なぜ、Webサイトが狙われるのか。そして何をすべきか。

わんくま同盟 東京勉強会 #27

上野 宣

上野 宣 （うえの・せん）

- 株式会社トライコーダ 代表取締役

- 著書など
- 今夜わかるTCP/IP
- 今夜わかるHTTP
- 今夜わかるメールプロトコル SMTP/POP3/IMAP4
- ネットワーク初心者のためのTCP/IP入門
- ネットでライフハック―仕事をらくらく片付ける超便利!ウェブツール
- 平成20年度 テクニカルエンジニア 情報セキュリティ試験によくでる午前問題集
- @IT、ScanNetSecurity、HackerJapanなどに連載中、その他多数
- セキュリティ＆プログラミングキャンプ講師

詳しくはWebで

また上野宣か

検索

安全なWeb？

安全な”車”なら想像がつく

今日の目的

- Webの安全について知ってもらうことが目的
- そのためには、現在のWebを取り巻く現状を知っていただき、どうやって守ればいいかを知る必要がある

- なぜ、皆さんがWebサイトを安全にしなければならないのか？
- 攻撃はどこのWebサイトでも請ける可能性があることを知って下さい
- ウチには狙われる資産がない？
  - そのサーバー自体を狙っている
  - 御社の信用力を利用しようとしている

- なぜ、御社のWebサイトが攻撃されるのか。
- そして、守るためには何をすればいいのか。
- これらを学びましょう。

なぜ、御社のWebサイトが攻撃されるのか？

前日惊闻日本政府不承认1937年南京大屠杀这一历史恶行

为强烈抗议日本政府的这一丑恶行为，特借你站传达中华民族的正义声音

本世纪三、四十年代，在日本军国主义发动的侵华战争中，中国军民伤亡３５００多万人，仅在“南京大屠杀”暴行中就有３０万以上的同胞遇难，这是人类文明史上最残暴、最黑暗的一页，・・・・・・我们要更加坚定振兴中华、维护世界和平决心与信念。

大和民族？一个不敢面对历史的民族，是亚洲的耻辱！

1937年，你们的祖先犯下了人类文明史上的滔天罪行，为什么这些bitchson不敢承认？

Japanese? As all peoples know, It's a folk which has no courageface to the truth of history It's the disgrace of Asion !

- これは
- 今は金

具体的な被害

- 換金できる情報を盗まれる
  - 個人情報やクレジットカードなどの情報
- フィッシング詐欺などの温床になる
  - Webにフィッシングサイトを構築されてしまい、訪れたユーザーから個人情報などを盗む
  - サイトにある個人情報などが欲しいわけではない
  - あなたのサイトの信用力が欲しい
  - 信用力を使って、フィッシングサイトを構築

mixii ミクシイ

永久完全無料 0円宣言!!

運営者からのお知らせ

・円滑なコミュニケーションをおこなうためにも他の人の迷惑となる行為は慎みましょう。
・完全無料登録でマイミクを増やそう。
・プロフィールを充実させましょう！
・動画をアップロードしよう！

真紀さん(23)

マイミク一覧



自己紹介

名前	佐藤　真紀
性別	女性
年齢	21歳
生年月日	3月9日
血液型	A型
趣味	男遊び♪
職業	大学生
所属	彼氏欲しい連合www
出身地	北海道

マイミク無料参加

PCアドレス
性別 男性
ニックネーム 名無しさん
年齢 秘密(18以上)
お住まい 東京
希望の部屋 秘密のセフレ探し☆
メッセージ 自分の住んでる地域を書けば、近所の方からの返信率が高くなります。

具体的な被害

- 次の攻撃の拠点（踏み台）に活用される
- スパム配信の拠点
- DDoS攻撃の拠点

- セキュリティ上の問題があるサーバーなら何でもいい

どうやって攻撃しているのか

-
-

攻撃成功後の行動

- 見つからないように潜伏
- こっそりサーバーを利用する
- こっそり個人情報やクレジットカード情報などを盗み続ける
- 攻撃されていることを運営者が気が付かないことも多い

