Engineer

多要素認証

異なる種類の証拠を二つ以上求めるログイン

どのようなものか

多要素認証は、通常MFAと略し、申告された身元を、複数の種類の証拠で裏付けた場合だけ認めるログインです。政府以外の多くの組織も従う、米国連邦政府の認証指針NIST SP 800-63B-4は、三種類を挙げます。パスワードなどの知っているもの、電話やセキュリティーキーなどの持っているもの、指紋などの本人の身体的特徴です。定義の要点は、一つの言葉にあります。「異なる」種類の要素を複数求める必要があります。パスワードと秘密の質問への答えは、どちらも知っているものなので、一要素にしかなりません。二要素認証、つまり2FAは、ちょうど二つを使う、よくある場合です。

指針は、二つの方法を認めます。一つ目は、別々の単要素認証器を組み合わせます。パスワードに、使い捨てコードを表示する電話のアプリか、印刷した予備コードの一覧を加えます。二つ目は、一つの多要素認証器を使います。例えば、秘密鍵を持ち、PINか指紋を確認した後でしか使わない、セキュリティーキーや電話です。このPINや指紋を、認証器の有効化要素と呼びます。端末内で確認し、サーバーは、検証できる署名という結果だけを受け取るため、一度触れるだけで、二要素を提示できます。

生体情報の役割は、限られています。NISTは、認証に物理的な認証器も関わる場合だけ、指紋や顔を認め、単独の要素には認めません。また、生体情報を使わない代替手段を常に用意するよう求めます。生体的な特徴は、秘密ではないとも明記します。顔は撮影でき、指紋は、本人が触れたものから採れます。

概念を判断に生かす

「多要素認証とは？」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

パスワードは、本人が気づかずコピーされる場合があります。本物らしい偽のログイン画面に入力したり、推測されたり、後でデータベースを盗まれるサイトで使い回したりする場合です。MFAなら、コピーしたパスワードだけでは足りず、攻撃者は、別の種類のものも必要になります。

NISTは、これを二つ目の保証レベルの基本とします。AAL2では、一つの認証器に組み合わせるか、別々に提示するかを問わず、二要素で認証する必要があります。組み合わせには、物理的な認証器と、パスワードか生体情報の照合を含めます。そのレベルの検証側は、少なくとも一つ、フィッシング耐性のある方法を提供する必要があります。連邦政府の機関は、職員、請負業者、提携先に、フィッシング耐性のある認証を必須としなければなりません。

同じ三文字で売られる方法同士の差は、MFAとパスワードだけの差より大きく、指針は明確に分類します。

- **SMSや音声通話で届くコード**は、NISTが制限付きと分類する、唯一の認証器の種類です。使用は可能ですが、検証側は、制限付きでない代替手段を用意し、利用者にリスクを伝え、そこから移る計画を持つ必要があります。登録電話番号の変更は、新しい認証器の登録とみなし、必要な確認をすべて行います。
- **アプリのコードと電話へのプッシュ承認**は、AAL2で認められますが、フィッシング耐性はありません。利用者が入力したり承認したりするものが、要求したサイトと結び付いていないからです。
- **応答をサイトと結び付ける暗号認証器**には、パスキーやスマートカードなどがあり、フィッシング耐性があります。AAL3では、秘密鍵をハードウェアから書き出せないものが必要です。

電子メールがないのは、意図的です。指針は、別経路での認証のチャネルとして認めません。

実際の活用例

認証アプリのコードは、最もよく使われる第二の要素で、2011年5月のRFC 6238が、数ページで定義します。登録時に、サーバーは秘密の共有鍵を作り、通常QRコードでアプリに渡すため、両者が同じ鍵を持ちます。コードを作るときは、それぞれ現在のUnix時刻を、既定では30秒の時間幅で割り、整数部分をカウンターとします。そのカウンターを鍵でHMAC計算し、結果を短い十進数へ切り詰めます。2005年のRFC 4226で定義されたHOTPの方法を使い、少なくとも六桁が必要です。Pythonでは、全計算が数行に収まります。

```
import hashlib, hmac, struct

def totp(key: bytes, unix_time: int, digits: int = 6, step: int = 30) -> str:
    counter = struct.pack(">Q", unix_time // step)
    mac = hmac.new(key, counter, hashlib.sha1).digest()
    offset = mac[-1] & 0x0F
    code = struct.unpack(">I", mac[offset:offset + 4])[0] & 0x7FFFFFFF
    return str(code % 10**digits).zfill(digits)

print(totp(b"12345678901234567890", 59, digits=8))  # 94287082
```

RFCのテストベクトルは、この20バイトのASCII鍵と八桁を使います。基準時点から59秒では、カウンターは1で、SHA-1を使う場合は94287082となり、関数もそれを表示します。アプリが表示する六桁のコードは、末尾の六桁、287082です。計算にネットワークは使わないため、機内モードの電話でも動きます。同じ理由で、サーバーにも、各利用者の鍵のコピーが必要です。

コードの確認は、サーバーで同じ計算をし、比べることです。RFC 6238は、ネットワーク遅延のために認める過去の時間幅を、多くても一つ前までとするよう勧め、認証に一度成功したコードを、再び受け入れることを禁じます。

