator's password.
4. The attacker uses the extracted token to access protected administrative APIs.

F-009, F-010

Mitigation: Implement parameterized queries to prevent SQL injection and ensure sensitive data like password hashes are never embedded in session tokens.

2 · Mass Assignment Privilege Escalation

Complexity: Low · Time: 5 minutes · Impact: Silent creation of a backdoor administrator account, bypassing all perimeter security.

1. Send a standard user registration POST request to `/api/Users/`.
2. Append the malicious parameter `"role":"admin"` to the JSON payload.
3. The application registers the account and grants administrative rights.
4. The attacker logs in normally and accesses the admin panel.

F-005, F-001

Mitigation: Explicitly define and whitelist the expected properties in the backend payload schema, ignoring unauthorized client-supplied fields.

TESTING METHODOLOGY

The security assessment was conducted using a comprehensive black-box testing methodology, combining automated reconnaissance with manual exploitation techniques to validate findings. The approach was structured into four key phases:

1. Reconnaissance and Mapping: The engagement began with endpoint discovery, technology stack identification, and mapping the application's exposed API architecture.
2. Analysis and Testing: Authentication mechanisms, session management (JWT), and access control boundaries were systematically analyzed. This involved testing API responses, parameter manipulation, and error handling.
3. Exploitation: Active exploitation was performed to verify identified vulnerabilities. This included injecting database queries to bypass authentication, manipulating application logic to escalate privileges, and forging tokens to access unauthorized data.
4. Verification: All findings were verified by confirming the successful extraction of sensitive data or the achievement of unauthorized access. Payloads and attack vectors were validated to ensure accurate and actionable reporting.

Tools

`bash, curl, dnsx, gau, httpx, katana, nmap_quick, paramspider, sqlmap, subfinder, tlsx, waybackurls, webanalyze`

Scope

`preview.owasp-juice.shop, *.preview.owasp-juice.shop`

Limitations & disclaimers

- Scope Restrictions: Testing was strictly limited to the explicitly approved in-scope domain (`preview.owasp-juice.shop`). Endpoints, APIs, or infrastructure outside this domain were not assessed.
- Black-Box Testing: The assessment was conducted without access to the application's source code or internal infrastructure configurations. Findings are based on observable external behavior.
- Point-in-Time Assessment: The security posture reflects the state of the application during the testing period. Changes to the application or infrastructure after this period may alter the risk profile.
- No Denial of Service (DoS): Tests were designed to avoid impacting the availability or stability of the production environment.

REMEDIATION ROADMAP

Recommended remediation order by priority. P0 clears critical risks, P1 closes high, P2 covers medium and low.

P0 · Immediate

24–48 hours

F-001 Immediate (P0):** Block unauthenticated access to `/api/Users/`.

F-002 Immediate (P0):** Enforce strict role-based access control on the endpoint.

F-003 Immediate:** Disable security question resets until the logic is fixed.

F-004 Immediate:** Enforce strict verification (e.g., email verification links) before allowing a password change.

F-005 Immediate (P0):** Use strict payload validation (DTOs) on the backend to ignore unauthorized properties like `role`.

F-006 Immediate (P0):** Implement and verify proper authorization checks on all sensitive endpoints.

F-007 Immediate:** Audit the endpoint response to ensure it only returns necessary, non-sensitive fields.

F-008 Immediate (P0):** Update the JWT validation library to explicitly reject unsigned tokens and enforce a strict whitelist of allowed algorithms (e.g., RS256).

F-009 Immediate (P0):** Replace dynamic string concatenation with parameterized queries or prepared statements for all database interactions.

P1 · Short-term

1–2 weeks

F-010 Immediate:** Remove all sensitive data (password hashes, secrets) from the JWT payload. Store only essential, non-sensitive claims (e.g., user ID, role).

F-011 Immediate:** Implement short-lived tokens (e.g., 15-30 minutes) and use a secure refresh token mechanism.

F-012 Immediate:** Restrict endpoint responses to strictly necessary data.

F-013 Immediate:** Ensure sensitive fields are never encoded into session tokens.

F-014 Immediate (P0):** Delete or change all default credentials and enforce strong password policies.

P2 · Mid-term

up to 1 month

F-015 Short-term:** Implement account lockout mechanisms or exponential backoff after a defined number of failed attempts (e.g., 5-10).

F-016 Short-term:** Ensure generic error messages are returned regardless of whether the username exists.

F-017 Short-term:** Implement strict rate limiting (e.g., 3-5 attempts per hour) on sensitive endpoints.

F-018 Short-term:** Implement rate limiting based on IP address and account name.

F-019 Immediate:** Restrict access to the `/metrics` endpoint to internal networks or authenticated administrators only.

F-020 Short-term:** Disable detailed error messages in production and return generic, standardized HTTP errors.

F-021 Short-term:** Catch exceptions and return generic error pages without backend details.

F-022 Short-term:** Implement global error handling to prevent backend stack traces or logic details from reaching the client.

F-023 Short-term:** Replace the wildcard `*` with a strict whitelist of trusted domains.

F-024 Short-term:** Validate input parameters strictly and suppress detailed ORM error messages.

F-025 Short-term:** Add the `Strict-Transport-Security` header to all HTTP responses.

F-026 Short-term:** Define and implement a robust Content-Security-Policy (CSP) and add a Referrer-Policy header.

Potential-impact estimates are forward-looking and are provided to help prioritise remediation.

Report generated by Backdoor scan-orchestration