SecurEyes
Infusing Security

Whitepaper

Exploiting PHP Upload Module of FCKEditor

Bypassing File-type Check

Anant Kochhar
Utkarsh Bhatt
Sabyasachi Samanta

TABLE OF CONTENTS

Abstract 4

Introduction 4

Exploits 4

The Attack Scenario 5

Vulnerable Versions 15

Impact 15

Recommended Resolutions 15

About The Authors 17

About SecurEyes 17

Addendum* 17

Abstract

The PHP file upload module in FCKEditor allows developers to offer file upload functionality to end users. This paper describes a vulnerability which allows attackers to bypass file-type checks in this module and upload malicious PHP code into the web servers.

Introduction

FCKEditor (*now CKeditor*) is an **open source WYSIWYG text editor** from CKSource that can be integrated into web applications, to give end users word processor like interface. FCK stands for 'Frederico Caldeira Knabben', the creator of the project, and the first version was released in 2003.

This editor supports many server side languages like ASP, ASP.NET, PHP etc. The PHP upload module, for PHP web applications, has a vulnerability which allows remote attackers to bypass file-type checks. This vulnerability was discovered during the course of our website audit work.

The vulnerability affects FCKEditor versions **2.6.4 and below**.

Note: A different recently discovered vulnerability in the ASP.NET connector file allows attackers to upload malicious ASP code into vulnerable servers.

Exploits

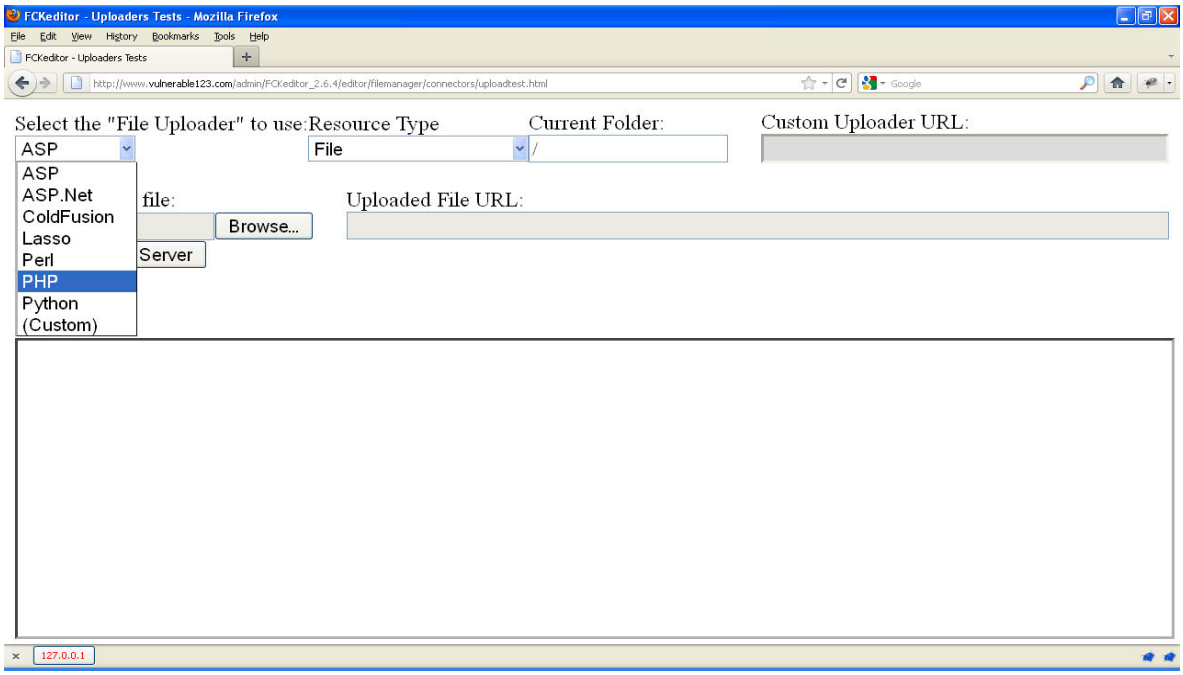
FCKEditor has built-in '**filemanager**' package, which allows developers to offer a file upload and management module to web site end users. For PHP web applications, one of the relevant files is '**upload.php**', which is available at the following location:
<site name><fckeditor>/editor/filemanager/connectors/php/upload.php
This file allows end users to upload files into the web server. It has a built in file-type checker which does not allow PHP files to be uploaded, but a new vulnerability allows remote attackers to bypass this vulnerability.

The Attack Scenario

A website, http://www.vulnerable123.com, has integrated FCKEditor version **2.6.4.** For the sake of this proof-of-concept, it is assumed that the test page, '**uploadtest.html**', packaged with FCKEditor, has not been removed. Also, assume that there is no session-based access control check on any of the FCKEditor files, including '**upload.php**', allowing remote attackers to access them without authentication.

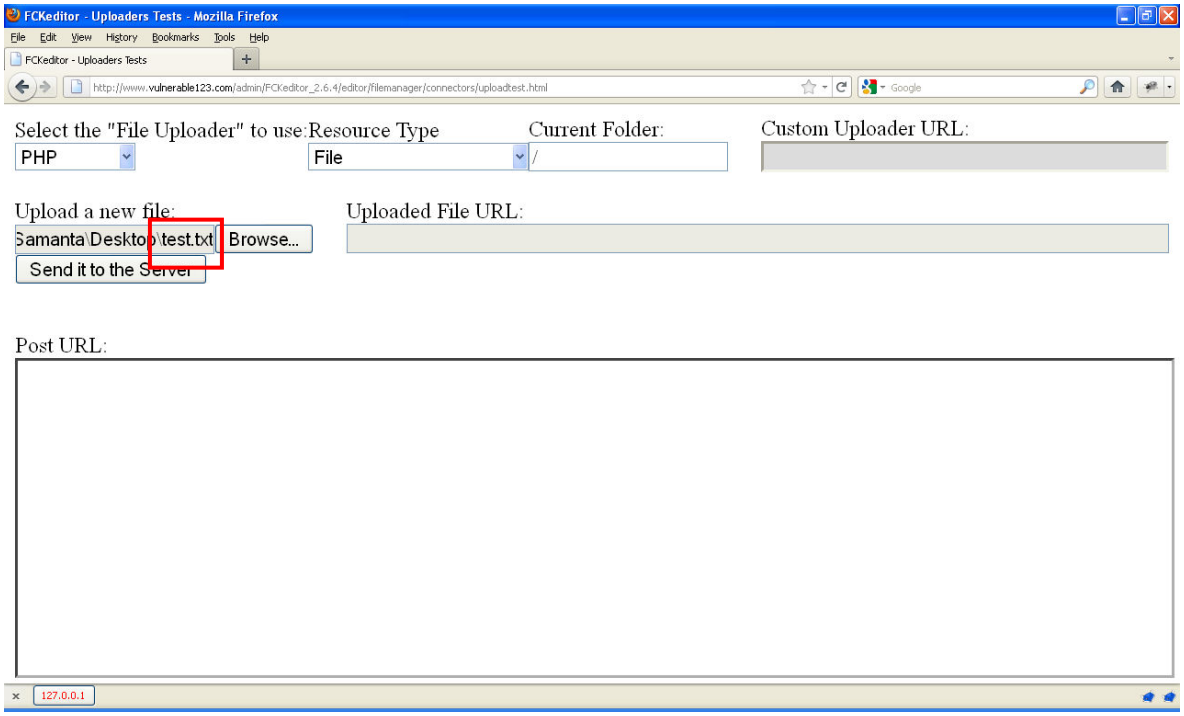
An attacker browses to the File Upload test page at the URL:

http://www.vulnerable123.com/admin/FCKeditor_2.6.4/editor/filemanager/connectors/uploadtest.html