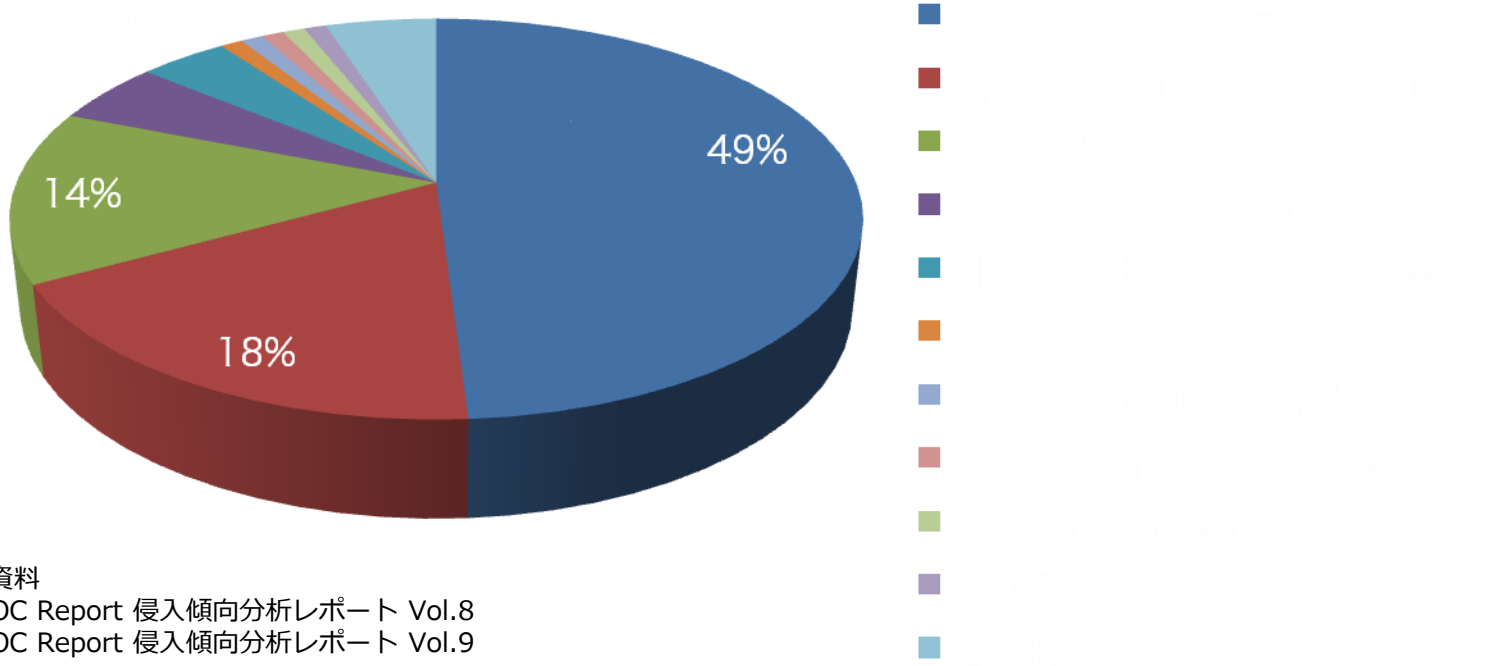
- 御社のWebサーバーに個人情報があろうがなかろうが狙われる時代になっている
- 攻撃者は誰でもいい
- 無差別に弱いWebサーバーを狙う

攻撃手法の変遷

- ターゲットはサーバーからWebアプリケーションへ
- サーバー、OS、市販アプリケーション
  - 厳しいセキュリティ要件とテストを実施
- Webアプリケーション
  - 低予算、機能優先であることが多い
  - 脆弱性のあるWebアプリケーション

2007年度の国内の侵入傾向

- Webアプリケーションへの攻撃が急増
  - 独自開発のWebアプリを狙った攻撃が増加
  - SQLインジェクションは前年度比約6倍



参考資料
・JSOC Report 侵入傾向分析レポート Vol.8
・JSOC Report 侵入傾向分析レポート Vol.9

脆弱性のほとんどはバグ

- 誰の問題？
  - 営業
  - プログラマ
  - 設計者
  - 経営者
- もし被害が起きた場合、発注者が責任を負うことになる

カカクコム事件

- 不正アクセスにより、22,511件のユーザーのメールアドレスが盗まれる
- SQLインジェクションによりWebページが改ざんされ、トロイの木馬型ウイルスを仕掛けられる
- 改ざん後はWebサイトを10日間閉鎖
  - 2005年5月14日～24日

