

PENTEST LAB

WEB APPLICATION SECURITY ASSESSMENT

脆弱性診断（¥100,000・税込）報告書サンプル

Webアプリケーション脆弱性診断報告書

OWASP Juice Shop（検証用アプリケーション）に対する脆弱性診断（動的／ブラックボックス）の結果

発行者	PENTEST LAB
診断対象	OWASP Juice Shop（検証用アプリケーション）
バージョン	v20.3.0（/metrics・エラーページ等から外部的に特定）
診断手法	脆弱性診断（動的／ブラックボックス）
実施期間	2026-10-12 21:33 JST ～ 2026-10-12 21:50 JST（実検査 約17分）
発行日	2026-10-12
レポートID	PL-JS-SAMPLE-20261012

本書は PENTEST LAB が検証用アプリケーション「OWASP Juice Shop」に対して、ソースコードを参照しない脆弱性診断（動的／ブラックボックス）として実際に実施した結果の**報告書サンプル**です。実在の顧客・案件の診断結果ではありません。実際の顧客診断は、対象・契約範囲により内容が異なります。

本サンプルは有料プラン「脆弱性診断（動的／ブラックボックス）」（¥100,000（税込））の成果物イメージです。CVSS値は本診断の文脈に基づく**参考値（v3.1・自社算定）**です。

CONTENTS

目次

本報告書は全19ページです。

p.1 表紙

p.2 目次

p.3 エグゼクティブサマリー

p.4 診断概要

p.5 検査対象と未検査対象

p.6 脆弱性一覧

p.7 脆弱性の詳細（A01・A02）

p.8 脆弱性の詳細（A03・A04）

p.9 脆弱性の詳細（A05・A06）

p.10 脆弱性の詳細（A07・A08）

p.11 脆弱性の詳細（A09・A10）

p.12 脆弱性の詳細（A11・A12）

p.13 脆弱性の詳細（A13・A14）

p.14 脆弱性の詳細（A15・A16）

p.15 脆弱性の詳細（A17・A18）

p.16 脆弱性の詳細（A19）

p.17 修正の優先順位

p.18 検証証跡（代表例）

p.19 診断の限界

EXECUTIVE SUMMARY

エグゼクティブサマリー

対象URLに対する外部（ブラックボックス）診断の一巡で、調査した26件の候補のうち19件を、外部から再現できる挙動として脆弱性を確認しました。

2

緊急 (Critical)

5

高 (High)

11

中 (Medium)

1

低 (Low)

認証を完全に回避できる重大な問題（SQLインジェクションによる管理者ログイン、JWT署名検証のバイパス）を含み、公開されているエンドポイントだけで管理者権限の奪取・全ユーザー情報や全注文の窃取が成立しました。いずれもソースコードを参照せず、外部からのHTTP／ブラウザ操作のみで実証しています。

特に優先度の高い所見 (Critical／High)

ID	深刻度	名称	CVSS
A01	緊急 (Critical)	SQLインジェクション — 認証バイパス (管理者としてログイン)	9.8
A02	高 (High)	UNION SQLインジェクション — 商品検索経由の認証情報・テーブル窃取	7.5
A03	緊急 (Critical)	JWT署名検証バイパス (alg=none + HS256 鍵取り換え)	9.8
A04	高 (High)	未ソルトのMD5パスワードハッシュ + 脆弱な既定管理者認証情報	5.9
A05	高 (High)	現在のパスワード確認なしでの変更 (+ GETによるCSRF)	8.1

各所見はOWASP Top 10 (2021)・WSTG・CWEに対応づけ、CVSS v3.1で採点した。ASVSは外部から検証できる範囲での対応づけに留める。

※ 検査で問題が見つからなかった範囲について、脆弱性が存在しないことを保証するものではありません。外部からは確認できず追加調査が必要な事項（4件）、誤検知として除外した事項（3件）は「診断の限界」に記載しています。

— METHODOLOGY

診断概要

本診断はソースコード・サーバー内部・ログを一切参照せず、外部クライアントから観測できる挙動のみを根拠とします。

診断手法	脆弱性診断（動的／ブラックボックス）
プラン	脆弱性診断（動的／ブラックボックス）（¥100,000（税込））
診断対象	OWASP Juice Shop（検証用アプリケーション） v20.3.0
対象URL	http://127.0.0.1:3001（ローカル限定）
アクセス方法	OWASP Juice Shop 公式配布版をローカルに起動し、HTTP経由で外部クライアント（ブラウザ／curl）からのみアクセスして診断。ソースコード・サーバー内部・ログには一切アクセスしない（純粋なブラックボックス）。
実施期間	2026-10-12 21:33 JST ～ 2026-10-12 21:50 JST（実検査 約17分）
診断範囲	対象URL（http://127.0.0.1:3001）に限定。非破壊。公開の登録・ログイン機能を使って検証用の一般アカウント（および権限昇格の実証用アカウント）を外部から作成して実施。サーバー内部・ソースコードは参照していない。
診断側環境	診断側クライアント環境: macOS 15.7.4（Darwin 24.6.0, x86_64）／ curl 8.7.1 ／ Chrome
CVSS	v3.1（参考値・自社算定）

使用した主なツール

- curl 8.7.1（HTTPリクエスト／レスポンスの送受信・検査）
- Chrome（ブラウザ上でのDOM型XSS実行の確認）
- Node v24.18.0（受領したJWTのデコード・署名検証バイパス検証用PoC）
- openssl（RS256署名の生成）／python3（証跡の整形）