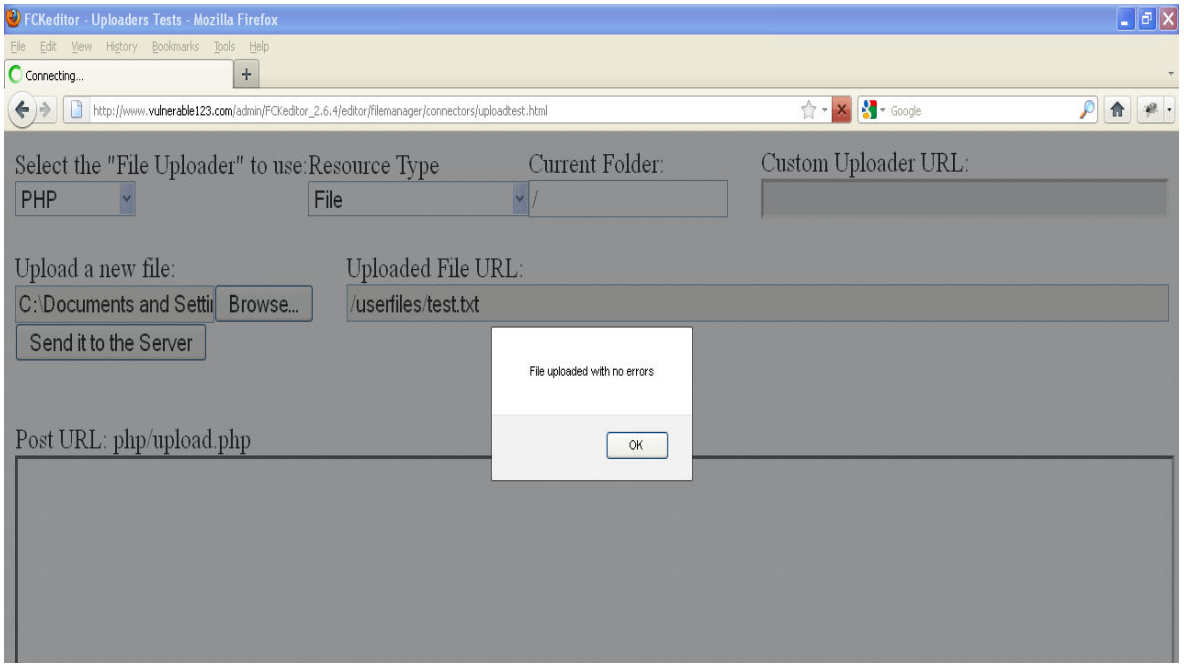


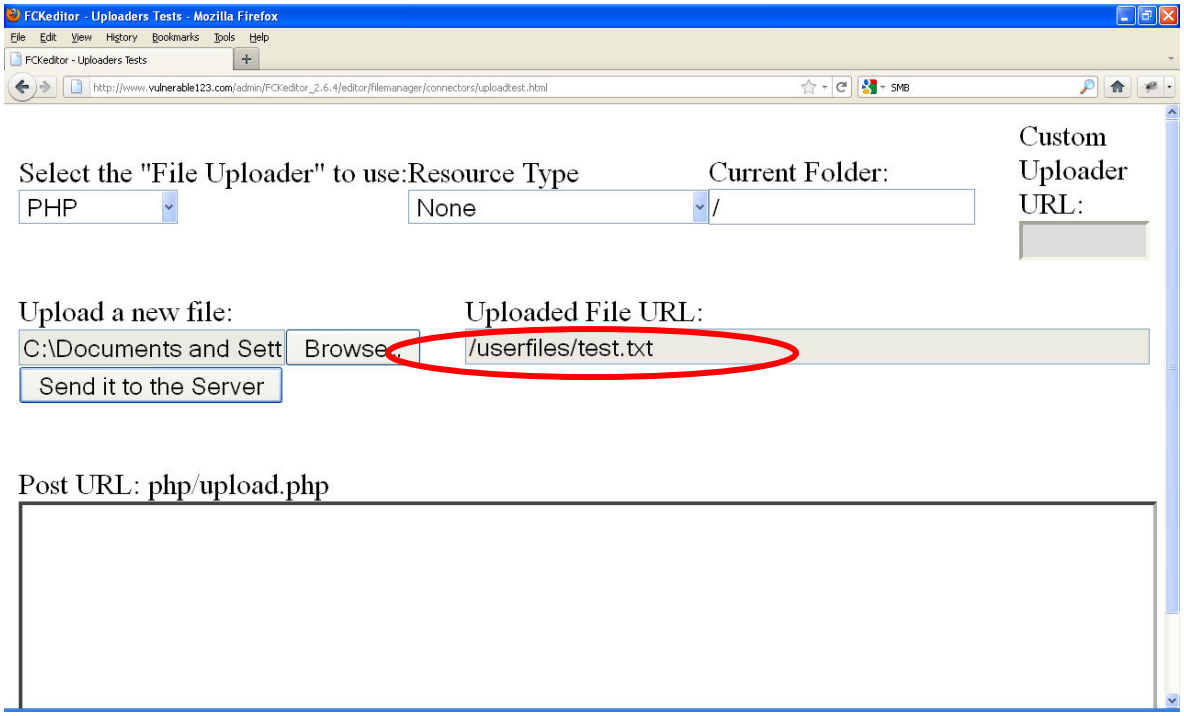
He selects the 'File Uploader' as PHP, 'Resource Type' as File and selects a '.txt' file.

Exploiting PHP Upload Module of FCKEditor



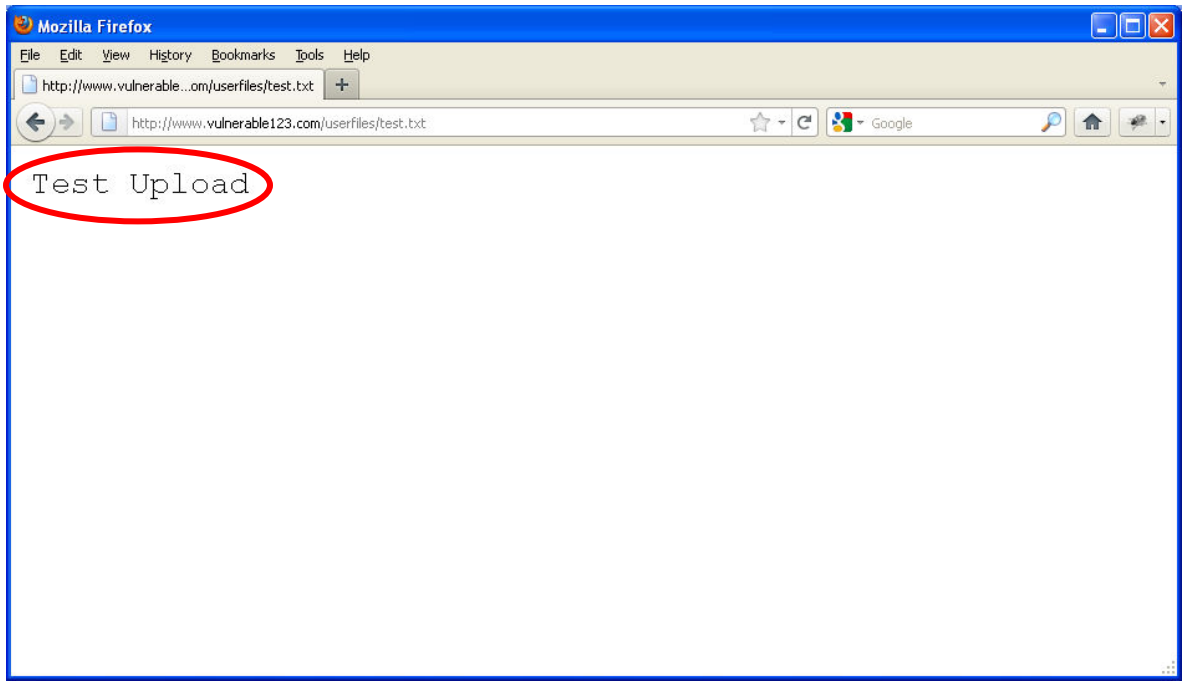
The file was successfully uploaded as shown below.