事件の被害

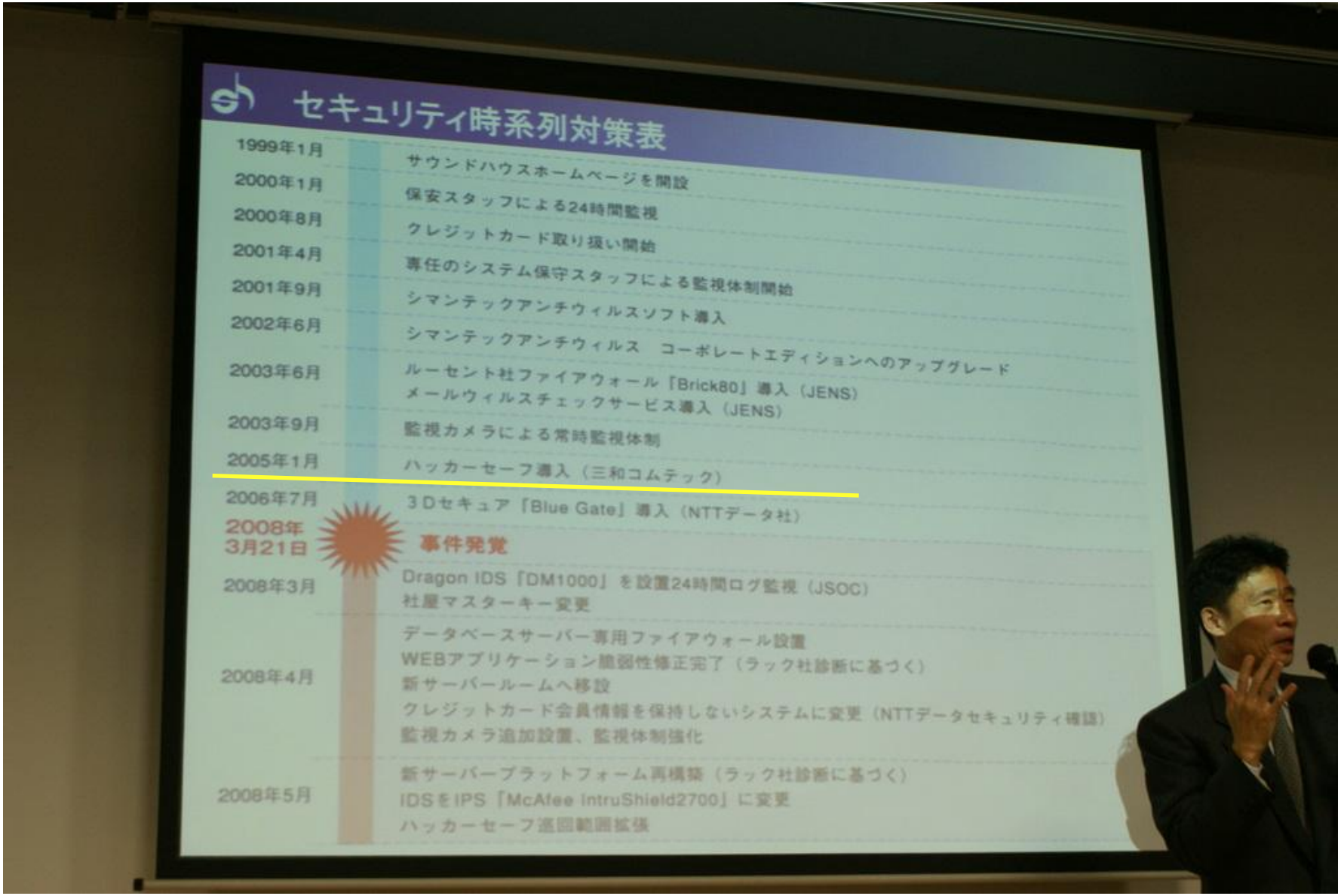
- 個人情報が盗まれたユーザーの被害
- ウイルスに感染した可能性のあるユーザーの被害
- 株式市場への影響
- 社員のモチベーションの低下
- サイト閉鎖によるビジネス上の損害

でも、

- Webアプリケーションに問題がなければ事件を防ぐことはできた

サウンドハウス事件

- 外部からの不正アクセスにより、
  最大97,500件の顧客データが流出した可能性
  - 2008年3月11日～22日に掛けて
- 約2年前にSQLインジェクションの痕跡



