[1-5.] hidden は危険(セッション変数を利用しよう)

Webページ間のデータの受渡しにフォームのhiddenフィールドは便利である。しかし，いったんWWWブラウザに渡った情報は外部に漏洩するのはもちろん，改竄のおそれが大きい。重要な情報はhiddenフィールドに入れてはならない。

● ● ●

hidden フィールドの利用

hiddenフィールドは画面には表示されず，複数のWebページ間でデータの受渡しを行う際に利用されるHTMLフォーム項目である。(hidden［ヒドゥン］は「隠された」の意)

画面１，**画面２**を見てほしい。これは，最初の「ログインページ」への入力でユーザの種別が識別され，その種別に応じて次の「記事メニューページ」から呼び出せる記事が制限されるという簡単な Web アプリケーションの例である。ユーザはログインページでパスワードを入力する。このパスワードが会員用パスワードであるか否かにより「会員」と「ゲスト」の２つの種別に分類される。会員は記事メニューページの全ての記事を見ることができ，ゲストが会員専用の記事を見ようとすると「会員登録ページ」が現れる。

ログインページのHTMLを**リスト１**に，そこからパスワードを受け取って記事メニューページを表示するプログラムを**リスト２**に，記事メニューページでの選択結果を受け取りユーザの種別に応じて表示を制限するページのプログラムを**リスト３**に示す。ここでは Web アプリケーションの記述に IIS の Active Server Pages (ASP) を使用している。

リスト２の４～８行目でパスワードからユーザ種別が決定され，その種別を表すコード１または０が 14 行目で記事メニューページの HTML に埋め込まれる。埋め込まれる先は MEMBERFLG という名の hidden フィールドである。**リスト３**のプログラムは，記事メニューページで行われた選択の結果を NEWSID 項目の値として受け取る。選ばれたものが会員専用の記事だった場合，hidden フィールド MEMBERFLG のユーザ種別が参照され，ゲストに対しては「会員登録ページ」が表示される。

画面１　ログインページ

画面２　記事メニューページ

リスト１ 「ログインページ」login.html

```
 1 <HTML>
 2   <TITLE>ログイン</TITLE>
 3   <BODY>
 4     会員の方は，パスワードを入力しログインしてください。<BR>
 5     <FORM METHOD="POST" ACTION="menu.asp">
 6       <INPUT TYPE="text" NAME="PASSWORD"><BR>
 7       <INPUT TYPE="SUBMIT" VALUE="ログイン">
 8       <INPUT TYPE="RESET" VALUE="クリア">
 9     </FORM>
10     会員以外の方は，<A HREF="menu.asp">こちら</A>からどうぞ。<BR>
11     会員登録は，<A HREF="addmember.html">こちら</A>からどうぞ。
12   </BODY>
13 </HTML>
```

リスト２ 「記事メニューページ」menu.asp

```
 1 <%
 2   Dim strPass, bolMember
 3   strPass = Request.Form("PASSWORD")
 4   If strPass = "pass" Then
 5     bolMember = 1
 6   Else
 7     bolMember = 0
 8   End If
 9 %>
10 <HTML>
11   <TITLE>記事メニュー</TITLE>
12   <BODY>
13     <FORM METHOD="POST" ACTION="news.asp">
14       <INPUT TYPE="HIDDEN" NAME="MEMBERFLG" VALUE=<%= bolMember %>>
15       <INPUT TYPE="RADIO" NAME="NEWSID" VALUE=0 CHECKED>最新情報<BR>
16       <INPUT TYPE="RADIO" NAME="NEWSID" VALUE=1>丸得！情報（会員のみ）<BR>
17       <INPUT TYPE="SUBMIT" VALUE="記事を読む">
18     </FORM>
19   </BODY>
20 </HTML>
```

hidden フィールドへの値の埋め込み

リスト３ 「記事表示ページ」news.asp

```
 1 <%
 2   If Request.Form("NEWSID") = 0 Then
 3     Server.Transfer "everyone.html"     '「最新情報ページ」へ
 4   Else
 5     If Request.Form("MEMBERFLG") = 0 Then
 6       Server.Transfer "addmember.asp"   '「会員登録ページ」へ
 7     Else
 8       Server.Transfer "memberonly.asp"  '「丸得！情報（会員のみ）ページ」へ
 9     End If
10   End If
11 %>
```

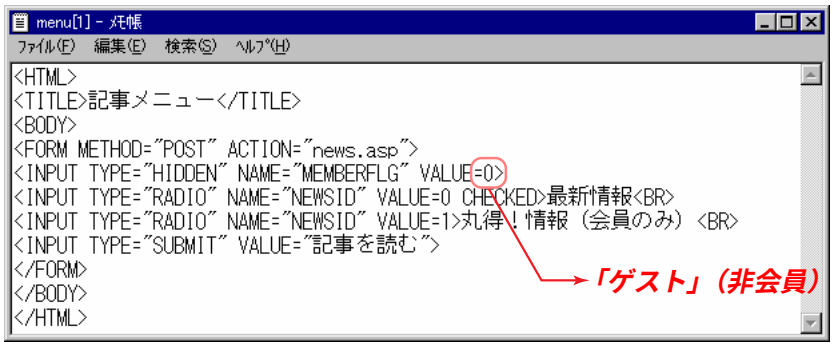
hidden フィールドの危険性

上記のようにフォームのhiddenフィールドを利用すると非常に簡単にWebページ間でデータの受渡しを行うことができる。しかし，このようなWebアプリケーションは大きな危険性を孕んでいる。