パスキーは、共有する秘密を、鍵の対に置き換えます。2026年8月25日からW3C勧告となったWeb Authentication Level 3では、認証器は、ドメインで識別するーサイト用の鍵の対を作り、サーバーには公開鍵だけを渡します。ログイン時には、ブラウザーが、実際につないだオリジンを渡し、認証器は、そのオリジンとサーバーのチャレンジを一緒に署名します。そのため、似せたドメインで作った署名は、本物のサイトでは使えません。鍵を使うためにPINか指紋が必要なら、パスキーは、一手順で使える多要素認証器になります。WebAuthnは、本人の複数端末にコピーするパスキーを、マルチデバイス認証情報と呼びます。NISTは、同期する鍵をAAL2では認め、AAL3では認めません。同期できる鍵は、定義上、書き出せる鍵だからです。

MFAを有効にするのは、簡単な部分です。誰に必須とし、どの場面でどの方式を認め、電話をなくしたらどうするかは、アカウント管理であり、アクセス全体の仕組みの一部です。このサイトのセキュリティーとアクセス管理サービスは、その仕組みを説明します。各人が仕事に必要なアクセスだけを持ち、退職する日に失うよう、アカウント、パスワード、権限を整えます。

問題が起きるところ

フィッシング画面が、コードを中継します。 使い捨てコードは、利用者が読んで入力する数値で、どのサイトを見ていたかは記録しません。偽の画面が、時間幅の間に、パスワードとコードを本物へ転送すれば、本人としてログインできます。そのためNISTは、出力を手入力する認証器は、どれもフィッシング耐性がないとし、偽サイトへどう到達したかも関係ないと付け加えます。対策は、応答を検証側と結び付ける認証器です。WebAuthnのようにドメイン名に結び付けるか、クライアント証明書やスマートカードのように、TLSチャネル自体に結び付けます。管理者には必須とし、ほかの全員にも提供します。

電話番号が要素になります。 SMSのコードは、その時点で、携帯電話事業者がその番号をつなぐ相手へ届きます。NISTの脅威一覧には、事業者を説得して被害者の番号を転送させ、コードを受け取る攻撃者も含まれます。SMSや音声で送る前に、検証側は、最近のSIM変更、電話番号の移転、端末交換といった兆候を調べるべきです。SMSは主手段でなく予備にし、登録番号の変更前に、ログイン手順の完了を求めます。

承認、承認、また承認です。 承認ボタンだけのプッシュ通知が求めるのは、証拠ではなく判断です。パスワードを既に持つ攻撃者は、利用者が通知に疲れて一つを認めるまで、送り続けられます。NISTは、認証疲労と呼び、単に承認する方式を、もはや認めません。一つのチャネルから、もう一つへ秘密を渡す必要があります。例えば、ログイン画面の番号を、本人が電話に入力します。短い一覧から選ぶだけでは、推測できるため不十分です。指針が勧めるように、最後の成功したログイン以後に送れる通知数も制限します。

復旧が第二の要素を飛ばします。 「電話をなくした」というリンクが、パスワードだけを求めたり、メールで再設定リンクを送ったりすれば、MFAは、攻撃者が断れる選択肢になります。NISTは、AAL2のアカウントの復旧に、異なる方法で得た二つの復旧コード、一つの復旧コードと、まだ登録されている単要素認証器の組み合わせ、または身元確認のやり直しを求めます。復旧のたびに、本人へ通知する必要があります。一般に、一要素で別種類の認証器を取得できてはいけません。パスワードだけで、新しい予備コードを印刷できてはなりません。

コードを二回、または長過ぎる期間受け入れます。 六桁のコードには百万通りがあり、強さは周りの規則から得ます。現在の前後にいくつもの時間幅を認めたり、使用済みを受け入れたりすれば、一つのコードを見た攻撃者に、再利用する期間を与えます。各利用者で最後に受け入れた時間幅を記録し、同じか、それ以前を拒否します。遅延のためにRFC 6238が認める一つ前の時間幅に絞り、連続失敗数にも上限を置きます。NISTは、認証器を無効にする前に認める失敗を、100回以下とします。

コード生成用の秘密を、パスワードの隣に置きます。 アプリのコードは、対称鍵方式です。サーバーは全利用者の鍵を持ち、コピーした者は、全アカウントの正しいコードを作れます。NISTは、鍵を強固に守り、必要なソフトウェア部品だけが読めるよう求めます。パスワードのハッシュの横に、平文の列として置きません。別の秘密保管場所に置く鍵で暗号化し、コードを確認するサービスだけに読ませます。

参考資料

- NIST SP 800-63B-4の定義と略語 — 現行の指針での、多要素認証、多要素認証器、フィッシング耐性の定義です。
- Computer Security Resource CenterのNIST SP 800-63B-4 — 2025年7月に公開され、SP 800-63Bを置き換えたことを示す、公表記録です。
- RFC 6238：TOTP — 時刻に基づく使い捨てパスワードのアルゴリズム、テストベクトル、時計のずれへの規則です。
- RFC 4226：HOTP — 基になる、カウンター方式のアルゴリズムです。六桁以上への切り詰めを含みます。
- Web Authentication Level 3 — オリジンに結び付く認証情報と同期パスキーを含む、パスキーの基礎となるW3C仕様です。

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

「多要素認証とは？」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/multi-factor-authentication/