WAS Forum 2008 「Web攻撃の脅威に立ち向かうには」
講演者：株式会社サウンドハウス　代表取締役社長　中島尚彦

でも、

- Webアプリケーションに問題がなければ事件は防ぐことができた

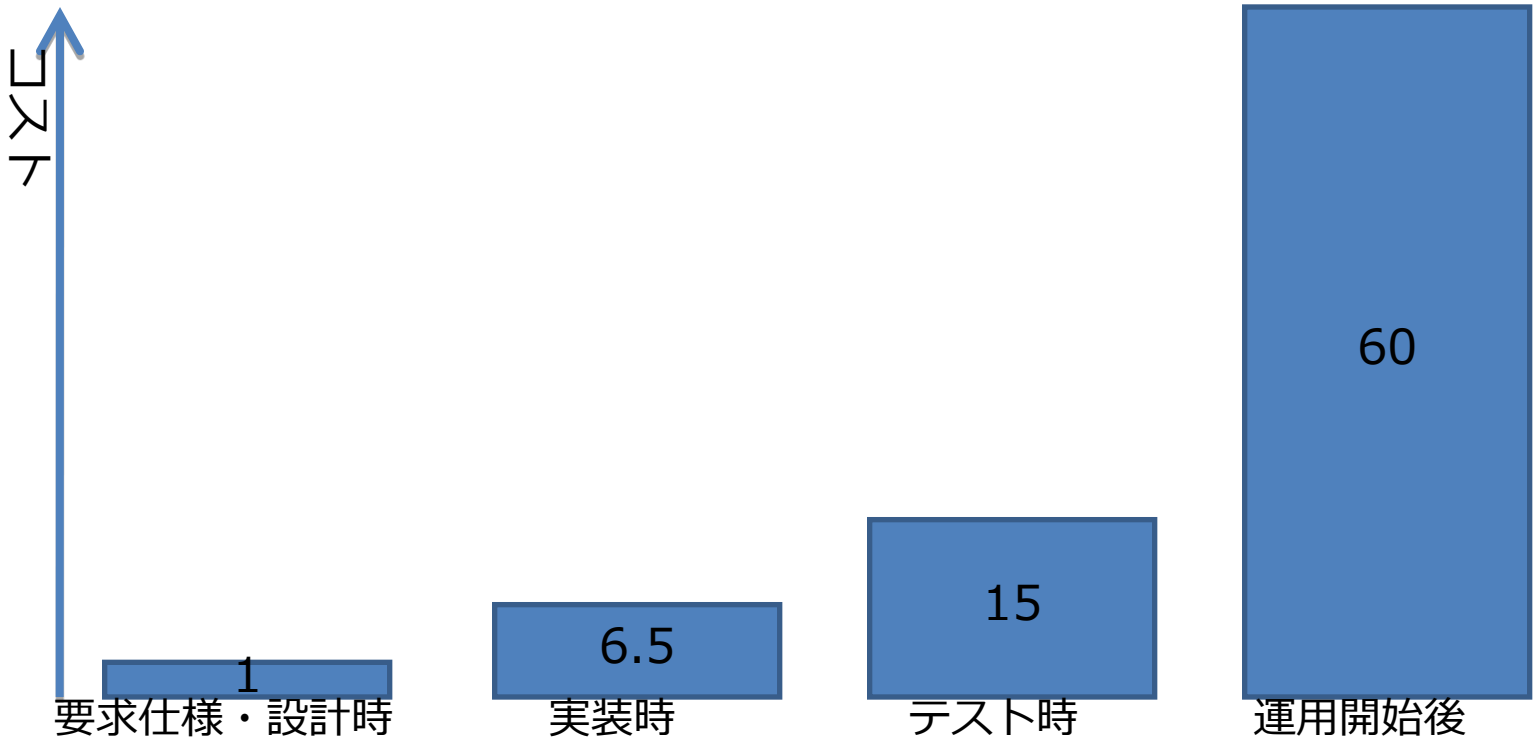
守るためには
何をすればいいか

要件です。設計です。

- 安全なWebサイトを作るには設計以前が重要
- 早い段階で対応する方がコストが少ない
  - 要求仕様や設計の不備は、後のフェーズ全てに影響する

安全なWEBサイト設計のために知っておくべきこと

セキュリティホールを修正するコスト（フェーズ別）



参考：Kevin Soo Hoo, “Tangible ROI through Secure Software Engineering.”Security Business Quarterly, Vol.1, No.2, Fourth Quarter, 2001.

安全でないWebサイトができる例

- 要求仕様や設計だけが重要なわけではない

- 開発に本来必要な工数が確保できていない
- 機能実装を優先し、異常系に不備がある
- 安全でないコードの再利用
- プログラマのスキル不足
- テストケースの不備

- 環境、運用の問題
  - ネットワーク・サーバ環境の不備
  - 運用開始後に出現した新たな脅威

開発に必要な工数の目安（割合）

要求仕様作成、設計	1/3
実装（コーディング）	1/6
単体テスト	1/4
統合テスト	1/4

- 守るべき資産に応じて、「どの程度、安全にすべきなのか」という基準を知っておく必要がある

- 守るべき資産とは何か？
  - 利用者の情報
  - サービスレベル（会社・サービスの信頼）
  - 売上
  - 会社

セキュリティはトレードオフ

- 何かを求めれば、何かを犠牲にする二律背反の関係

- コスト
  - 実際の費用
    - 開発費、維持費、運用費
    - 教育費、間接費（営業、総務、地代など）など
  - 機能を実装しなかったことによる機会損失
    - 実装していれば増えたはずの売上
- 利便性
- リスク

