実務説明・診断範囲整理用　|　2026 年 8 月版

> **要点**　URL 診断は「外部から実際に成立する問題」の確認に強く、ソースコード診断は「未到達の処理を含む潜在的な欠陥」と「発生原因」の特定に強みがあります。最も確実なのは両方を組み合わせる方法です。

1. 診断方法の違い

URL 診断（ブラックボックス診断）は、攻撃者と同じ外部視点から、公開されている画面・API・通信を検証します。ソースコード診断（ホワイトボックス診断）は、入力から重要処理までの流れ、権限判定、例外経路、設定・依存関係を内部から確認します。

比較項目	URL だけの診断	ソースコード診断
主な目的	外部から問題が実際に成立するか確認	危険な実装と原因を網羅的に特定
得意な範囲	公開画面、API、通信、認証・認可の挙動	非公開処理、例外経路、権限分岐、設定、依存関係
主な成果	再現手順、影響、外部露出の確認	問題箇所、データフロー、横展開範囲、修正案
主な限界	内部処理や到達しにくい条件を見落とす可能性	本番環境固有の設定・実際の攻撃成立性は別確認が必要

診断結果の読み方

- **URL 診断で発見：**現在の公開環境で再現できる可能性が高く、優先的な修正・緩和が必要です。
- **コード診断で発見：**現時点で外部から再現できなくても、条件変更や別画面から到達する潜在リスクとして対処します。
- **両方で発見：**実装上の原因と実際の影響を結び付けられるため、修正範囲と再テスト条件を明確にできます。

2. URL だけで発見しやすい脆弱性

以下は、公開 URL、画面操作、HTTP 通信、API 応答などから確認しやすい代表例です。認証が必要な機能は、診断用アカウントと権限別アカウントがあるほど検証範囲を広げられます。

1. HTTPS・TLS 設定の不備

- **発見できること：**HTTP 通信の残存、証明書の期限・ホスト名不一致、弱いプロトコルや暗号スイート、HTTPS への強制転送不足を確認できます。
- **具体例：**ログイン画面は HTTPS でも、画像・API・ダウンロードが HTTP のままになっており、通信内容や Cookie が盗聴される可能性がある状態です。
- **URL 診断の限界：**ロードバランサーや内部区間の暗号化、証明書更新処理の設計までは通常確認できません。

2. セキュリティヘッダーの不足

- **発見できること：**CSP、HSTS、frame-ancestors、X-Content-Type-Options、Referrer-Policy などの有無と設定内容を確認できます。
- **具体例：**画面が外部サイトの iframe 内に表示でき、利用者に見えない形でボタンを押させるクリックジャッキングが成立する状態です。
- **確認ポイント：**画面ごとのヘッダー差異、リダイレクト応答、エラー画面、静的ファイルにも同じ方針が適用されているかを確認します。

3. Cookie・セッション設定の不備

- **発見できること：**Cookie の Secure、HttpOnly、SameSite、有効期限、スコープ、ログアウト後のセッション無効化を確認できます。

- **具体例：**ログアウト後も取得済みセッション ID が利用できる、または JavaScript から認証 Cookie を読み取れる状態です。
- **確認ポイント：**ログイン前後でセッション ID が変わるか、パスワード変更や権限変更後に既存セッションが失効するかも検証します。

4. 認証機能の不備

- **発見できること：**ログイン、パスワード再設定、多要素認証、アカウントロック、エラーメッセージの問題を確認できます。
- **具体例：**存在するメールアドレスだけ異なるエラーが表示され、登録済み利用者を第三者が判別できるアカウント列挙です。
- **確認ポイント：**初期パスワード、招待 URL、再設定トークンの期限・再利用、認証済み操作での再認証要否を確認します。

5. 認可不備・IDOR

- **発見できること：**URL や API の ID を書き換えた際に、他人・他部署・他組織のデータを閲覧、更新、削除できないかを確認できます。
- **具体例：**請求書 URL の番号を変更すると、別利用者の請求書 PDF を取得できる状態です。
- **確認ポイント：**画面表示だけでなく、ダウンロード、CSV 出力、添付ファイル、履歴、管理 API まで権限境界を横断して確認します。

