Operating Systems

ユーザとセキュリティの管理

2020-13

ユーザの権限

- マルチユーザシステム
  - 複数の「ユーザ」による利用に対応したOS
  - 個人用OSでも，通常使用，管理用などのモードとして「ユーザ」がある
- 管理者ユーザ
  - UNIX, macOS: スーパーユーザ（root），ユーザ番号0のユーザ
  - Windows: Administrator （Winサーバ: Domain Admins）
  - もし管理者モードで操作ミスやウイルス感染等をすると致命的
    ⇒ 管理者モードでの作業は最小限にしたほうがよい（sudoの利用）
- ユーザの権限管理
  - スーパーユーザ方式: 管理者にモードになった者が全権限を行使
  - POSIXケーパビリティ: 権限を細かく分割して個別にユーザに付与
  - ACL（アクセス制御リスト）: ファイル側にユーザと可能な操作を登録

グループ

- グループ
  - 通常のOSでは，ユーザは1つ以上の「グループ」に所属する
  - グループ単位でも権限を付与できる
- UNIX系のアクセス管理
  - ユーザ，グループ，その他に対してアクセス権を付与できる
  - 読み（r）書き（w）実行（x）に対して各1ビット ⇒ 8進数3桁で表す
  - 他にプログラムを所有者の権限で実行するsetuidビットなどがある

```
% ls -l test1
% -rw-r--r--  1 shiozawa  staff  819  6 18  2018 test1
```

ユーザ　グループ　その他　ユーザ名　グループ名

```
% chmod 764 test1
% ls -l test.c
% -rwxrw-r--  1 shiozawa  staff  819  6 18  2018 test1
```

111 110 100 = rwx rw- r--

POSIXケーパビリティ

- スーパーユーザ権限を細分化
  - それぞれのユーザに必要な権限のみを付与できる
- Linux 3.6のケーパビリティ
  - 以下のリストでは，語頭の「CAP_」は省略している
  - CHOWN, DAC_OVERRIDE, DAC_READ_SEARCH, FOWNER, FSETID, KILL, SETGID, SETUID, SETPCAP, LINUX_IMMUTABLE, NET_BIND_SERVICE, NET_BROADCAST, NET_ADMIN, NET_RAW, IPC_LOCK, IPC_OWNER, SYS_MODULE, SYS_RAWIO, SYS_CHROOT, SYS_PTRACE, SYS_PACCT, SYS_ADMIN, SYS_BOOT, SYS_NICE, SYS_RESOURCE, SYS_TIME, SYS_TTY_CONFIG, MKNOD, LEASE, AUDIT_WRITE, AUDIT_CONTROL, SET_FCAP, MAC_OVERDRIVE, MAC_ADMIN, SYSLOG, WAKE_ALARM, BLOCK_SUSPEND

Windows ACL

- オブジェクト側(ファイル等)が保持
  - ファイルに,各ユーザ/グループへのアクセス許可(ACE)を設定
  - 上位フォルダから下位フォルダ・ファイルへのACLの「継承」が可能
- 通常のアクセス許可
  - フルコントロール,変更,読み取りと実行,フォルダ内容の一覧表示,読み取り,書き込み
- アクセス制御エントリ(ACE)
  - フォルダのスキャン/ファイルの実行,フォルダの一覧/データの読み取り,属性の読み取り,拡張属性の読み取り,ファイルの作成/データの書き込み,フォルダの作成/データの追加,属性の書込み,拡張属性の書込み,サブフォルダとファイルの削除,削除,アクセス許可の読み取り,アクセス許可の変更,所有権の取得,同期

Androidパーミッショングループ

- アプリごとに権限を付与

グループ	パーミッション
CALENDAR	READ_CALENDAR, WRITE_CALENDAR
CAMERA	CAMERA
CONTACTS	READ_CONTACTS, WRITE_CONTACTS, GET_ACCOUNTS
LOCATION	ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION
MICROPHONE	RECORD_AUDIO
PHONE	READ_PHONE_STATE, CALL_PHONE, READ_CALL_LOG, WRITE_CALL_LOG, ADD_VOICEMAIL, USE_SIP, PROCESS_OUTGOING_CALLS
SENSORS	BODY_SENSORS
SMS	SEND_SMS, RECEIVE_SMS, READ_SMS, RECEIVE_WAP_PUSH, RECEIVE_MMS
STORAGE	READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE

