「CSIネットワークマスター虎の穴」 第7回セミナー

安全なWebサイト構築のためのポイント

三井物産セキュアディレクション株式会社
中村 隆之

2006年12月14日

三井物産セキュアディレクション とは

- 設立
  - 2001年3月23日
- 本社所在地
  - 東京都千代田区神田錦町3－26 一ツ橋SIビル8F
- 従業員数
  - 111名（2006年4月1日現在）
- 主な業務内容
  - セキュリティ製品（サーバ、クライアント）の取り扱い
  - ISMSやJ-SOX対応に関するコンサルティング
  - Webアプリケーションに関する教育、脆弱性検査、コンサルティング
  - ネットワーク脆弱性検査
  - ネットワーク監視

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

目次

- Webアプリケーションの脆弱性とその対策
- 安全なWebアプリケーション開発のための基礎知識
  - 認証について
  - セッション管理について
  - セッションIDについて
  - 画面遷移の管理について
  - 入力値チェック
- Webサイトの企画から運用までの各フェーズにおける対策

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Webアプリケーションの脆弱性とその対策

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Webセキュリティ対策は社会的要請に

- 経済産業省　平成18年3月2日(木) 公表
  「安心･安全な情報経済社会の実現のための行動計画」 p.11～p.12 より引用

> **SQLインジェクション等により個人情報の漏洩が多発している最近の状況にかんがみ、政府は、個人情報取扱事業者に対して、自らの保有する情報システムの点検を行い、必要な対策を施すよう、要請する。なお、本要請後にSQLインジェクション等による個人情報漏洩が発生した場合は、個人情報保護法に基づき、必要に応じて、これら事業者に厳格な対応を行うこととする。**

※SQLインジェクション･･･Webアプリケーションの脆弱性のひとつ。詳細は後述。

- Webサイトの開発、運営を行う事業者は、利用者が安心して利用できる安全なサイトにするための対策をとるべきである

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

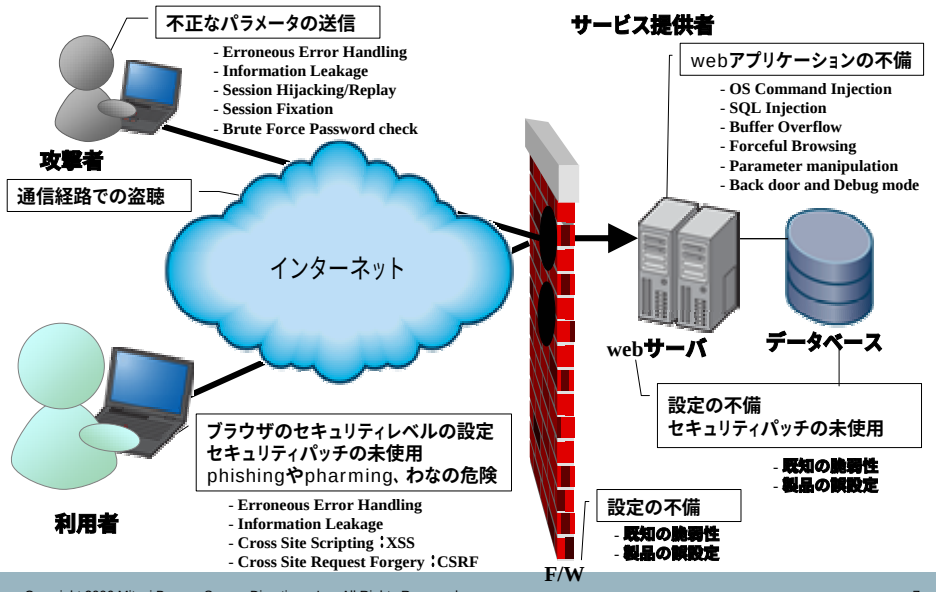
対策を取らないと、どんな被害が起こるのか？

- 個人情報リストの漏洩
- サービス停止
- データ破壊
- サービスの不正利用
- 金銭的被害（オンラインバンキングなど）
- ソースコード漏洩
- 他サイト攻撃のための踏み台　　など

> **被害レベルは、サービスの種類によって異なる**

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Webアプリケーションの脆弱性とは



Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Webアプリケーションの脆弱性の種類

アプリケーション別の脆弱性

- Buffer Overflow
- Cross-Site Scripting
- Parameter Manipulation
- Backdoor & Debug Options
- SQL Injection
- OS Command Injection
- Client Side Comment
- Error Codes
- Forceful Browsing
- Unnecessary Information
- HTTPS Misuse
- Cross-Site Request Forgeries

サイト全体の脆弱性

- Unnecessary File
- Server misconfiguration
- Insecure Cookies
- Session Hijack
- Session Replay
- Session Fixation
- Known Vulnerability

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Buffer Overflow

- 不正なプログラムを実行させることができる
  - バイナリデータを扱う言語(C言語やC++など)で発生する

- 2001年のCodered が有名

  GET/default.ida?NNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0

- 原因
  - ユーザ入力の長さチェックが行われていない

- 対策
  - 長さチェックを行う

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

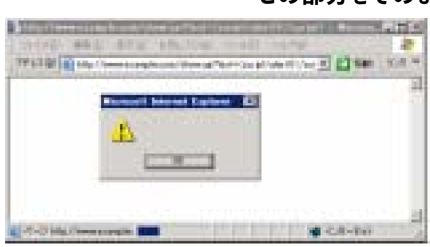
Cross-Site Scripting

- ユーザのブラウザ上で不正なスクリプトを動作させ、不正なページを表示させたり、認証情報を奪取することができる

- 原因
  - ページを出力する際、ユーザからの入力をそのまま出力してしまっている

http://www.example.com/show.cgi?text=<script>alert()</script>

この部分をそのまま出力してしまうと



Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Cross-Site Scripting （Cont'd）