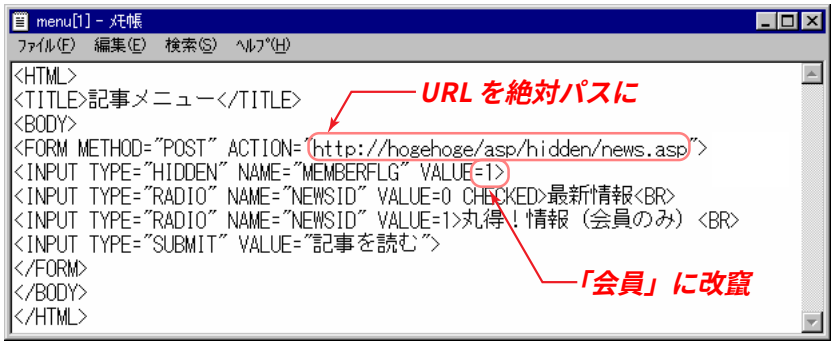
hiddenフィールドで指定したデータはWWWブラウザには表示されないため安全であるかのように錯覚しがちであるが，実際には他のデータと同様にブラウザに送られている。WWWブラウザに送られたデータは，ユーザによって容易に参照・改竄される可能性があることを認識する必要がある。

画面３，**画面４**を見てほしい。これらは，いかに簡単にフォームのhiddenフィールドで指定されたデータを参照・改竄できるかを示す例である。**画面３**は，ログインページでゲストユーザとしてログインし，表示された記事メニューページのHTMLソースを表示した結果である。5行目のhiddenフィールドにMEMBERFLGとして0が設定されていることが確認できる。また，このHTMLソースを**画面４**のように編集した後ブラウザで開き，「丸得！情報（会員のみ）」を選択して「記事を読む」ボタンをクリックすると，会員用パスワードを知らなくとも会員専用ページを参照することができるのである。

画面３　「記事メニューページ」のHTMLソースをエディタで開いたところ



画面４　「記事メニューページ」のHTMLソースを編集



セッション変数

Webページ間のデータ受け渡しにフォームのhiddenフィールドを利用する場合の問題は，次のページに受け渡されるデータが一度WWWブラウザを経由することにある。WWWブラウザに預けられたデータは，常にユーザによって参照・改竄される危険性が伴う。では，どのようにしてWebページ間でデータを受け渡せばよいのだろうか？　その答えはWebアプリケーション実行環境が提供するセッションメカニズムを利用することである。

多くのWebアプリケーション実行環境は，複数Webページ間の連続性を維持するためのセッションメカニズムを備えている。そこではそれぞれのセッションごとに専用の記憶領域が提供されていて，セッションが続いている限りデータを保持できるようになっている。このような記憶領域のことを「セッション変数」と呼ぶ。

セッション変数に設定されたデータはサーバ側で保持され，WWWブラウザに流出することがないため，安全にWebページ間のデータ受け渡しを行うことができる。

Active Server Pages（ASP）のセッション変数

ASPにはSessionという組み込みオブジェクトが用意されていて，次のような形でセッション変数が使用できる（プログラミング言語がVBScriptの場合）：

```
Session("名前") = 式                          ' セッション変数への値の格納
Set Session("名前") = オブジェクト            ' セッション変数へのオブジェクトの格納

変数 = Session("名前")                        ' セッション変数の値の参照
Set オブジェクト変数 = Session("名前")        ' セッション変数のオブジェクトの参照
```

Java Servletのセッション変数

Java ServletにはHttpSessionというクラス（厳密にはインタフェース）が用意されていて，次のような形でセッション変数が使用できる（Servlet 2.2，2.3の場合）：

```
HttpSession session = request.getSession(true); // Servletへのリクエストrequestからの
                                                // セッションインタフェースの取得
session.setAttribute("名前", 式);               // セッション変数への オブジェクト の格納
型 変数 = (型)session.getAttribute("名前");     // セッション変数の オブジェクト の参照
```

セッション変数による対策

問題のあった**リスト２**，**リスト３**のプログラムをセッション変数を利用した形に修正したコードをそれぞれ**リスト４**，**リスト５**に示す。まず，**リスト４**の9行目でユーザ種別を判別するための値をセッション変数Session(″MEMBERFLG″)にセットしている。**リスト５**の5行目では，**リスト４**で設定されたセッション変数から値を取得しユーザ種別の判定を行っている。

リスト４，**リスト５**の対策反映後，再度ゲストでログインして表示された記事メニューページのソースを表示してみると，フォーム項目MEMBERFLGが参照できないことが分かる（**画面５**）。