各所見はOWASP Top 10（2021）・WSTG・CWE に対応づけ、深刻度はCVSS v3.1の参考値で示します。ASVSは外部から検証できる範囲での対応づけに留めます。本サンプルに含まれる値・識別子は検証用のダミー、または伏字です。

— COVERAGE

検査対象と未検査対象

脆弱性診断として実施した領域と、本診断の範囲では未実施の領域を明示します。「実施」は外部から確認した内容、未実施は正直に未実施と記載します。

#	領域	実施した内容／未実施の範囲
1	認証・セッション管理	SQLi認証バイパス・JWT偽造・現在パスワード無しの変更・総当たりを実施。TOTP/2FA・OAuth・セッション固定は未実施。
2	アクセス制御・認可 (IDOR)	カゴBOLA・ユーザー列挙(BFLA)・マスアサインメントを実施。認可マトリクスの網羅は未実施。
3	入力値検証	負の数量を実施。全パラメータのファジングは未実施（部分的）。
4	インジェクション	ログインSQLi・UNION SQLi・NoSQLiを実施。ブラインド/時間差・書き込み/スタック・コマンド/LDAPは未実施。
5	XSS	DOM反射型XSSをブラウザで1件実証。他のXSS課題は未実施。
6	セキュリティ設定	ヘッダー・CORS・/ftp・/encryptionkeys・/metricsを実施。TLSはプレーンHTTPのため未検査。
7	API	認可・マスアサインメント・NoSQLi・列挙を実施。全エンドポイントのレート制限・スキーマファジングは未実施。
8	機密情報の露出	SQLiダンプ・acquisitions.md・Null Byte経由バックアップ・鍵ディレクトリ・metrics・JWTハッシュを実施。ソースマップ/git採掘は未実施。
9	エラーハンドリング	冗長なSQL/バージョンエラーを実施。負系の網羅は未実施。

本診断は特定時点・非破壊の一巡です。TLS・2FA/OAuth・網羅的なファジング・ブラインド/時間差インジェクション等は本サンプルの範囲外です（契約範囲により拡張可能）。

FINDINGS

確認した脆弱性（19件）

外部から再現・実証できた脆弱性の一覧です。詳細は次ページ以降に記載します。

ID	名称	深刻度	該当箇所	CVSS
A01	SQLインジェクション — 認証バイパス（管理者としてログイン）	緊急 (Critical)	POST /rest/user/login (email パラメータ)	9.8
A02	UNION SQLインジェクション — 商品検索経由の認証情報・テーブル窃取	高 (High)	GET /rest/products/search?q=	7.5
A03	JWT署名検証バイパス (alg=none + HS256 鍵取り換え)	緊急 (Critical)	Authorization: Bearer (認証付きREST) 例: /rest/user/authentication-details/	9.8
A04	未ソルトのMD5パスワードハッシュ + 脆弱な既定管理者認証情報	高 (High)	アカウント認証情報の保存 (Users テーブル)	5.9
A05	現在のパスワード確認なしでの変更 (+ GETによるCSRF)	高 (High)	GET /rest/user/change-password?new=&repeat=	8.1
A06	ログインのレート制限・自動化対策が無い（認証情報の総当たり）	中 (Medium)	POST /rest/user/login	5.3
A07	IDOR / BOLA — 他人の買い物カゴへの不正アクセス	中 (Medium)	GET /rest/basket/:id	6.5
A08	マスアサインメント — 登録時の権限昇格 (role=admin)	高 (High)	POST /api/Users/	8.1
A09	機能レベル認可の不備 — 全ユーザーの列挙	中 (Medium)	GET /api/Users/ ・ /api/Users/:id ・ /rest/user/authentication-details/	4.3
A10	NoSQLインジェクション — 注文追跡（全注文の開示）	中 (Medium)	GET /rest/track-order/:id	5.3
A11	商品検索におけるDOMベースの反射型XSS	中 (Medium)	GET /#/search?q= (検索結果画面・クライアント側描画)	6.1
A12	/ftp のディレクトリ一覧 + Poison Null Byte パストラバーサル	高 (High)	GET /ftp ・ GET /ftp/<file>.bak%2500.md	7.5
A13	機密文書の露出 (/ftp/acquisitions.md)	中 (Medium)	GET /ftp/acquisitions.md	5.3
A14	暗号鍵ディレクトリの露出 (/encryptionkeys/)	中 (Medium)	GET /encryptionkeys/ ・ /encryptionkeys/jwt.pub ・ /encryptionkeys/premium.key	5.3
A15	Prometheus /metrics の未認証公開	中 (Medium)	GET /metrics	5.3
A16	セキュリティ設定不備 — セキュリティヘッダー欠落 + ワイルドカードCORS	中 (Medium)	全レスポンス	4.7
A17	不適切な入力検証 — カゴ数量の負値	中 (Medium)	PUT /api/BasketItems/:id (quantity)	5.4
A18	冗長なエラー処理による情報開示	中 (Medium)	GET /rest/products/search?q=<不正値> (汎用Expressエラーハンドラ)	5.3
A19	クライアント側JWTへの機微データ混入（パスワードハッシュ）	低 (Low)	POST /rest/user/login (発行されるJWTのペイロード)	2.6

A01 SQLインジェクション — 認証バイパス（管理者としてログイン）**緊急 (Critical)**

CVSS 9.8

POST /rest/user/login (email パラメータ)