- 対策
  - ユーザ入力を次のページで出力する際は、必ずHTMLエスケープを行う
    （必ず ブラウザへの出力時に行うこと！）
- HTMLエスケープの変換ルール

&	→	&amp;
<	→	&lt;
>	→	&gt;
“	→	&quot;
‘	→	&#39;

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Parameter Manipulation

- パラメータを書き換えて送信することで、想定外の動作を起こさせることができる
- 原因
  - ユーザ側に渡す必要のないパラメータが、ユーザのフォーム内（もしくはURLパラメータ内）に含まれてしまっている
    例えば、ショッピングサイトで商品を購入する際、価格をユーザ側から渡す必要はない

http://www.example.com/buy.cgi?id=30&num=3&price=123

必要ない

- 対策
  - URL内のパラメータ、送信フォームのパラメータ（特にhidden）は、本当にユーザ側に渡す必要があるものか確認する
  - 改ざんされると問題があるパラメータについては、サーバ側で保持する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Backdoor & Debug Options

- 開発者がこっそりと残したもの ・・・ バックドア
- 開発者が消し忘れたもの ・・・ デバッグ オプション

- 攻撃者に発見された場合は、非常に深刻な被害につながる可能性がある

- バックドアに関しては、開発者が作りこむことを防ぐことは難しいが、開発工程において、チーム内でソースコードレビューを実施したりすることで、検出できることもあるのではないだろうか

- デバッグオプションについては、開発時の方針で、デバッグ用コードの扱い方を予め決定しておくことで、消し忘れがあったとしても、リリース前に一括して取り除くことが可能となると思われる

- 対策
  - コーディング規則の作成と徹底

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

SQL Injection

- 不正なＳＱＬ文をＷｅｂアプリケーション経由でデータベース（ＤＢ）に送り、対象外データの閲覧、改ざん、破壊を行うことができる
  最近の個人情報漏洩事件は、ほとんどがこの脆弱性だと思われる

- ＤＢの内容が画面に表示されなくても、不正な入力に対するアプリケーションの挙動によって、ＤＢの内容を抜き取る攻撃もある（Ｂｌｉｎｄ ＳＱＬ Ｉｎｊｅｃｔｉｏｎ）

- 原因
  - ユーザからの入力値をそのまま使用してＳＱＬ文を作成し、ＤＢに渡してしまっている

- 対策
  - ＤＢにユーザ入力値を渡す場合は、ＳＱＬ文で使用される特殊文字をエスケープする
  - または、バインド変数を必ず使用する （こちらを推奨）

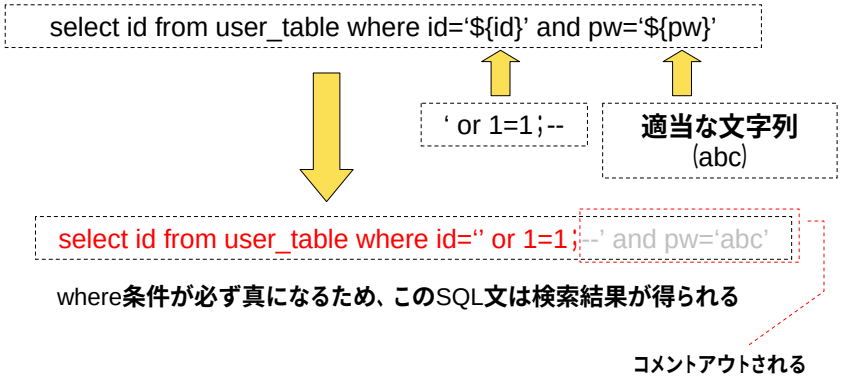
Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

SQL Injection （デモ）

- シングルクォートの入力でエラーが表示される

- 認証を回避する

  指定したID／PWのアカウント情報がDB内に存在すれば検索結果が得られるSQL



where**条件が必ず真になるため、このSQL文は検索結果が得られる**

コメントアウトされる

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

SQL Injection （デモ）

- SQL特殊文字の入力によりエラーは表示されないが、挙動が変化する
  - 常に真となる条件 (xxx' and 1=1--) を与えると、正常遷移
  - 常に偽となる条件 (xxx' and 1=0--) を与えると、エラーが発生

  and 以降の条件が真偽を判定することが可能となる
  これを利用することで、DB内の情報を抜き出すことができる
  (これが Blind SQL Injection)

- Blind SQL Injection のツールにより、DB内のテーブル情報を取得する
  - absinthe
    http://www.0x90.org/releases/absinthe/

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

OS Command Injection

- 任意のＯＳコマンドを実行することができる
- Ｗｅｂサーバのコンソールを奪うことに等しいため、深刻な被害に繋がる可能性がある

- 原因
  - ＯＳコマンドの実行が可能な、「危険な関数」を使用している
  - ユーザからの入力値に含まれる特殊文字のエスケープ処理を行っていない

- 対策
  - ＯＳコマンドを呼び出さないと実行できない処理なのか確認する
  - 「危険な関数」は使用しない
  - 想定している入力の文字種、フォーマットに合致しているか、入力値チェックを行う
  - ユーザ入力をシェルに渡す前に、シェル上での特殊文字をエスケープする

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Client Side Comment

- ブラウザのメニューから「ソースの表示」で表示されるＨＴＭＬソースの中に含まれるコメントの中に、有益な情報が含まれていることがある
- 直接被害に繋がることはないが、場合によっては深刻な被害に繋がる可能性がある（過去の検査にて、個人情報リスト漏洩の被害に繋がった例あり）

- コメントの例
  - 更新履歴
  - 送信パラメータについての情報
  - 内部ロジックに関する情報
  - 使用していないアプリケーションのＵＲＬ（古いフォーム）

