No.09 ノウハウ・全10ページ

検出と「成立しないことの確認」は別もの

「診断済み」でも事故が起きる理由



SAST/DAST・WAF・認可 SDK が担う範囲と、業務上の操作が成立しないことを確かめる工程の空白

Securiy

INSIGHT

この資料でわかること

WAF も脆弱性診断も入れているのに、なぜ他社のデータが見える事故が起きるのか。製品が担う範囲と、空いている工程を分けて説明します。読み終えたら、自社で3つの操作を試せます。

想定読者

WAF と年次診断を入れている開発責任者・情シス

セキュリティ専任がいない SaaS・Fintech・EC 事業者。「診断済み」の根拠に不安がある方。

扱うこと

製品が担う範囲、担わない範囲

WAF（境界で攻撃の通信を見張る）、IdP（ログインを一元管理する）、SAST/DAST（コードや画面を自動で調べる診断ツール）、認可 SDK（権限判定の部品）がそれぞれ担う範囲と、どの製品も担わない「業務上の操作が成立しないことの確認」。

扱わないこと

製品の選び方・設定の手順

WAF のルールや診断ツールの設定は扱いません。既存の対策を外す提案もしません。

1 事故の起き方

診断から5か月後、何が起きたか P.03

製品が担う範囲と、空白 P.04

空白で起きる事故の2つの型 P.05

2 なぜ見つからないか

診断項目と、業務上の操作 P.06

「検出」と「成立しないことの確認」の違い P.07

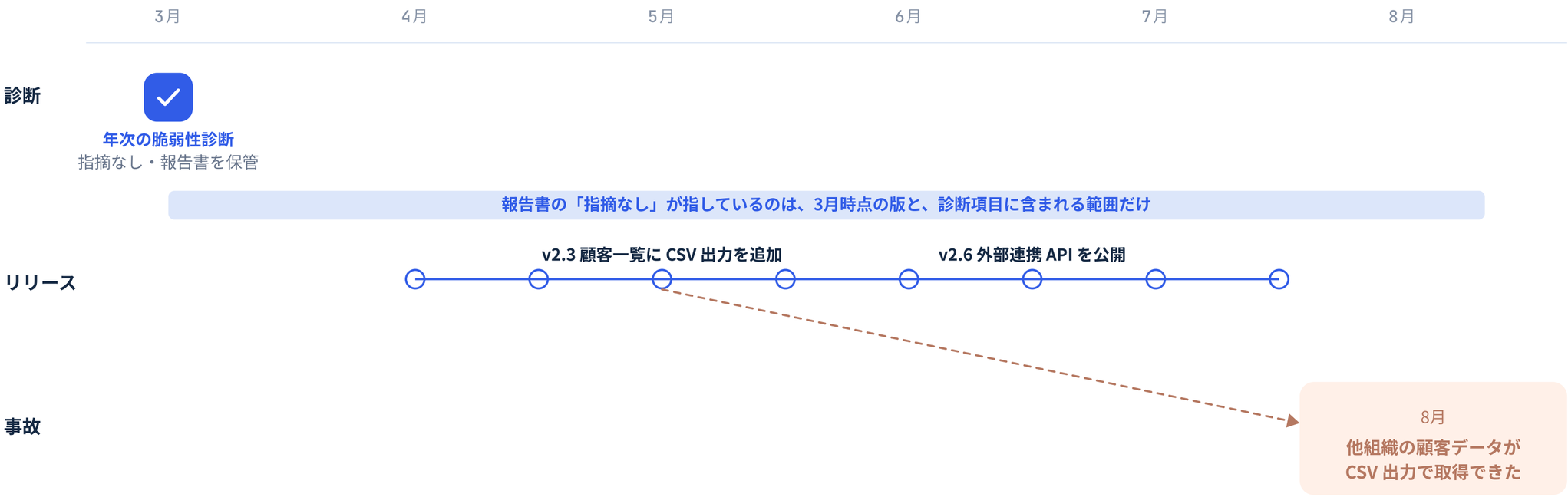
3 自分でやるなら

3つの操作で始める最小の確認 P.08

API に直接送る手順と点検表 P.09

診断から5か月後、何が起きたか