概要

ログイン処理の email 項目がSQL文へ文字列連結されており、正規の認証情報なしで任意アカウント（管理者）としてログインできる。

技術的原因（外部観測からの推定）

email 値をプレースホルダ（バインド変数）を使わずSQL文へ直接連結しているため、`' OR 1=1--` のような入力でWHERE条件を無効化できる。

想定される影響

認証の完全バイパス。攻撃者は管理者を含む任意ユーザーとしてログインし、アプリケーション全体を掌握できる。

検証証跡（外部からの再現）

```
POST /rest/user/login
body: {"email":"' OR 1=1--","password":"anything"}
-> HTTP 200 (認証情報なしで user id 1 / admin / role=admin の有効なJWTを取得)
JWT header: {"typ":"JWT","alg":"RS256"} / payload.data.role = "admin"
(トークン文字列・パスワードハッシュは伏字)
```

修正方法

ログイン検索をパラメータ化クエリ／バインド変数で行い、文字列連結でSQLを組み立てない。email入力の検証・正規化を加える。

CWE CWE-89 **Top10** A03:2021 Injection **WSTG** WSTG-INPV-05 **ASVS** V5.3.4 / V5.3.5**CVSS** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H (9.8) **対応状況** 未対応**A02 UNION SQLインジェクション — 商品検索経由の認証情報・テーブル窃取****高 (High)**

CVSS 7.5

GET /rest/products/search?q=

概要

検索の q パラメータが生のSQL（LIKE句）へ注入され、UNION SELECT により任意テーブルを読み出せる。

技術的原因（外部観測からの推定）

検索語をバインドせずにSQL文へ連結しているため、列数を合わせたUNION SELECTで他テーブル（Users）の内容を検索結果に混入できる。

想定される影響

パスワードハッシュを含む全ユーザーテーブルの開示。原理的には任意のDBテーブルを読み取り可能。

検証証跡（外部からの再現）

```
GET /rest/products/search?q=aaa')) UNION SELECT id,email,password,'4'..'10' FROM Users--
列数=10 と特定 (8/9/11 は SQLITE_ERROR、10 で成功)
-> HTTP 200, Usersテーブル24行が検索結果JSONに混入
id=1 admin@juice-sh.op <パスワードハッシュ (MD5相当) : 伏字>
id=2 jim@juice-sh.op <パスワードハッシュ: 伏字>
... 全24行のemail・権限・ハッシュ項目を取得 (emailはJuice Shopの固定シード値／ハッシュ値は本報告書では非掲載)
```

修正方法

検索クエリをパラメータ化し、DBアクセスは最小権限で行う。汎用的なエラー処理を導入する。

CWE CWE-89 **Top10** A03:2021 Injection **WSTG** WSTG-INPV-05 **ASVS** V5.3.4 **CVSS** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N (7.5)**対応状況** 未対応

A03 JWT署名検証バイパス（alg=none + HS256 鍵取り換え）

緊急 (Critical)

CVSS 9.8

Authorization: Bearer（認証付きREST）例: /rest/user/authentication-details/

概要

JWT検証がアルゴリズムの許可リストを指定しておらず、alg=none や、公開されたRSA公開鍵でHS256署名した偽造トークンが受理される。

技術的原因（外部観測からの推定）

検証側が受理するJWTのアルゴリズムを限定していないと考えられる。実際に、署名なし（alg=none）のトークンと、公開されているRSA公開鍵をHMAC鍵として用いて署名したHS256トークンの双方が検証を通過することを外部から確認した。署名用のRSA公開鍵は GET /encryptionkeys/jwt.pub から誰でも取得できる（A14参照）。

想定される影響

攻撃者は任意ユーザー（管理者を含む）の有効トークンを生成でき、完全ななりすまし／認証バイパスに至る。

検証証跡（外部からの再現）

```
Authorization: Bearer <forged> (alg=none / 空署名)          -> HTTP 200
Authorization: Bearer <forged> (HS256 を公開鍵で署名)      -> HTTP 200
トークン無し                                              -> HTTP 401
（対照）ゴミ文字列 / 誤った鍵で署名したHS256             -> HTTP 401
公開鍵は GET /encryptionkeys/jwt.pub (HTTP 200) から取得。偽造トークンは伏字。
```

修正方法

検証側でアルゴリズムを固定する（jwt.verify(token, publicKey, {algorithms:['RS256']}））。none を拒否し、署名鍵はWebルート外に置く。

CWE CWE-347 **Top10** A02:2021 Cryptographic Failures / A07:2021 Identification & Authentication Failures

WSTG WSTG-SESS-10（JSON Web Tokens） **ASVS** V3.5.3 / V9 **CVSS** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H（9.8） **対応状況** 未対応

A04 未ソルトのMD5パスワードハッシュ + 脆弱な既定管理者認証情報

高 (High)

CVSS 5.9

アカウント認証情報の保存（Users テーブル）

概要

A02 のSQLiで窃取した管理者のパスワードハッシュが、ソルト無し的高速ハッシュ（MD5相当）であり、かつ既定の管理者パスワードが脆弱（admin123）であることを、オフライン解析により外部から実証した。

技術的原因（外部観測からの推定）

窃取したハッシュが、よく知られた脆弱なパスワードのMD5値とそのまま一致したことから、保存されているパスワードがソルト無し的高速ハッシュで、辞書・レインボーテーブル攻撃に弱いと外部から判断できる（ソースコードは参照していない）。