6. SQL インジェクション等の入力注入

- **発見できること：**検索、絞り込み、ログイン、並び替え、API パラメータなどに不正な入力を与え、DB 処理へ影響するかを確認できます。

- **具体例：**検索条件の解釈が変わり、本来表示されないレコードが返る、または DB エラー情報が応答へ露出する状態です。
- **URL 診断の限界：**エラーを隠す実装や到達しにくい処理では検出が難しく、危険な SQL 生成箇所を網羅できない場合があります。

7. クロスサイトスクリプティング（XSS）

- **発見できること：**URL、フォーム、プロフィール、検索結果、管理画面、エラー表示などへ入力した文字列が、ブラウザでスクリプトとして解釈されないかを確認できます。
- **具体例：**問い合わせ内容を閲覧した管理者のブラウザで不正処理が実行され、管理者権限の操作につながる格納型 XSS です。
- **確認ポイント：**HTML 本文、属性、URL、JavaScript、CSS など、出力される文脈ごとに安全性を確認します。

8. CSRF（意図しない操作の実行）

- **発見できること：**ログイン中の利用者を外部ページへ誘導し、設定変更、送信、削除などを意図せず実行させられないかを確認できます。
- **具体例：**攻撃者が用意したページを開くだけで、登録メールアドレスや送金先が変更される状態です。
- **確認ポイント：**CSRF トークン、Origin 検証、SameSite Cookie、重要操作の再認証が組み合わされているかを確認します。

9. ファイルアップロードの不備

- **発見できること：**許可されていない形式、偽装 MIME、巨大ファイル、危険なファイル名、公開領域への保存などを確認できます。

- **具体例：**画像としてアップロードしたファイルが実行可能な状態で公開され、サーバー上の処理実行につながる構成です。
- **確認ポイント：**拡張子、MIME、実内容、サイズ、保存名、保存先、ダウンロード時の Content-Type を確認します。

10. パストラバーサル・任意ファイル参照

- **発見できること：**ダウンロード名、画像名、テンプレート名などを操作して、想定外のファイルを読み書きできないかを確認できます。
- **具体例：**ダウンロード API のファイル名を変更すると、設定ファイルやログを取得できる状態です。
- **確認ポイント：**URL エンコードや二重エンコード、Windows／Linux の区切り文字差、シンボリックリンクの影響も考慮します。

11. SSRF（サーバーからの不正アクセス）

- **発見できること：**URL プレビュー、画像取得、Webhook、PDF 生成などが、利用者指定 URL へサーバー側からアクセスする場合の制限を確認できます。
- **具体例：**外部 URL 取得機能を使って、インターネットから直接到達できない社内管理画面へアクセスできる状態です。
- **確認ポイント：**内部 IP、localhost、名前解決後の IP、リダイレクト先、許可プロトコルを検証しているかを確認します。

12. オープンリダイレクト

- **発見できること：**ログイン後や通知リンクの遷移先を外部の任意サイトへ変更できないかを確認できます。

- **具体例：**正規サイトの URL から偽ログイン画面へ転送でき、フィッシングの信頼性を高めてしまう状態です。
- **確認ポイント：**完全 URL、相対 URL、プロトコル相対 URL、エンコード済み URL などの解釈差を確認します。

13. CORS・API 公開範囲の不備

- **発見できること：**外部オリジンから認証付き API を呼べる設定、不要な HTTP メソッド、過剰なレスポンス項目を確認できます。
- **具体例：**攻撃者のサイトから利用者の認証情報付きで API を呼び、個人情報を読み出せる状態です。
- **確認ポイント：**Origin の部分一致、null Origin、資格情報許可、プリフライト応答、エラー応答の差を確認します。

14. 情報漏えい・不要ファイルの公開

- **発見できること：**デバッグ画面、スタックトレース、バックアップ、ソースマップ、.git、設定ファイル、ログ、API 仕様書などの露出を確認できます。
- **具体例：**エラー応答に内部パス、SQL 文、ライブラリのバージョン、アクセストークンの一部が表示される状態です。
- **確認ポイント：**既知のファイル名だけでなく、HTML・JavaScript・robots.txt・公開ディレクトリから参照先を洗い出します。

15. レート制限・自動化対策の不足

- **発見できること：**ログイン、OTP、パスワード再設定、検索、問い合わせ、クーポン、API などを大量実行できないか確認できます。

- **具体例：**同一アカウントへのログイン試行を無制限に行え、パスワード総当たりのリスクが高い状態です。
- **確認ポイント：**IP だけでなくアカウント・端末・操作単位の制限、解除条件、分散アクセスへの耐性を確認します。