ユーザやシステムの情報管理

- 各種システム設定
  - UNIX系：/etc に各ソフトウェアが設定ファイルを作るのが伝統的
  - Windows：レジストリAPIをOSが提供（regedit.exeで編集できる）
  - 近年では，GUIの設定ソフトウェアが充実している
- ユーザ・グループの設定
  - UNIX系：/etc/passwd, /etc/group ファイル等
  - Windows：GUIの「設定」，「net user」コマンド等
- ホームディレクトリ
  - 各ユーザが所有者となっているデフォルトのカレントディレクトリ
  - Windowsでは，レジストリの内部にもユーザごとの領域がある

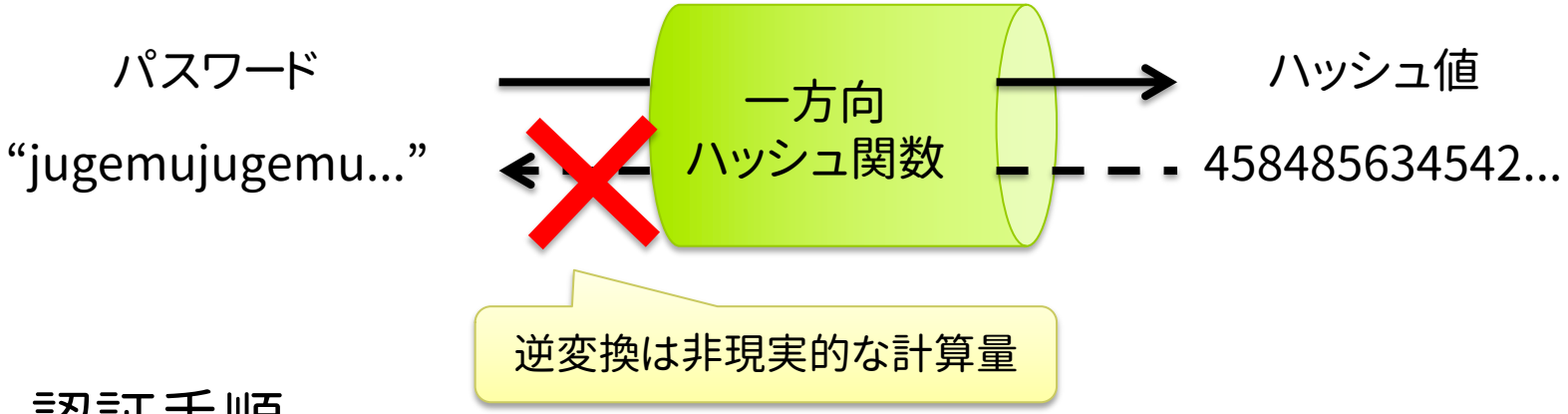
ユーザの認証

- パスワード，暗証番号（PIN），パスフレーズ
  - 最も一般的で，長いものから短いものまで種々の方式がある
- ワンタイムパスワード（使い捨てパスワード）
  - 毎回または一定時間ごとに，パスワードや暗証番号が変わっていく
  - 「パスワード計算機」（スマホアプリもある）を持ち歩く必要がある
- ハードウェアキー
  - 磁気カード，ICカード，RFIDなど，物理的な鍵となるもの
  - ドングル（USBコネクタなどに指す）
- 生体認証
  - 指紋，手のひらや指の静脈パターン，指の長さ，眼の光彩など…
  - 顔認識，音声認識，脳波の反応（本人が意図しない特定にも）

パスワード認証

何百万件ものパスワード漏洩がよくニュースになる

- パスワードの格納方式
  - パスワードを平文のまま保存すると，盗聴や情報漏洩に弱い
  - 対策として，パスワードを高度な「一方向ハッシュ関数」（暗号学的ハッシュ関数）によって，ハッシュ値（整数）に変換して保存する



- 認証手順
  - 入力されたパスワードを，ハッシュ関数でハッシュ値に変換する
  - 変換されたハッシュ値と，システムが持つハッシュ値を比較する
  - 両者が一致していれば，認証成功（本人確認）とする

チャレンジ&レスポンス認証

- 盗聴対策をしたパスワード認証
  - ネットワークサービスなどで，パスワードの盗聴防止に用いられる
  - 下記の「チャレンジ」を事前にサーバとクライアントで共有しておけば，ワンタイムパスワードの一方式として利用できる
  - ただし，サーバにパスワードを平文で保存すると漏洩の危険があり，ハッシュ化してもハッシュ値が漏洩すれば不正アクセスの危険はある