想定される影響

ハッシュが一度でも漏えいすると（本件ではA02のSQLiで現に漏えい）、ソルトが無く高速なため大量の認証情報回復が容易になる。

検証証跡（外部からの再現）

```
A02 の UNION SQLi で管理者アカウントのパスワードハッシュ（32桁の16進＝MD5相当の長さ）を取得
取得したハッシュ値が md5('admin123') と完全一致（ハッシュ値・一致の計算結果は本報告書では非掲載）
=> ソルト無し・高速ハッシュ + 脆弱な既定管理者パスワードであることを外部から確認
```

修正方法

MD5をbcrypt/scrypt/Argon2id（ユーザーごとのソルト＋適切なワークファクタ）へ置き換える。パスワードポリシーを強制し、シード/既定パスワードを更新する。

CWE CWE-916 / CWE-759 / CWE-521 **Top10** A02:2021 Cryptographic Failures **WSTG** WSTG-ATHN-07 / WSTG-CRYP **ASVS** V2.4.1 / V2.1

CVSS CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N（5.9） **対応状況** 未対応

A05 現在のパスワード確認なしでの変更（+ GETによるCSRF）

高 (High)

CVSS 8.1

GET /rest/user/change-password?new=&repeat=

概要

パスワード変更エンドポイントが現在のパスワードを要求・検証せず、状態変更をCSRFトークン無しのGETで受け付ける。

技術的原因（外部観測からの推定）

current パラメータの検証が無く、かつ副作用のある操作をGETで実行しているため、トークンの盗用・再送やクロスサイトからの強制実行が成立する。

想定される影響

トークンが盗まれると即座にアカウント乗っ取り。GET+CSRFトークン無しのため、罠サイト経由で被害者のパスワードを強制変更できる。

検証証跡（外部からの再現）

```
GET /rest/user/change-password?new=NewPass123!&repeat=NewPass123!  
(current パラメータ無し / Bearer=diaguser2)  
-> HTTP 200 (現在のパスワード確認なしで変更成立)  
その後、新パスワードでのログイン -> HTTP 200
```

修正方法

現在のパスワードの入力・検証（または再認証／ステップアップ）を必須にする。POST/PUT化し、CSRFトークンと SameSite Cookie を用いる。

CWE CWE-620 / CWE-352 **Top10** A07:2021 Identification & Authentication Failures / A01:2021

WSTG WSTG-ATHN-09 / WSTG-SESS-05 (CSRF) **ASVS** V2.5 / V4.2.2 **CVSS** CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N (8.1)

対応状況 未対応

A06 ログインのレート制限・自動化対策が無い（認証情報の総当たり）

中 (Medium)

CVSS 5.3

POST /rest/user/login

概要

連続したログイン失敗に対して、ロックアウト・スロットリング・CAPTCHA・429 のいずれも無い。

技術的原因（外部観測からの推定）

ログイン試行回数を制限・監視する仕組みが実装されていない。

想定される影響

オンラインでの認証情報の総当たり／パスワードプレーが無制限に可能。

検証証跡（外部からの再現）

```
POST /rest/user/login を誤ったパスワードで12回連続  
応答: 401 ×12  
429・アカウントロックアウト・遅延・CAPTCHA はいずれも観測されず
```

修正方法

アカウント単位+IP単位のレート制限、段階的な遅延／ロックアウト、N回失敗後のCAPTCHAを導入する。

CWE CWE-307 **Top10** A07:2021 Identification & Authentication Failures **WSTG** WSTG-ATHN-03 **ASVS** V2.2.1

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3) **対応状況** 未対応

A07 IDOR / BOLA — 他人の買い物カゴへの不正アクセス

中 (Medium)

CVSS 6.5

GET /rest/basket/:id

概要

JWTの主体と要求されたカゴIDの所有関係を確認しておらず、他人のカゴを閲覧できる。

技術的原因（外部観測からの推定）

オブジェクトレベルの認可（所有者チェック）がサーバー側に無く、:id を変えるだけで他人の資源にアクセスできる。

想定される影響

任意の顧客がIDを総当たりして他ユーザーのカゴを閲覧（および操作）できる。

検証証跡（外部からの再現）

```
Bearer = diaguser1 (userId=25、自分のカゴは id 6)
GET /rest/basket/1 -> UserId=1(admin) のカゴ: Apple / Orange / Eggfruit Juice (3点)
GET /rest/basket/2 -> UserId=2(jim) のカゴ: Raspberry Juice
=> 所有者チェックが無く :id の総当たりで他人のカゴを閲覧可能
```

修正方法

オブジェクトレベルの認可を強制する。カゴIDが認証済みユーザーに属することをサーバー側で必ず確認する。

CWE CWE-639 **Top10** A01:2021 Broken Access Control / API1:2023 BOLA **WSTG** WSTG-ATHZ-04 **ASVS** V4.2.1

CVSS CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N (6.5) **対応状況** 未対応

A08 マスアサインメント — 登録時の権限昇格 (role=admin)

高 (High)

CVSS 8.1

POST /api/Users/

概要

登録エンドポイントがクライアント指定の role 項目を受理し、そのまま保存する。

技術的原因（外部観測からの推定）

登録APIがクライアント送信の role 項目をそのまま受理・保存しており、サーバー側で書き込み可能な項目を制限（許可リスト化）していないことが外部から確認できる。