何を調整してセキュリティに割り当てるか

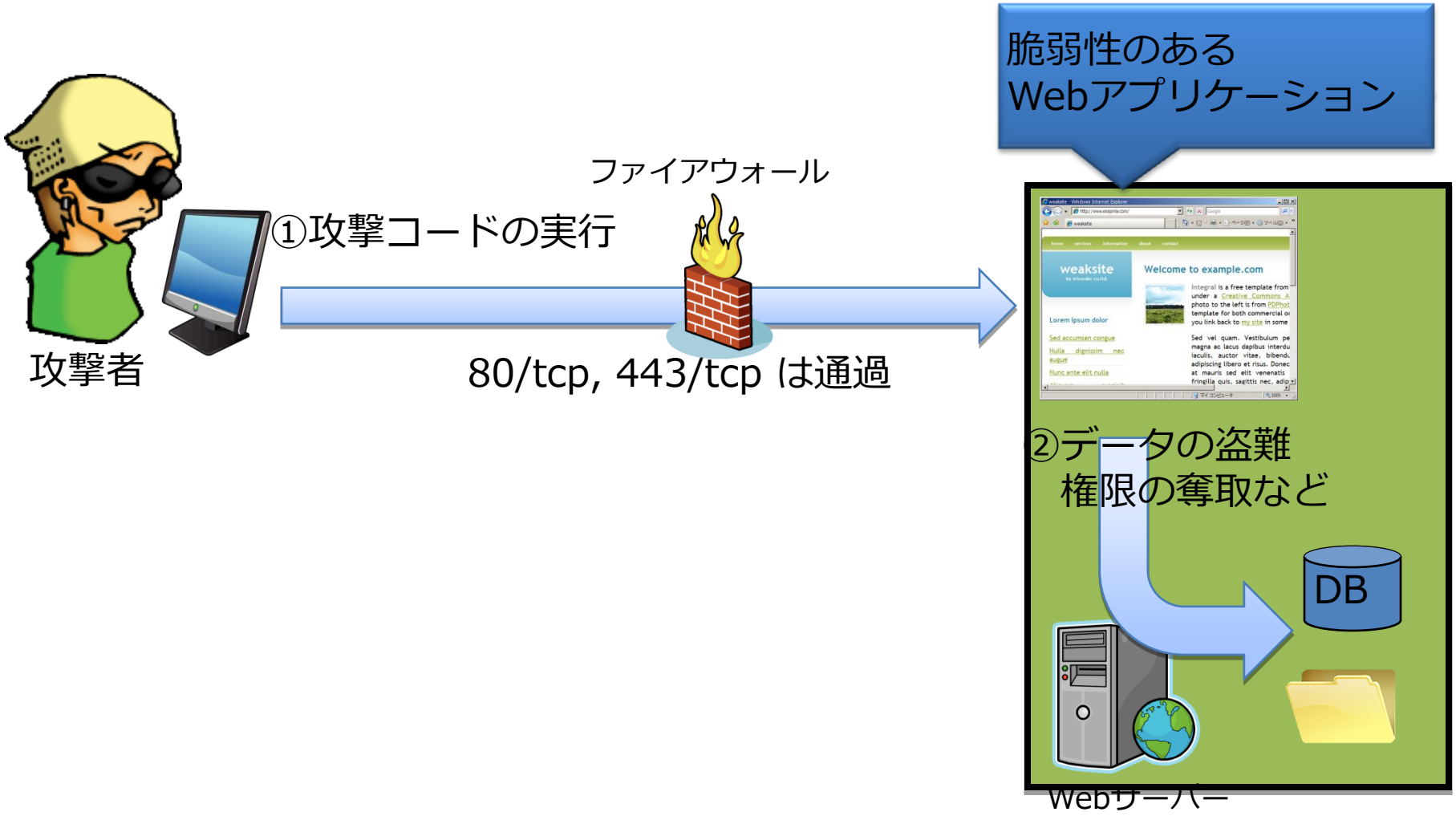
- コスト
  - セキュリティのためのコスト増加は容認されないこともある
- 利便性
  - セキュリティ向上のために利便性を犠牲にできるのか？
- リスク
  - 調整できるのはリスク！
  - リスクの緩和を戦略として取り入れる

