FCKeditor .NET File Upload Vulnerability
분석 보고서

2010. 04. 19

넷시큐어테크놀러지(주)
NetSecure Technology

Copyrights

이 보고서에서는 기술적으로 부정확한 설명, 인쇄상의 잘못 등이 있을 가능성이 있습니다.
넷시큐어테크놀러지㈜에서는 예고 없이 이 보고서의 내용 또는 프로그램을 수정 또는 변경할 수 있음을 양지 하시기 바랍니다.

목 차

1. 개요

1.1 목적

본 문서의 목적은 FCKeditor .NET(version 2.6.3)에 존재하는 File Upload 취약점을 분석한 후, 이를 이용한 공격에 대응하는 방법을 제시하는 것임

1.2 분석 내용

❍ FCKeditor .NET 2.6.3 버전에서 확장자 우회를 통해 동일한 파일명으로 업로드 할 경우 파일명에 대한 필터링을 제대로 하지 못해 웹 스크립트로 파일 업로드가 가능함

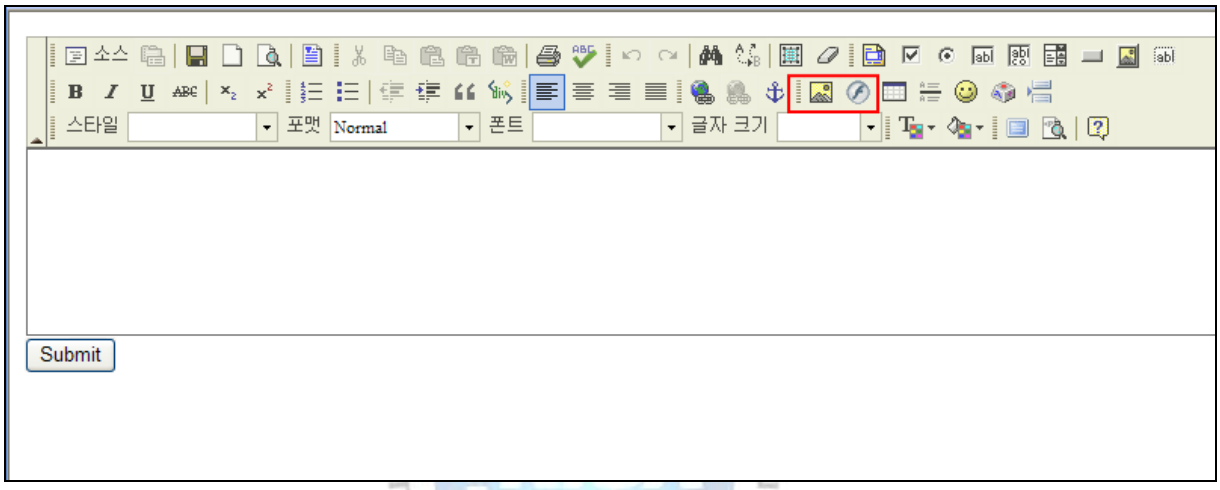
구분		상세 내용
FCKeditor .NET File Upload Vulnerability	설명	서버에 존재하는 동일 파일명으로 파일 업로드 시 업로드 인증 로직 오류로 인하여 웹 스크립트로 실행될 수 있는 파일이 업로드 됨
	버전	FCKeditor .NET 2.6.3
	동작 환경	Windows Server 2003 IIS 6.0 (.NET)
	URL	http://[host]/[path]/fckeditor/editor/filemanager/browser/default/browser.html?Type=Image&Connector=http://[host]/[path]/fckeditor/editor/filemanager/connectors/aspx/connector.aspx
	대응 방안	- aspx connector 비활성화 - 업로드 인증 로직 소스 코드 수정

2. 상세 분석

2.1 FCKeditor .NET File Upload Vulnerability

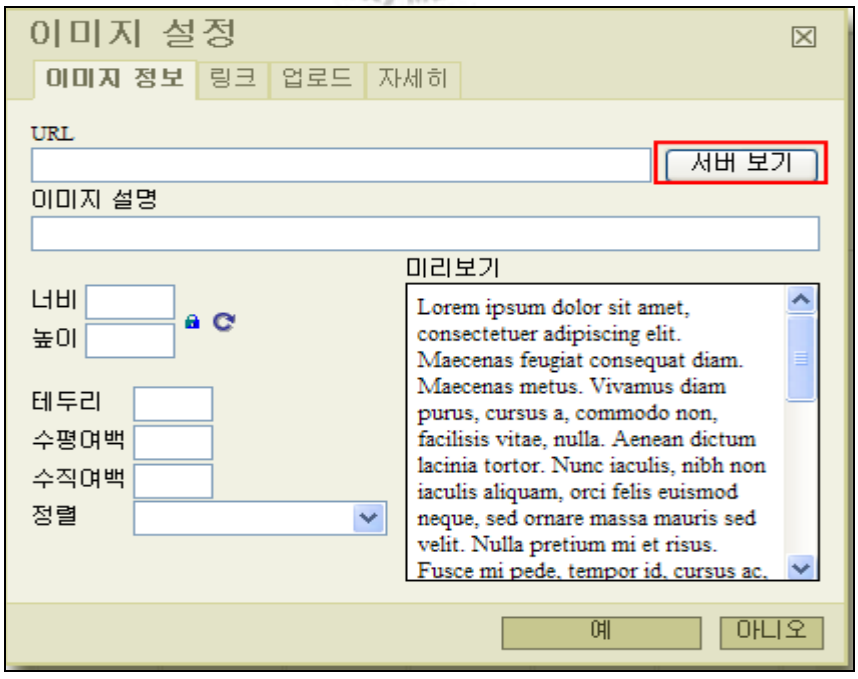
FCKeditor 업로드 기능에 존재하는 '서버 보기' 기능을 통해 파일 업로드가 가능함

■ FCKeditor의 게시물 등록 폼에 파일(file, image, flash, Media) 업로드 기능이 존재하며 그 중 image 업로드 기능을 이용하여 파일 업로드를 시도함