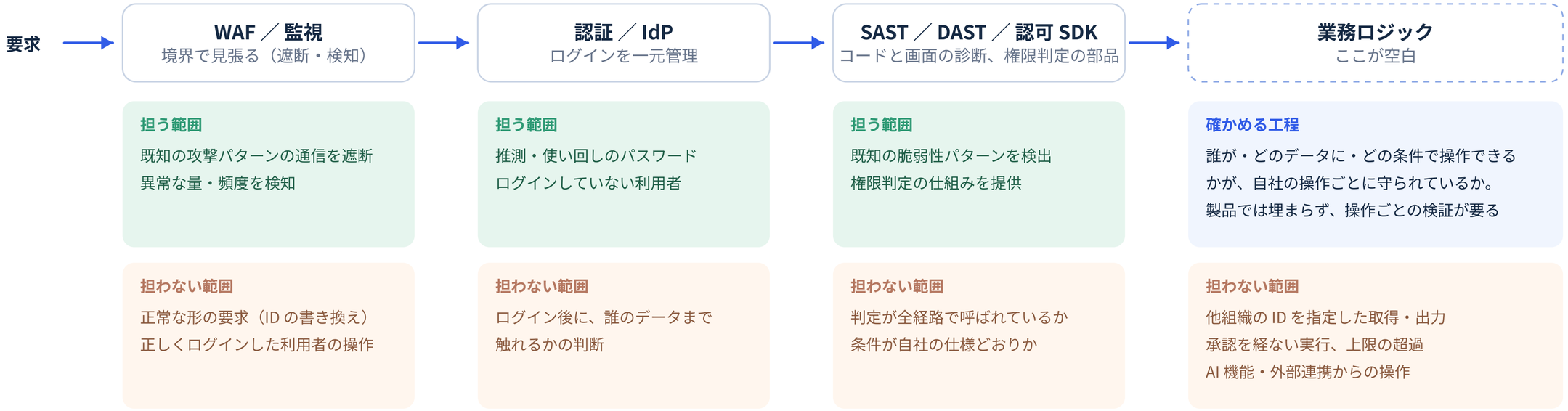
説明のための例です。3月に年次診断を受け、指摘なし。その後に機能を足し、8月に他組織の顧客データが CSV 出力で取得できる状態になっていた。診断にも WAF にも落ち度はありません。



診断は正しく行われました。事故は、診断の後に足した機能で、診断後に確かめていない操作で起きています。

製品が担う範囲と、空白

要求が通る順に、各製品が担う範囲と担わない範囲を並べます。右端の業務ロジックだけ、担当する製品がありません。認可 SDK は判定の仕組みを提供しますが、全経路で呼ばれているかは製品の外です。



各製品は自分の範囲を正しく守っています。ここで扱う事故は、どの製品の範囲にも入らない「業務上の操作」で起きます。

この資料では、この空白を埋める工程を「成立しないことの確認」と呼びます。

診断済みでも起きる事故の2つの型

この資料が扱うのは2つの型です。型1は業務上の操作の空白、型2は診断の後に公開された脆弱性という時間の空白。どちらも要求の形は正常で、WAF と認証を通過します。P.06 以降は型1を扱い、型2は No.10 で扱います。認証情報の盗用や設定ミスは別の資料で扱います。

型1: 認可・テナント分離の不備

組織 A の一般ユーザー
GET /api/customers?org=B

↓

WAF: 正常な通信として通過

↓

認証: ログイン済みの利用者

↓

B 社の顧客データが返る

なぜ通るか

要求の形は正常。ID を1つ変えただけ。
WAF の攻撃パターンには一致しない。
ログインは本人なので認証も通る。
『この要求がこの組織のものか』を
業務ロジックで照合していない。