リスクの緩和を戦略として取り入れる

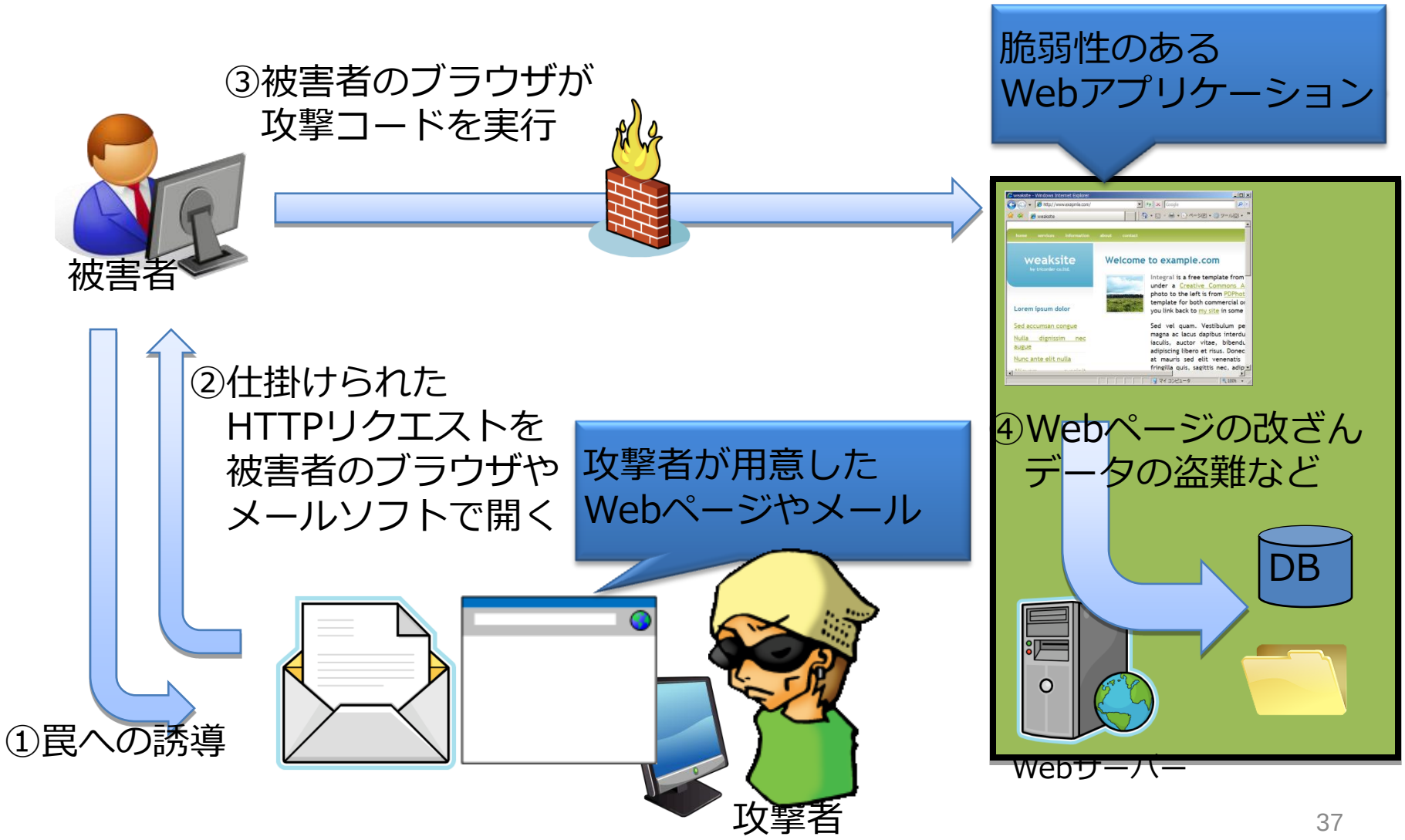
- リスク前提で運用を続ける
  - リスクを受け入れる、許容範囲内に納める
- リスクを回避する
  - 機能をあきらめる、システムを停止する
- リスクの限定
  - 影響を一定レベルに抑制する、インシデントの検知、サポート体制を整える
- リスク計画
  - リスク緩和プランを策定して、リスク管理を行う
  - 事前合意を得ておく、復旧計画を立てる
- リスクの委譲
  - 保険に加入（個人情報漏洩保険など）、損失を他で補う