想定される影響

セルフサービスでの管理者アカウント作成（権限昇格）。アクセス制御の破綻につながる。

検証証跡（外部からの再現）

```
POST /api/Users/ body に "role":"admin" を付与
-> HTTP 201, data.id=27 / data.role=admin
=> 自己登録したアカウントに管理者権限が付与された（本来は customer 固定であるべき）
```

修正方法

書き込み可能フィールドをサーバー側で許可リスト化する。自己登録では role='customer' を強制し、リクエストボディを直接モデルにバインドしない。

CWE CWE-915 **Top10** A01:2021 Broken Access Control / A08:2021 **WSTG** WSTG-BUSL-01 **ASVS** V5.1.1 / V5.1.2

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N (8.1) **対応状況** 未対応

A09 機能レベル認可の不備 — 全ユーザーの列挙

中 (Medium)

CVSS 4.3

GET /api/Users/ • /api/Users/:id • /rest/user/authentication-details/

概要

任意の認証済み顧客が全ユーザーアカウント（email・role・lastLoginIp・isActive）を一覧・参照できる。

技術的原因（外部観測からの推定）

ユーザー一覧／管理系エンドポイントが管理者に限定されておらず、自分以外のレコードも返す。

想定される影響

低権限ユーザーによるアカウント／権限の列挙とメールアドレス（PII）の収集。

検証証跡（外部からの再現）

```
未認証: GET /api/Users/ -> 401
顧客トークン: GET /api/Users/ -> 200 (26名の全ユーザー: id/email/role/lastLoginIp/isActive)
GET /api/Users/2 -> jim@juice-sh.op を取得 (password項目は省略／マスク)
```

修正方法

ユーザー集合／管理エンドポイントを管理者に限定し、自己参照系は呼び出し元自身のレコードのみ返す。

CWE CWE-285 / CWE-639 **Top10** A01:2021 Broken Access Control / API5:2023 BFLA **WSTG** WSTG-ATHZ-02 **ASVS** V4.1.3

CVSS CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N (4.3) 対応状況 未対応

A10 NoSQLインジェクション — 注文追跡（全注文の開示）

中 (Medium)

CVSS 5.3

GET /rest/track-order/:id

概要

orderId がNoSQLの条件式（\$where 相当）に注入され、全注文を返すよう改変できる。

技術的原因（外部観測からの推定）

ユーザー入力が無条件にNoSQLの条件式として評価されており、常に真となる式を与えることで絞り込みを回避できる（外部からの注入で全件返却を確認）。

想定される影響

全顧客の注文レコードの開示。フィルタのバイパス。

検証証跡（外部からの再現）

```
GET /rest/track-order/abc123 -> 1件 (echo)
GET /rest/track-order/' || true || ' (URLエンコード) -> 保存済みの全注文3件を返却
orderId / totalPrice / products / マスク済メール
```

修正方法

安全なクエリ演算子とバインド値を用い、\$where/JS評価にユーザー入力を渡さない。orderId の形式を検証する。

CWE CWE-943 **Top10** A03:2021 Injection **WSTG** WSTG-INPV-05 (NoSQL) **ASVS** V5.3.4

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3) 対応状況 未対応

A11 商品検索におけるDOMベースの反射型XSS

中 (Medium)

CVSS 6.1

GET /#/search?q= (検索結果画面・クライアント側描画)

概要

検索語がクライアント側でHTMLとして解釈・描画され、攻撃者のマークアップ／スクリプトがアプリのオリジンで実行される。

技術的原因（外部観測からの推定）

検索語に対するクライアント側の出力エスケープ（サニタイズ）が不十分で、入力したHTMLがそのまま描画・実行される。ブラウザ上で任意のJavaScriptがアプリのオリジンで実行されることを外部から確認した。

想定される影響

JWTがブラウザのlocalStorageに保存されているため、トークン窃取／アカウント乗っ取り、UIの偽装、被害者コンテキストでのリクエスト偽造が可能。

検証証跡（外部からの再現）

```
http://127.0.0.1:3001/#/search?q=<iframe src="javascript:console.log('XSS_PWNED_'+document.domain)">
ブラウザ(Chrome)コンソール: [LOG] XSS_PWNED_127.0.0.1
=> アプリのオリジンで任意のJavaScriptが実行された
(alert(`xss`)) でも同様に動作。ブロックを避けるため console.log を使用)
```

修正方法

ユーザー入力をHTMLとして信頼せず、テキストとして描画するか適切にエスケープ／サニタイズする。フレームワーク既定のエスケープを無効化しない。厳格なCSPを導入してスクリプト実行を制限する。

CWE CWE-79 **Top10** A03:2021 Injection (XSS) **WSTG** WSTG-CLNT-01 (DOM XSS) **ASVS** V5.3.3

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N (6.1) **対応状況** 未対応

A12 /ftp のディレクトリ一覧 + Poison Null Byte パストラバーサル

高 (High)

CVSS 7.5

GET /ftp ・ GET /ftp/<file>.bak%2500.md

概要

/ftp がディレクトリ一覧可能であり、%00 (Poison Null Byte) + .md 拡張子の付加で拡張子の許可リストを回避し、ブラックリスト対象の.bak ファイルを取得できる。

