PHPによる Web アプリケーションとセキュリティ

概 要

Web アプリケーションで端末での入力データを用いて、サーバ側処理を実行する CGI や PHP 等のプログラムの作成を通し、World Wide Web の動作を理解する。さらに問題のあるプログラムの実例や改良を通して、セキュリティに対する意識を培う。

1 概要

WWW(World Wide Web) は、物理研究の資料などの効率良い情報交換のために、ティム・バーナーズ・リーにより開発され、HTML(Hyper Text Markup Language) と呼ばれる文字データと絵や音を含めたレイアウト情報を、HTTP(Hyper Text Transfer Protocol) と呼ばれる簡単な方法な通信が行われる。

端末では、ブラウザに URL という相手サーバとサーバ内のファイル位置情報を入力し、HTML 情報を読み出しレイアウト情報によって分かりやすく表示される。ブラウザでは、HTML の文中に埋め込まれた他の文書へのリンクを簡単に参照できるため、広く普及している。

```
http://www.ei.fukui-nct.ac.jp/ t-saitoh/index.html
```

意味	`http://`	スキーマ (情報交換のプロトコル名)
	`www.ei.fukui-nct.ac.jp`	接続相手のコンピュータ名
	`/~t-saitoh/index.html`	サーバ内のファイルの保存場所

表 1: URL の意味

WWW では、ユーザからの情報を用いて、サーバで処理を加えた情報を返す機能として CGI (Common Gateway Interface) という手法が普及している。しかし、きちんとしたセキュリティ対策をとらなければ、サーバから意図しない情報が読み取られる。本実験では、簡単な WWW アプリケーションの製作と不備のあるプログラムのセキュリティ問題について考える。

なお、この実験は本来セキュリティ面で問題のあるプログラムを通して行う。このため学内でのみ利用可能なサーバにて行い、実験が終ったらファイルをすぐに削除するので注意せよ。またここで指摘した手法を、実験サーバ以外で試すと**不正アクセス行為の禁止等に関する法律**[1] にて処罰をうける可能性がある。

[1] `http://www.ipa.go.jp/security/ciadr/law199908.html`

1.1 HTML

HTML は、文字データとタグと呼ばれるレイアウト情報から構成される。ページ作成の経験があれば、概要を理解している人が多いので、実験で利用する主要なタグを表 2 にて示す。これ以外については、http://www.zspc.com/htmlref/(Super HTML Reference) 等を参照。

タグ	意味
<HTML>〜</HTML>	HTML 文書の開始と終了
<HEAD>〜</HEAD>	ヘッダ部
<BODY>〜</BODY>	本文の開始と終了
<H1>〜</H1>…<H7>〜</H7>	見出し (普通フォントサイズも変わる)
<P>〜</P>	段落
 	強制改行
<UL><LI>〜<LI>〜</UL>	記号の箇条書 (UL), 項目 (LI)
<OL><LI>〜<LI>〜</OL>	番号の箇条書 (OL), 項目 (LI)
<A HREF="URL">〜</A>	他の文書へのリンク
<FORM METHOD="方式"	入力値を CGI に渡す。
ACTION="URL">〜</FORM>	方式は GET or POST
<INPUT TYPE="方式"	入力形式
NAME="変数名">	入力された値は変数名に対応づけ
TYPE=text	文字入力
TYPE=submit	送信ボタン

表 2: HTML タグの抜粋

1.2 CGI

CGI とは、端末ブラウザの入力した値に応じてサーバでプログラムを実行し、その結果に応じてページを表示する機能である。ページの<FORM>〜</FORM>によって入力された値はサーバに送られ、ACTION で指定したプログラムを起動する。この時、環境変数と標準入力を通じて値が渡され、CGI プログラムでは値を読み取り変数値に応じてプログラムを実行する。このプログラムの標準出力は、サーバプログラムを通じて端末ブラウザに送られる。

表 3 に、C 言語で CGI を記述した例を示す。この例では、POST メソッドのデータをまとめて表示しするだけであるが、実際の CGI では、様々な文字列処理が必要となる。このため、一般的な CGI では、Perl というスクリプト型言語を利用する事が多い。