診断の見積りに、組織をまたぐ
取得が入っていないと、試されない。

型2: 既知の脆弱性の悪用

外部の攻撃者
依存ライブラリの公開済み脆弱性を突く

↓

WAF: 正常な通信として通過

↓

認証: ログイン済みの利用者

↓

認証を迂回して管理画面に入る

なぜ通るか

脆弱性は公開情報にある。
診断時点では未公開、または診断後に
ライブラリを更新していない。
WAF のルールが追いついていない間は
正常な通信に見える。

自社の構成のどこに該当するかを
照合する工程が止まっている。

型1は「権限・承認・上限の照合」、型2は「公開情報との照合」。別の資料（No.11 設計／No.10 3つの壁）で、それぞれの対策を扱います。

診断項目と、業務上の操作は別のもの

脆弱性診断は、どのアプリにも共通する既知のパターンを網羅的に調べます。認可の項目もありますが、試せるのは診断時点の版と、渡したアカウント・見積りに入れた画面と API の範囲までです。見積りに入れなかった経路（連携キー＝外部ツール用の API キー・AI 機能）、診断後に足した操作、仕様を渡さなかった業務操作は残ります。範囲は見積書の項目で確かめられます。

脆弱性診断の項目（例）

どのアプリにも共通する、既知のパターン

- SQL インジェクション、OS コマンド注入
- クロスサイトスクリプティング（XSS）
- CSRF、クリックジャッキング
- 認証・セッション管理の不備
- 認可制御の不備（渡したアカウントと、見積りに入れた画面・API の範囲で）
- セキュリティヘッダー、TLS の設定
- 既知の脆弱性を持つライブラリの検出

網羅性が高く、再現性がある。ここは診断に任せてよい。

業務上の操作（自社固有）

診断の範囲・期間・仕様の共有次第で、十分に試されないことがある

- 他組織の ID を指定した顧客データの取得・出力
- 承認を経ない送金先の変更・返金の実行
- 上限を超える金額・件数の処理
- 一般ユーザーによる自分の権限の引き上げ
- 外部連携の API キーによる全データの取得
- AI チャットへの指示による業務データの操作

見積りに入れなかった経路（連携キー・AI 機能）と、診断後に足した操作は残る。操作ごとに試す。

「診断済み」は主に左の列の話です。右の列は、診断時点の版と渡した範囲でしか試されていないため、自社の操作ごとに確かめ直す必要があります。

「検出」と「成立しないことの確認」の違い

同じ「セキュリティの確認」でも、問いが違います。検出は「脆弱性があるか」、確認は「この操作が成立してしまうか」。どちらも要りますが、後者は自社にしか決められません。

	検出（脆弱性診断）	成立しないことの確認
対象	既知のパターン（共通）	**自社の業務上の操作（固有）**
問い	脆弱性があるか	**この操作が成立してしまうか**
誰が決める	診断事業者の項目表と、渡した範囲	**自社（守りたいデータと操作）**
やり方	パターンを網羅的に当てる	**正当なアカウントで、範囲外の操作を試す**
頻度	年1回・リリース前	**操作を決めたら、変更のたびに同じ手順で**
結果の形	検出一覧（重大度つき）	**操作ごとの 拒否／成立 と再現条件**
「なし」の意味	そのパターンは見つからなかった	**その操作は、その版で成立しなかった**

「指摘なし」と「成立しなかった」は別の報告です。取引先に説明するときも、この2つを分けて書くと、何を確かめたのかを読み違えられません。

3つの操作で始める最小の確認

攻撃の手口は要りません。検証環境に2つのテスト組織とテストアカウントを作り、正当なアカウントで「範囲外の操作」を試すだけです。結果を版・日付つきで1操作1行の表に残せば、それが検証記録になります。