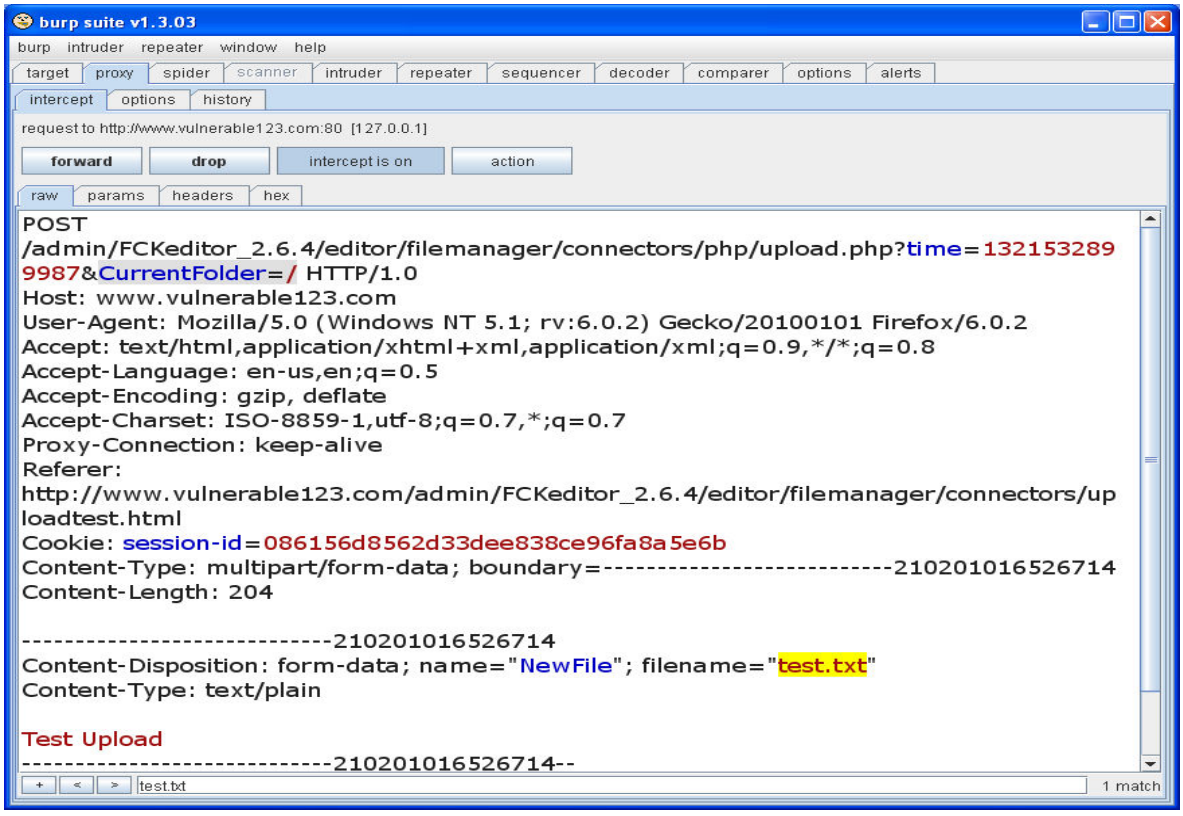


The uploaded file location can be seen by browsing to the above encircled location.

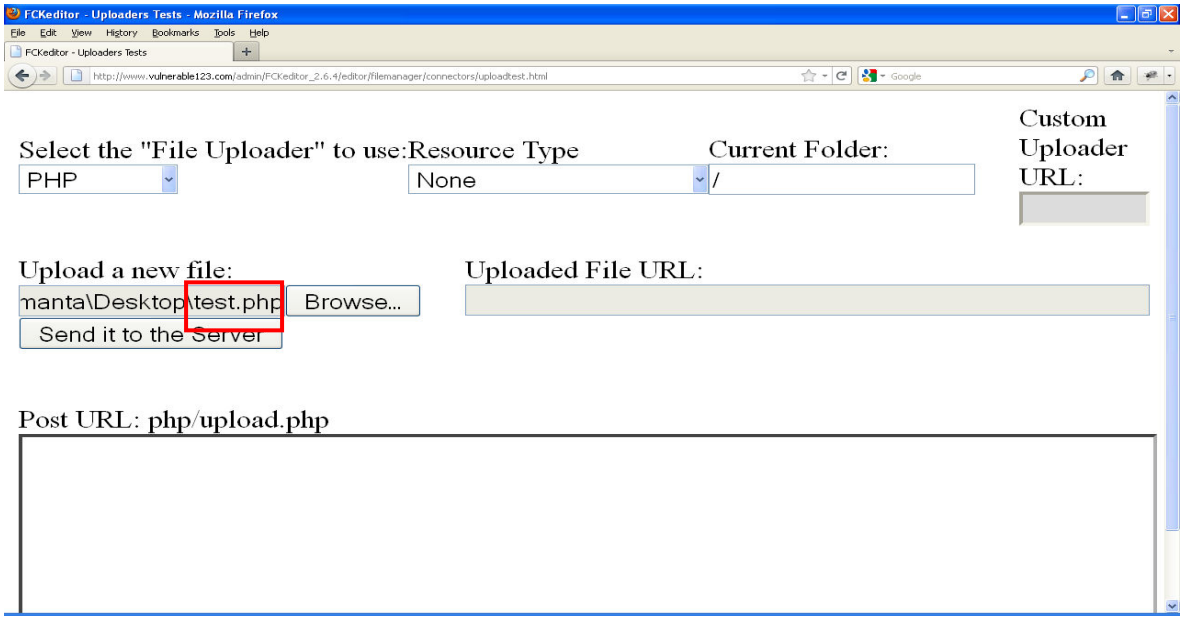
http://www.vulnerable123.com/userfiles/test.txt



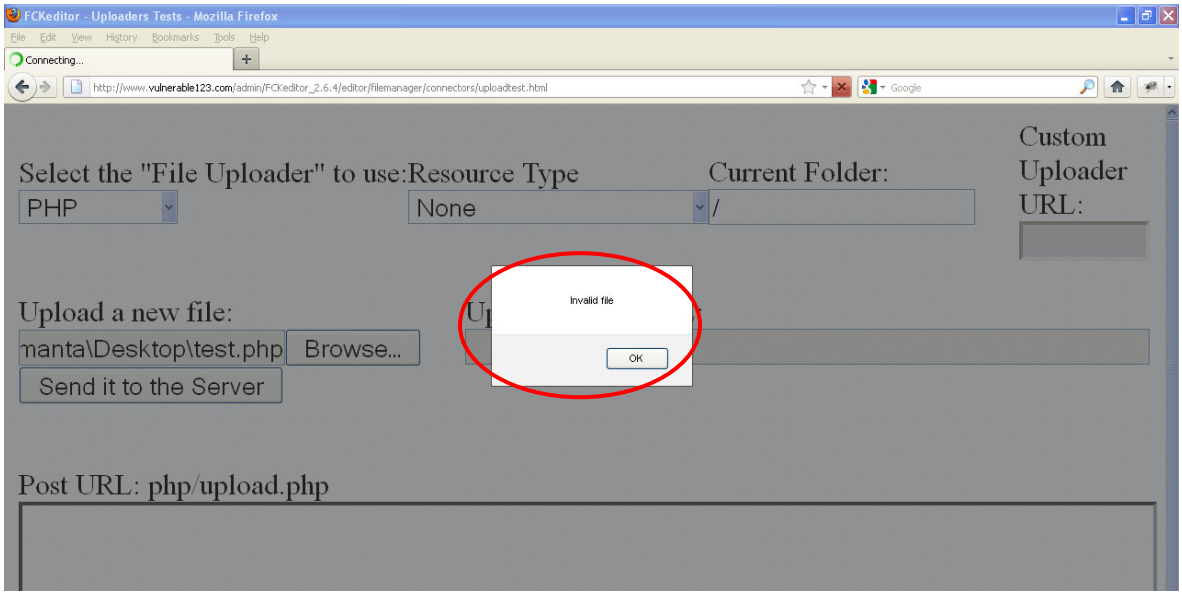
The raw HTTP traffic for this transaction, as intercepted in an HTTP proxy, has the following structure:



Next, the attacker tries to upload a PHP file.