技術的原因（外部観測からの推定）

拡張子チェックがデコード後のフルパスではなく末尾のみを見ているため、null バイトで本来の拡張子を隠蔽できる。ディレクトリ一覧も有効。

想定される影響

バックアップ／秘匿ファイル（クーポンコード、依存関係バージョン＝CVE標的化の手掛かり）の露出。

検証証跡（外部からの再現）

```
GET /robots.txt -> Disallow: /ftp
GET /ftp -> ディレクトリ一覧 (acquisitions.md / coupons_2013.md.bak / package.json.bak ...)
GET /ftp/coupons_2013.md.bak -> 403 "Only .md and .pdf files are allowed!"
GET /ftp/coupons_2013.md.bak%2500.md -> 200 (Null Byteで拡張子チェックを回避)
GET /ftp/package.json.bak%2500.md -> 200 (依存関係マニフェストを取得)
```

修正方法

ディレクトリ一覧を無効化し、バックアップ等の静的ファイルをWebルートから除去する。null バイトを拒否し、末尾ではなくデコード後のフルパスを検証する。

CWE CWE-22 / CWE-158 **Top10** A01:2021 Broken Access Control / A05:2021 **WSTG** WSTG-ATHZ-01 (path traversal) **ASVS** V12.3.1 / V5.1.1

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N (7.5) **対応状況** 未対応

A13 機密文書の露出 (/ftp/acquisitions.md)

中 (Medium)

CVSS 5.3

GET /ftp/acquisitions.md

概要

「confidential」と明記された文書が、認証なしで直接ダウンロードできる。

技術的原因（外部観測からの推定）

機密文書が公開の静的ディレクトリから配信されており、アクセス制御が無い。

想定される影響

機微なビジネス情報（M&A計画）の漏えい。

検証証跡（外部からの再現）

GET /ftp/acquisitions.md → HTTP 200（認証なし）

本文: "# Planned Acquisitions ... This document is confidential! Do not distribute!"

修正方法

機密文書を公開静的ディレクトリから配信しない。文書アクセスに認可を要求する。

CWE CWE-200 / CWE-540 **Top10** A05:2021 Security Misconfiguration / A01:2021 **WSTG** WSTG-CONF-04 **ASVS** V12.3

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3) **対応状況** 未対応

A14 暗号鍵ディレクトリの露出 (/encryptionkeys/)

中 (Medium)

CVSS 5.3

GET /encryptionkeys/ • /encryptionkeys/jwt.pub • /encryptionkeys/premium.key

概要

鍵ディレクトリが一覧可能で、鍵素材を認証なしで配信している。

技術的原因（外部観測からの推定）

署名に関わる公開鍵等の鍵素材をWebルート配下に置き、ディレクトリ一覧も有効になっている。

想定される影響

jwt.pub はJWT偽造（A03）を直接可能にする。premium.key はプレミアムコンテンツ保護を露出する。

検証証跡（外部からの再現）

GET /encryptionkeys/ → 200 ディレクトリ一覧: jwt.pub, premium.key

GET /encryptionkeys/jwt.pub → 200 (JWT署名用RSA公開鍵 → A03のforgeryを可能にする)

GET /encryptionkeys/premium.key → 200

修正方法

鍵素材をWebルート外へ移す。ディレクトリ一覧を無効化し、公開が必須のもの以外（特に署名関連）は配信しない。

CWE CWE-548 / CWE-200 **Top10** A05:2021 Security Misconfiguration **WSTG** WSTG-CONF-04 / WSTG-CONF-05 **ASVS** V14.3.2

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3) **対応状況** 未対応

A15 Prometheus /metrics の未認証公開

中 (Medium)

CVSS 5.3

GET /metrics

概要

メトリクスのエンドポイントが公開されており、ランタイムのバージョンや運用カウンタを開示する。

技術的原因（外部観測からの推定）

/metrics に認証・ネットワーク制限がかかっていない。

想定される影響

正確なランタイムバージョンはCVE標的化を助け、運用・ビジネス上のカウンタも漏えいする。

検証証跡（外部からの再現）

```
GET /metrics -> 200 (認証なし)
nodejs_version_info v24.18.0 / process_start_time_seconds /
http_requests_count / users_registered / orders_placed などを公開
```

修正方法

/metrics に認証・ネットワーク制限をかけ、バージョンラベルを除去する。

CWE CWE-200 **Top10** A05:2021 Security Misconfiguration (Observability) **WSTG** WSTG-CONF-05 **ASVS** V14.3

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3) **対応状況** 未対応

A16 セキュリティ設定不備 — セキュリティヘッダー欠落 + ワイルドカードCORS

中 (Medium)

CVSS 4.7

全レスポンス

概要

CSP / HSTS / Referrer-Policy / Permissions-Policy が無く、APIで Access-Control-Allow-Origin: * を返す。

技術的原因（外部観測からの推定）

防御的なレスポンスヘッダーが設定されておらず、CORSが任意オリジンを許可している。

想定される影響

CSPの欠落はXSS (A11) の悪用を容易にする。ワイルドカードCORSは任意オリジンからのAPI読み取りを許す (Allow-Credentials が無いため資格情報付き読み取りは限定的)。

検証証跡（外部からの再現）

```
レスポンスヘッダ: Access-Control-Allow-Origin: * (APIで反射)
欠落: Content-Security-Policy / Strict-Transport-Security / Referrer-Policy / Permissions-Policy
存在: X-Frame-Options: SAMEORIGIN, X-Content-Type-Options: nosniff
Access-Control-Allow-Credentials は無し
```

修正方法

厳格なCSP、HSTS (TLS下)、Referrer-Policy、Permissions-Policy を付与する。CORSは信頼するオリジンの許可リストに限定する。

CWE CWE-693 / CWE-942 / CWE-1021 **Top10** A05:2021 Security Misconfiguration **WSTG** WSTG-CONF-07 / WSTG-CLNT-07

