認証・認可

認証と認可

- 認証（Authentication）：ユーザ自身であることを証明する
- 認可（Authorization）：ユーザに権限を与えること
- 認証後に認可されてファイルにアクセスできる

認証

- 知識情報（自分しか知らないこと）
  - パスワード認証
- 生体情報・所持情報（自分しか持っていないもの）
  - 指紋認証，顔認証，携帯電話による二段階認証
- MFA (Multi-Factor Authentication): 多要素認証
  - 上記のうち二つ以上で認証する

- パスワード認証の脆弱性
  - 漏洩の危険性：ユーザ側から，管理者側から
  - 管理者側からの漏洩を防ぐためハッシュを用いる
  - 逆変換できない関数 $x' = h(x)$
    - x と y の値を知ることなく，x, y が同じかどうかを判定するためには，x' と y' が同じかどうかを確認する

$$x' = h(x), \quad y' = h(y)$$

  - 暗号化されていない通信で生パスワードが流れる可能性

認可

- リソースアクセスの権限を与える
  - 電車の切符（「誰」は関係なく切符を持っている人なら電車に乗れる）
  - 鍵（「誰」は関係なく鍵を持っている人なら家に入れる）
- 通常は，認証で「誰」を特定した後に権限を付与
- 典型的な権限
  - read: 読み込み権限（データを見ることができる）
  - write: 書き込み権限（データを見て変更することができる）
  - execute: 実行権限（データ？を実行することができる）

認可による認証

- 「この家の鍵を持っている人はこの家の人だ」（所持情報による認証）
- OAuth認証
  - 鍵 ↔️ アクセストークン（アプリケーションパスワードとも）
  - アクセストークンはアクセス権が限定されているため，万が一漏洩しても，認証のためのパスワード漏洩よりも影響範囲が一般的に狭い

データ保護のベストプラクティス

- 不必要な人をデータから遠ざける
  - 「誰」ならどこまでアクセスしてよいか
  - セキュリティポリシーの作成
- 安全な鍵の管理
  - 鍵の漏洩をふせぐ
  - 「アクセストークン」でユーザと権限を疎結合にする
- データを暗号化する
  - データの暗号化，改ざん防止のための署名
- データのメンテナンス
  - バックアップ・暗号化を継続的に実施（自動実行）