- 原因
  - コーディング規則で、コメントについて決められていない

- 対策
  - 挙動に関するコメント類は、プログラム言語のコメント機能を使用する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Error Codes

- プログラミング言語の処理系やミドルウェアのエラーメッセージには、システム情報が含まれていることが多いため、攻撃者に有益な情報となる

- 原因
  - エラーハンドリングに不備がある。エラーハンドリングは、セキュリティ対策というよりアプリケーション開発での基本であるため、必ず行うべきである
  - 攻撃を受けたときだけでなく、例えばサーバの負荷が上がって内部処理が失敗した場合のエラーもあるため、パラメータ操作だけのテストでは検出できないものもある

- 対策
  - きちんとエラーハンドリングを行う
  - エラーが発生した場合は、カスタマイズしたエラー画面を表示する
    但し、詳しいエラーメッセージを含めないよう注意！
    （これをやると、Unnecessary Information の脆弱性になる）

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Forceful Browsing

- 複数の画面を経由する機能における途中の画面（例えば認証後の画面）に直接アクセスすることができる

- 原因
  - セッション管理の仕組みにおける不備
  - 個々のアプリケーションにおいて、統一したステータスチェックが行われていない

- 対策
  - 設計段階で画面遷移図を作成し、
    - ✓ 認証無しでＯＫの画面
    - ✓ 認証が必要な画面
    - ✓ 再認証が必要な画面

    といった情報を記入する。
  - 認証状態のチェックは全てのアプリケーションで共通のコードとすればミスは減るはず

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Unnecessary Information

- ユーザに必要のない情報を表示してしまっている
- 原因
  - エラー画面で出力するメッセージが親切すぎる
- 例
  - ログイン失敗時に「パスワードが違います」と表示する
  - パスワードリカバリで「このメールアドレスは登録されていません」と表示する
- 対策
  - 「IDまたはパスワードが違います」(ログイン失敗時)
  - 「指定したメールアドレス宛に新しいパスワードを送信しました」(パスワードリカバリ)

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

HTTPS Misuse

- 本来、HTTPS(SSL)を使用すべき箇所において、HTTPを使用している
- HTTPSにより実現可能となることは以下の2つ
  - 通信経路の暗号化
  - 電子証明書による認証(サーバ認証、クライアント認証)
- HTTPSを利用する場合の注意点
  - 画面構成
  - サーバ負荷
  - フォーム入力画面もHTTPSで保護
  - HTTPSのサイトでのセッション管理でCookieを使用する場合は、secure属性を付けて発行する(後述のInsecure Cookieを参照)

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Cross-Site Request Forgeries

- 処理のコミット（登録完了、メール送信、振込完了、購入完了、等）を強制的に実行させられてしまう
- 発生する被害は、実行させられてしまう機能に依存する

- 原因
  - 送信ＵＲＬ、送信パラメータを攻撃者が推測可能である

    ショッピングサイトでの購入完了リクエストの例

    http://www.example.com/buy.cgi?productid=AX123&num=10&action=finish

    すべて推測可能

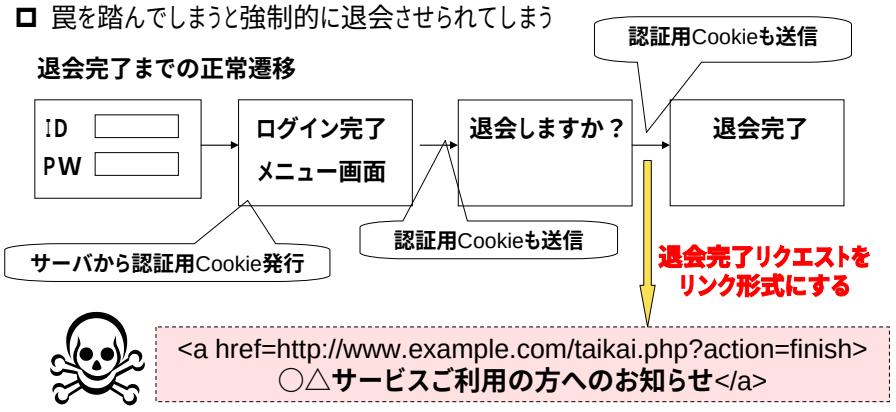
- 対策
  - 送信するパラメータの中に、攻撃者が推測できない値を入れておく

    例えば、
    - ✓ Ｃｏｏｋｉｅに入っているセッションIDと同じ値を hidden に入れておく
    - ✓ ワンタイムトークンを生成し、hidden に入れておく

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Cross-Site Request Forgeries （デモ）

- 罠を踏んでしまうと強制的に退会させられてしまう



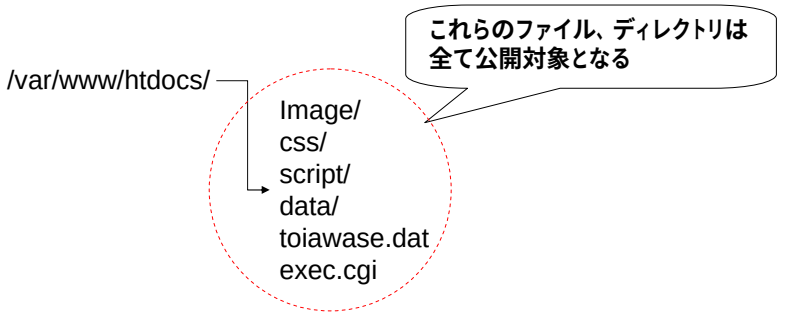
- 上記の罠ＵＲＬを、ログイン中のユーザが踏んでしまったらどうなるか？
  - → 認証済みのユーザが踏むと、認証用Ｃｏｏｋｉｅも一緒に送信されるため、退会処理が完了する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Unnecessary File