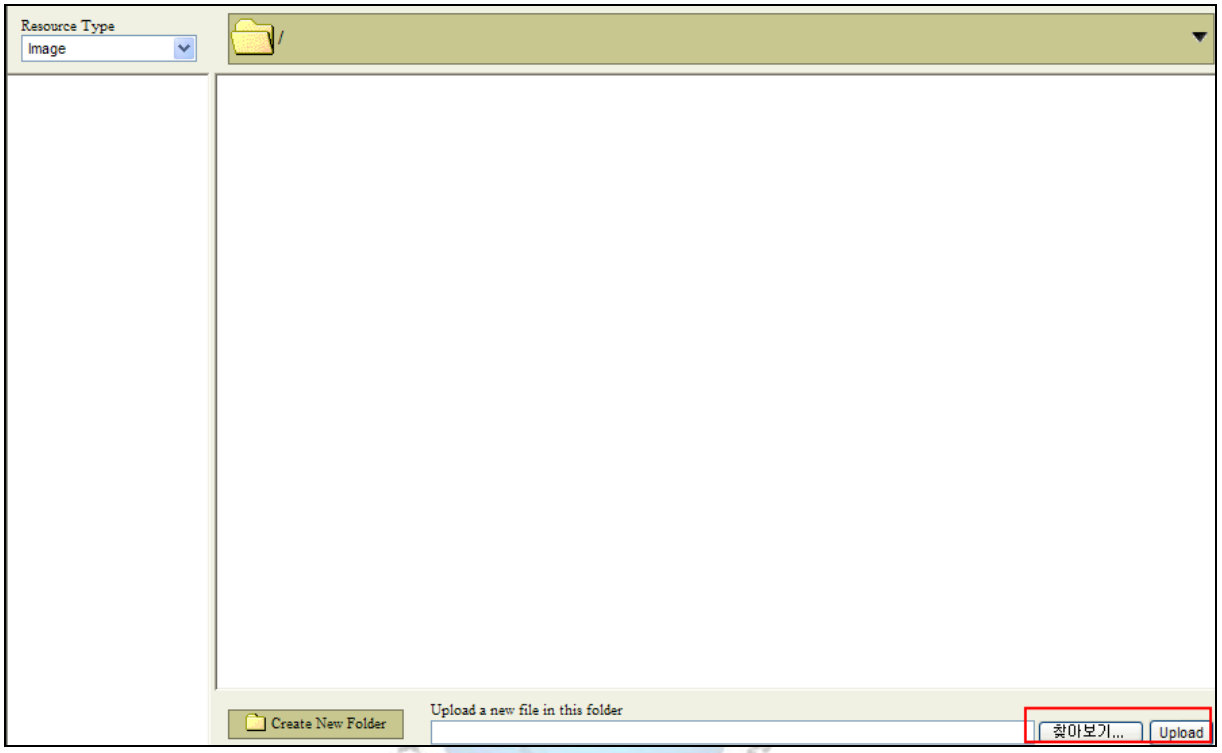
<그림> 이미지 파일 업로드 탭

■ 아래와 같이 이미지 업로드 설정 창에서 '서버 보기' 버튼이 보여지며 해당 버튼을 통해 'File Browser'에 접근이 가능함



<그림> 이미지 설정 창

■ '서버 보기'를 통해 FCKeditor .NET 'File Browser'에 접근이 가능하며 Upload 기능을 이용하여 파일 업로드가 가능함

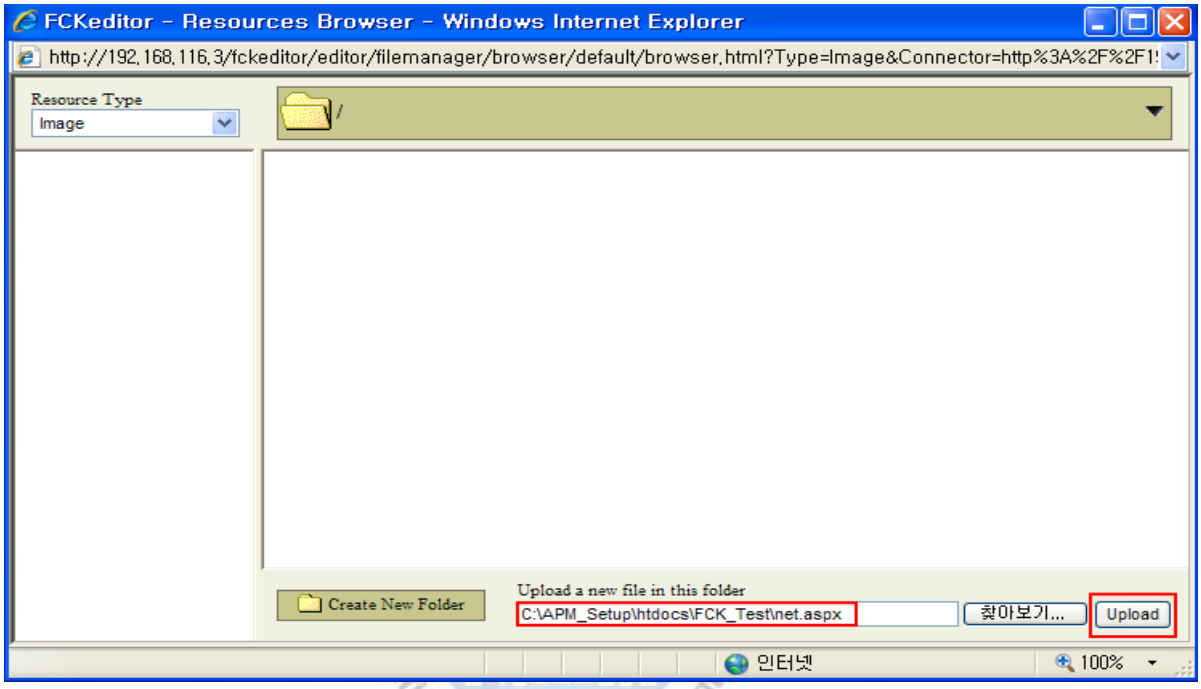


<그림> File Browser

2.1.1 웹 스크립트 파일 업로드(asp, aspx)

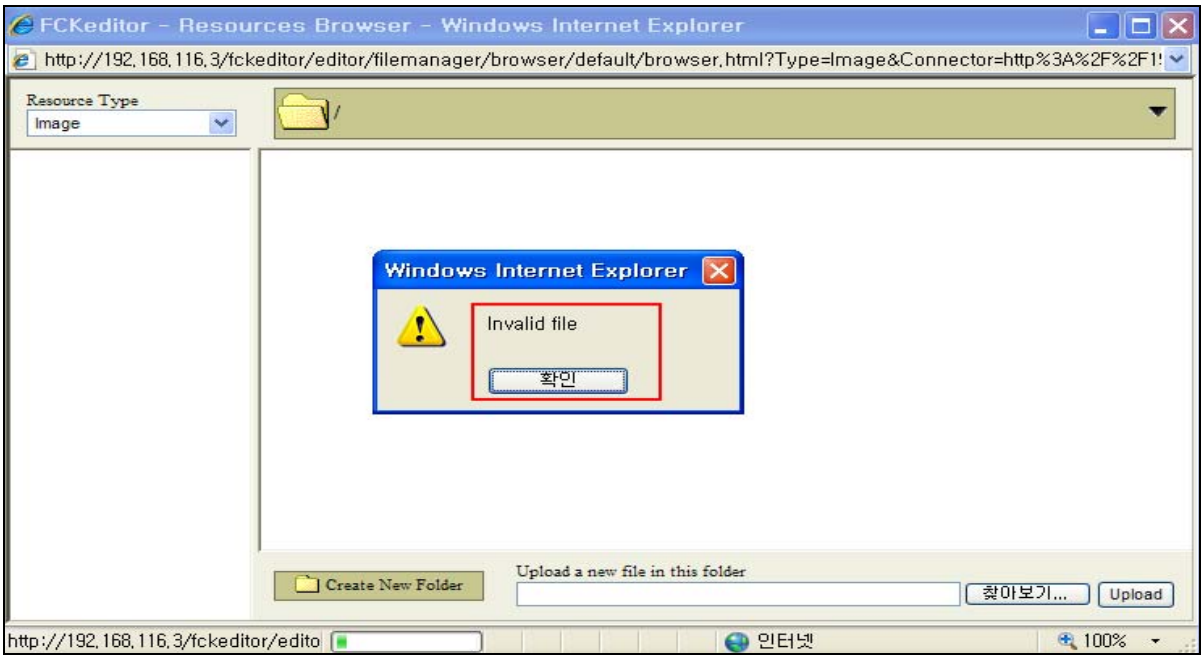
웹 스크립트 파일 필터링 여부를 확인하기 위해 관련 파일 업로드 테스트를 진행함