16. キャッシュ制御・ブラウザ保存の不備

- **発見できること：**個人情報や管理画面が共有キャッシュ、ブラウザ履歴、戻る操作などから再表示されないかを確認できます。
- **具体例：**共有端末でログアウト後に戻るボタンを押すと、前利用者の個人情報が表示される状態です。
- **確認ポイント：**Cache-Control、CDN キャッシュキー、ユーザー別応答、ダウンロードファイルの扱いを確認します。

17. 業務ロジック・処理順序の不備

- **発見できること：**金額、数量、ステータス、割引、承認手順などを通常と異なる順序や値で操作し、不正な結果にならないか確認できます。
- **具体例：**画面では変更できない単価を API で書き換え、低い金額で注文を確定できる状態です。
- **確認ポイント：**リクエスト改変、処理の再送、同時実行、途中ステップの省略、期限切れ操作を確認します。

18. 既知の脆弱性・技術情報の露出

- **発見できること：**HTTP 応答、JavaScript、エラーページ等から判別できる製品・ライブラリのバージョンと既知脆弱性を確認できます。

- **具体例：**公開 JavaScript に脆弱性が報告されている旧ライブラリが含まれ、影響を受ける機能が実際に利用されている状態です。
- **URL 診断の限界：**バージョンが非表示の場合や、サーバー側依存関係は外部から判別できないことがあります。

3. ソースコードがある場合に見つけられる脆弱性

ソースコード診断では、外部から見える現象だけでなく、入力値がどの処理を経由して DB・OS・ファイル・外部サービスへ到達するかを追跡できます。同じ危険な実装パターンを全体検索し、未公開・未到達の機能を含めて修正対象を洗い出せる点が大きな利点です。

A. 入力・出力・インジェクション

1. 入力値検証の不足

- **追跡できること：**フォーム、JSON、Cookie、HTTP ヘッダー、ファイル名、外部 API 応答が、型・長さ・形式・許可値で検証される全経路を確認できます。
- **具体例：**画面側では数値限定でも API は任意文字列を受け取り、後段の SQL やテンプレート処理へそのまま渡している実装です。
- **修正判断：**拒否リストではなく許可リスト、境界での検証、正規化後の再検証が必要かを特定できます。

2. SQL／NoSQL インジェクション

- **追跡できること：**SQL 文字列連結、動的な列名・並び順、プレースホルダー漏れ、ORM の生クエリ利用を横断検索できます。
- **具体例：**検索条件だけは安全でも、ORDER BY に利用者入力を直接連結している箇所が残っている実装です。
- **修正判断：**値はバインドし、識別子は許可リストから選択するなど、文脈別の修正を提示できます。

3. OS コマンドインジェクション

- **追跡できること：**exec、system、shell_exec、proc_open 等へ利用者入力・ファイル名・URL が到達するデータフローを確認できます。
- **具体例：**画像変換コマンドへ元ファイル名を文字列連結し、特殊文字をシェルが解釈する実装です。
- **修正判断：**シェルを介さない API、固定引数、許可リスト、専用ライブラリへの置換を検討できます。

4. XSS の出力エスケープ漏れ

- **追跡できること：**HTML 本文、属性、URL、JavaScript、CSS など各文脈で、適切なエスケープ関数が使われているか確認できます。
- **具体例：**通常画面は安全でも、管理画面の監査ログ表示だけが未エスケープで、格納型 XSS になる実装です。
- **修正判断：**出力直前の文脈依存エスケープ、危険な HTML のサニタイズ、DOM API の安全な利用へ修正できます。

5. テンプレートインジェクション

- **追跡できること：**テンプレート本文や式として利用者入力を評価する箇所、動的テンプレート名、危険なフィルターを確認できます。
- **具体例：**メールテンプレートの編集機能で、入力された式をサーバー側テンプレートエンジンが評価する実装です。
- **修正判断：**変数値としてのみ渡す、利用可能な式を制限する、サンドボックスを有効にする対応を検討できます。

6. XML 外部エンティティ（XXE）

- **追跡できること：**XML、SVG、Office 文書等の解析で外部エンティティ・DTD・ネットワーク参照が有効になっていないか確認できます。
- **具体例：**外部エンティティを無効化せず XML を解析し、サーバー内ファイル参照や SSRF につながる実装です。
- **修正判断：**安全なパーサー設定、不要な DTD 禁止、入力形式の限定を確認できます。