- 公開する必要のないファイルがサーバ上に置かれていることがある
  - バックアップファイル
  - 開発中のファイル
  - アプリケーションが使用しているデータ（例：問い合わせデータ）
- 「サーバ上」とは、Webサーバの公開ディレクトリ上のことを指す

例



Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Unnecessary File （Cont'd）

- 対策
  - 本番サーバ上では開発・編集作業は行わない
  - 不要なファイルがないかどうか定期的にチェックする
  - 一時的なキャンペーン等で使用したアプリケーションは、サーバから除去する
    （HTML内でコメントアウトされたフォームから発見されてしまう）
  - サーバ製品の初期インストール状態では、サンプルアプリケーションやマニュアル類も一緒に入ることが多いため、これらがインストールされた場合は除去する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

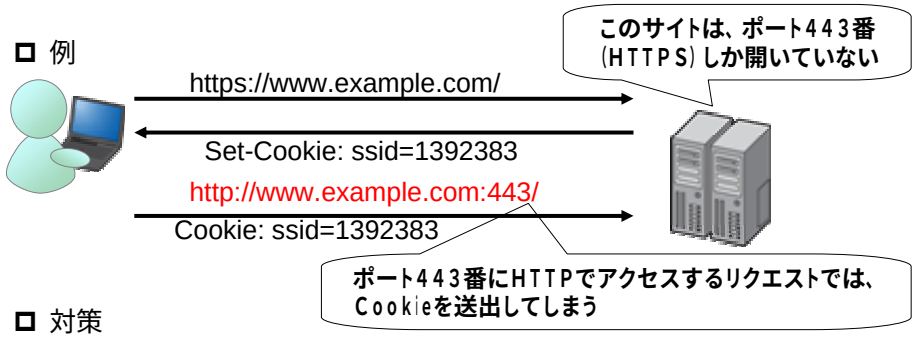
Server misconfiguration

- Webサーバ、アプリケーションサーバ、DBサーバなどの設定ミスによる問題
- 例
  - ディレクトリ内のファイル一覧が表示されてしまった
  - CGIが実行されずに、CGIのソースコードをそのまま出力してしまった
  - 不要なHTTPメソッド（WebDAVなどのメソッド）が動作している
- 対策
  - マニュアルを必ず読み、設定項目を理解して設定を行う
  - 設定変更時に毎回チェックする

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Insecure Cookies

- HTTPSのサイトで発行されるCookieにsecure属性が付いていないため、HTTPでサイトにアクセスするときに、平文でCookieを送信してしまう
- 例



- 対策
  - HTTPSだけで使用するCookieには、secure属性を付けて発行する

```
Set-Cookie: ssid=HKHD3kSA31GAK9F; secure
```

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Session Hijack

- ユーザになりすましてアクセスすることができる
- 原因
  - セッションIDが推測可能
- セッションハイジャックが可能なセッションIDの例
  - 年月日・時間だけで生成した数字列
    550612170411 （2006年12月4日11時55分17秒 を、分年月秒日時 に変換）
- 対策
  - 推測不可能で十分長い値を使用する
  - ミドルウェアが発行する値を使用する

詳細は、後述する「セッション管理について」で解説

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Session Replay

- 何らかの方法で入手した他人のセッションIDをそのまま再利用することで、サイトにアクセスすることができる
- 主な原因
  - セッションIDが不変情報だけで生成されている
  - 有効期限がない、もしくは長すぎる
  - セッションIDをURLパラメータ（http://server/aa.cgi?ssid=XXXXXX）に含めている
- 対策
  - セッションIDの生成要素には、可変情報を含める
  - 有効期限は必ず設ける
  - ログアウト機能では、サーバ側でセッションIDを無効化し、ユーザ情報を切り離す
  - 重要な処理を行う箇所（登録情報の表示・変更、商品購入、退会、など）では、必ずIDとパスワードを求める（再認証）

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Session Fixation

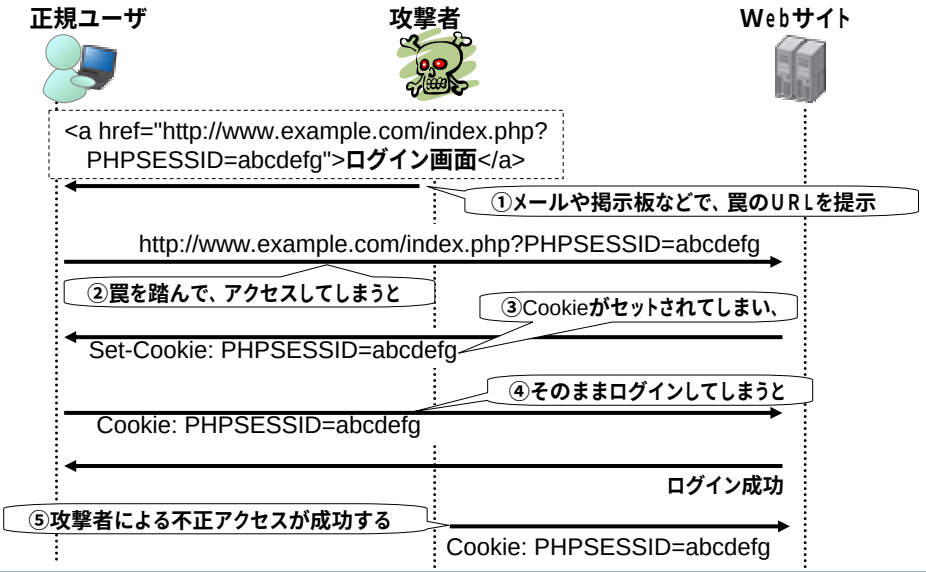
- 攻撃者が指定したセッションIDを、正規ユーザに使用させることができる
- 例
  - URLからセッションIDを受け付ける仕様のミドルウェアも存在する

```
http://www.example.com/index.php?PHPSESSID=abcdefg
http://www.example.com/index.jsp;jsessionid=jk3lsw8fjhgyt10
```