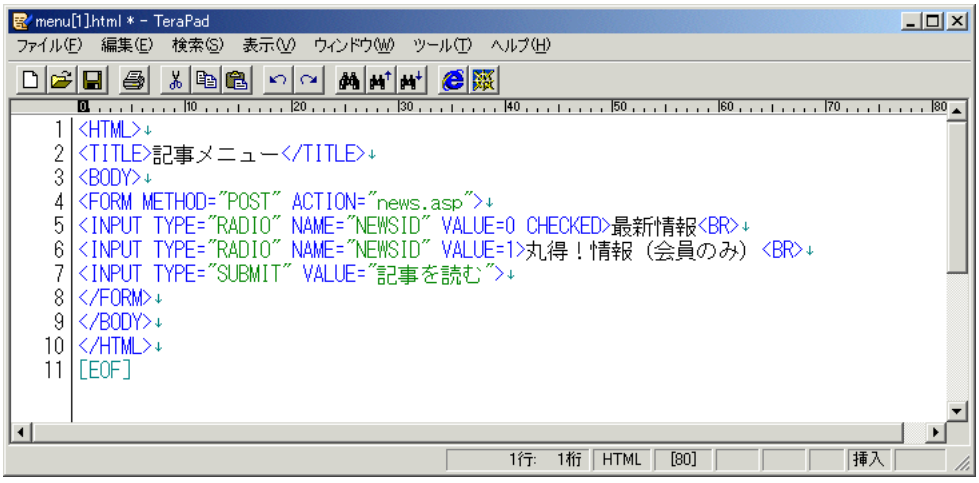
リスト４　「記事メニューページ」menu.asp の修正版

```
 1 <%
 2   Dim strPass, bolMember
 3   strPass = Request.Form("PASSWORD")
 4   If strPass = "pass" Then
 5     bolMember = 1
 6   Else
 7     bolMember = 0
 8   End If
 9   Session("MEMBERFLG") = bolMember          ← 追加
10 %>
11 <HTML>
12   <TITLE>記事メニュー</TITLE>
13   <BODY>
14     <FORM METHOD="POST" ACTION="news.asp">
15       <INPUT TYPE="RADIO" NAME="NEWSID" VALUE=0 CHECKED>最新情報<BR>
16       <INPUT TYPE="RADIO" NAME="NEWSID" VALUE=1>丸得！情報（会員のみ）<BR>
17       <INPUT TYPE="SUBMIT" VALUE="記事を読む">
18     </FORM>
19   </BODY>
20 </HTML>
```

リスト５　「記事表示ページ」news.asp の修正版

```
 1 <%
 2   If Request.Form("NEWSID") = 0 Then
 3     Server.Transfer "everyone.html"     '「最新情報ページ」へ
 4   Else
 5     If Session("MEMBERFLG") = 0 Then                    ← 変更
 6       Server.Transfer "addmember.asp"   '「会員登録ページ」へ
 7     Else
 8       Server.Transfer "memberonly.asp"  '「丸得！情報（会員のみ）ページ」へ
 9     End If
10   End If
11 %>
```

画面５　リスト４，リスト５による対策後の「記事メニューページ」の HTML ソース



セッション変数とクッキー

セッション変数を実現するセッションメカニズムも，実はWebページ間の連続性を追跡するための小さなデータをWWWブラウザに預けることで実現されている。この小さなデータは「セッションID」と呼ばれ，多くの場合HTTPプロトコルが備えているクッキー（Cookie）という仕組みを使ってWWWサーバとWWWブラウザの間でやりとりされる。クッキーはWWWブラウザのバグの悪用や「クロスサイトスクリプティング」といった不正な手法で他人に盗まれることがあり，Webアプリケーションの設計にあたってはその点も考慮しておく必要がある。

まとめ

「WWWブラウザの画面に表示されない＝安全」と言う図式は成り立たない。WWWブラウザに送られた全てのHTMLタグをユーザが参照・改竄することは容易である。干渉されては困る重要なデータをWebページ間で受け渡す場合，フォームのhiddenフィールドではなくセッション変数を利用すべきである。その際，クッキーの安全対策にも考慮が必要だ。

参考文献

『HTML クイックリファレンス　<INPUT> タグ』
http://www.htmq.com/html/input.shtml

『HTML 4.0 日本語リファレンス　<INPUT> タグ』
http://www.htmlhelp.com/ja/reference/html40/forms/input.html

『msdn online Web Workshop - Web ファームにおけるセッション状態の管理 -』
http://www.microsoft.com/japan/developer/workshop/server/feature/webfarm3.asp

『ZD Net JAPAN Chapter 8... プレゼンテーション層の構築 ～ IIS と ASP による Web 用ユーザーインタフェースの構築～』，http://www.zdnet.co.jp/help/howto/win/win2000/0007complus_vb/chap08/01.html

『Javadoc: HttpSession』
http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/http/HttpSession.html

『Javadoc: HttpServletRequest getSession』
http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/http/HttpServletRequest.html#getSession(boolean)