7. 危険なデシリアライズ

- **追跡できること：**unserialize、pickle、ObjectInputStream 等で、利用者が変更できるデータをオブジェクト復元していないか確認できます。
- **具体例：**Cookie に保存したシリアライズ済みオブジェクトを署名検証せず復元する実装です。
- **修正判断：**単純な JSON 構造への変更、型の固定、完全性検証、危険クラスの排除を検討できます。

8. HTTP ヘッダーインジェクション

- **追跡できること：**Location、Content-Disposition、メールヘッダー、独自レスポンスヘッダーへ入力値を設定する箇所を確認できます。
- **具体例：**ダウンロードファイル名を無検証で Content-Disposition へ入れ、改行文字が混入できる実装です。
- **修正判断：**改行拒否、値の生成規則、フレームワーク API の安全な利用を確認できます。

B. 認証・認可・セッション

9. パスワード保存の不備

- **追跡できること：**平文、復号可能な暗号、弱いハッシュ、固定ソルト、コスト不足、独自ハッシュの使用を確認できます。
- **具体例：**古い SHA 系ハッシュだけでパスワードを保存し、漏えい時に高速な総当たりが可能な実装です。
- **修正判断：**Argon2id や bcrypt 等、言語標準のパスワードハッシュ関数と段階的移行を検討できます。

10. ログイン処理の分岐不備

- **追跡できること：**ユーザー存在、無効化、ロック、退会、パスワード比較、多要素認証の判定順序を確認できます。
- **具体例：**パスワード認証後、多要素認証完了前にセッションへログイン済みフラグを設定する実装です。
- **修正判断：**認証状態を段階化し、全要素完了後にだけ正式セッションを発行するよう整理できます。

11. 認可チェックの欠落

- **追跡できること：**ルート、コントローラー、サービス、DB クエリのどこで権限確認するかを一覧化し、未実装経路を探せます。
- **具体例：**画面メニューは管理者だけに表示される一方、管理 API 自体には管理者判定がない実装です。
- **修正判断：**UI ではなくサーバー側の共通認可層で、操作・資源・所属組織を毎回検証する設計にできます。

12. IDOR・テナント境界違反

- **追跡できること：**ID を指定する全クエリで、所有者 ID・組織 ID・テナント ID が検索条件に含まれるか確認できます。
- **具体例：**SELECT 条件が document_id だけで、tenant_id を含まないため、ID を知れば他組織の文書を取得できる実装です。
- **修正判断：**認可済みスコープからのみデータを取得するリポジトリ設計や複合条件へ修正できます。

13. セッション固定・失効不備

- **追跡できること：**ログイン時の ID 再生成、ログアウト・パスワード変更・権限変更時の失効、サーバー側セッション削除を確認できます。
- **具体例：**ログアウトはブラウザ Cookie だけを削除し、サーバー側セッションは有効なまま残る実装です。
- **修正判断：**セッションの世代管理、全端末失効、短い有効期限と更新方式を検討できます。

14. パスワード再設定・招待フロー

- **追跡できること：**トークンの乱数強度、保存方法、期限、単回使用、対象ユーザーとの関連付け、使用後失効を確認できます。
- **具体例：**再設定トークンを DB へ平文保存し、利用後も期限まで再利用できる実装です。
- **修正判断：**十分な乱数、ハッシュ保存、短い期限、単回使用、完了後の全セッション失効へ整理できます。

15. JWT・署名トークンの検証不足

- **追跡できること：**署名、アルゴリズム固定、exp、nbf、iss、aud、鍵選択、失効処理を確認できます。
- **具体例：**署名は検証するものの audience を確認せず、別システム用トークンを受け入れる実装です。
- **修正判断：**期待値の固定、鍵ローテーション、短期トークン、失効可能な更新トークンを検討できます。

16. CSRF 対策の実装漏れ

- **追跡できること：**状態変更ルートに CSRF 検証が共通適用されているか、例外指定や GET 更新処理がないか確認できます。
- **具体例：**通常フォームは対策済みでも、JSON 以外も受理する旧 API だけが検証対象外の実装です。
- **修正判断：**フレームワークの共通ミドルウェア、Origin 検証、Cookie 属性、重要操作の再認証を組み合わせられます。