WEBアプリケーションへの攻撃

Webアプリケーションへの攻撃



受動的攻撃


③被害者のブラウザが
攻撃コードを実行
脆弱性のある
Webアプリケーション
被害者
②仕掛けられた
HTTPリクエストを
被害者のブラウザや
メールソフトで開く
攻撃者が用意した
Webページやメール
④Webページの改ざん
データの盗難など
DB
①罠への誘導
攻撃者
Webサーバー

Webアプリケーション脆弱性ランキング

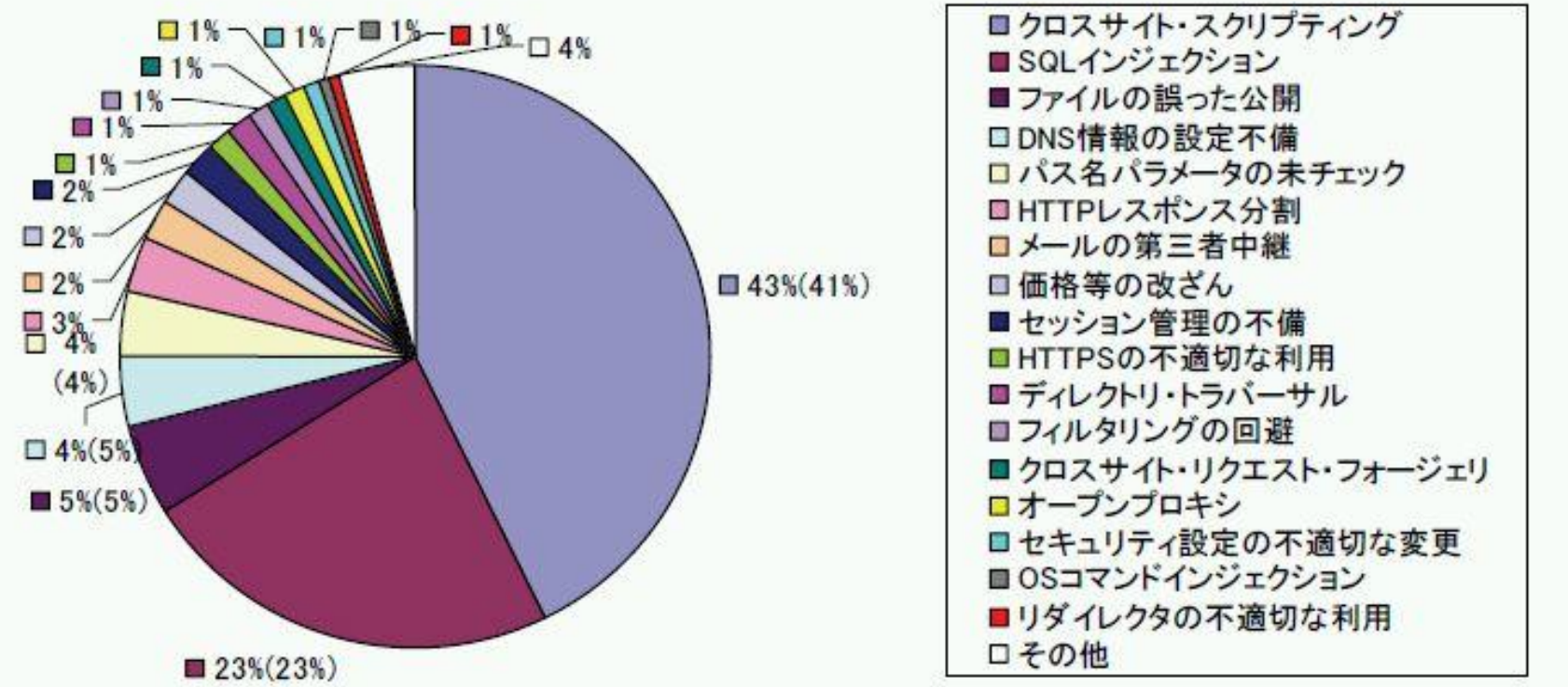


図 3-1 ウェブアプリケーションの脆弱性種類別内訳（届出受付開始から2006年12月末まで）

情報セキュリティ白書 2007年版
（http://www.ipa.go.jp/security/vuln/documents/2006/ISwhitepaper2007.pdf）より

攻撃手法（１）

- Webアプリケーションに依存しない
  - 既知の脆弱性
  - 製品の設定ミス
  - パスワード攻撃
  - バッファオーバーフロー
- 構成や設定の不備
  - 強制ブラウズ
  - 情報の意図せざる流出
  - 不十分なエラー処理
  - システムのバックドアやデバッグモードの残存