- 対策
  - ログイン成功後に新しいセッションIDを発行する
  - ユーザから送られてきたセッションIDを受け付けないようにミドルウェアの設定を行う

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Session Fixation （デモ）



Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

安全なWebアプリケーション開発のための基礎知識

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

- 認証
- セッション管理
- セッションID
- 画面遷移の管理
- 入力値チェック

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

認証について

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Webアプリケーションにおけるユーザ認証方法

- 認証（Authentication）とは
  - ログインしようとするユーザが、本当にそのユーザであるか確認する方法
- 考えられる方法
  - パスワード
    - ✓ 実装は容易。文字種や長さ、有効期限などに気をつける必要がある
  - クライアント証明書
    - ✓ 非常に強固。B2Bのような、特定少数の相手との通信では有効である
  - IPアドレス
    - ✓ ユーザまでは識別できないが、特定の企業からのアクセスを許可する、といった使い方では有効
  - バイオメトリクス（指紋、静脈など）
    - ✓ 入退室やキャッシュカードの認証用に使われており、強固である。但し、生体情報は変更できないため、偽装されてしまっても変更がきかないため注意が必要である

（参考）金融取引における生体認証について　2005.4　横浜国立大学 松本教授
http://www.fsa.go.jp/singi/singi_fccsg/gaiyou/f-20050415-singi_fccsg/02.pdf

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

安全なパスワードを使用させるには

- パスワードは、文字種、長さ、含まれる文字列に制限を設ける
- 文字種
  - 英数字＋記号
    （英字のみ、数字のみ　ではダメ）
- 長さ
  - 攻撃速度とパスワード空間、パスワードの寿命から求める
  - 8文字以上が安全（但し、BufferOverflowを防ぐため、最大長の制限も設定する）
- 含まれる文字列
  - ユーザIDや生年月日の情報が含まれている場合や、辞書に記載されている単語、その他、よく使用されるパスワードについては、パスワードとして使用させない

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

パスワードクラックに必要な時間（参考）

ネットワーク経由でパスワードクラックを行った結果（100MbpsLANを使用）平均的PC1台使用時

文字数	時間
1	
2	3分30秒
3	3時間30分
4	9日
5	**1年半**
6	**95年**
7	5,900年
8	36万年

パスワードファイルをローカルマシンでクラック

文字数	時間
1	0.00062秒
2	0.038秒
3	2.38秒
4	2.46分
5	2.54時間
6	6.57日
7	13.58月
8	**69年**

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

パスワードの設定方法

- ユーザ自身が設定する
  - 前述した制限（文字種、長さ、含まれる文字列）に引っかからないように設定させる
- サーバ側で発行したものを使わせる
  - 複雑なパスワードを設定できるため、安全性は向上する。但し、ユーザがパスワードをどこかに書き留めてしまう可能性がある
- 完全にランダムなものより、ユーザが覚えていられる文字列、かつ、制限（文字種、長さ、含まれる文字列）に引っかからないようなパスワードが好ましい

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

パスワードのハッシュ化

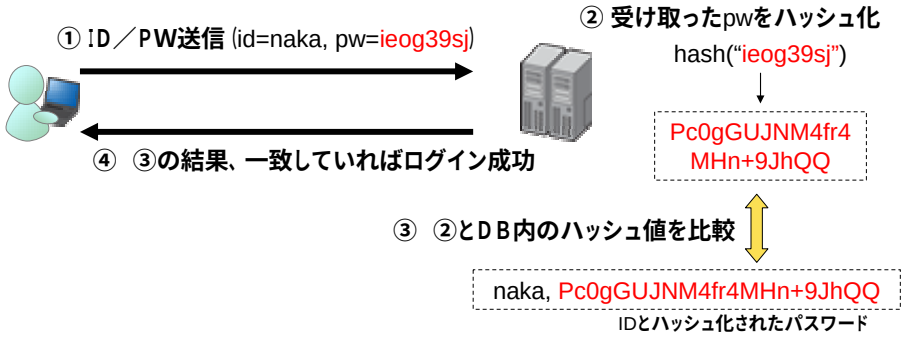
- サーバ側で記録しておくパスワードはハッシュ化しておく
- これにより、ＤＢの情報漏洩があっても、パスワードが漏洩することがない
- ハッシュ関数
  - データを受け取り、ある一定範囲の値（ハッシュ）を生成する関数
  - 似たデータから近いハッシュが生成されない
  - ハッシュが同じとなる元データの生成が困難
- ハッシュアルゴリズムについて
  - ＭＤｘやＳＨＡ－ｘといったアルゴリズムがある（例えば、ＭＤ５やＳＨＡ－１）
  - ＭＤ５は脆弱性が指摘されているため、使わないこと
  - ＳＨＡ－１も、新たな攻撃方法が見つかっており、当初よりは安全性が低下している
  - 今後は、ＳＨＡ－２に移行していくようである

（ハッシュについての詳細は、暗号に関する文献を参照のこと）

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

パスワードのハッシュ化（Cont'd）

- ハッシュ化したパスワードを使用して認証を行う方法



Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッション管理について

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッション管理の必要性

- セッション管理が必要な理由
  - 同じユーザの連続したアクセスを追跡するため
    - ✓ ログインしていなくても、同じユーザからの接続であることを維持する必要がある場合もある
      例： 買い物かごの中身の維持
  - 認証状態を維持するため
    - ✓ ID／PWを送信するのは、ログインのときだけであり、以降は何らかの方法でその状態を維持する必要がある