1.3 PHP

一般的な Perl を用いた CGI プログラムは、繁雑な部分や特殊な Perl 文法を理解する必要があるため、以後の実験では PHP という言語を利用する。

PHP では、HTML 文法の中に PHP 文法によるプログラムを混在して記述でき、基本的な文法も C 言語に類似しているため、WWW アプリケーションを簡単に作成できる。

```
=====test.html=====
<HTML>
<HEAD><TITLE>CGI の実験</TITLE></HEAD>
<BODY>
<FORM METHOD="POST" ACTION="env.cgi">
<INPUT TYPE=TEXT   NAME="A">
<INPUT TYPE=SUBMIT VALUE="送信します">
</FORM></BODY></HTML>
=====env.c=====
#include <stdio.h>
#include <stdlib.h>
char buff[ 100 ] ;
void main()
{   /* POST メソッドで送られてきたデータ量 */
    /*   getenv:環境変数を読出す          */
    char* clen = getenv( "CONTENT_LENGTH" ) ;
    int   len  = atoi( clen ) ;
    /* 返すデータが HTML 形式であることを指定 */
    printf( "Content-Type: text/html\n\n" ) ;
    /* 動作確認のためにデータ量情報を表示 */
    printf( "%d,%s<BR>\n" , len , clen ) ;
    /* POST メソッドで送られたデータを読み込む */
    fgets( buff , len + 1 , stdin ) ; /* セキュリティ問題あり */
    printf( "%s\n" , buff ) ;      /* データをそのまま表示 */
}
=====プログラム作成=====
$ cd public_html
$ gcc -o env.cgi env.c
```

表 3: C 言語による CGI プログラムの例

2 CGI,PHP による WWW アプリケーションの実験

以下に PHP による短い WWW アプリケーションを 2 つ示す。これらのプログラムは、問題点を検討するために、セキュリティ的に不備のあるプログラムとなっている。

データベースプログラム 表 5 は、データベースの例でユーザの入力した電話番号と名前を、1 件 1 ファイルで保存する。[2]

外部プログラムの利用 表 6 は、カレンダーを表示する外部プログラムを、system() 文により実行し、その結果によりカレンダーをページに表示する。

[2] 問題点：保存したファイル名を URL で指定すると内容が読める。

```
=====sample.php=====
<HTML>
<HEAD><TITLE></TITLE></HEAD>
<BODY>
<FORM METHOD=POST ACTION="sample.php">
<INPUT TYPE=text NAME="A">          // 変数 $_REQUEST[ "A" ]
＋
<INPUT TYPE=text NAME="B">          // 変数 $_REQUEST[ "B" ]
＝
<?php                               // <?php ～ ?>の間は PHP の文法
    if ( $_REQUEST[ "A" ] != "" && $_REQUEST[ "B" ] != "" ) {
                                    // 値が代入されていれば
       print $_REQUEST[ "A" ] + $_REQUEST[ "B" ] ;
                                    //    合計値を表示
    } else {                        // そうでなければ
       print "<INPUT TYPE=submit>" ; //  送信ボタンを表示
    }
?>
</FORM></BODY></HTML>
```

表 4: PHP プログラムの例

3 実験手順

1. 上記のプログラムを入力 (もしくはサンプルをコピー) し、プログラムの動作を確認せよ。

2. 実験時に指示する手法[3] で、セキュリティ問題を実際に発生させ、どの様な問題があるか、手法と問題点を記録せよ。

[3] WWW 公開した実験資料の悪用を防ぐため

4 考察

1. 前実験で示した問題点について、その対処方法について調査し、一般的な手法としてどの様な方法が取られるか説明せよ。

2. 前に示した対策を元に、実験で示したプログラムに対して問題が発生しないように対策を施せ。対策後の結果では、対策概要・プログラムリストを添付せよ。

3. 最も簡単なサーバでのパスワードの管理では、(a) 認証のために誰でも読める必要があり、(b) 一方でパスワードを変更するため、本人だけが書き込みができる必要がある。(c) さらに単純に誰もがパスワードを読める状態は、暗号化パスワードを読み出してパスワードの総当たりチェックを実行される危険性がある。