C. ファイル・URL・外部連携

17. ファイルアップロード検証不足

- **追跡できること：**拡張子、MIME、マジックバイト、サイズ、画像再エンコード、保存名、保存場所、実行権限の処理順を確認できます。
- **具体例：**クライアント申告の Content-Type だけで画像判定し、元ファイル名のまま公開ディレクトリへ保存する実装です。
- **修正判断：**サーバー側内容判定、ランダム名、非公開領域、別ドメイン配信、危険形式拒否へ修正できます。

18. パストラバーサル

- **追跡できること：**入力パスの結合、正規化、realpath 確認、ベースディレクトリ内判定、シンボリックリンク対策を確認できます。
- **具体例：**文字列から../を除くだけで、エンコードや区切り文字の違いを回避できる実装です。
- **修正判断：**利用者入力をパスに使わず ID から解決する設計、正規化後の包含判定を検討できます。

19. SSRF

- **追跡できること：**HTTP クライアントへ URL が到達する全機能と、DNS 解決、リダイレクト、IP 制限、プロトコル制限を確認できます。
- **具体例：**最初の URL だけ外部ドメインか確認し、その後のリダイレクト先を再検証しない実装です。
- **修正判断：**許可先固定、名前解決後 IP 検査、リダイレクトごとの再検査、専用ネットワーク分離を検討できます。

20. Webhook 受信の検証不足

- **追跡できること：**署名検証、時刻差、リプレイ防止、生のリクエスト本文利用、イベント ID の重複排除を確認できます。
- **具体例：**JSON を再シリアライズした後の文字列で署名検証するため、正しい検証にならない実装です。
- **修正判断：**生本文での定数時間比較、期限確認、イベント ID の一意制約へ修正できます。

21. OAuth／外部ログインの不備

- **追跡できること：**state、nonce、PKCE、redirect_uri、トークン交換、アカウント紐付け条件を確認できます。
- **具体例：**外部 IdP から返されたメールアドレスだけで既存アカウントへ自動紐付けする実装です。
- **修正判断：**IdP の一意識別子、検証済みメール、明示的な連携確認、CSRF 対策を整理できます。

22. メールヘッダー・宛先制御

- **追跡できること：**To、CC、BCC、Reply-To、件名、本文、添付ファイル名へ入力が入る経路を確認できます。
- **具体例：**問い合わせフォームのメールアドレスを Reply-To へ無検証で設定し、改行を含められる実装です。
- **修正判断：**メールライブラリのアドレス API、形式検証、改行拒否、送信先固定を検討できます。

23. 外部 API 応答の信頼しすぎ

- **追跡できること：**外部 API の値を HTML、SQL、ログ、ファイルパス、権限判定へ利用する箇所を追跡できます。
- **具体例：**外部サービスから返るファイル名を安全とみなし、そのままローカル保存パスへ連結する実装です。
- **修正判断：**外部応答も非信頼入力として検証し、失敗時の安全な既定動作を定義できます。

D. 秘密情報・暗号・個人情報

24. API キー・認証情報の直書き

- **追跡できること：**ソース、設定、テスト、サンプル、CI 定義、コメント、ログ出力に含まれる秘密情報を検索できます。
- **具体例：**本番 API キーが PHP や JavaScript へ直書きされ、ブラウザ配信物や Git 履歴から取得できる状態です。
- **修正判断：**環境変数・シークレット管理へ移し、漏えいしたキーを失効・再発行する範囲を特定できます。

25. .env・設定ファイルの保護不足

- **追跡できること：**.env の配置場所、Web サーバー拒否設定、バックアップ名、エラー出力、リポジトリ除外設定を確認できます。
- **具体例：**Web 公開ディレクトリ直下の.env にアクセス拒否設定がなく、設定次第でダウンロード可能な構成です。
- **修正判断：**公開領域外へ置き、必要に応じて.htaccess 等で明示拒否し、.gitignore と配布手順も整備できます。

26. 暗号化・ハッシュの誤用

- **追跡できること：**固定 IV、弱い乱数、ECB、認証なし暗号、独自暗号、同一鍵の多目的利用を確認できます。
- **具体例：**毎回同じ IV で個人情報を暗号化し、同じ平文が同じ暗号文になる実装です。
- **修正判断：**AEAD、暗号学的乱数、用途別鍵、鍵バージョン管理へ移行できます。

27. 乱数・トークン生成の弱さ

- **追跡できること：**パスワード再設定、招待、API キー、CSRF、ファイル共有 URL に予測可能な乱数や連番が使われていないか確認できます。
- **具体例：**現在時刻とユーザーID から共有トークンを生成し、第三者が推測可能な実装です。
- **修正判断：**十分なビット長の暗号学的乱数と、期限・単回使用・ハッシュ保存を組み合わせます。