- HTTPについて
  - Webで使用されるプロトコルのこと
  - 基本的に、1リクエストに対して1レスポンスを返して終了（※）
  - そのためセッションを維持するためには、別の仕組みが必要
  - HTTPSとは、SSL上でHTTPを使用すること

（※）通信効率を上げるために、通常は1回の送受信でコネクションが切断されることはない （Keep-Alive）

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

HTTPのリクエストとレスポンス （参考）

```
GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-
  flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: ja
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.1; .NET
  CLR 1.1.4322)
Host: www.example.com
Connection: Keep-Alive
```

```
HTTP/1.0 200 OK
Date: Wed, 29 Nov 2006 08:21:21 GMT
Server: Apache
Content-Length: 17463
Connection: close
Content-Type: text/html; charset=Shift_JIS

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN“
（以下省略）
```

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッション管理の方法

- BASIC認証
- Digest認証（BASIC認証を改良したもの）
- CookieにセッションIDを入れる
- hiddenにセッションIDを入れる
- HTTPSのクライアント認証を使う

あまり推奨できない方法

- URLパラメータにセッションIDを入れる

ダメな方法

- ログイン後はパラメータに含まれるIDだけでユーザを識別する
- IDとパスワードをパラメータに含めて、アクセス毎に認証モジュールを経由させる

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

BASIC認証

- HTTPの仕様に含まれる認証方法であり、HTTPヘッダに認証情報を含めることで毎回認証を行うことにより、セッションを維持する
- 改良版であるDigest認証があるので、そちらを使うべき

```
GET /auth/index.html HTTP/1.1
Accept: */*
Accept-Language: ja
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.1; .NET
CLR 1.1.4322)
Host: www.example.com
Connection: Keep-Alive
Authorization: Basic Z3Vlc3Q6Zm9vYmFy
```

認証情報

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

BASIC認証 (Cont'd)

- 利点
  - Webサーバの設定により簡単に実現可能
  - 拡張モジュールを使用することで、DBやLDAPでユーザ情報を管理できる
  - HTTPSと組み合わせれば、安全性は高くなる
- 欠点
  - 認証時にダイアログボックスがポップアップするため、見た目が良くない
  - 「ログアウト」機能が、仕様に含まれていないため、ログアウトできない
  - HTTPヘッダ内に含める認証情報が、IDとPWをBase64エンコードしただけであるため、ネットワーク盗聴により、ID／PWが漏洩する (HTTPS未使用時)

```
$ perl -MMIME::Base64 -e 'print MIME::Base64::decode_base64("Z3Vlc3Q6Zm9vYmFy");'
guest::foobar  ←――― 簡単に逆変換可能
```

  - XST (Cross-Site Tracing) 攻撃により、認証情報が奪われる可能性がある

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Digest認証

- HTTP／1.1 で策定された、Challenge＆Response方式による認証

```
GET /auth2/index.html HTTP/1.1
Accept: */*
Accept-Language: ja
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.1; .NET
  CLR 1.1.4322)
Host: www.example.com
Connection: Keep-Alive
Pragma: no-cache
Authorization: Digest username="guest", realm="sec", qop="auth", algorithm="MD5",
  uri="/auth2/index.html",nonce="yFgKVbIjBAA=2a8b5d8f5e1d4f4f342a457d8b604640f5
  1347d5", nc=00000001, cnonce="bfae40a7d3fdeb264eb078563de43943",response=
  "44985ca43eb4c7001c9948efcfa23a1f"
```

認証情報

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Digest認証 (Cont'd)

- 利点
  - Webサーバの設定により簡単に実現可能
  - ハッシュ化した認証情報を使用するため、BASIC認証での欠点である、XSTやネットワーク盗聴により、認証情報が漏洩するという欠点が解消されている
- 欠点
  - 認証時にダイアログボックスがポップアップするため、見た目が良くない
  - 「ログアウト」機能が、仕様に含まれていないため、ログアウトできない

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

CookieにセッションIDを入れる

- 標準的な実装方法
- 利点
  - JavaやASP，PHPなどのミドルウェアの仕組みを使用することで簡単に実現可能
  - HTTPSを使い、Cookieにsecure属性を設定することで、安全性が高くなる
- 欠点
  - ブラウザの設定でCookieを受け付けないようにしているユーザには対応できない
  - XSSによりCookieが漏洩する可能性がある

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

hiddenにセッションIDを入れる

- 重要なアプリケーションでは、この方法を取られていることが多い
- 利点
  - Cookieと異なり、セッションIDが漏洩する可能性が低い
  - Cookieにセッション IDを入れる方法と組み合わせることで、より強固になる
- 欠点
  - 全てのリンクをPOSTメソッドでアクセスすることになるため、JavaScriptを使用してフォームをSUBMITすることになる

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

HTTPSクライアント認証

- サーバとSSLコネクションを確立する際に、クライアント証明書を要求し、正規に発行したクライアント証明書だけを受け付ける
- 利点
  - 正しいクライアント証明書を持つユーザだけしかアクセスできないため、なりすまし攻撃を受ける可能性が非常に低い
- 欠点
  - 証明書の発行はお金がかかるため、不特定多数のサービスには向かない
  - ユーザのブラウザにクライアント証明書のインストールを強要しなければならない
  - 正規のユーザでも、クライアント証明書が入ってない別のPCからはサービスを利用できなくなる

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッション管理方式のまとめ

- Digest認証
  - 特定の集団にのみアクセスを許可させたいサイト
    - ✓ 通常は、ID／PW情報をサーバ上のファイルで管理するため、多くのユーザを登録したい場合には向いてないので、グループに対して1つのID／PWを発行する方法となる
  - ログアウトが必要ないサービスを提供するサイト
- Cookieにセッション IDを入れる
  - 個人情報を扱うが、それほど重要性が高くないサイト
- hiddenにセッション IDを入れる
  - 機密性の高い情報を扱うサイト
- HTTPSクライアント認証を使う
  - B2Bなど、特定少数のユーザ向けのサービスで、重要な情報を扱うサイト