❍ .NET(aspx) 파일 업로드를 시도함



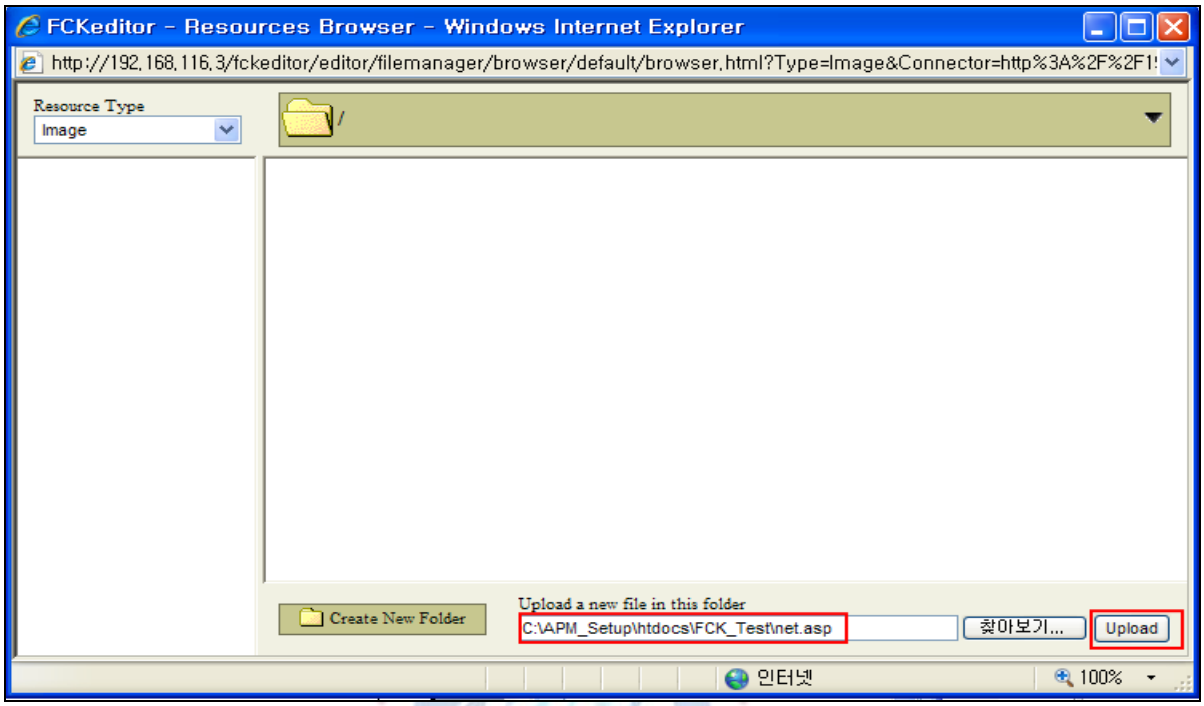
<그림> 파일 업로드

❍ .NET(aspx) 확장 필터링으로 인한 파일 업로드가 실패함



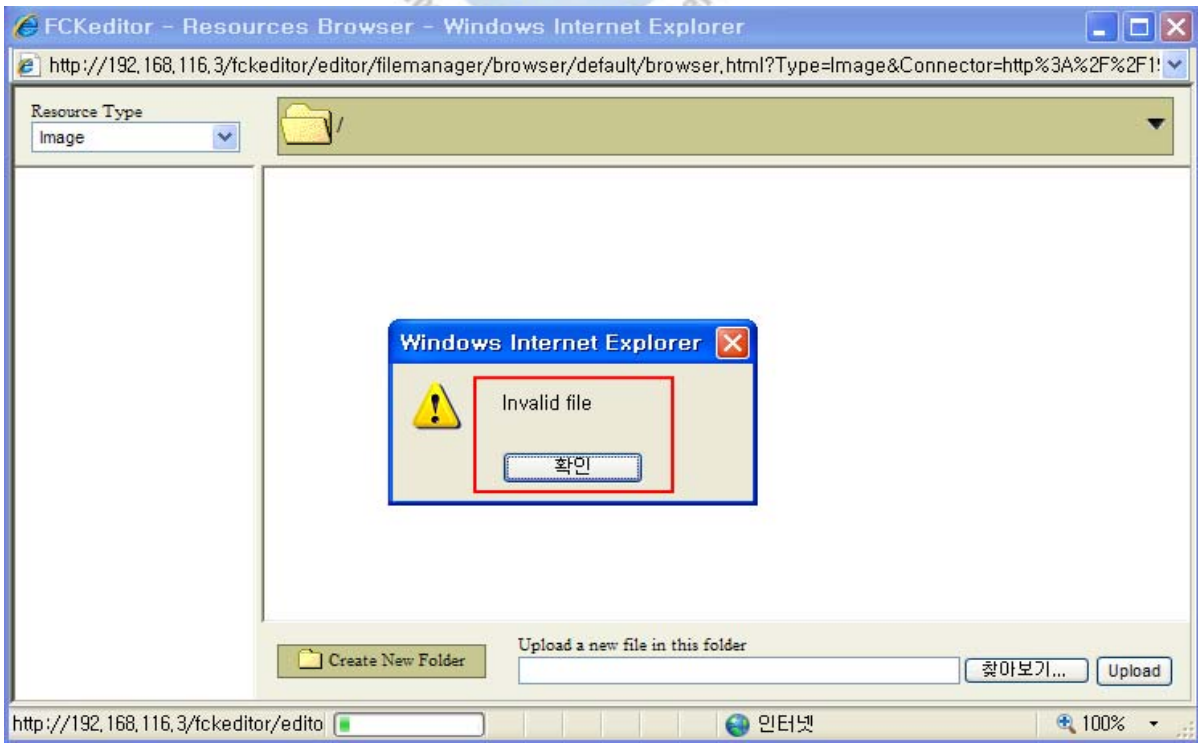
<그림> 파일 업로드 실패

❍ ASP 파일 업로드를 시도함



<그림> 파일 업로드

❍ ASP 확장 필터링으로 인한 파일 업로드가 실패함

