No.19 ノウハウ・全10ページ

ZERO-DAY

API

情報がなくても効く防御を設計する

ゼロデイ前提の不正操作対策

未知の脆弱性を全部見つける前提を捨て、権限・承認・上限の確認で成立させない設計

SaaS

AI AGENT

Securiy

この資料について

この資料でわかること

未知の脆弱性（ゼロデイ）は、公開情報を照合しても見つかりません。それでも事故を小さくする方法はあります。入られた後に「アプリの操作として何ができてしまうか」を設計で絞る、2系統目の考え方です。業界では「侵入を前提にする（Assume Breach）」「最小権限」と呼ばれる考え方を、SaaS の業務操作に当てはめたものです。

想定読者

「全部見つけるのは無理」と感じている開発責任者・CTO

ライブラリ更新と診断は回しているが、それでも不安が残る SaaS・Fintech・EC 事業者。

扱うこと

入られた前提で効く設計と、効く範囲・効かない範囲

攻撃者の目的を4つに整理する見方、権限・承認・上限で何を成立させないか、照合が効かない侵入の補い方、公表された日の動き。

扱わないこと

侵入の検知と防止、設計の組み込み方

EDR（端末やサーバーの不審な動きを検知する製品）や WAF（境界で攻撃の通信を見張る仕組み）の検知ルールは扱いません。照合をどう組み込むかは No.11 で扱います。

1 前提を変える

「全部見つける」が成り立たない理由 P.03

入口は塞ぎきれない。アプリを通る被害は4つ P.04

2 成立させない設計

自社の4つの操作を書き出す P.05

照合が効かない侵入と、その補い方 P.06

公表から修正版までの間に、損害の上限を下げる P.07

3 公表日に備える

公表日の机上演習 P.08

点検表 P.09

「全部見つける」前提が成り立たない理由

脆弱性が公開されるまでの間、公開情報を照合する系統では何も起きません。攻撃者が先に見つけていれば、その間は無防備です。
だから、アプリの操作を経由する被害については、情報がなくても効く系統を別に持ちます。

A ／ 公開情報で追いつく系統

脆弱性が作り込まれる → 攻撃者が先に見つける → 公開される → 照合・修正

この区間、A の系統では何も起きない（情報がないので照合できない）

ここからは A が効く

B ／ 情報がなくても効く系統

入られた後の業務操作（データの取得・金銭の移動・権限の奪取・設定変更）を、権限・承認・上限の照合で成立させない

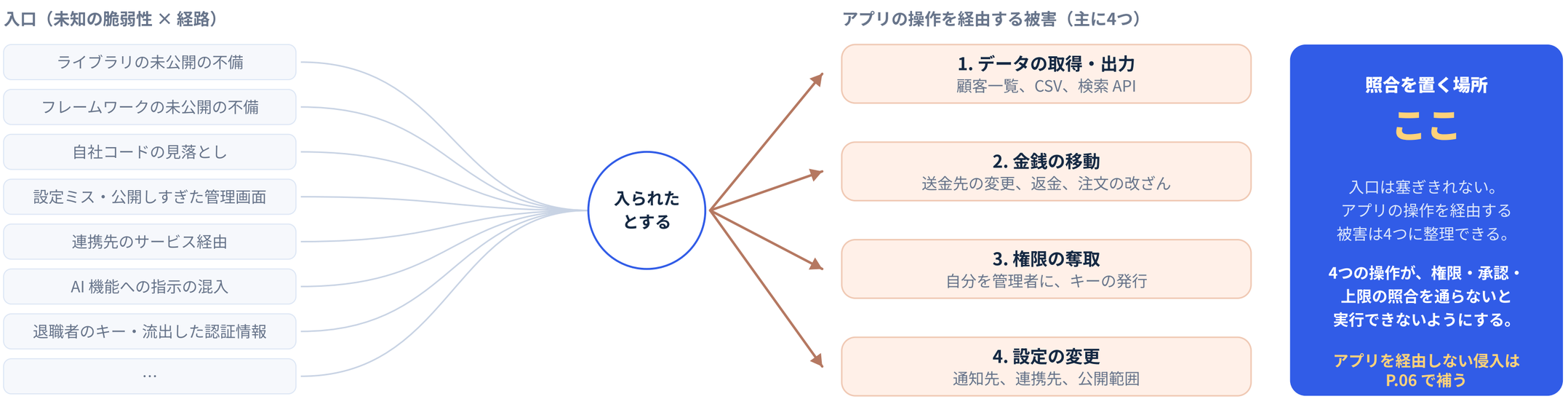
アプリの操作を経由する被害なら、照合そのものに不備がない限り、入口の脆弱性が未知でも既知でも効く