ASVS V14.4 / V14.5 **CVSS** CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N (4.7) **対応状況** 未対応

A17 不適切な入力検証 — カゴ数量の負値

中 (Medium)

CVSS 5.4

PUT /api/BasketItems/:id (quantity)

概要

カゴ内商品の数量が負の値を受理し、そのまま保存されることを外部から確認した。

技術的原因（外部観測からの推定）

数量に対するサーバー側の値域検証が無く、負の値がそのまま保存される（会計・合計金額への反映可否は本診断の範囲では未検証）。

想定される影響

負の数量が有効となるため、会計ロジック次第では注文合計の不正操作・割引悪用につながり得る（想定される影響）。

検証証跡（外部からの再現）

```
POST /api/BasketItems/ {"ProductId":1,"BasketId":"6","quantity":1} -> itemId=9
PUT /api/BasketItems/9 {"quantity":-100} -> HTTP 200
GET /api/BasketItems/9 -> 保存された quantity = -100（検証後に1へ戻した）
```

修正方法

数量は正の整数かつ妥当な範囲であることをサーバー側で検証し、合計はサーバー側で再計算する。

CWE CWE-20 / CWE-840 **Top10** A04:2021 Insecure Design **WSTG** WSTG-BUSL-01 / WSTG-INPV-01 **ASVS** V5.1.3

CVSS CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N (5.4) **対応状況** 未対応

A18 冗長なエラー処理による情報開示

中 (Medium)

CVSS 5.3

GET /rest/products/search?q=<不正値>（汎用Expressエラーハンドラ）

概要

サーバーエラー時に、生のDBエラーメッセージとフレームワーク／バージョンをHTMLエラーページで返す。

技術的原因（外部観測からの推定）

本番相当のエラー処理が設定されておらず、内部エラーの詳細を利用者に返している。

想定される影響

DBエンジン・フレームワークのバージョンを明かし、インジェクション箇所の特定を助ける。

検証証跡（外部からの再現）

```
GET /rest/products/search?q=<列数不一致のUNION>
-> HTTP 500, エラーページに以下を露出:
"SQLITE_ERROR: SELECTs to the left and right of UNION do not have the same number..."
"OWASP Juice Shop (Express ^4.22.1)"
（本件の漏えいはエラーメッセージとフレームワーク名・バージョンに限られ、ファイルパスを含むスタックトレース一覧の表示は確認されなかった → C02参照）
```

修正方法

汎用的なエラー応答を返し、詳細はサーバー側のみに記録する。本番用のエラー処理を設定する。

CWE CWE-209 **Top10** A05:2021 Security Misconfiguration **WSTG** WSTG-ERRH-01 / WSTG-ERRH-02 **ASVS** V7.4.1

CVSS CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N (5.3) **対応状況** 未対応

A19 クライアント側JWTへの機微データ混入（パスワードハッシュ）

低 (Low)

CVSS 2.6

POST /rest/user/login（発行されるJWTのペイロード）

概要

発行されるJWTの data オブジェクトにユーザーのパスワードハッシュ項目が含まれ、ブラウザのlocalStorageに保存される。

技術的原因（外部観測からの推定）

JWTに最小限の主張のみを入れる設計になっておらず、ユーザーレコードをほぼそのまま格納している。

想定される影響

XSS（A11）や未ソルトMD5（A04）と組み合わせると、クライアント側でのハッシュ露出が認証情報の侵害を助ける。

検証証跡（外部からの再現）

```
POST /rest/user/login が返すJWTの data に password (MD5ハッシュ) 項目を含む
{"data":{"id":1,"email":"admin@juice-sh.op","password":"<伏字>","role":"admin",...}}
JWTはブラウザのlocalStorageに保存 (XSS A11 で読み取り可能)
```

修正方法

JWTには最小限かつ非機微な主張（id, role 等）のみを入れ、パスワードハッシュや秘密を埋め込まない。

CWE CWE-312 / CWE-522 **Top10** A02:2021 Cryptographic Failures / A04:2021 **WSTG** WSTG-SESS-10 **ASVS** V3.5 / V8.1

CVSS CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:N/A:N (2.6) **対応状況** 未対応

— REMEDIATION PLAN

修正の優先順位

深刻度に加え、悪用のしやすさ・影響を受ける情報の重要性・他の問題を可能にする度合いを考慮した実務順です。

第1段階 — 即時

最優先・即時に着手

A01 / A03 / A14

認証の完全バイパス（SQLi認証バイパス・JWT署名バイパス）と、それを直接可能にする鍵ディレクトリの露出。アプリ全体の掌握につながるため最優先。A14（公開鍵の露出）はA03の前提を断つため同時に対応する。

第2段階 — 高（早期）

数日～短期で対応

A08 / A12 / A02 / A05 / A04

未認証での権限昇格（マスアサインメント）、機微ファイルの取得（Null Byte トラバーサル）、認証情報の窃取（UNION SQLi）、現在パスワード確認なしの変更、未ソルトMD5。被害が重大・悪用も比較的容易なため早期に対応。

第3段階 — 中（計画的）

改修計画に組み込み対応

A07 / A11 / A16 / A10 / A09 / A17 / A13 / A15 / A18 / A06

アクセス制御（IDOR・列挙）、XSSとその悪用を助けるヘッダー/CORS不備、NoSQLi、業務ロジック（負の数量）、情報露出、レート制限不足。単独の影響は限定的でも、連鎖で影響が拡大するため計画的に対応する。

