# セキュリティ診断レポート

Security Assessment Report — Sample

**ちいさなIT屋さん**
発行日：2026年3月7日
対象：架空アプリケーション（サンプル）
種別：ソースコード診断

| 診断対象 | 診断種別 | 使用技術 | 発行者 |
|---|---|---|---|
| **架空のWebアプリケーション（サンプル）** | **ソースコードレビュー（手動）** | **Laravel / Next.js** | **ちいさなIT屋さん** |

## 診断結果サマリ

| CRITICAL | HIGH | MEDIUM | LOW |
|---|---|---|---|
| 3 件 | 2 件 | 2 件 | 0 件 |

本レポートはソースコードレビューに基づく診断であり、すべての脆弱性の検出や安全性の保証を行うものではありません。 ヒアリング情報およびコード精査の範囲内で確認できた事項を報告しています。

## 指摘事項

### Critical 1. アクセストークンが平文で保存されている

| | |
|---|---|
| 問題箇所 | DBの access_tokens テーブルにトークン値が平文保存されている |
| なぜ問題か | DB漏洩・ログ漏洩が発生した場合、トークンをそのまま不正ログインに悪用される |
| 改善の方向性 | トークンは復元不能な形（ハッシュ等）で保管し、失効・期限管理を行う |

### Critical 2. ユーザー入力HTMLが無検証で表示される（XSS）

| | |
|---|---|
| 問題箇所 | プロフィール・投稿本文がサニタイズなしでレンダリングされている |
| なぜ問題か | 管理画面を含む任意ユーザーのブラウザ上でスクリプトが実行され、トークン窃取・画面操作が可能になる |
| 改善の方向性 | 表示コンテキストに応じたエスケープ／サニタイズ方針を定義し、許可タグのみを通す制御を実装する |

### Critical 3. 保護APIの認可（Authorization）テストが存在しない

| | |
|---|---|
| 問題箇所 | API疎通とシナリオテストは存在するが、権限違いによる拒否ケースのテストが不足している |
| なぜ問題か | 認可漏れは気づきにくく、本番公開後にデータ漏洩や不正操作につながりやすい |
| 改善の方向性 | 重要APIについて「許可される／拒否される」両面のテストを用意し、ロール・テナント境界を含めて検証する |

### High 4. Laravelで生SQLが多用され、入力処理次第でSQLインジェクションリスク

| | |
|---|---|
| 問題箇所 | 文字列結合によるクエリ生成が複数箇所に散見される |
| なぜ問題か | パラメータ処理の揺れによりSQLインジェクションの温床となる。保守性の低下も伴う |
| 改善の方向性 | クエリ生成を統一し（パラメータバインド徹底、ORM / QueryBuilder等の利用）、危険パターンを集中排除する |

### High 5. CORS設定が Access-Control-Allow-Origin: *（全オリジン許可）

| | |
|---|---|
| 問題箇所 | 全エンドポイントでワイルドカード許可が設定されている |
| なぜ問題か | フロント分離構成では意図しないオリジンからのAPIアクセスや誤設定誘発につながる |
| 改善の方向性 | 許可オリジンを限定し、認証方式（Cookie / Token）と整合したCORS設計に変更する |

### Medium 6. Next.jsのバージョンが古く、既知脆弱性の影響範囲が不明

| | |
|---|---|
| 問題箇所 | package.json 上で長期間更新されていない |
| なぜ問題か | 既知脆弱性が修正されていない可能性があり、依存関係も連鎖的に古い状態になっている |
| 改善の方向性 | 更新計画（頻度・担当・手順）を定め、段階的にアップデートする |

### Medium 7. パスワードポリシー・防御（レート制限等）が不足

| | |
|---|---|
| 問題箇所 | 長さ・複雑性・ログイン失敗回数制限などの設定が不十分 |
| なぜ問題か | 総当たり攻撃・リスト型攻撃に対して脆弱な状態になっている |
| 改善の方向性 | パスワードポリシーと防御機能（レート制限・アカウントロック・監視）をセットで設計する |