A を止める必要はありません。A は「既知に追いつく」、B は「未知でも業務操作を成立させない」。役割が違います。

A の回し方は No.10「既知の脆弱性が放置される3つの壁」。この資料は B だけを扱います。
B はカード業界の基準（PCI DSS）や米国の政府機関（NIST・CISA）が「侵入を前提にする」として勧める考え方と同じ向きです。

入口は塞ぎきれない。アプリを通る被害は4つに整理できる

攻撃者が使う入口は塞ぎきれず、増え続けます。一方で、アプリの操作を経由して起きる被害は4つに整理できます。守る場所を入口から目的に移すのが、この資料の考え方です。



入口を全部塞ぐ設計は終わりません。アプリの操作を経由する被害は4つに整理できるので、そこに照合を置けば、その4つは入口がどこでも成立しにくくなります。

アプリを経由せずにサーバーやデータベースへ直接届く侵入（任意のコード実行、DB の直接読み出し）には、この照合は効きません。サービス停止・暗号化・破壊も対象外です。

自社の4つの操作を書き出す

4つの目的が、自社ではどの操作に当たるかを書き出します。これが最初の作業です。右の3列は、その操作に権限・承認・上限のどれを効かせるかの例で、組み込み方は No.11 にあります。

目的	自社ではどの操作か（記入）	権限 誰が・何を	承認 承認どおりか	上限 どの条件で
データの取得・出力	例: 顧客一覧の CSV 出力、検索 API	他組織の ID を指定した取得を拒否	一括出力・大量取得に承認を要求	件数・回数の上限で量を抑える
金銭の移動	例: 返金、送金先の変更	担当者以外の送金・返金を拒否	承認した宛先・金額と一致しない実行を拒否	1件・1日の金額上限で損害を抑える
権限の奪取	例: ロール変更、連携キーの発行	自分の権限の引き上げを拒否	権限変更に別の人の承認を要求	連携キーの範囲と期限で広がりを抑える
設定の変更	例: 通知先・連携先の変更	管理者以外の設定変更を拒否	通知先・連携先の変更に承認を要求	変更回数を絞り、旧い宛先にも通知する

書き方

1行に1〜2操作。「顧客」「返金」のような機能名ではなく、「組織 B の ID を指定して顧客一覧を取得」のように、誰が・何を・どの条件で、まで書きます。

使い方

この4行が、No.11 の設計の対象であり、No.24 でリリースごとに試す操作の候補です。初回相談ではこの4行から2操作を選びます。

照合が効かない侵入と、その補い方

権限・承認・上限の照合は、アプリの操作を経由する被害にだけ効きます。効かない侵入を先に認めて、別の手で補います。

照合が効かない侵入	なぜ効かないか	補い方
サーバー上でコードを実行される	アプリの照合を通らず、DB や設定に直接触れる	**アプリの実行権限を最小にし、DB の接続ユーザーを用途ごとに分ける。DB 側で接続元と権限を絞る**
DB の認証情報が漏れる	アプリを経由せず DB を直接読まれる	**DB 側の権限と接続元の制限。重要な列の暗号化。接続の記録と通知**
盗まれた正規の認証情報で操作される	本人として照合を通ってしまう	**権限の範囲内の被害に抑える（最小権限・上限）。ログイン通知、普段と違う操作の記録と通知**
記録そのものを消される・書き換えられる	何が起きたか分からなくなる	**記録はアプリとは別の場所に保管し、追記のみにする**
サービス停止・暗号化・破壊	業務操作ではない被害	**この資料の対象外。バックアップと復旧の手順で備える**