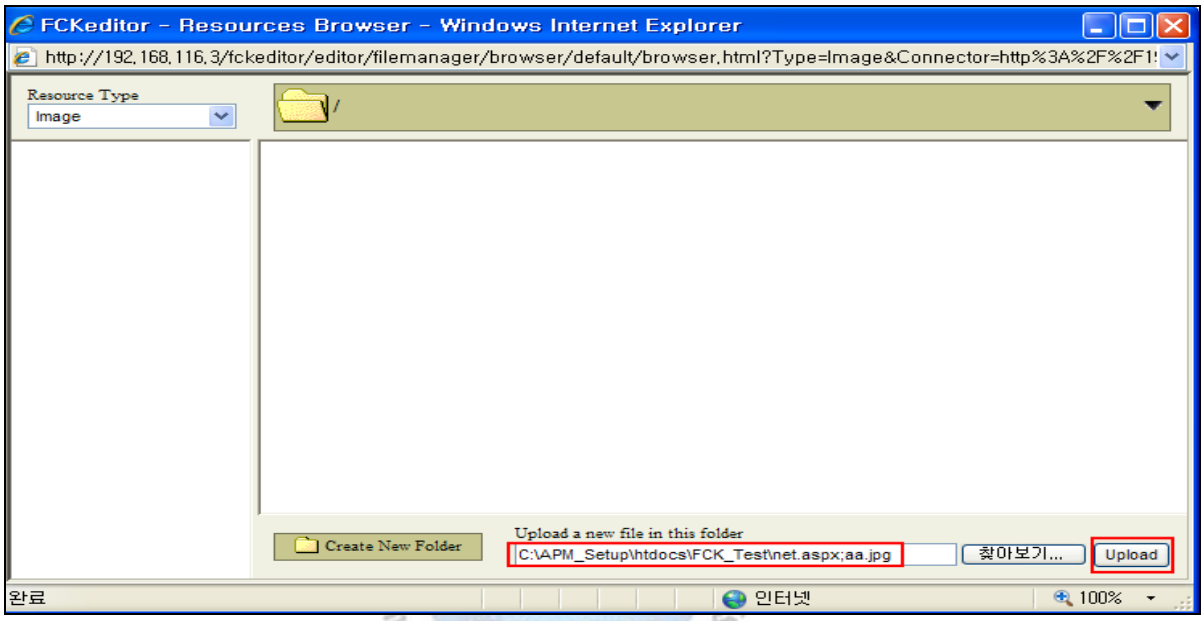


<그림> 파일 업로드 실패

2.1.2 IIS 6.0 확장자 파싱 취약점을 이용한 웹 스크립트 파일 업로드(asp, aspx)

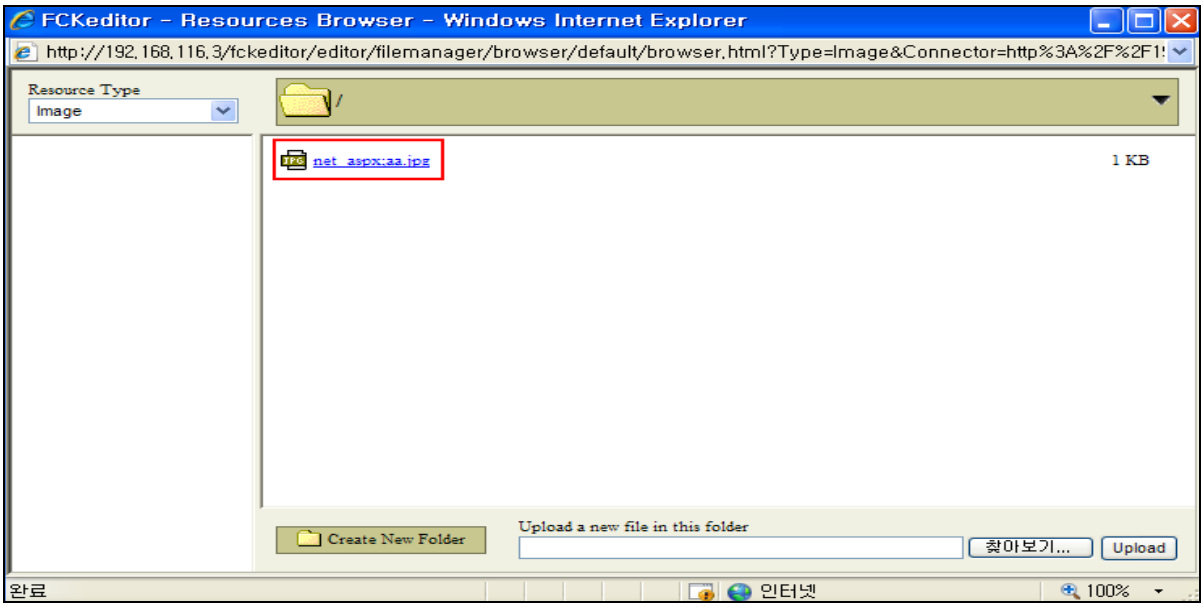
IIS 6.0 확장자 파싱 취약점(※별첨 참조)을 이용하여 확장자 우회를 통해 파일 업로드 테스트를 진행함