28. ログへの秘密情報・個人情報出力

- **追跡できること：**リクエスト全体、ヘッダー、Cookie、Authorization、API 応答、ファイル内容をログへ記録する箇所を確認できます。
- **具体例：**障害調査用ログに認証トークンと申請者の個人情報がそのまま残る実装です。
- **修正判断：**項目単位の許可リスト、マスキング、保持期間、閲覧権限、監査を整備できます。

29. 過剰なデータ返却

- **追跡できること：**DB モデル全体を JSON 化する処理、非表示フィールド、内部 ID、権限情報、削除済みデータの返却を確認できます。
- **具体例：**画面では氏名しか表示しないのに、API は住所・電話・内部メモまで返す実装です。
- **修正判断：**用途別 DTO、明示的な返却項目、フィールド単位認可へ修正できます。

30. データ保持・削除処理の不備

- **追跡できること：**退会・削除時に本体、添付、検索インデックス、キャッシュ、バックアップ参照がどう処理されるか確認できます。

- **具体例：**画面上は削除済みでも、直接 ID を指定すると API から取得できる論理削除の実装です。
- **修正判断：**削除済み条件の共通適用、匿名化、保持期限、バックアップからの復旧方針を整理できます。

E. 設定・依存関係・運用

31. デバッグ機能の本番残存

- **追跡できること：**debug フラグ、テストルート、開発者メニュー、診断 API、初期アカウント、サンプルデータを確認できます。
- **具体例：**環境変数が未設定の場合に debug=true となり、例外情報と環境変数一覧を表示する実装です。
- **修正判断：**本番では安全側へ倒れる既定値、ビルド時除外、起動時検査を導入できます。

32. セキュリティヘッダー設定の分散

- **追跡できること：**Web サーバー、フレームワーク、CDN、各ルートでのヘッダー設定と上書き関係を確認できます。
- **具体例：**通常画面は CSP 適用済みでも、ファイル閲覧用サブアプリだけ設定が抜ける構成です。
- **修正判断：**共通ミドルウェアとインフラ設定へ集約し、例外を明示管理できます。

33. CORS 設定の危険な生成

- **追跡できること：**Origin の文字列比較、正規表現、資格情報許可、許可ヘッダー・メソッドの生成を確認できます。

- **具体例：**許可ドメイン名を endsWith で比較し、攻撃者所有の類似ドメインも許可する実装です。
- **修正判断：**URL を構文解析して scheme・host・port を完全一致させる許可リストへ修正できます。

34. 依存ライブラリの既知脆弱性

- **追跡できること：**composer.lock、package-lock、requirements 等から実際のバージョンと到達可能な利用箇所を確認できます。
- **具体例：**脆弱なライブラリが含まれるだけでなく、問題となる画像解析機能をアップロード処理で利用している状態です。
- **修正判断：**影響範囲、更新互換性、暫定回避策、未使用依存の削除を判断できます。

35. 危険な初期値・フェイルオープン

- **追跡できること：**設定取得失敗、外部認証障害、権限サービス停止、タイムアウト時に処理を許可する分岐を確認できます。
- **具体例：**認可 API がタイムアウトすると、可用性優先でアクセスを許可する実装です。
- **修正判断：**重要処理は拒否側へ倒し、再試行・監視・限定的な縮退動作を設計できます。

36. 例外処理による情報漏えい

- **追跡できること：**例外をそのままレスポンスへ返す箇所、スタックトレース、SQL、内部パス、外部 API 応答の露出を確認できます。
- **具体例：**API 障害時に外部サービスの Authorization ヘッダーを含むデバッグ情報を返す実装です。
- **修正判断：**利用者向け固定エラーと内部ログを分離し、追跡 ID で関連付ける設計にできます。

37. 監査ログの不足・改ざん耐性不足

- **追跡できること：**ログイン、権限変更、閲覧、出力、削除、設定変更の記録内容と失敗時の扱いを確認できます。
- **具体例：**管理者が個人情報を CSV 出力しても、実行者・対象・件数が記録されない実装です。
- **修正判断：**必要イベント、時刻、主体、対象、結果、相関 ID、保護・保持期間を定義できます。