- 認証手順
  - クライアントから通知されたユーザIDに対し，サーバは「チャレンジ」と呼ばれるデータを乱数等によって生成し，クライアントに送信する
  - クライアントは，そのチャレンジとユーザが入力したパスワードを合成して暗号化（ハッシュ化）し，「レスポンス」としてサーバに返信する
  - サーバも，同じチャレンジと自分が持つユーザのパスワードで同様に暗号化を行い，レスポンスと照合して一致すれば認証成功とする

デジタル署名による認証

- デジタル署名，電子印鑑
  - 一般的に公開鍵暗号系を使う
  - 盗聴による「リプレイ攻撃」に弱いので，チャレンジ&レスポンス方式等と組み合わせる（サーバが毎回生成する乱数を暗号化して返信）
- 認証手順
  - ユーザは，事前にシステムに自分の「公開鍵」を登録しておく
  - ユーザは自分の「秘密鍵」で情報を暗号化し，システムに送信する
  - システムは，ユーザの「公開鍵」でそれを復号化して，認証する
- 公開鍵暗号系
  - 暗号化の鍵（秘密鍵）と復号化の鍵（公開鍵）が異なる暗号方式
  - つまり，暗号化の方法が分かっても，復号化の方法は分からない
  - 秘密鍵と公開鍵が正しく対応しているときだけ復号化できる

システムセキュリティ

- 設計面からの対策
  - 強い暗号アルゴリズムを使用する
  - 1カ所を破られても致命的な影響が生じないような設計を採用する
- 開発面からの対策
  - 影響が大きいセキュリティ問題の原因は，ソフトウェアのバグが多い
  - 常に裏をかかれる可能性を考え，「防衛的なコーディング」をする
  - 特に，ユーザやネットワークからの入力は，疑って入念にチェックする
- 運用面からの対策
  - システムの脆弱性・セキュリティホールを発見したらすぐに修正する
  - 弱いパスワードを利用しているユーザなどへ警告・啓蒙を行う
  - 人間的な不注意や無知による情報漏洩に注意する（ゴミ捨て等も）

マルウェア(1)

- マルウェア(malware)
  - 悪意のあるソフトウェア(malicious software)のこと
- コンピュータウィルス
  - 狭義では,自己増殖機能(感染機能)を備えるマルウェアのこと
  - 例えば,PCの中で実行ファイルに次々に感染していく
- トロイの木馬
  - 一見有用なプログラムを装い,ユーザを騙してシステムに侵入する
  - 侵入した後は,そのユーザの権限を乗っ取る(「踏み台」にする)
- スパイウェア
  - システムを密かに監視し,ユーザのプライベートな情報を収集する
  - すべてのキー入力を収集するキーロガーなどがある

マルウェア(2)

- ワーム(ネットワークワーム)
  - ネットワークを介して,マシンからマシンへと自動感染して広がる
  - 主にネットワークプロトコルの脆弱性(弱点)を攻撃して広がる
  - インターネットの初期には,ワームで機能不全になったこともある
- バックドア(裏口)
  - システムに侵入するために,何者かによって仕掛けられた裏口
  - プログラマ(開発者)によって,こっそり仕掛けられていることもある
  - 侵入者がバックドアを設置して残したり,ユーザから侵入を発見されないように偽装するためのソフトウェアを「ルートキット」と呼ぶ
- ランサムウェア
  - 感染するとデータを暗号化するなど,システムの通常利用を制限し,解除のためには身代金(ransom)の支払いを要求する

悪意のあるコンテンツ

- スパム（迷惑メール）
  - インターネットのメールの少なくとも90%以上はスパム
  - 膨大な数のマシンが「ボットネット」で操られ，スパムを送信している
  - 法律で禁止されているので，発信元が明白な場合はIPAに通報する
- フィッシング（Phishing）サイト
  - 信頼できるWebサイトに偽装して，パスワード等を収集する
  - 対策として，URLは自分で入力するか，信頼できるところからたどる
  - バナー広告や検索結果や紛れ込んでいる場合もあるので注意
- 架空請求，ワンクリック請求等
  - メール等で，身に覚えのない請求書が届く
  - Webページのリンクをクリックすると，「ハイ契約したね，金払え」
  - 無視するか，警察や裁判所の連絡先を“自分で”調べて確認する

