Engineer

アクセス制御

🚪 リソースを利用する要求を一件ずつ許可または拒否する仕組み

どのようなものか

アクセス制御は、「この呼び出し元は、今、このリソースに、この操作をしてよいか」という問いに、一秒間に何度も答えるシステムの部分です。呼び出し元は、人、サービス、定期実行のジョブなどです。リソースは、ファイル、データベースの行、APIのエンドポイント、Gitリポジトリ、扉などが該当します。その判断を行う個々の仕組みをアクセス制御と呼び、NISTの用語集では、この考え方全体の同義語にauthorization（認可）を挙げています。定義は、仕事の両面を含みます。「主体にシステムのオブジェクトへのアクセスを許可または拒否する判断」と、情報や関連する処理サービスの取得・利用、または特定の物理的な施設への立ち入りについて「個別の要求を許可または拒否する手続き」です。

認証と混同しやすいものの、二つは別の手順です。NIST SP 800-162は、「認証はアクセス制御と同じではない」と明記しています。認証は、誰が要求しているかを確かめます。アクセス制御は、その身元情報と、要求について分かっているその他の情報を使い、呼び出し元が何をしてよいかを判断します。本人を完全に確認できても、二つ目の問いを投げかけていなければ、他人の記録を読ませてしまう可能性があります。

同じガイドは、仕事を二つに分けています。ポリシー決定点が答えを計算し、ポリシー実施点が要求の経路上でそれを適用します。両方を同じプロセスに置けますが、判断を実施するものがなければ何も守れないため、この区別は覚えておく価値があります。NIST SP 800-53は、二つの管理策に記載しています。AC-24は、実施前に各アクセス要求へ判断を適用することを求め、AC-3は、承認された認可を実施することを求めます。

ルールには、各オブジェクトの許可ID一覧、RBACのようなロール、SP 800-162で扱う呼び出し元・対象・操作・環境の属性など、いくつかの形があります。どの形でも、すべての要求で、その問いを投げかけるものが必要です。

概念を判断に生かす

「アクセス制御とは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 アカウント、パスワード、権限を整理し、業務に必要なアクセスだけを付与し、退職者の権限は当日に解除します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 Linuxサーバーを守る1,000件のRBACルールを作成 — 1,000 セキュリティ上の穴を塞ぐアクセスルール

Linuxサーバーを守る1,000件のRBACルールを作成 お問い合わせ →

なぜ重要なのか

ログイン後に何へアクセスできるかは、ほぼすべてアクセス制御で決まります。弱いアクセス制御の前に強い認証を置いても、データを持ち出したアカウントを記録するだけです。他者の情報を保持する事業では、誰が何を見られるかという約束が守られるかどうかは、ここで決まります。

NISTは、理想的な実施の仕組みをリファレンスモニターと呼びます。SP 800-53の管理策AC-25は、改ざんできず、必ず呼び出され、分析とテストが可能なほど小さいものを求めています。この三つは、実際の製品の検査を評価する基準にもなります。呼び出し元が判断の根拠となる入力を変更できれば、一つ目を満たしません。検査を通らずデータに到達する経路があれば、二つ目を満たしません。ロジックが何百ものハンドラーに散らばり、誰も全体を読めなければ、三つ目を満たしません。

OWASP Application Security Verification Standardは、これをチームが検証できる要件にしています。認可の章では、機能へのアクセスを明示的に許可された呼び出し元に限定し、個々のデータ項目へのアクセスも同様に限定することを求めています。また、ブラウザーで動くJavaScriptなど、信頼できないクライアントが変更できるものに頼らず、信頼できるサービス層でルールを実施することを求めます。判断に使う値の変更は直ちに反映し、それが不可能な場合は、権利を失った後の操作をリスク緩和策で検出して通知することも求めています。

代償は、要求の処理経路にかかる負荷です。判断には事実が必要で、事実を得るには、その都度情報の照会が必要です。照会を省こうとすると、最も負荷の小さい場所、多くの場合は呼び出し元が操作できる場所へ、検査を移してしまいがちです。

実際の活用例

一人が複数の組織の代理として操作できる海事プラットフォームでは、誰が判断するかを、その判断をどこで行うかで定めました。選択中の組織はクライアントに保持し、アクセス判断では一切信用しません。OrganizationContextが現在の選択をlocalStorageに保存するため、切り替えは即座に行えます。サーバー側のCookieにも反映し、サーバーで描画するページも同じ組織を参照します。認可は、すべての要求でサーバーが再確認します。そのページは、境界を一文で示しています。何が許されるかを決める唯一の主体はサーバーであり、localStorageを改変しても、自分の画面の表示が変わるだけです。サーバーは引き続き拒否するためです。