❍ net.aspx;aa.jpg 파일업로드를 시도함



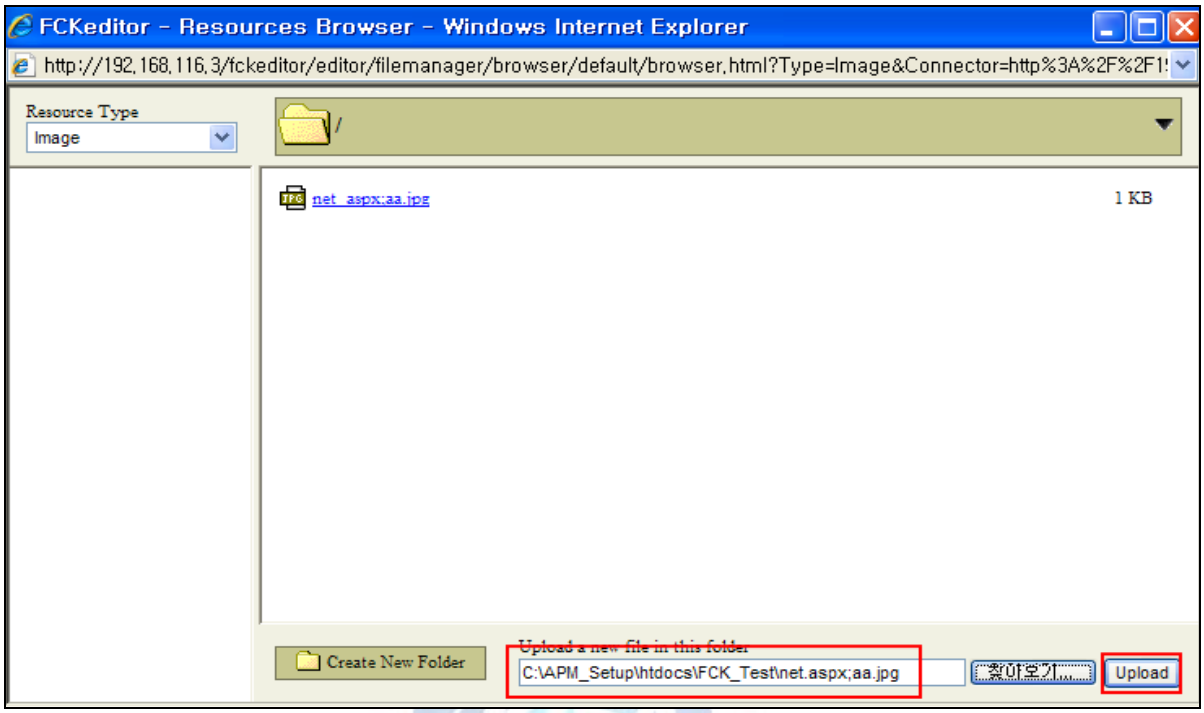
<그림> 파일 업로드

❍ 해당 파일은 정상적으로 업로드되며 파일명 필터링으로 인해 net_aspx;aa.jpg 로 변경되어 업로드됨



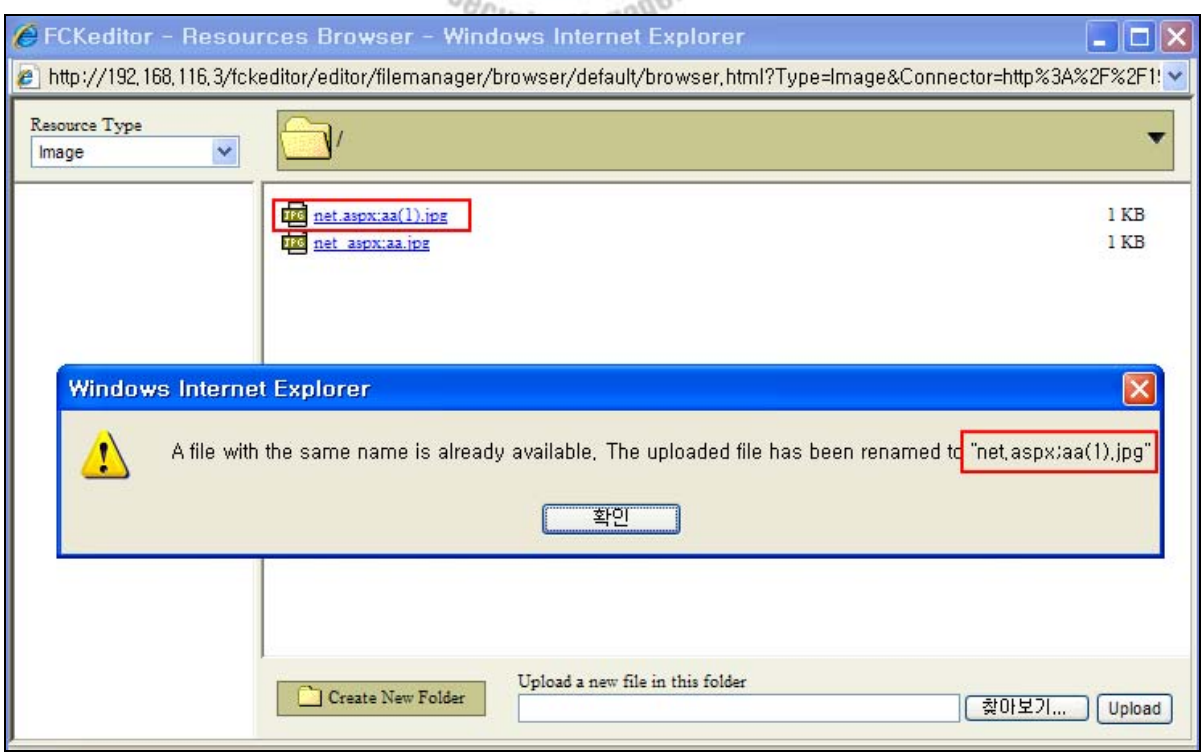
<그림> 파일 업로드

❍ 동일한 파일명(net.aspx;aa.jpg)으로 다시 파일 업로드를 시도함



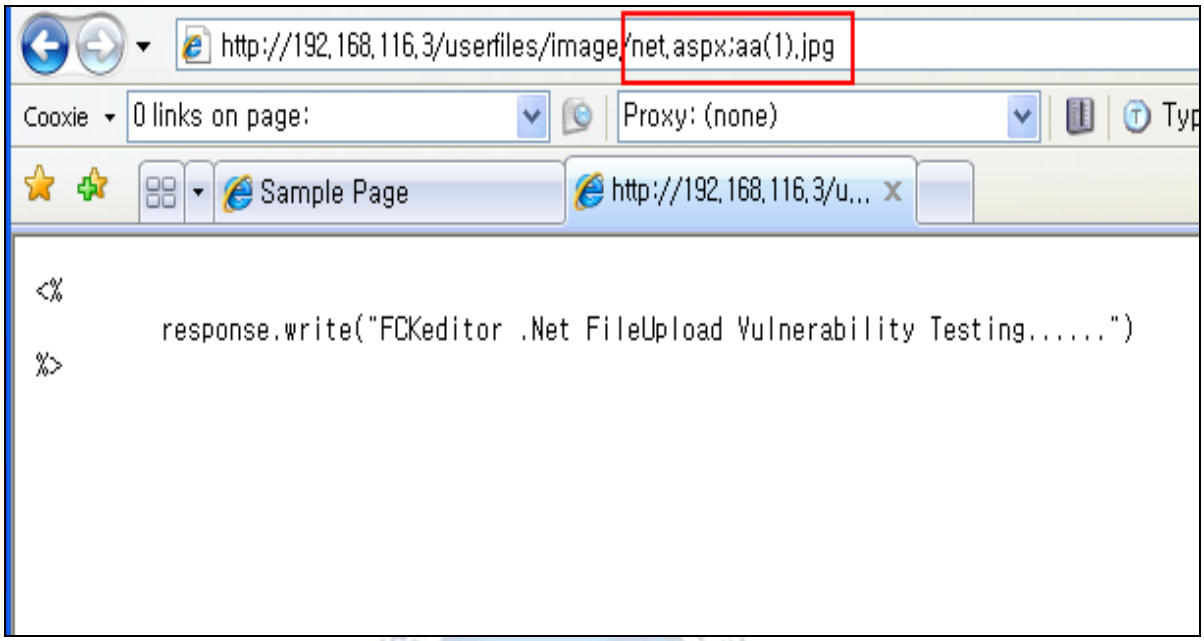
<그림> 파일 업로드

❍ 동일 파일명으로 업로드 할 경우 파일명 필터링이 제대로 이루어 지지 않아 net.aspx;aa(1).jpg 로 업로드 됨



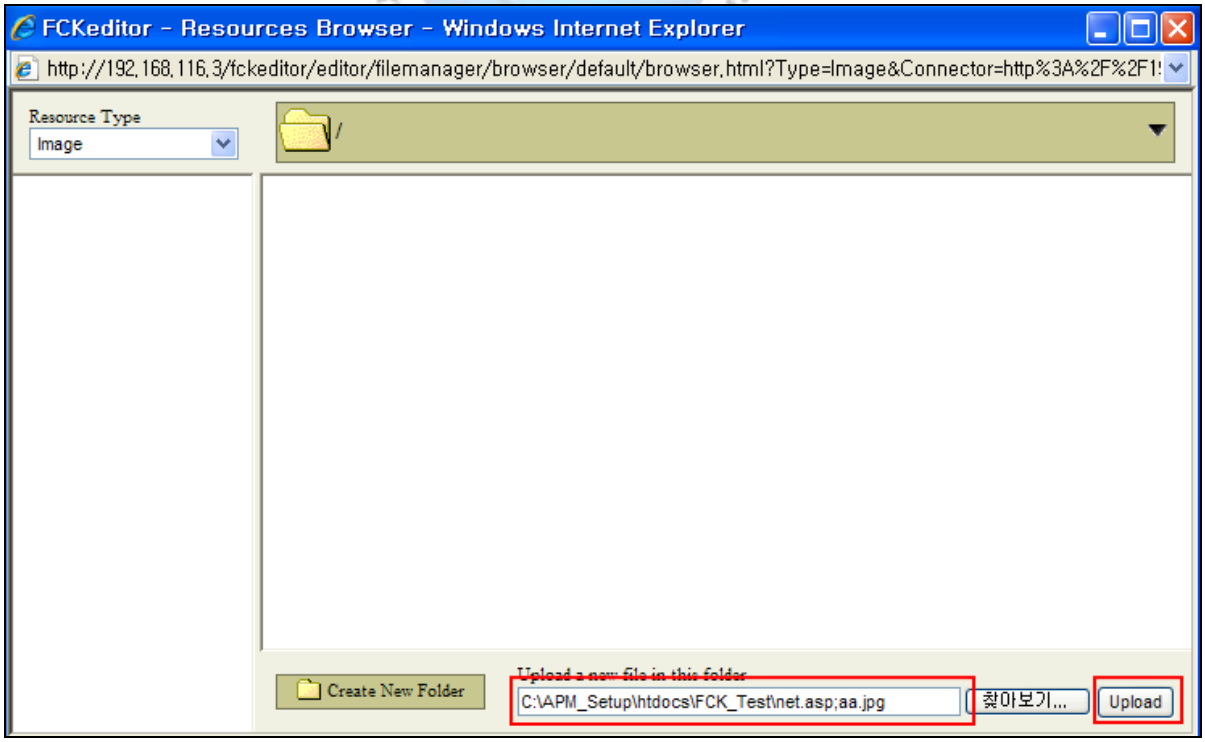
<그림> 파일 업로드

❍ 해당 파일(net.aspx;aa(1).jpg) 실행 시 웹 스크립트로 실행되지 않음



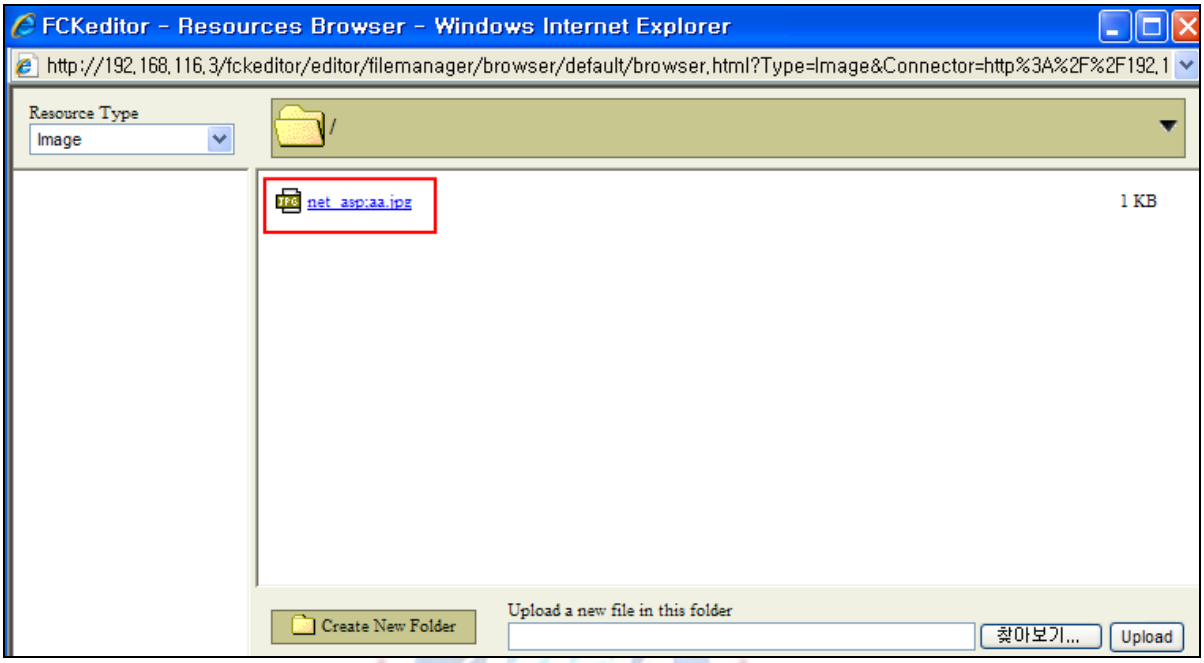
<그림> 웹 스크립트 실행 실패

❍ net.asp;aa.jpg 파일업로드를 시도함



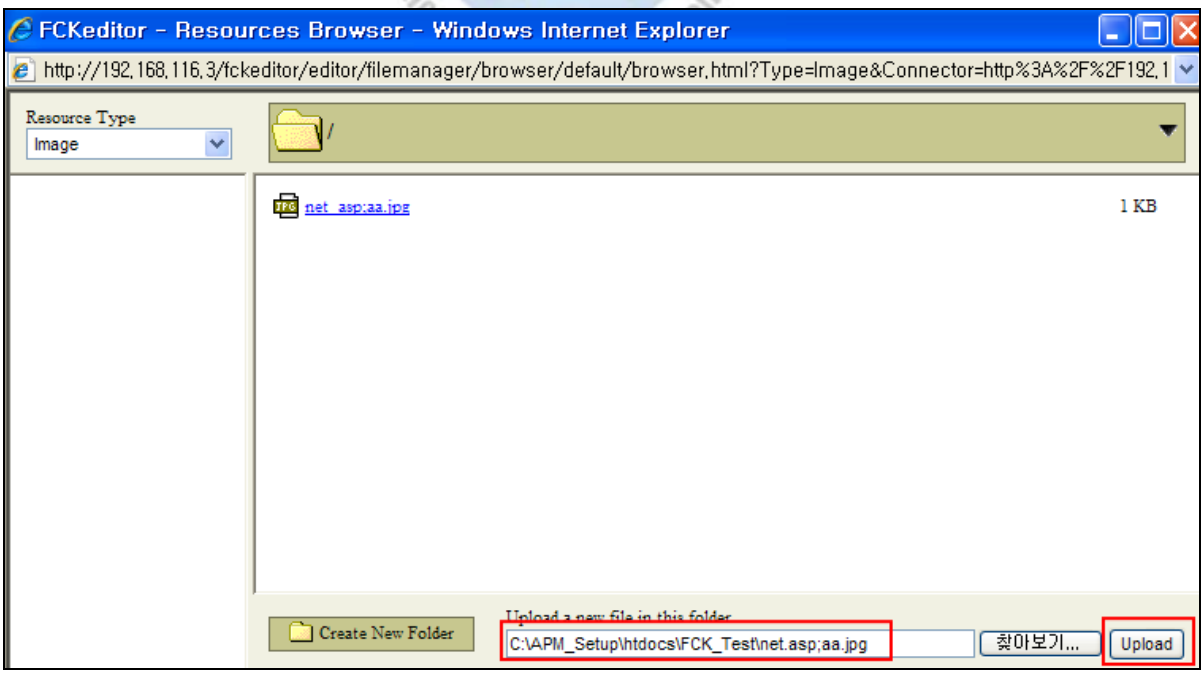