攻撃手法（２）

- 入力チェックの不備
  - パラメータ操作
    - (1) URLパラメータの改ざん
    - (2) hiddenフィールド値の操作
    - (3) Cookie値の操作
  - ステルスコマンド操作
    - (1) SQLインジェクション
    - (2) OSコマンドインジェクション
  - クロスサイトスクリプティング
  - クロスサイトトレーシング
  - HTTPヘッダインジェクション
  - ディレクトリトラバーサル

攻撃手法（３）

- 認証の不備（ロジックエラー）
  - セッションハイジャック／リプレイ
  - セッション固定
  - CSRF（Cross Site Request Forgery ）

安全なWebアプリケーションのためのセキュリティ要件

- 開発者、開発業者任せ
- 「セキュリティを考慮」という要件
- これでは解決しない

- 要求(発注)仕様書にセキュリティ要件を明記することが必要
  - 要件がないものは誰も作らない
  - 要件がないと納品検収時に確認できない

セキュリティ要件の参考資料

- IPA 安全なウェブアプリケーション発注のあり方
  http://www.ipa.go.jp/security/vuln/event/documents/200612_5.pdf
- JNSAセキュアシステム開発ガイドライン「Webシステム　セキュリティ要求仕様（RFP）」編　β版
  http://www.jnsa.org/active/2005/active2005_1_4a.html

- 弊社（株式会社トライコーダ）も
  セキュアWebアプリケーション構築の教育をやっています。

まとめ

- なぜ、御社のWebサイトが攻撃されるのか？
  - 大半は自動ツールによる無差別攻撃
- そして、守るためには何をすればよいのか？
  - リスク管理
  - 適切なセキュリティ要件
    - Webアプリケーションに共通で使える部分は多い

Webアプリケーションのセキュリティ要件（１）

- 画面設計
  - 画面遷移図の提出
  - 新たなウィンドウを開かない
  - Webブラウザのデフォルト設定で動作すること
  - フレームを使用しない
- ドメイン名
  - システムが使用するURLのドメイン名は1つにする

Webアプリケーションのセキュリティ要件（２）

- SSL
  - 暗号化なしに送受信してはならない情報を明確にする
  - 入力欄のある画面をhttps:// にする
  - サーバ証明書は警告の出ないものを使用する
  - SSL2.0を使用しない
- キャッシュ
  - キャッシュを禁止するようにする

Webアプリケーションのセキュリティ要件（３）

- CSRF対策
  - 重要な変更を実施する機能を持つ画面はPOSTメソッドによるアクセスに限定（またはパスワードの再入力）
  - 重要な変更を実施する機能を持つ画面に遷移する前の画面には秘密情報をhiddenパラメータに挿入し次の画面で確認を行う
- クロスサイトスクリプティング（XSS）対策
  - HTMLの出力は直接プリントしない
  - HTMLタグを含む入力を認める機能を設けない

Webアプリケーションのセキュリティ要件（４）

- SQLインジェクション対策
  - SQL呼び出しをするコードは一か所にまとめて抽象化すること
  - SQL文を直接組み立てることをせず、Prepared Statementを使用すること
- セッション
  - ログイン中のユーザーの特定は、セッションIDで行うこと
  - ログインが成功する前の段階でセッションIDを発行しない
  - セッションIDは、cookieまたは、POSTアクセスのhiddenパラメータにのみ格納すること
  - セッションIDはログインする都度、乱数により生成する
  - セッションIDは80ビット以上とする

Webアプリケーションのセキュリティ要件（５）

- ログイン・ログアウト
  - ログイン時にパスワードを数回間違えると、数時間ログインできないロックアウト機構を設ける
  - ログアウト機能を設け、ログアウト機能が呼び出されたらセッションを破棄する
  - 一定時間操作がなかったときに、セッションを破棄（自動ログアウト）すること

Webアプリケーションのセキュリティ要件（６）

- リダイレクタ
  - パラメータで指定されたURLへのリダイレクト機能を設けない
- パラメータ
  - セッションID以外の値を格納するcookieを発行しない
  - URLパラメータに秘密情報を含めない
  - パラメータにファイル名（パス名）を一切含めない

Webアプリケーションのセキュリティ要件（７）

- 実装
  - シェル呼び出しを使用しない
  - 外部コマンド呼び出しを使用しない
  - C言語等のバッファオーバフロー系の脆弱性が所持得る言語を使用しない
  - eval()など指定された文字列を言語として実行する機能を使用しない
  - 暗号を使用する場合は、暗号アルゴリズムにCRYPTRECの電子政府推奨暗号のみを用いること
  - 乱数を用いる場合は、暗号学的に安全な擬似乱数生成系により乱数を生成する