同種のプラットフォームでは、支払いの受け付けと権限の付与を、二つのシステムとして構築しました。Stripeのカード決済と、Apple・Googleのアプリ内購入を、9テーブルと34関数からなる決済スキーマに集約しています。別の11テーブルと34関数からなるアクセス用スキーマが、どのストアで支払ったかを知らずに、「このアカウントは、この操作をしてよいか」に答えます。その答えで企業名簿、検索、データ出力を制限し、すべての要求でサーバーが再確認します。ボタンを隠すことは配慮であって、制御ではないためです。ストアの追加は決済側だけに影響し、アクセスに関するサポートでは、三つではなく一つのテーブルを見れば済みます。そのページは、代償も示しています。二つのスキーマを維持し、そうでなければ追加の処理が不要な要求でも、利用権限を照会する必要があります。

会社がソースコードを自社のハードウェアへ移したときは、Gitフォージを、原則として拒否する設定にしました。そのページは、フォージの運用に必要なことの一つに、アクセス制御を挙げています。管理用ウェブインターフェースがないことを選定理由としたのは、インターフェースが少ないほど守る対象も減るためです。方針は拒否から始まります。IDがなければ何も与えず、鍵がなければ何も与えず、宣言したすべてのリポジトリに非公開フラグを付けます。秘密を保つ根拠を、見つかりにくさではなくポリシーに置いています。到達できる経路も、同じ方針の一部です。ポート23231はループバックでだけ応答し、外部からの入り口を踏み台ホスト一つに限定します。公開ポートからGitを使えるようにする多重化の仕組みもコードにはありますが、スイッチを切っています。誰も使わないリスナーも、攻撃対象になるためです。

問題が起きるところ

検査が画面側にしかありません。 ボタンを隠したり無効にしたりしても、その先のエンドポイントが直接の呼び出しに誰でも応答します。サーバーは、呼び出し元が編集できない事実を基に、すべての要求で判断し直す必要があります。

機能は確認しても、記録を確認しません。 要求が請求書、文書、アカウントをIDで指定すると、サーバーは請求書を閲覧できることだけを確認し、誰のものかを調べず指定の記録を返します。ASVSは、この欠陥を、安全でない直接オブジェクト参照（insecure direct object reference、IDOR）、またはオブジェクト単位の認可の不備（broken object level authorization、BOLA）と呼びます。防御する側は、サーバーが要求内のIDをそのまま使っていることを手掛かりに、この欠陥を見つけられます。修正は、すべてのオブジェクトを、呼び出し元に許された範囲の中で検索することです。複数テナントの製品では、テナントを要求から受け取らず、必ずサーバー側のセッションから取得します。

新しいものが、閉じるまで公開されます。 新しい経路、テーブル、リポジトリは、既定の状態で始まります。既定が許可なら、誰かが気付くまで穴が開いたままです。SP 800-53は、逆のネットワーク原則を「原則拒否、例外として許可」と定めています。同じルールは、あらゆる層に適用できます。与えられていない権限は、拒否します。

権限が増える一方です。 SP 800-162は、時間の経過とともにアクセスを削除・失効させなければ、利用者に権限が蓄積すると指摘しています。異動、プロジェクトの終了、委託先の離任は、それぞれ何かを取り除くべき時点です。数時間後のセッション終了まで取り消しが効かないのも、同じ欠陥の緩やかな形です。

要求を許可した理由を、誰も説明できません。 判断が各所で行われ、どこにも記録されなければ、調査で読めるものがありません。許可と拒否を、その判断に使ったルールとともに記録し、ログも保護します。SP 800-162は、判断のパターンから、要求の背後にいる人の属性が分かる可能性を警告しています。

参考資料

- NIST CSRC用語集のアクセス制御：FIPS 201-3とCNSSI 4009の定義。認可を同義語として掲載しています。
- NIST SP 800-53 Rev. 5：アクセス制御の管理策群。AC-3の実施、AC-24の判断、AC-25などを含みます。
- NIST SP 800-162：属性ベースのアクセス制御と、ポリシー決定点・実施点を説明しています。
- OWASP Application Security Verification Standard：機能、データ、フィールド単位で検証できる認可要件を示します。

用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「アクセス制御とは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/access-control/