<그림> 파일 업로드

❍ 해당 파일은 정상적으로 업로드되며 파일명 필터링으로 인해 net_asp;aa.jpg 로 변경되어 업로드 됨



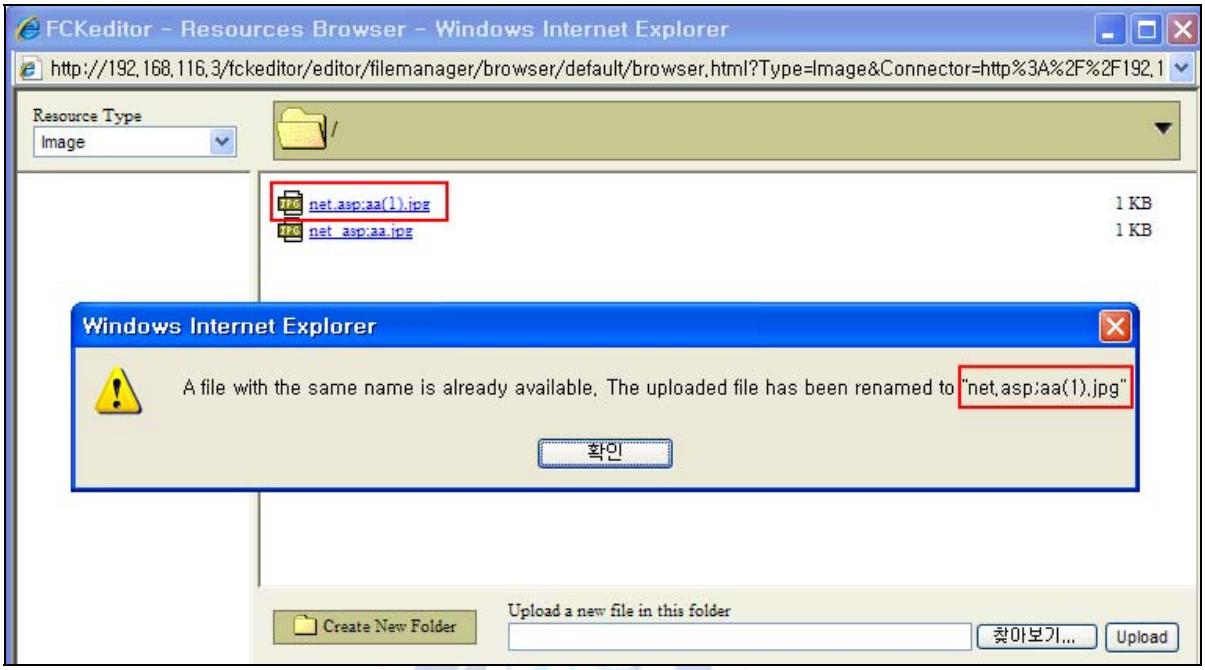
<그림> 파일 업로드

❍ 동일한 파일명(net.aspx;aa.jpg)으로 다시 파일 업로드를 시도함



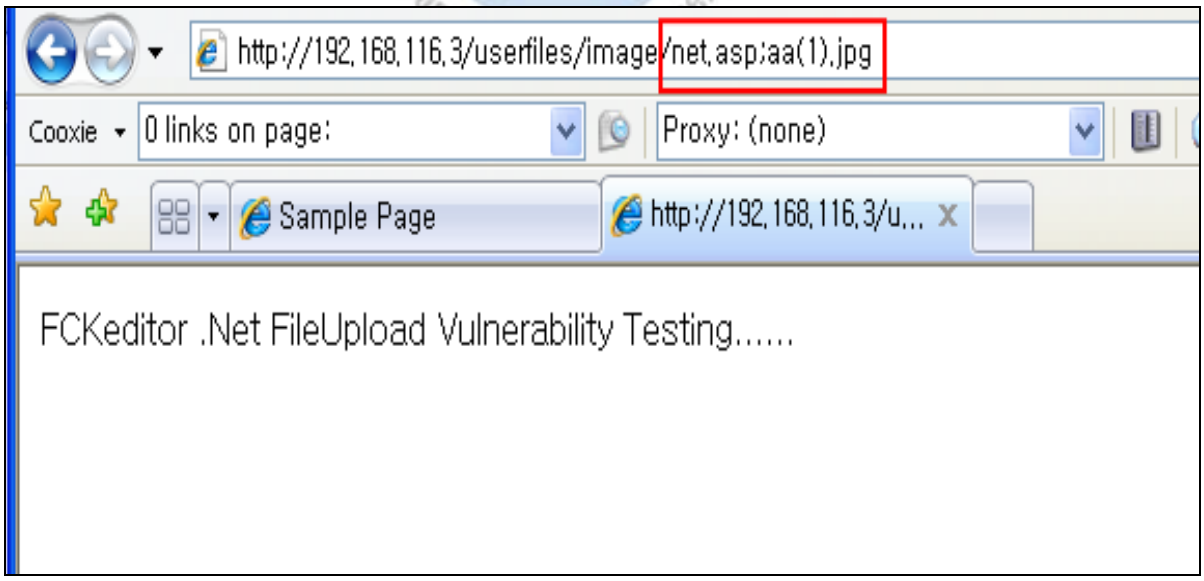
<그림> 파일 업로드

❍ 동일 파일명으로 업로드 할 경우 파일명 필터링이 제대로 이루어 지지 않아 net.asp;aa(1).jpg 로 업로드 됨



<그림> 파일 업로드

❍ 해당 파일(net.aspx;aa(1).jpg) 실행 시 웹 스크립트로 실행됨



<그림> 웹 스크립트 실행

2.1.3 FCKeditor .NET 파일 업로드 소스코드 분석

❍ 파일 업로드 시 파일명에 대한 필터링을 통해 '.'이 존재할 경우 '_'로 치환함

```
FredCK.FCKeditorV2.FileBrowser.FileWorkerBase                    CheckNonHtmlFile(HttpPostedFile file)

        // Remove . \ / | : ? * " < >
        return Regex.Replace( folderName, @"[.\\/|:?*""<>\p{C}]", "_", RegexOptions.None );
    }

    // Do a cleanup of the file name to avoid possible problems
    private string SanitizeFileName( string fileName )
    {
        // Replace dots in the name with underscores (only one dot can be there... security issue).
        if ( Config.ForceSingleExtension )
            fileName = Regex.Replace( fileName, @"\.(?![^.]*$)", "_", RegexOptions.None );

        // Remove \ / | : ? * " < >
        return Regex.Replace( fileName, @"[\\/|:?*""<>\p{C}]", "_", RegexOptions.None );
    }
```