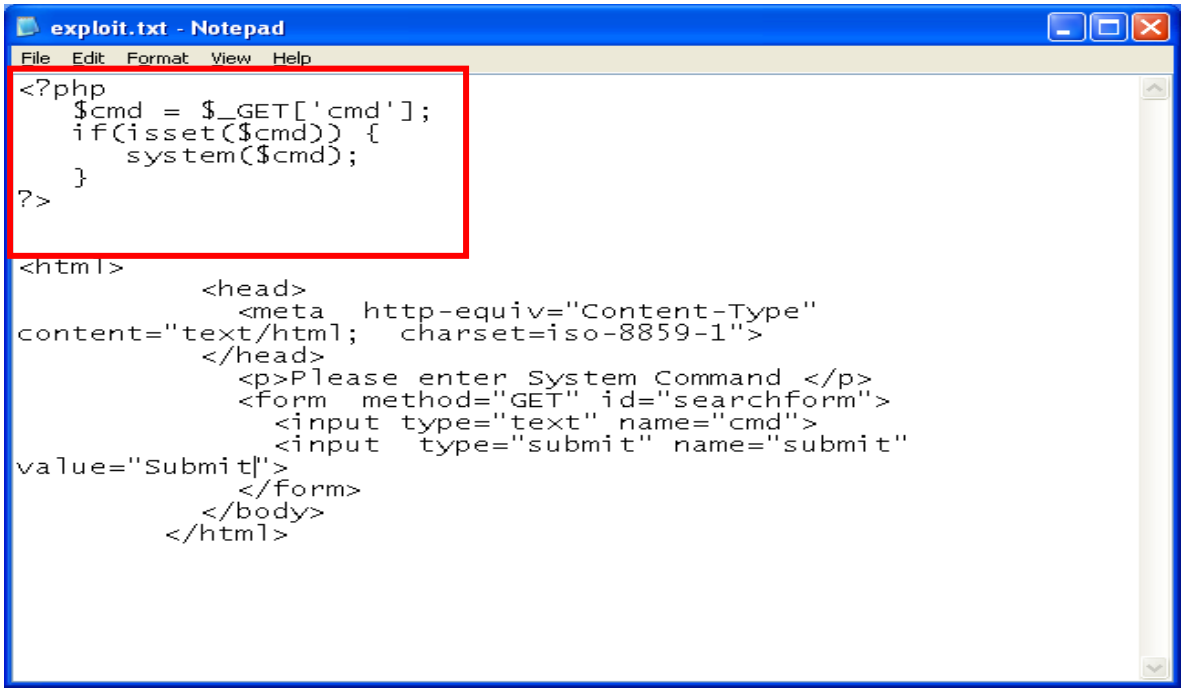


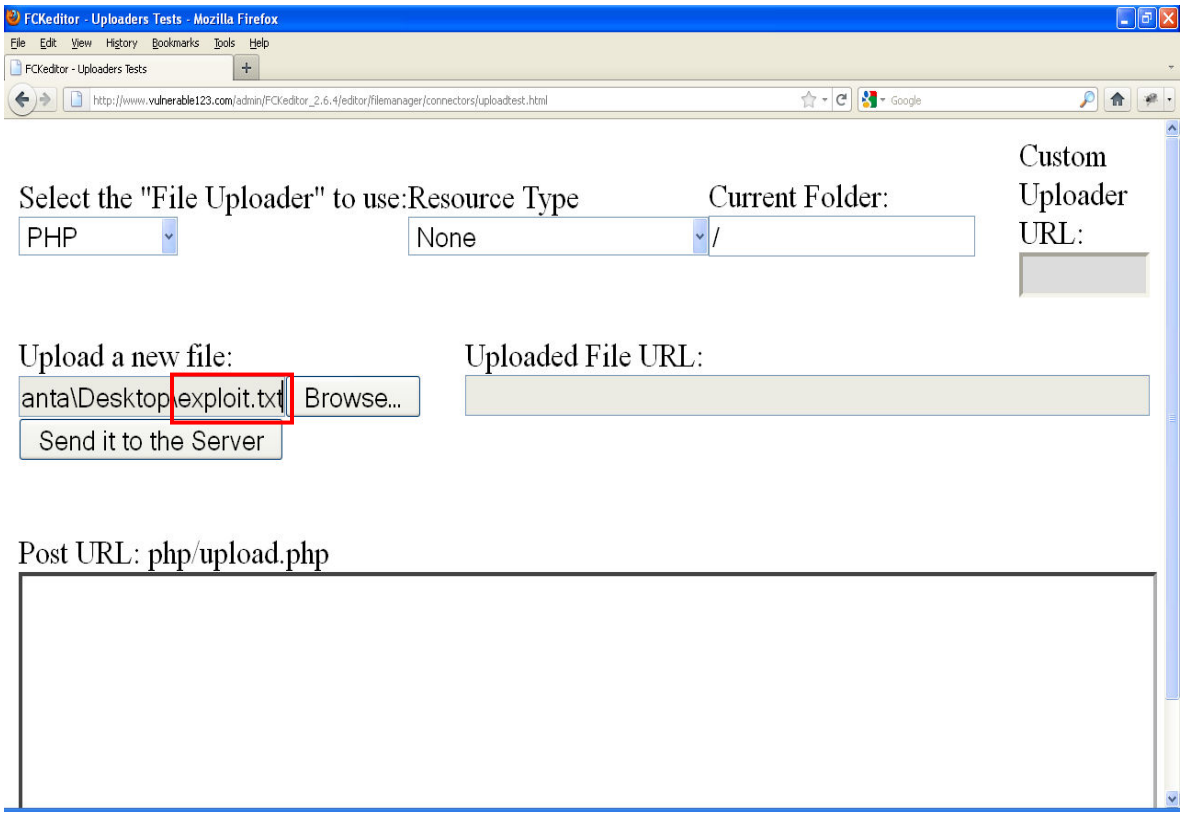
On clicking the 'Send it to the Server' button, the file doesn't get uploaded and the following 'Invalid File' message was displayed:



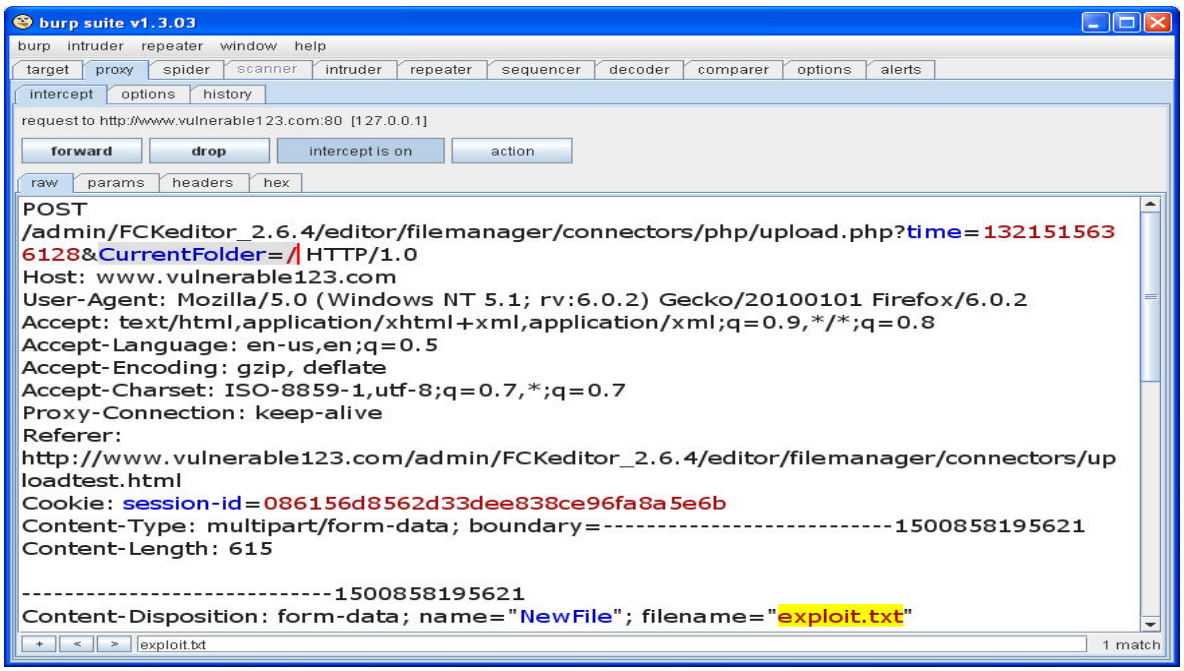
From the above screen, it is evident that '**upload.php**' implements a file-type check on all uploaded files. This check can be bypassed in the following way:

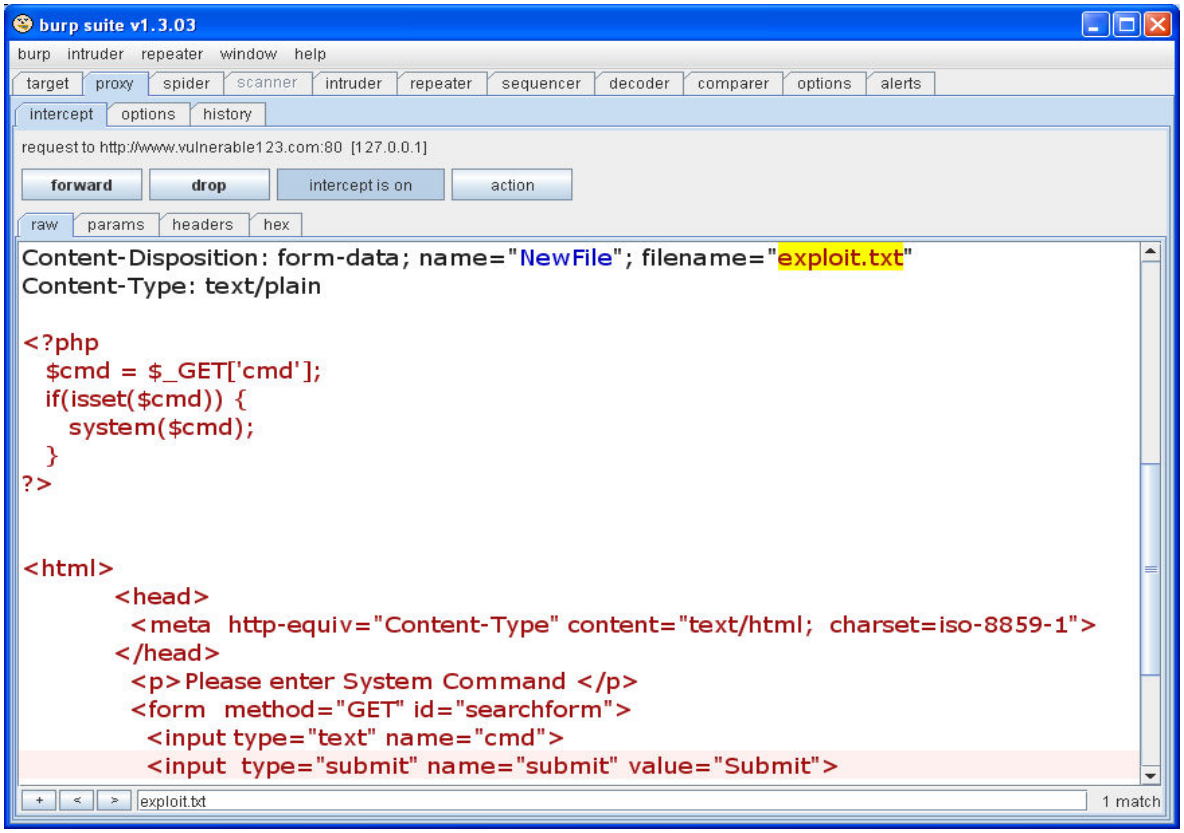
Step#1: The attacker enters malicious PHP code into a '.txt' file, which is valid file-type. Following is the malicious PHP content in the 'exploit.txt' file.





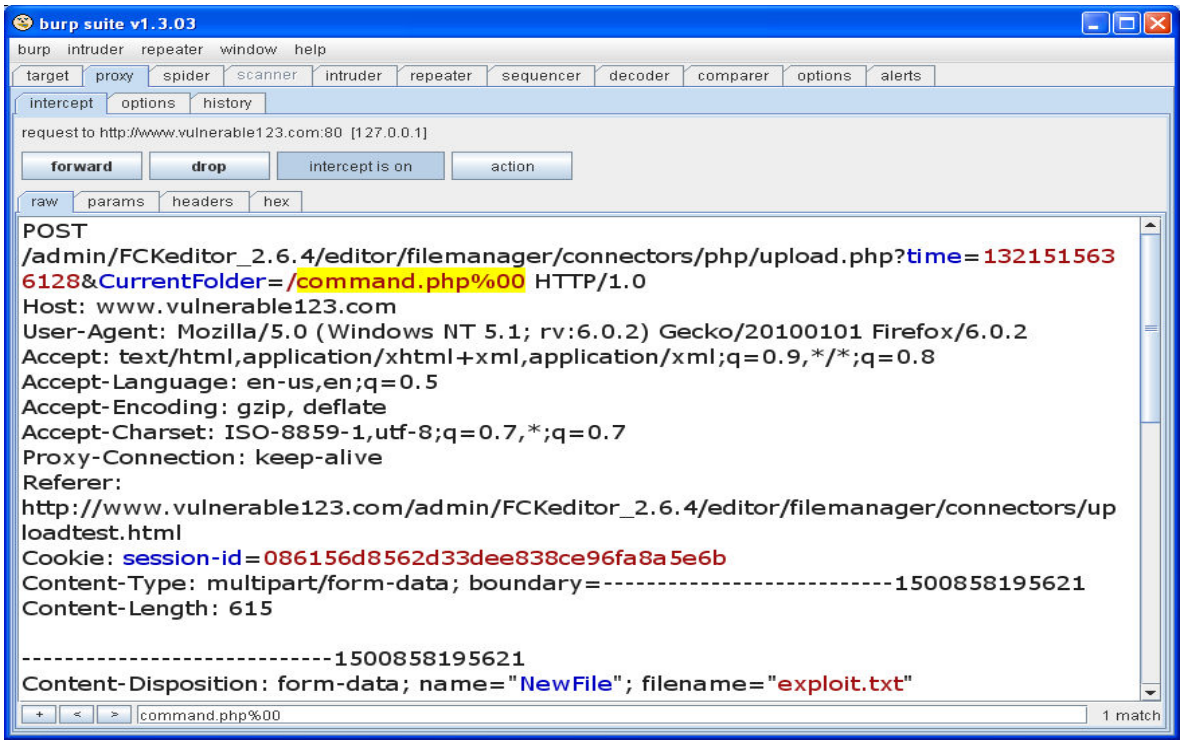
Step#2: The attacker clicks on the 'Send it to the Server' button and captures the request in an HTTP proxy:





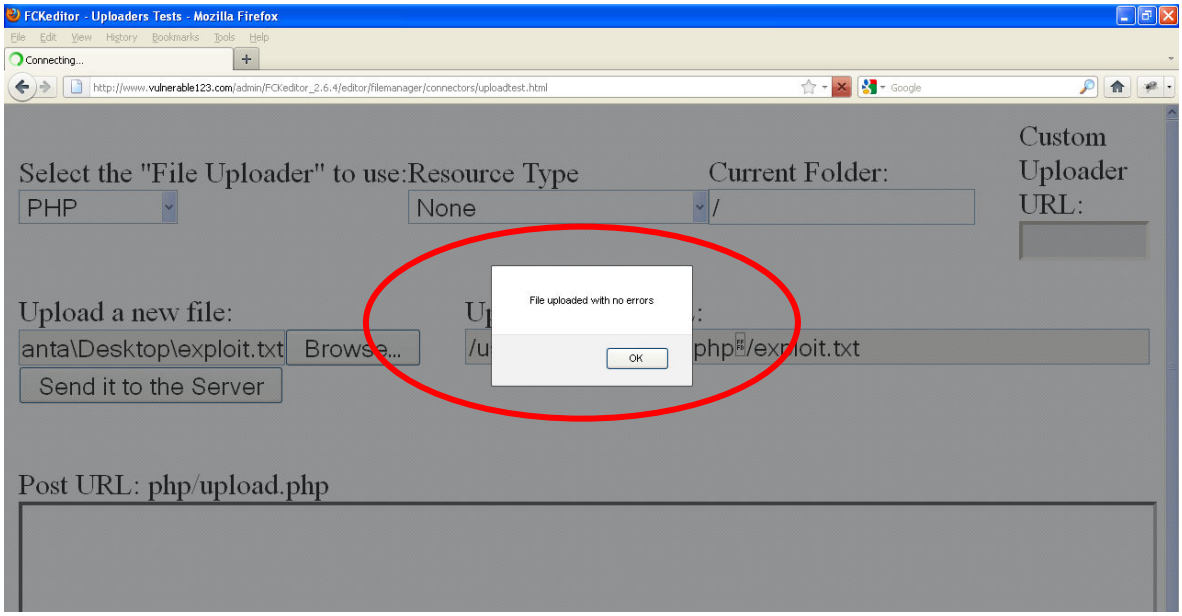
He appends the following string to the value of the URL parameter 'Current Folder'.

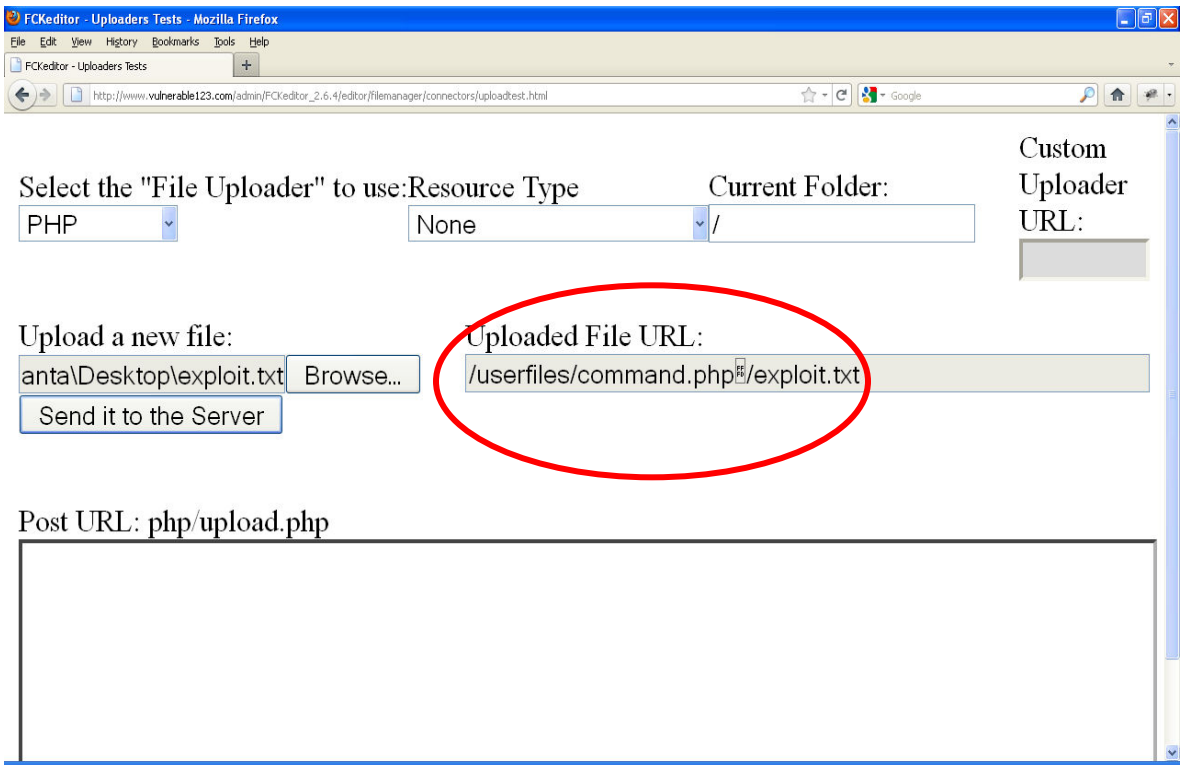
command.php%00



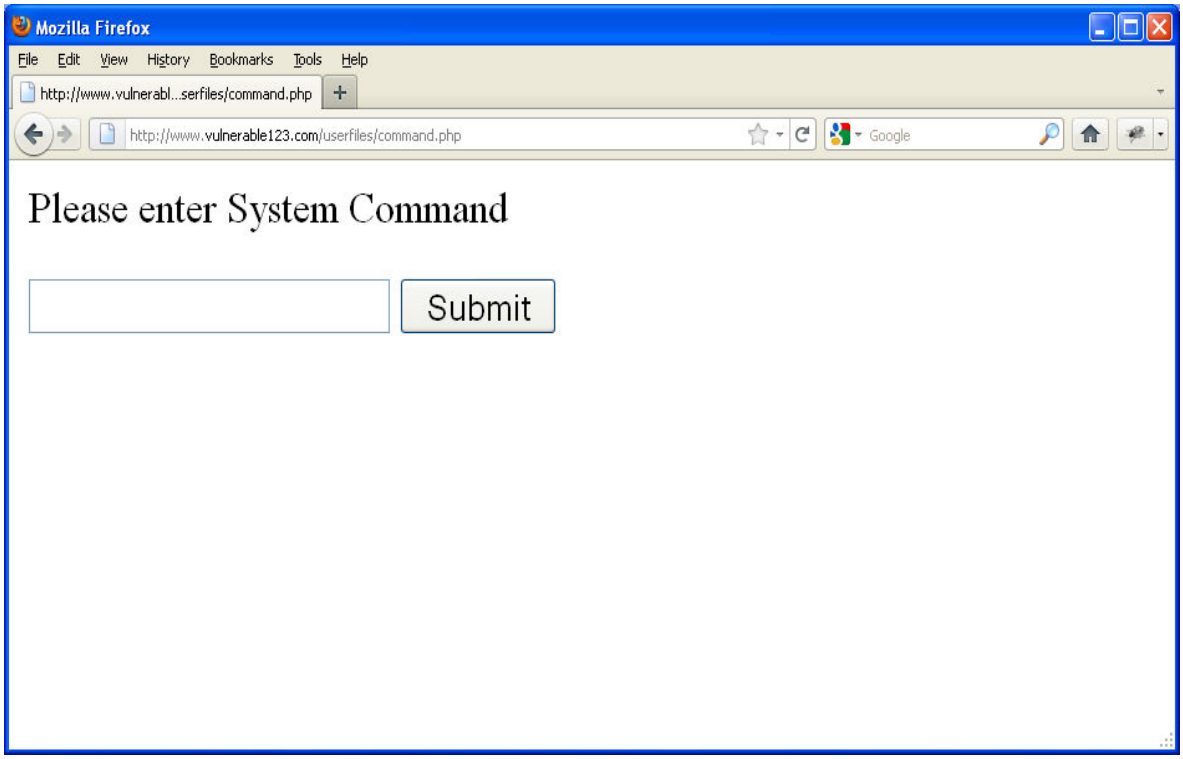
Note: In the above screenshot '**%00**' in the string is the Null Byte.