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッション IDについて

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッションIDの生成アルゴリズム

- JavaやASP、PHPなどのミドルウェアが生成するセッションIDを使用するのであれば、アルゴリズムについて特に気にする必要はない
  (最新バージョンを使用していれば、まず問題ない)
- 独自に生成したセッションIDを使用する方法は、なるべく避ける
- もしセッションIDを独自生成したいのであれば、以下の点に注意する
  - 必ずハッシュを使用する (解読できないようにするため)
  - 生成には可変情報を取り入れる (毎回同じ値が生成されないようにするため)
  - 十分長い値を生成する (総当り攻撃を不可能にするため)

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッションIDの発行タイミング

- 毎回同じセッションIDを使用する ×
  - 一度漏洩したら、必ずなりすましに成功されてしまうため、推奨できない
- ログイン毎に異なるセッションIDを発行する
  - 一般的な方法であり、推奨できる
  - セッションIDの生成に可変情報が使われていれば必ず異なるセッションIDが生成される
- 認証毎に異なるセッションIDを発行する
  - 登録情報参照や、重要な処理のコミット時には再認証を求めることが多いが、このときに、もうひとつのセッションIDを発行し、Cookieによるセッション管理と、hiddenによるセッション管理を併用すると、安全性が高くなる
- CookieとhiddenをIを組み合わせた方法で、hiddenの値をアクセス毎に変化させる

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

セッションIDの発行タイミング （Cont'd）

- 必ず **ログイン成功後** に発行しなければならない
- 最初にTOPページにアクセスしたタイミングで発行するのではダメ？

  → **攻撃者にセットされたセッションIDでログインしてしまうと、Session Fixation 成功となる**

- ログイン実行時に発行するのではダメ？

  → **ログイン失敗時に発行されるセッションIDを攻撃者にセットされてしまうと、Session Fixation 成功となる**

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

画面遷移の管理について

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

画面遷移管理の必要性

- 画面遷移管理とは
  - ①入力フォーム、②確認、③送信完了、という画面遷移において、直接「③送信完了」のリクエストを送信させないよう、必ず「②確認」の画面を経由させるようにする仕組み
- なぜ必要か
  - 処理のコミットを行うリクエストを直接送信させられてしまう脆弱性「ＣＳＲＦ」の対策となる
- 厳密に管理すると、ユーザビリティが低下する
  - Webアプリケーションの場合、ブラウザを複数開くことが一般的であり、1つのセッションにおいて、複数の画面状態が存在することになるため、厳密に管理しようとすると、不正な画面遷移が多発してしまう

利便性と安全性のバランスを、サイトの重要度を考慮して決定する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

画面遷移の管理方法

- Refererを使う
  - 以下の理由のため推奨できない
    - ✓ Refererを送信しないブラウザもある → サービスが提供できない
- サーバ側でセッションIDと画面状態を管理する
  - 利点
    - ✓ 正確な管理が可能
  - 欠点
    - ✓ 複数のブラウザを開いた場合の正常遷移には対応できない
    - ✓ 正しい画面遷移を行っての攻撃は防げない

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

画面遷移の管理方法 （Cont'd）

- 必ず経由させたい画面のみで発行するパラメータを導入する
  - 確認画面を経由していることを保障するには、確認画面を経由しないと生成されないパラメータを追加し、処理の確定時に、このパラメータが有効かどうかチェックすればよい

確認画面

<input type="hidden" name="sec_key" value="ALG2K9Q0E">

この値をセッション変数にも格納しておく

  - 利点
    - ✓ 実装は簡単。但し、パラメータ生成方法には注意が必要
  - 欠点
    - ✓ 正確な画面遷移までは追跡できない

（この方法は、厳密に管理する目的の方法ではないので、欠点とまでは言えないかも）

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

入力値チェック

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

入力値チェック

- ユーザから渡されるデータ（HTTPリクエスト全体）は、不正なデータが含まれている可能性があるため、処理する前に必ずチェックしなければならない
  例： HTTPリクエストヘッダ内の User-Agent
- 入力値の文字種、フォーマット、長さが事前に想定可能な場合は、その情報を用いてチェックを行い、想定外の入力については、エラーとして処理すべき
- 特にフォーマットがない場合でも、制御文字や記号類については受け付ける必要がない場合が多いはず
- 正規表現を正しく理解すること
  - 数字だけを許可する場合

✕ `$str =~ /¥d+/;` **これだと、文字列の一部に数字が入っていればいいことになる**

⇨ ○ `$str =~ /^¥d+$/;`

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Webサイトの企画から運用までの各フェーズにおける対策

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Webサイトの開発から運用までの各フェーズにおける対策

- 企画
- 発注
- 設計
- 開発
- 検収
- 運用

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

企画時のポイント

- Webサイトの分類の明確化
  - B to B （企業間取引システム）
  - B to C （企業対一般消費者取引システム）
  - C to C （一般消費者間取引システム）
  - 社内向けシステム
- サービス内容の明確化
  - 問い合わせ
  - オークション
  - ショッピング ・・・ etc
- 対象ユーザの明確化
  - 不特定多数
  - 特定多数
  - 特定限定

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

企画時のポイント （Cont'd）

- サイトで扱うデータ（資産）の明確化
  - 会員情報（氏名、住所、電話番号）
  - 決済情報
  - 問い合わせ情報
  - 有料情報（音楽、映像、画像）

- どのような脅威が考えられるか？
  - 不正にサービスを利用される
  - 会員情報が漏洩する
  - 決済情報が改ざんされる
  - 有料情報が盗まれる　など