攻撃の方法

- パスワードクラック（解読）
  - 辞書攻撃，総当たり攻撃（brute force attack），GPUによる並列処理
  - 「ソーシャル・エンジニアリング」が有効（誕生日やペットの名前を調査）
- サービス不能攻撃（DoS: Denial of Services）
  - 想定外（例えば大量）のデータを送り，システムを機能不全にする
  - 例） scanf("%d", &n); ←「a」と入力したらどうなる？
  - 多地点・多数のマシンから攻撃するものを，DDoS（分散DoS）と呼ぶ
- 標的型攻撃 / ソーシャル・エンジニアリング
  - 標的型攻撃：特定の組織を攻撃対象としてカスタマイズした攻撃（例：長年の顧客を装ったメールで特別なトロイの木馬を送る）
  - ソーシャル・エンジニアリング：技術的な攻撃ではなく，ニセ電話など人間による情報収集（他に，掃除係になりすまして建物に侵入など）

攻撃の技術(1)

- バッファー・オーバーフロー(オーバーラン)
  - 非常に長い文字列などで,プログラムの入力バッファをあふれさせる
  - 最善でもDoSの被害を受け,最悪ならシステムを乗っ取られる
  - スタック上のバッファをオーバーフローさせれば,変数の値や関数の戻り先アドレスを書き換えてプログラムの挙動を変えることもできる
  - 例) char buf[100];

    scanf("%s", buf); ← 10000文字入力したらどうなる?

- 整数値オーバーフロー/アンダーフロー
  - 想定外の入力値によって,整数型による演算の結果(変数値など)をその整数型の範囲外(あるいはマイナス値)にさせる
  - プログラミング言語(CやJava)によっては,計算結果が範囲外になってもエラーにならず,下の桁だけを有効として動作し続ける
  - 想定外の変数値によって,想定外の実行結果が起きてしまう

攻撃の技術(2)

- コード・インジェクション(注入)
  - SQLインジェクション,クロスサイトスクリプティング(XSS)など
  - 入力文字列を工夫してSQL文などを“注入”し,それを実行させる
  - プログラムが,ユーザからの入力データを他のソフトウェア(データベース,OSのファイル処理など)に渡す仲介処理のすきを突く
  - 対策としては,外部からの入力データの「汚染チェック」を徹底する

- レースコンディション(競合状態)
  - 排他制御(セマフォやロック)の不備や隙を突き,整合性を壊す
  - ファイルの差し替えなどによって,プログラムの動きを操れることも
  - 例) ファイルを確認して読み込む瞬間に,他へのリンクに差し替える
  - 例) 同じファイルにアクセスするプログラムを,複数同時に実行して,不具合を生じさせる

ネットワークからの攻撃

- 盗聴・傍受
  - バス型ネットワークや無線ネットワークでは盗聴が可能
  - 新しい方式で暗号化しても，解読方法といたちごっこになっている
- なりすまし・偽装
  - 通信プロトコルにおける発信元アドレスを偽装し（IP Spoofing），他のマシンになりすまして情報を取得するなど
  - ルータなどの通信の中継点に侵入し，プロトコルや暗号化を操作して攻撃するものは，Man-in-the-middle attack（中間者攻撃）という
- ポートスキャン
  - ネットワークにどんなマシンがあってどんな通信を受け付けているか，手当り次第に通信を試みてスキャンする
  - これ自体は攻撃ではないが，攻撃対象を絞り込むのに使われる

演習課題

- 課題 13a セキュアプログラミング
  - この課題の狙いは，OS等のシステムプログラミングで特に重要になるセキュリティへの配慮について考察することである
- 手順
  - 入力として，代表者の名前（文字列），参加者の人数（整数），参加者全員分の年齢（整数）のデータを取得し，20歳以上の参加者が何人いるか出力するプログラムについて考察する
  - そのプログラムの動作を意図的に混乱させるような，可能な限りいろいろな想定外（※1）の入力を考え，具体的な入力値の例を示せ
  - さらに余裕があれば，このプログラムを実際にC言語で作成し，異常な入力が与えられてもなるべく適切に対処できるようにしてみよ
  - （※1）異常に長い文字列，空白や特殊文字の含まれた文字列，桁数が多い数値，範囲外の数値，データ型が違う入力など

次回：小テストとプログラミング実習

- 小テスト（45分?）
  - 第8回～第13回に関する知識・用語問題
- プログラミング実習
  - 第14回の演習課題の説明と実習
  - 第8回～第14回の演習課題の再確認と実習
  - レポート提出はその次の第15回