The file was successfully updated as shown below:



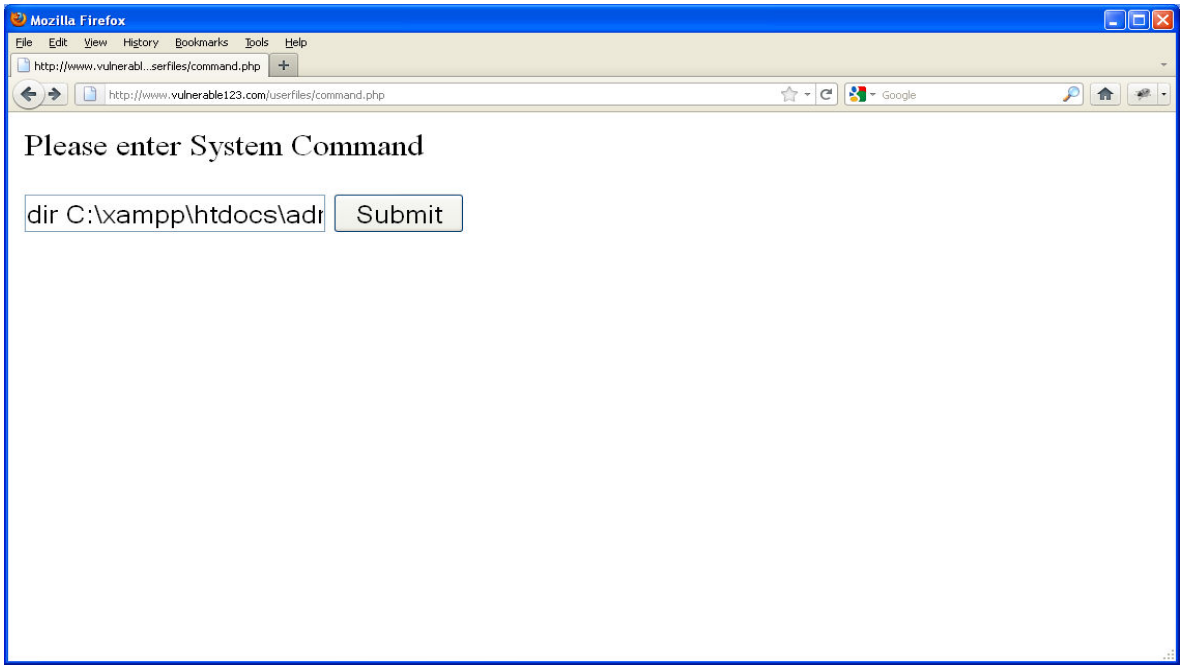


The attacker browses to the uploaded file URL and confirms that the malicious file has been uploaded: http://www.vulnerable123.com/userfiles/command.php

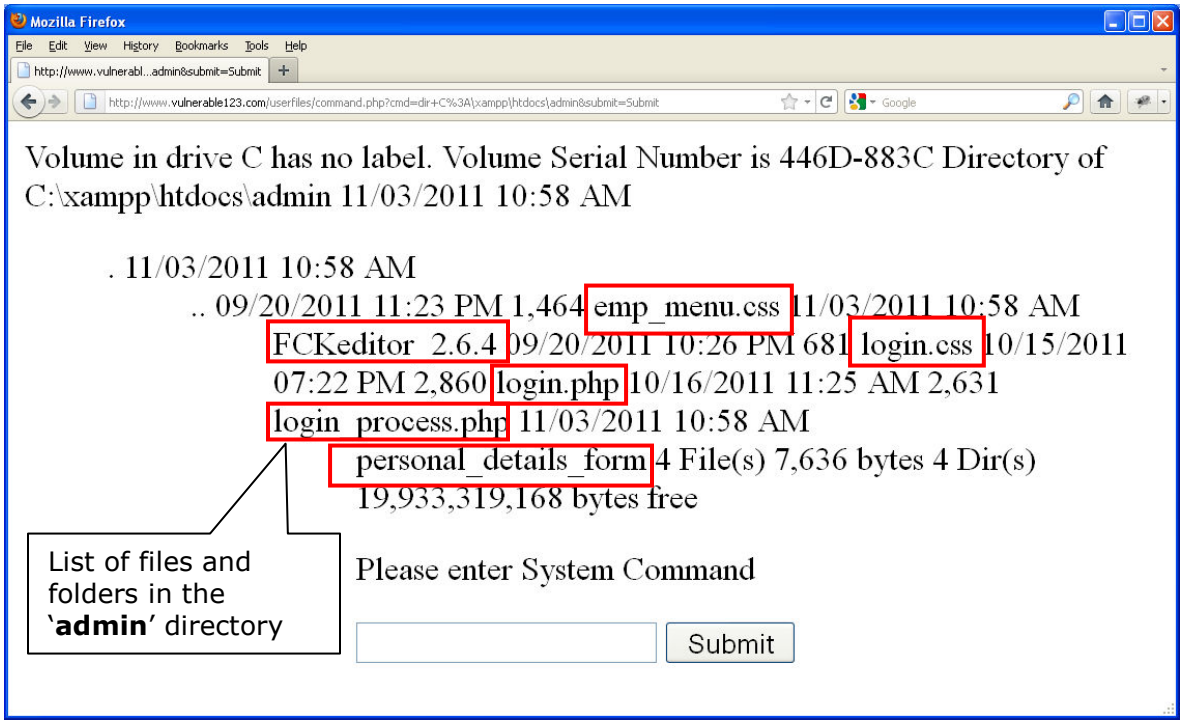


He enters a system command as shown below and clicks on the 'Search' button.

dir C:\xampp\htdocs\admin



He is shown the following details:



The above screenshot shows that the command displayed the directory listing of the mentioned folder.

What's going on?

1) There is a proper validation for the file type in the 'filename' parameter.
2) The 'exploit.txt' file is uploaded successfully because '.txt' is an allowed file-type.
3) The text file contains malicious PHP code, but since the server does not execute text files, it does not pose a security risk.
4) Instead of the 'exploit.txt' file, however, 'command.php' file is created on the server.
5) Moreover, the content of 'command.php' is the same as the content of 'exploit.txt' file.

This happens because:

1) The 'currentfolder' URL parameter value gives the name of a sub-folder, in the upload folder where the file will be uploaded.
2) By inserting the '%00' null byte character at the end of the sub-folder name, it is possible to create a new file instead of a new folder.

3) This happens because the path of the file to be created looks like the following: **/root/mysite/userfiles/command.php%00/exploit.txt**
4) As the server ignores everything after the null byte, a new file is created called 'command.php' with the following location: **/root/mysite/userfiles/command.php**

Vulnerable Versions

This vulnerability affects FCKEditor versions **2.6.4** and below.