   こういった問題に対応するために一般的な OS では、どの様な対策がとられているのか、参考文献を元に説明せよ。

補足資料

Windows 環境で実験をする場合：

1. ブラウザを起動し、http://www.ei.fukui-nct.ac.jp/~t-saitoh/exp/webphp/ をアクセスすれば、資料と同じファイルがダウンロードできる。

2. Windows のファイル共有にて、

   \\10.120.21.210\ゲストアカウント名\public_html

   にアクセスし、この中に自分の実験用のディレクトリを作成し、この中にダウンロードしたファイルを保存せよ。

   例：作成した自分用ディレクトリ名が test の場合、そのディレクトリ内の index.html を参照するなら、URL は

   http://10.120.21.210/~ゲストアカウント名/test/index.html

   となる。

3. データベース実験db.php では、実験用ディレクトリへの他人[4] からの書き込み権限が必要である。このため

   『プロパティ→セキュリティ→Everyone →フルコントロール』

   をチェックし、書き込み権限等を設定する。同様の設定を unix 環境で実行するなら、

   『スタート→プログラム→アクセサリ→コマンドプロンプト』
   C:> telnet 10.120.21.210
   tsaitoh login: ゲストアカウント名
   Password: パスワード
   $ cd public_html/
   $ chmod 777 自分用のディレクトリ名　←この設定もセキュリティ的に NG

[4] 正確には WWW サーバアカウント (www-data)

```
=====db.php=====
<HTML>
<HEAD><TITLE>データベース実験</TITLE></HEAD>
<BODY>
<H1>データベース実験</H1><HR>
<H2>データ登録</H2>
<FORM METHOD="POST" ACTION="db.php">
  名前<INPUT TYPE=text NAME=user><BR>
  電話番号<INPUT TYPE=text NAME=phone>
  <INPUT TYPE=submit VALUE="登録">
</FORM>
<?php
      $user = $_REQUEST[ "user" ] ;
      $phone = $_REQUEST[ "phone" ] ;
      if ( $user != "" && $phone != "" // 登録データが与えられたときだけ
           && ($fp = fopen( $user , "w" )) != FALSE ) {
          fputs( $fp , $phone ) ;       // ファイルに電話番号を１行で出力
          fclose( $fp ) ;               // ファイルを閉じる
      }
?>
<H2>データ検索</H2>
<FORM METHOD="POST" ACTION="db.php">
  名前<INPUT TYPE=text NAME=search><BR>
  <INPUT TYPE=submit VALUE="名前で検索">
</FORM>
<?php
      $search = $_REQUEST[ "search" ] ;
      if ( $search != "" ) {           // 検索データが与えられたときだけ
         if ( file_exists( $search )      // 名前のファイルが存在するか？
           && ($fp = fopen( $search , "r" )) != FALSE ) {
            $answer = fgets( $fp , 999 ) ;// １行データを読み込む
            fclose( $fp ) ;              // ファイルを閉じる
            print "$search さんの電話番号は $answer です。" ;
         } else {
            print "$search さんは見つかりません。" ;
         }
      }
?>
</BODY></HTML>
```

表 5: データベースプログラムの例

```
=====prog.php=====
<HTML>
<HEAD><TITLE>PHP プログラムの応用例</TITLE></HEAD>
<BODY>
<FORM METHOD=POST ACTION="prog.php">
表示したい月,年を入力して下さい。
<INPUT TYPE=text NAME="A"> // 変数 $A
</FORM>
<H1>
<?php
      $A = $_REQUEST[ "A" ] ;
      print $A ;            // 入力値を表示(セキュリティ問題あり)
?>
</H1>
<PRE>
<?php
      system( "/usr/bin/cal $A" ) ;
                            // cal はカレンダー表示プログラム
                            // system はプログラムを起動する関数
                            // プログラム起動のセキュリティ問題あり
?>
</PRE></BODY></HTML>
```

表 6: 外部プログラム呼び出しの例