補い方の多くは、クラウドや DB の標準機能で設定できます。この表の左列が自社に当てはまるかを、No.11 の設計と一緒に確かめます。

公表から修正版までの間に、損害の上限を下げる

修正版が出るまでの間、何ができるか。B の系統が入っていれば、「上限を下げて待つ」という選択肢が生まれます。誰が上限を変えてよいかは、平時に決めておきます。



「上限を下げる」の中身: 修正版を待つ間に、動かせる量と範囲を絞る

該当機能の一括出力を一時停止、または件数上限を 1/10 に
送金・返金の1日上限を下げ、承認者を2人に
該当機能に関わる連携キーを一時無効化
4つの目的に当たる操作の通知を「即時」に切り替え

やらないこと

影響を確かめずに全面停止する（ベンダーや当局が切断・停止を推奨している場合は従う）
該当を確かめずに全ライブラリを更新する（壊れて戻せなくなる）
対応を記録せずに終える（取引先に説明できない）

これができるのは、上限・承認・通知が設計に入っていて、誰が上限を変えてよいかを平時に決めてあるときだけです。
設計がなければ、修正版が出るまでの手は、機能の停止と接続の制限に限られます。

A と B を両方持っていると、公表から修正版の適用までの空白期間を、アプリの操作を経由する被害については「上限を下げて待つ」で埋められます。

公表日の机上演習: 平時に一度だけ試す

公表された日に慌てないために、上限を下げる手順を平時に一度だけ試して、この表を埋めておきます。埋まらない行が、設計に足りないものです。攻撃の手口は要りません。

No.	手を打つ内容	誰が判断	どの画面・コマンドから	何分で	どう戻すか
01	**該当機能の一括出力を止める**				
02	**件数・金額の上限を 1/10 に下げる**				
03	**該当機能に関わる連携キーを無効にする**				
04	**4つの目的の操作の通知を「即時」にする**				
05	**取引先・利用者に状況を知らせる（必要な場合）**				
06	**上限を元に戻し、記録する**				

やり方

検証環境で、01〜06 を実際に操作します。「どの画面から」が書けない行は、設定画面がないか、権限が管理者にしかない行です。

記録

埋めた表は版・日付つきで保管し、設定画面が変わったら更新します。取引先の「インシデント時の対応手順はありますか」に、そのまま添付できます。

3 ／ 公表日に備える

点検表: 入られても、4つの操作が成立しにくいか

7つが揃えば、未知の脆弱性で入られても、アプリの操作を経由する被害は成立しにくくなります。Q1 だけでも、いまの状態がはっきりします。

No.	点検すること	目安	状態	担当
Q1	**4つの目的が、自社のどの操作に当たるか書き出した（P.05）**	機能名ではなく、誰が・何を・どの条件で	ある ／ 一部 ／ ない	
Q2	**その操作に、権限・承認・上限の照合がサーバー側にある**	画面のチェックだけではない（No.11）	ある ／ 一部 ／ ない	
Q3	**照合が効かない侵入の補い方を決めた（P.06）**	DB 側の権限、記録の別保管、バックアップ	ある ／ 一部 ／ ない	
Q4	**連携キーに範囲と期限がある**	退職者・流出時に自動で止まる	ある ／ 一部 ／ ない	
Q5	**4つの目的に当たる操作が記録・通知される**	誰が・何を・結果・根拠	ある ／ 一部 ／ ない	
Q6	**上限を下げる手順を平時に試し、表を埋めた（P.08）**	誰が・どこから・何分で・どう戻すか	ある ／ 一部 ／ ない	
Q7	**公表を知る経路が決まっている**	ベンダー通知・JVN・KEV のどれを誰が見るか（No.10）	ある ／ 一部 ／ ない	

自社で回るもの

4つの操作の書き出し、効かない侵入の補い方、机上演習は、この資料の手順で自社でできます。決めた条件で画面と API から試すところも自社でできます（No.09 P.09）。

外に出す価値があるもの

書き出した操作を、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）まで広く試し、試さなかったものも未検証範囲として書いた記録にするところ。共同検証プログラムでは、4行から2操作を選んで始めます。

見つける前提を捨てて、成立させない設計へ。

4つの操作を書き出し、補い方を決め、机上演習をし、決めた条件で画面とAPIから試すところまでは自社でできます（No.09 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。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

※ 本資料は事故を小さくする設計を扱うものです。侵入の防止や検知を代替するものではなく、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

4つの目的が自社のどの操作に当たるかを一緒に書き出し、初回に検証する2操作を決めます。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

B の系統の設計と、A の系統の回し方はこちら。