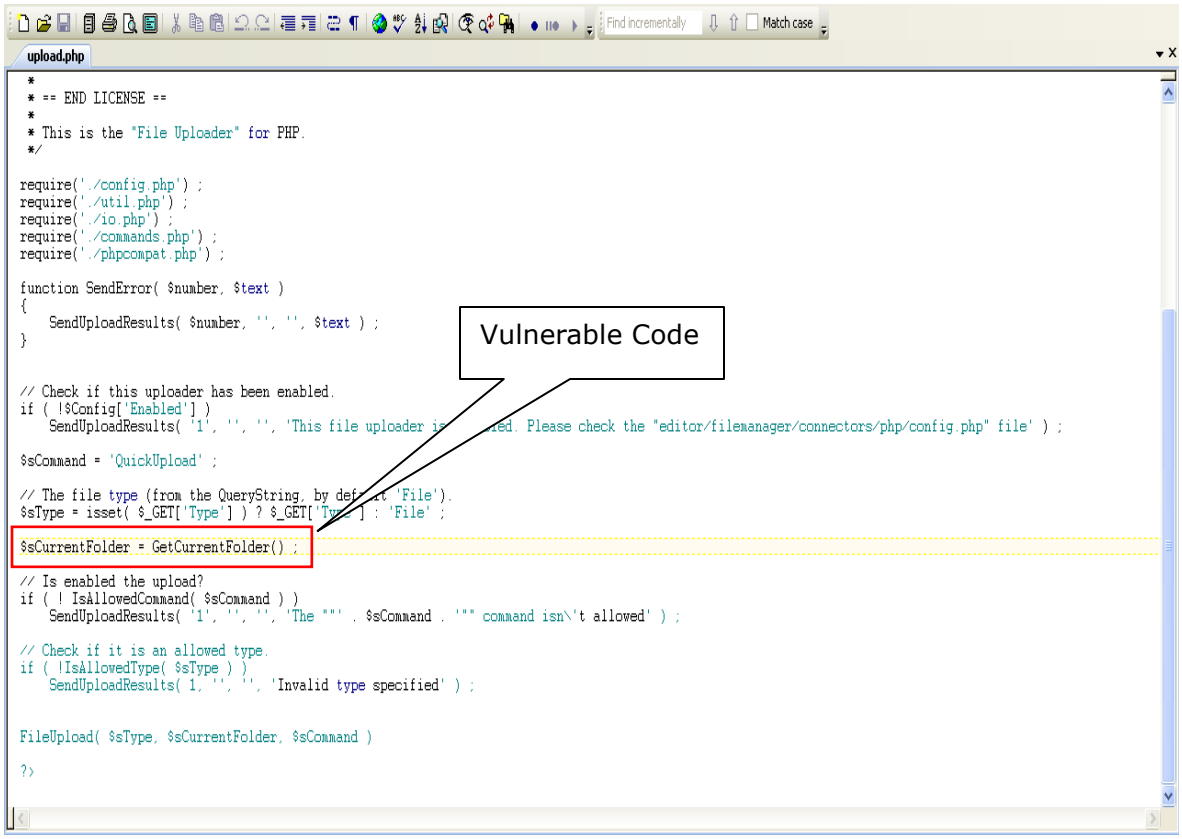
Impact

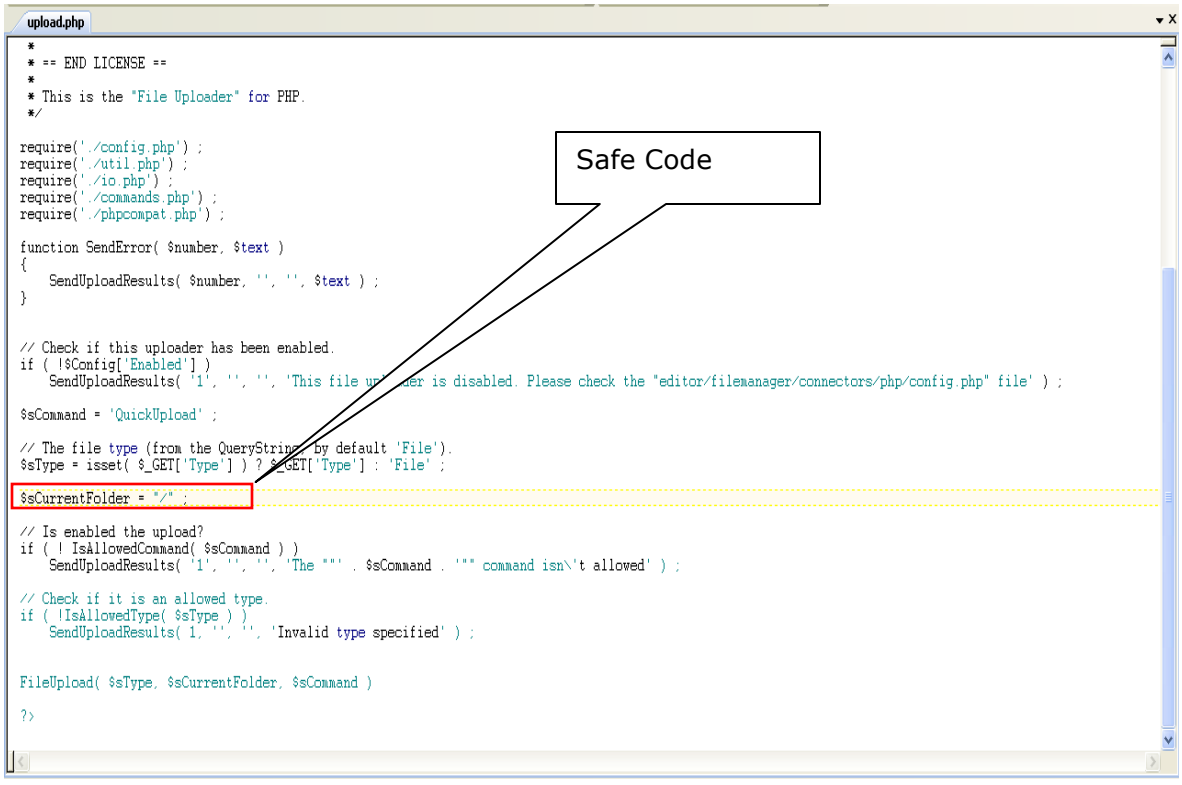
The impact of the above findings is **SEVERE** since attackers can upload malicious files, like web shells and completely compromise affected web servers.

Recommended Resolutions

These are some of the measures that should be taken to remove this vulnerability

1. Older versions of FCKEditor should be replaced with latest version of FCKEditor (CKEditor 3.6.6) to thwart the above vulnerabilities.
2. For FCKEditor of version 2.6.4 and less, the 'currentfolder' parameter can be disabled with the following code changes in the upload.php file:
   '**sCurrentFolder = GetCurrentFolder()'** should be replaced with **'sCurrentFolder = "/"'**


upload.php
Vulnerable Code
$sCurrentFolder = GetCurrentFolder() ;


upload.php
Safe Code
$sCurrentFolder = "/" ;

About The Authors

Anant Kochhar, Utkarsh Bhatt and Sabyasachi Samanta are Information Security Consultants and they have, between them, secured several web applications. They can be reached at anant.kochhar@secureyes.net, utkarsh.bhatt@secureyes.net and sabyasachi.samanta@secureyes.net respectively.

About SecurEyes

SecurEyes is a Bangalore based firm specializing in IT security. SecurEyes offers a wide range of security services and products to its clients. For more information, please visit our website: http://www.secureyes.net/.

Addendum*

While we discovered this vulnerability independently and during the course of our work, it has recently come to our attention that this vulnerability was already in the public realm. This was an honest oversight on our part and we did not intend to take the credit away from those who discovered and disclosed this vulnerability first. More information is available at: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-2265