<그림> 파일 업로드 로직

❍ 파일 업로드 시 동일한 파일 명이 존재할 경우 '.'에 대한 치환이 이루어지지 않고 업로드 되는 파일명을 그대로 사용함

```
FredCK.FCKeditorV2.FileBrowser.FileWorkerBase                    CheckNonHtml

        // Get the uploaded file name.
        sFileName = System.IO.Path.GetFileName( oFile.FileName );
        sFileName = this.SanitizeFileName( sFileName );

        string sExtension = System.IO.Path.GetExtension( oFile.FileName );
        sExtension = sExtension.TrimStart( '.' );

        if ( !this.Config.TypeConfig[ resourceType ].CheckIsAllowedExtension( sExtension ) )
        {
            this.SendFileUploadResponse( 202, isQuickUpload );
            return;
        }

        if ( this.Config.CheckIsNonHtmlExtension( sExtension ) && !this.CheckNonHtmlFile( oFile ) )
        {
            this.SendFileUploadResponse( 202, isQuickUpload );
            return;
        }

        int iErrorNumber = 0;
        int iCounter = 0;

        while ( true )
        {
            string sFilePath = System.IO.Path.Combine( sServerDir, sFileName );

            if ( System.IO.File.Exists( sFilePath ) )
            {
                iCounter++;
                sFileName =
                    System.IO.Path.GetFileNameWithoutExtension( oFile.FileName ) +
                    "(" + iCounter + ")." +
                    sExtension;

                iErrorNumber = 201;
            }
            else
            {
                oFile.SaveAs( sFilePath );
                break;
            }
        }
```

sFileName : 필터링된 파일명
oFile.FileName : 필터링 되지 않은 파일명

<그림> 파일 업로드 로직

3. 대응 방안

✓ FCKeditor .NET 기능 중 File Upload 기능을 이용하지 않을 경우 환경설정 파일(config.ascx)의 CheckAuthentication 값을 false로 설정함으로써 업로드 기능을 비활성화 함

FileBrowser₩FileWorkerBase.cs | **config.ascx**

```
 * Configuration file for the File Browser Connector for ASP.NET.
--%>
<script runat="server">

    /**
     * This function must check the user session to be sure that he/she is
     * authorized to upload and access files in the File Browser.
     */
    private bool CheckAuthentication()
    {
        // WARNING : DO NOT simply return "true". By doing so, you are allowing
        // "anyone" to upload and list the files in your server. You must implement
        // some kind of session validation here. Even something very simple as...
        //
        //      return ( Session[ "IsAuthorized" ] != null && (bool)Session[ "IsAuthorized" ] == true );
        //
        // ... where Session[ "IsAuthorized" ] is set to "true" as soon as the
        // user logs in your system.

        return false;
    }
```

<그림> config.aspx

✓ File Upload 기능을 이용해야 할 경우 FileWorkerBase.cs 파일을 수정하여 동일 파일명 업로드 시 파일명 필터링을 통해 파일을 업로드 하도록 수정함
(소스 파일 수정 후 컴파일하여 해당 DLL을 적용)

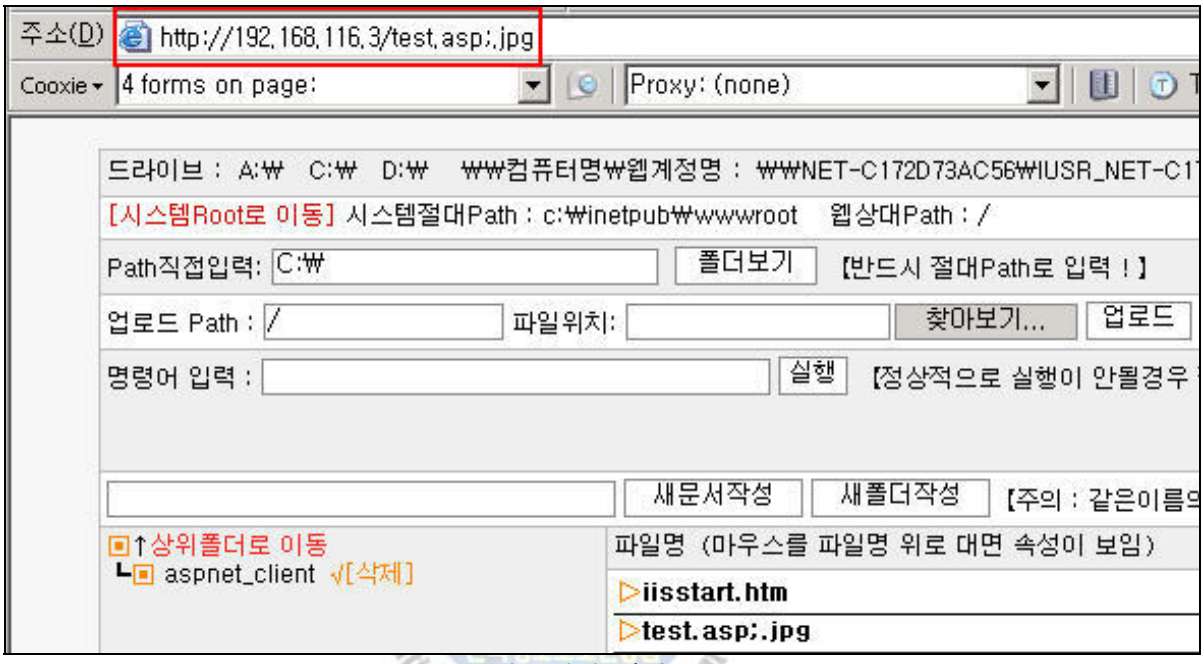
```
while ( true )
{
    string sFilePath = System.IO.Path.Combine( sServerDir, sFileName );

    if ( System.IO.File.Exists( sFilePath ) )
    {
        iCounter++;
        sFileName =
            System.IO.Path.GetFileNameWithoutExtension( sFileName ) +
            "(" + iCounter + ")." +
            sExtension;

        iErrorNumber = 201;
    }
    else
    {
        oFile.SaveAs( sFilePath );
        break;
    }
}
```

<그림> FileWorkerBase.cs

※ 별첨

Microsoft IIS 6.0에서 세미콜론(;)을 포함한 웹 페이지 요청 시 asp.dll을 이용하는 parser 제대로 parsing 을 하지 못해 웹쉘 등의 악성 웹 스크립트를 구동시킬 수 있는 취약점이 존재함

✓ IIS 6.0(Windows Server 2003)에서 테스트한 결과 'test.asp;jpg' 파일이 웹 스크립트로 실행됨



<그림> 웹쉘 실행

✓ IIS 5.0(Windows Server 2000)에서 테스트한 결과 'test.asp;jpg' 파일은 텍스트 파일로 실행됨

<그림> 웹 스크립트