38. クラウド・コンテナ設定との不整合

- **追跡できること：**Dockerfile、IaC、CI/CD、権限設定、公開ポート、実行ユーザー、シークレット注入方法をコードと合わせて確認できます。
- **具体例：**アプリは非公開ストレージを想定している一方、IaC では保存先バケットを公開設定にしている状態です。
- **修正判断：**アプリコードとデプロイ設定を一体でレビューし、最小権限と安全な既定値を整備できます。

F. 業務ロジック・並行処理・設計

39. クライアント側だけの重要値検証

- **追跡できること：**金額、単価、割引、権限、状態遷移を JavaScript だけで制御し、サーバーが再計算・再検証しているか確認できます。
- **具体例：**注文合計をブラウザから受け取り、その値を正しいものとして決済へ渡す実装です。
- **修正判断：**サーバー側で信頼できる商品・契約データから再計算する構成へ変更できます。

40. 状態遷移・承認手順の抜け

- **追跡できること：**申請、承認、公開、取消等の状態変更を行う全ルートと、遷移元・権限・前提条件を確認できます。
- **具体例：**未承認状態から公開済み状態へ直接更新できる API 分岐が残っている実装です。
- **修正判断：**許可された状態遷移表を一元化し、DB 更新前に必ず検証する設計にできます。

41. 競合状態・二重実行

- **追跡できること：**在庫、残高、クーポン、発番、決済、申請の読み取りと更新が原子的か、再送に耐えるか確認できます。
- **具体例：**残数確認と減算が別処理で、同時注文により在庫以上の受付が成立する実装です。
- **修正判断：**DB トランザクション、行ロック、条件付き更新、一意制約、冪等キーを組み合わせます。

42. マルチテナント分離の設計不備

- **追跡できること：**テナント識別子の取得元、DB 条件、キャッシュキー、ファイルパス、ジョブ、検索インデックスの分離を確認できます。
- **具体例：**DB はテナント分離済みでも、キャッシュキーに tenant_id がなく、他組織の結果を返す実装です。
- **修正判断：**データアクセス層での強制スコープと、全保存層のテナントキー統一を検討できます。

43. 非同期処理・キューの権限引継ぎ

- **追跡できること：**ジョブ投入時と実行時の権限、対象 ID、テナント情報、再試行、重複実行の扱いを確認できます。
- **具体例：**投入時に権限確認しただけで、実行時には退職・権限剥奪後でも処理を続行する実装です。
- **修正判断：**実行時再認可、最小限のジョブデータ、冪等性、失敗キューの保護を整備できます。

44. 安全でないキャッシュキー・共有状態

- **追跡できること：**ユーザー別・権限別・言語別の応答が同一キャッシュキーに混在しないか、共有メモリの扱いを確認できます。
- **具体例：**URL だけをキーに管理画面の応答をキャッシュし、別利用者へ前利用者の内容を返す実装です。
- **修正判断：**認証応答の非キャッシュ化、Vary 相当の分離、主体・権限を含むキー設計へ修正できます。

4. 組み合わせて実施する場合の流れ

STEP 1　対象範囲を確定

ドメイン、画面、API、アカウント権限、除外対象、実施時間、禁止操作を明確にします。

STEP 2　コードと構成を把握

入口、認証・認可、重要データ、外部連携、保存先、依存関係を整理します。

STEP 3　ソースコードを横断分析

危険関数、秘密情報、権限判定、入力から重要処理までのデータフローを確認します。

STEP 4　許可された環境で再現確認

コード上の問題が外部から成立するか、影響を抑えた方法で検証します。

STEP 5　修正と再テスト

根本原因を修正し、同種箇所の横展開と外部からの再検証を行います。

診断時に用意すると精度が上がる資料

- **対象一覧：**ドメイン、サブドメイン、API、管理画面、モバイル連携先、外部連携先
- **権限別アカウント：**未ログイン、一般利用者、組織管理者、全体管理者など
- **ソース一式：**アプリ本体、API、バッチ、ジョブ、設定、IaC、CI/CD、依存ロックファイル
- **設計情報：**画面・API 一覧、データフロー、認証方式、権限表、ネットワーク構成
- **運用情報：**ログ方針、鍵管理、バックアップ、障害時動作、脆弱性対応・更新手順

> **結論**　URL 診断は『外から攻撃できるか』、ソースコード診断は『なぜ危険か・同じ欠陥がどこにあるか』を明らかにします。公開前・大規模改修時・重要機能追加時は、両方を組み合わせると最も効果的です。