確かめること	誰として	何を指定して試すか	期待する結果
他組織のデータが見えないか	組織 A の一般ユーザー	API に組織 B の ID を指定して顧客一覧・CSV 出力・検索を要求	**3操作とも拒否される（403 など）**
承認なしで実行できないか	担当者（承認権限なし）	承認待ちの送金先変更・返金を、API に直接「承認済み」の状態で送る	**拒否され、承認の記録が残る**
上限を超えられないか	承認済みの担当者	上限を1円・1件だけ超える金額・件数で実行	**停止し、申請に回る**

準備

検証環境に、組織 A・B と各1アカウント

本番の認証情報や実データは使いません。ステージングがない場合は、本番に検証用の組織を作る方法もあります。その場合は他の顧客のデータに触れない範囲と時間帯を先に決め、許可を取ってから（No.24 P.05）。

試し方

画面ではなく、API に直接送る

画面だけで試すと、この抜けは見つからないことがあります。ブラウザーの開発者ツールで要求をコピーし、ID や状態を書き換えて送り直します。手順は P.09。

記録

日付・版・誰として・何を・結果・確かめていない範囲

6列の表に1操作1行。リリースのたびに同じ3行を追記すれば、「変更時に確認しているか」にも答えられます。

3 ／ 自分でやるなら

API に直接送る手順と、点検表

P.08 の3操作を試す手順です。攻撃ツールは要りません。ブラウザーの開発者ツールで、画面が送った要求をコピーして書き換えるだけです。2つのアカウントで同じ要求を送り比べる作業は、OWASP ZAP の Access Control Testing や Burp Suite の Autorize で半自動にできます。

01
開発者ツールを開く
Chrome で F12 →［ネットワーク］タブ。組織 A の一般ユーザーでログインし、顧客一覧を開く

→

02
要求をコピーする
一覧の該当行を右クリック →「cURL としてコピー」

→

03
書き換えて送る
org=A を org=B に、status を approved に、amount を上限＋1 に。ターミナルに貼って実行

→

04
応答を記録する
応答コード（403／200）と中身。日付・版・誰として・何を・結果を1行で

No.	点検すること	状態	担当・期限
Q1	**診断後に足した機能の一覧（版と日付）**	ある ／ 一部 ／ ない	
Q2	**守りたいデータと、範囲外の操作3つを決めた表**	ある ／ 一部 ／ ない	
Q3	**P.08 の3操作を試した記録（版・日付つき）**	ある ／ 一部 ／ ない	
Q4	**画面と API の両方で試した記録**	ある ／ 一部 ／ ない	
Q5	**診断の見積書で、認可・業務ロジックの項目と渡す範囲を確かめた**	ある ／ 一部 ／ ない	

「成立しないことの確認」を、3つの操作から。

P.08 の3操作を、決めた条件で画面と API から試すところまでは自社で繰り返せます（P.09）。外に出す価値があるのは、まだ決めていない条件と経路（承認内容の書き換え・連携キー・AI 機能）を広く洗い出して試し、試さなかったものも未検証範囲として書いた、取引先に出せる記録にするところです。共同検証プログラムでは、3操作のうち2操作から始めます。初回のご相談は30分・無料です。

お申し込み・お問い合わせ　**securiy-preview.pages.dev**　のフォームから
Tempura technologies株式会社

※ 本資料は既存の対策を外すことを勧めるものではありません。すべての攻撃の網羅や「安全」の総合判定は保証しません。金融の規制適合審査、決済基盤全体の監査、カード不正判定は対象外です。

無料相談

30分無料相談

守りたいデータと気になる操作を伺い、初回に検証する2操作を一緒に決めます。本番の認証情報や顧客データは不要です。

相談を申し込む →

共同検証プログラムとは

検証テーマを一緒に作る企業に、初回スコープで提供する枠組み

1アプリ・1検証環境・2操作、準備完了後10営業日・合計40時間（修正案と再検証は2件・12時間まで）。費用は着手前に提示します。

次に読む

No.11 操作に権限・承認・上限を組み込む設計
No.10 既知の脆弱性が放置される3つの壁

型1（認可の不備）と型2（既知脆弱性）それぞれの対策はこちら。