第4段階 — 低

上記対応に合わせて実施

A19

JWTへのパスワードハッシュ混入。JWT設計の見直し（A03対応）に合わせて最小クレーム化する。

— EVIDENCE

検証証跡（代表例）

外部から再現できたリクエスト／レスポンスの抜粋です（秘密情報・ハッシュ値・トークンは伏字）。全所見の証跡は各詳細ページに記載しています。

A01 SQLインジェクション — 認証バイパス（管理者としてログイン） POST /rest/user/login (email パラメータ)

```
POST /rest/user/login
body: {"email":"' OR 1=1--","password":"anything"}
-> HTTP 200 (認証情報なしで user id 1 / admin / role=admin の有効なJWTを取得)
JWT header: {"typ":"JWT","alg":"RS256"} / payload.data.role = "admin"
(トークン文字列・パスワードハッシュは伏字)
```

A03 JWT署名検証バイパス (alg=none + HS256 鍵取り違い) Authorization: Bearer (認証付きREST) 例: /rest/user/authentication-details/

```
Authorization: Bearer <forged> (alg=none / 空署名)          -> HTTP 200
Authorization: Bearer <forged> (HS256 を公開鍵で署名)      -> HTTP 200
トークン無し                                              -> HTTP 401
(対照) ゴミ文字列 / 誤った鍵で署名したHS256              -> HTTP 401
公開鍵は GET /encryptionkeys/jwt.pub (HTTP 200) から取得。偽造トークンは伏字。
```

A08 マスアサインメント — 登録時の権限昇格 (role=admin) POST /api/Users/

```
POST /api/Users/ body に "role":"admin" を付与
-> HTTP 201, data.id=27 / data.role=admin
=> 自己登録したアカウントに管理者権限が付与された（本来は customer 固定であるべき）
```

A12 /ftp のディレクトリ一覧 + Poison Null Byte パストラバーサル GET /ftp • GET /ftp/<file>.bak%2500.md

```
GET /robots.txt -> Disallow: /ftp
GET /ftp -> ディレクトリ一覧 (acquisitions.md / coupons_2013.md.bak / package.json.bak ...)
GET /ftp/coupons_2013.md.bak -> 403 "Only .md and .pdf files are allowed!"
GET /ftp/coupons_2013.md.bak%2500.md -> 200 (Null Byteで拡張子チェックを回避)
GET /ftp/package.json.bak%2500.md -> 200 (依存関係マニフェストを取得)
```

— LIMITATIONS

診断の限界

脆弱性診断（ブラックボックス）として実証できた範囲と、できなかった範囲を正直に区別します。

追加調査が必要な事項（4件・外部からは影響を確定できず）

B01 プロフィール画像URL取得によるSSRFの可能性 **CWE-918**

POST /profile/image/url (imageUrl, cookie認証)

サーバー側がユーザー指定URLを取得する挙動を確認（ローカルURLで302）。ただし内部限定リソースへの到達や情報持ち出しは未確認。影響の実証には帯域外／内部限定の標的が必要。

B02 アプリケーション設定の未認証開示 **CWE-200**

GET /rest/admin/application-configuration

サーバー設定オブジェクト（baseUrl・chatbotモデル・ctfブロック等）を認証なしで返す。Juice Shopでは設計上公開だが、実デプロイの設定に秘密が含まれるかで深刻度が変わるため、意図・内容を要確認（内容は保存回避のため非ダンプ）。

B03 SQLiの書き込み／スタッククエリ／二次注入の可能性 **CWE-89**

ログイン・検索の注入点

注入は読み取り（認証バイパス+UNION読み出し）までを実証。書き込み／スタッククエリ／二次注入は理論上あり得るが未実証（sqlite3ドライバは通常マルチステートメントを許可しない）。追加の的を絞った検証が必要。

B04 ワイルドカードCORSによる資格情報付きクロスオリジン読み取り **CWE-942**

API の ACA0:*

ワイルドカードCORSは確認（A16）。ただし Access-Control-Allow-Credentials が無いため、ブラウザからの資格情報付き読み取りは自動的に成立しない。実害にはトークン漏えいの別経路が必要で、デプロイ毎に要確認。

誤検知・対象外（3件）

C01 B2Bファイルアップロード経由のXXE

v20.3.0 では脆弱でない。悪意あるXMLを POST /file-upload すると HTTP 410（機能は廃止）。当初の想定を実測で否定した。

C02 フルパスのスタックトレース開示

本環境では該当なし（エラーページにファイルパスを含むスタックトレース一覧の表示は確認されず）。外部から観測できた漏えいはエラーメッセージ+フレームワーク名・バージョンのみで、A18として正確に記録した。過大申告を回避。

C03 ブラウザ限定／外部サービス／暗号／ステガノ／コーディング系の課題

curl+ブラウザによる脆弱性診断（外部・ブラックボックス）の一巡では対象外。一部機能（外部web3サービス連携・外部LLMに依存するチャットボット）は本環境で外部サービスが応答せず検査不可。深い暗号・ステガノグラフィ・SPA操作に強く依存する課題は未実施。

本診断は特定時点・非破壊の脆弱性診断（外部・ブラックボックス）です。「安全である」ことを保証するものではなく、未検査・未実施の領域に脆弱性が無いことを示すものでもありません。ソースコードレビューを伴う内部診断や、契約範囲を広げた継続的な診断で、より深い検証が可能です。