MOAUB

Abysssec Research

1) Advisory information

Title	: IfNuke Multiple Remote Vulnerabilities
Affected	: IfNuke 4.0.0
Discovery	: www.abysssec.com
Vendor	: http://www.ifsoft.net/default.aspx
Impact	: Ciritical
Contact	: shahin [at] abysssec.com , info [at] abysssec.com
Twitter	: @abysssec

2) Vulnerability Information

Class

1- **Upload arbitrary file**

2- **Persistent XSS**

Impact

An attacker may leverage this issue to have arbitrary script code execute in the browser of an unsuspecting user. This may help the attacker steal cookie-based authentication credentials and launch other attacks.

Also it's possible to upload a malicious script and run arbitrary command on target server.

Remotely Exploitable

Yes

Locally Exploitable

No

3) Vulnerabilities detail

1- Arbitrary Upload file:

Using this vulnerability you can upload any file with these two ways:

```
1- http://Example.com/Modules/PreDefinition/PhotoUpload.aspx?AlbumId=1   (the value of AlbumId is necessary)
```

Your files will be in this path:

```
http://Example.com/Users/Albums/
```

With this format (for example):

```
Shell.aspx ---> img_634150553723437500.aspx
```

That 634150553723437500 value is DateTime.Now.Ticks.ToString() and will be built in this file :

```
Ln 102 :
http://Example.com/Modules/PreDefinition/PhotoUpload.ascx.cs
fileName = "img_" + DateTime.Now.Ticks.ToString() + "." + GetFileExt(userPostedFile.FileName);
```

It's possible to do same thing here:

```
2- http://Example.com/modules/PreDefinition/VideoUpload.aspx
```

And the same vulnerable code is located here:

```
Ln 39 :
http://Example.com/Modules/PreDefinition/VideoUpload.ascx.cs
string createdTime = DateTime.Now.ToString("yyyyMMddHHmmssffff");
string newFileNameWithoutExtension = Path.GetFileNameWithoutExtension(fileName) + "_" +
createdTime;
string uploadFilePath = Server.MapPath(VideoHelper.GetVideoUploadDirectory(CurrentUser.Name) +
newFileNameWithoutExtension + Path.GetExtension(fileName));
```

2- Persistent XSS Vulnerabilities:

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

```
1- Module name    : Article
  Fields        : Title , Description
  Valnerable Code: ...\Modules\PreDefinition\Article.ascx.cs
  ln 106:
        if (S_Title.Text.Trim() != string.Empty)
        {
          parameters.Add("@Title", S_Title.Text.Trim());
          parameters.Add("@Description", S_Title.Text.Trim());
          parameters.Add("@Tags", S_Title.Text.Trim());
        }
```

```
2- Module name    : ArticleCategory
  Field         : Name
```

```
   Valnerable Code: ...\Modules\PreDefinition\ArticleCategory.ascx.cs
   ln 96:
        entity.Name =
((TextBox)lstSearch.Rows[lstSearch.EditIndex].FindControl("txtCategoryName_E")).Text.Trim();
```

```
 3- Module name    : HtmlText
   Field         : Text
   Valnerable Code: ...\Modules\PreDefinition\HtmlText.ascx.cs
   ln 66:
        entity.Content = txtContent.Value.Trim().Replace("//",string.Empty);
```

```
 4- Module name    : LeaveMessage
   Fields        : NickName , Content
   Valnerable Code: ...\Modules\PreDefinition\LeaveMessage.ascx.cs
   ln 55:
        entity.NickName = txtNickName.Text.Trim();
        entity.Content = txtContent.Text.Trim();
```

```
 5- Module name    : Link
   Field         : Title
   Valnerable Code: ...\Modules\PreDefinition\Link.ascx.cs
   ln 83:
        entity.Title =
((TextBox)lstSearch.Rows[lstSearch.EditIndex].FindControl("txtTitle_E")).Text.Trim();
```

```
 6- Module name    : Photo
   Field         : Title
   Valnerable Code: ...\Modules\PreDefinition\Photo.ascx.cs
   ln 280:
        entity.Title = txtTitle_E.Text.Trim();
```