- 資産と脅威から、サイトの重要度を決定する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

発注時のポイント

- サイトの重要度に応じて、業者に対する条件を設定する
  - 開発経験（開発サイト数）
  - セキュリティに関する知識を持っている人の割合

- 脆弱性が作りこまれないように、要件（非機能要件）を設定する
  - 以下の対策を施すこと
    - ✓ 不正な入力に対する入力値チェックおよび無害化
    - ✓ ＳＱＬインジェクション
    - ✓ ＯＳコマンドインジェクション
    - ✓ 認証回避
    - ✓ セッション乗っ取り

      など、脆弱性に対する対策
    - ✓ 機密情報の漏洩防止
    - ✓ なりすましによる不正アクセスの防止

      など、被害を防ぐ対策

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

設計時のポイント

- サイトの重要度によって、アプリケーションの仕様も変わってくる
  - ユーザ登録の方法
    - ✓ オンラインで登録し、即登録完了
    - ✓ オンラインで申込書申請、申込書を郵送
    - ✓ 窓口で登録
  - 認証方法
    - ✓ 「認証について」を参照。Webサイトの分類（B2B，B2Cなど）によって変わる
  - セッション管理方法
    - ✓ 「セッション管理について」を参照。Webサイトの重要度によって変わる

- 設計書のレビューには、セキュリティに詳しい人間を必ず含める
  - 設計の根本に脆弱性があると作り直しになってしまうため、大きな問題点は、この時点で指摘、修正する必要がある

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

設計時のポイント　～安全性を保つ設計方針

- 外部からの入力は全てチェック
  - 入力形式が決まっているなら、その形式であるかどうかの判断を行う

- 不要な情報をユーザ側から受け取らない
  - ログインしているユーザのIDは、セッション情報が取得できる
  - 商品の価格は、商品IDだけ提示してもらえればサーバ側の商品テーブルから取得できる
  - 選択画面では、選択肢の番号だけ提示してもらえればよい

- 必要最小限の権限しか与えない
  - OS、ファイルシステム、DB 、アプリケーションなどにおける権限は最小限に設定する

- フェイルセーフ
  - システムに問題が発生した場合は、誤動作で想定外の動作をしないよう、サービスを停止するといった動作をさせる
  - 想定外の動作は異常状態と判断する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

設計時のポイント　～安全性を保つ設計方針（Cont'd）

- 多重防御
  - ファイアウォールで不正な接続は拒絶する
  - OS、Webサーバなどのバージョンは全て脆弱性対策済み（最新版or最新パッチ）
  - 会員向けサービスでは、全ての処理で認証済みであることを確認する
  - 不正な入力は拒絶する
  - アプリケーションからアクセスできるファイルシステムは一部に制限する
  - アプリケーションをOS上で実行するユーザの権限は最小限に制限する
  - アプリケーションからアクセス可能なDBの範囲は一部のテーブル、コマンドに限定する
  - ネットワーク上で、Webサーバからアクセス可能なホストを制限する

- 既にあるものは新規に開発せず、それを使用する
  - セッション管理の仕組み（Java、ASP、PHPなど）
  - 暗号アルゴリズム

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

開発時のポイント

- 必ずコーディング規則を作成し、遵守させる
  - ソースコードレビューの負荷を軽くする
  - コードを見るだけで、間違いが分かるようにする
    - ✓ HTMLエスケープに関するルール
      - 必ず出力時にエスケープする
      - 未エスケープの文字列の変数名の接頭後を「xss_」とする、など
    - ✓ SQL文生成時のルール
      - 必ず、バインド変数を使う。それ以外の記述方法は許可しない、など

- 認証、セッション管理、入力値チェックなど、重要なコンポーネントは、少数精鋭のチームが開発し、全てのアプリケーションに同じルールで実装させる
  - これらのルールについても、コーディング規則に含めて徹底する

- 単体テストや結合テストなどでは、セキュリティに関するテストも実施する

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

検収時のポイント

- 必要とされる機能が動くことは当然。それよりも、不正操作時の対策が取られているかどうかが重要
- 発注側でも、セキュリティに関するテストを行い、挙動を確認する
  - "><script>alert()</script> と入力した場合、スクリプトが動作しないように、HTMLエスケープが行われているか
  - 「'」（シングルクォート）を入力した場合、おかしな挙動をしたり、DBのエラーが出力されたりしないかどうか
- 重要度の高いサイトの場合は、外部のセキュリティ検査ベンダーに検査を依頼し、その結果に応じて、検収の合否判定を行うという方法もある

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

運用時のポイント

- OSやWebサーバ、データベースなどのサーバアプリケーションのバージョン管理
- Webアプリケーションの機能追加や改修手順
  - 追加する機能の情報や影響範囲に関する情報を提出させる
  - 機能追加後に必ずセキュリティに関するテストを実施する
- ログの管理
  - どこまで記録できるのかを知っておく必要がある
    - ✓ Webサーバのログ ・・・ アクセス時刻、メソッド、ステータスコード、URLパラメータ
    - ✓ アプリケーションのログ（作り込んでいるならば） ・・・ ログイン失敗、不正入力
  - 保管
    - ✓ 一定期間保管することが望ましい

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

運用時のポイント （Cont'd）

- 不正アクセス発覚時の対応手順
  - バグによるシステム停止と同じように、不正アクセスされた場合の手順も明確にしておく
  - 可能であれば、フォレンジックについての知識も身に付けておくとよい
    - ✓ フォレンジック ・・・ 法的手段の証拠とするために、不正アクセスの痕跡を調査する手段
  - 不適切な手順の場合、復旧までのコストや期間の増加だけでなく、再び攻撃される場合もある
- 重要なサイトの場合、セキュリティ監視サービスを受ける方法もある

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.

Q

Copyright 2006 Mitsui Bussan Secure Directions, Inc. All Rights Reserved.