security-assessment.com

White Paper

Title: 0x00 vs ASP file upload scripts

Prepared by: Brett Moore
Network Intrusion Specialist
Security-Assessment.com

Date: 13 July 2004

Copyright Security-Assessment.com Ltd 2004

Abstract

The affects of the `Poison NULL byte` have not been widely explored in ASP, but as with other languages the NULL byte can cause problems when ASP passes data to objects.

Many upload systems written in ASP suffer from a common problem whereby a NULL byte can be inserted into the filename parameter leading to any extension, after the null byte, being ignored when writing the file.

This means that in some cases it is possible to bypass checks for valid extensions, even if one is appended by the application.

This is very similar to attacks against perl and PHP, the difference being how the null byte is sent to the application.

This problem arises when data is compared and validated in ASP script but passed to the *FileSystemObject* without checking for NULL bytes.

This document will discuss how ASP upload scripts can be affected by the Poison NULL byte attack.

Scope

The information in this document is based on research done using upload systems that incorporate multipart/form-data posts and the Scripting.FileSystemObject object.

Throughout this document we focus on the CreateTextFile method, which is used to create a file for writing, but it is possible that other objects functions are vulnerable to the same type of problem.

A %00 or NULL can not be sent through the URL or a normal form post as the web server registers this as the end of the string, but does not store it in the filename variable.

When a filename is sent using a multipart/form-data post the null byte will be included in the filename variable, thus affecting calls to the FileSystemObject.

File Uploading

File uploading is commonly done using an input object of type *file* and an encoding type of *multipart/form-data.*

The content type "application/x-www-form-urlencoded" is inefficient for sending large quantities of binary data or text containing non-ASCII characters. The

Copyright Security-Assessment.com Ltd 2004

content type "multipart/form-data" should be used for submitting forms that contain files, non-ASCII data, and binary data.

A "multipart/form-data" message contains a series of parts, each representing a successful control. The parts are sent to the processing agent in the same order the corresponding controls appear in the document stream.

```
<form method=post enctype="multipart/form-data" action=upload.asp>
        Your Picture:<BR><input type=file name=YourFile><BR><BR>
        <input type=submit name=submit value="Upload">
</form>
```

When submitted the forms data will be sent in the multipart/form-data format. This allows for the transfer of all bytes, including nulls, within the forms posted data.

Upon receiving the post, the target ASP page needs to process and decode the posted data into a useable state.

File Saving

At some point in the uploading process, the file will be saved to a file location. The following is some commonly used code to do this.

```
Public Sub Save(filename)
        Dim objFSO, objFSOFile
        path=server.MapPath("/uploads/")
        Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
        Set objFSOFile = objFSO.CreateTextFile(path + "\" + filename)
        objFSOFile.Write <file contents>
        objFSOFile.Close
End Sub
```

When the filename parameter is passed to the CreateTextFile() function, it may contain NULL bytes. This can affect the name of the created file as the CreateTextFile only reads up to the NULL byte when creating the file.

```
Set objFSOFile = objFSO.CreateTextFile(path + "\" + filename)
```

If filename contains a NULL byte, anything after that byte will be ignored.

Copyright Security-Assessment.com Ltd 2004

Null Byte

The NULL byte can be inserted manually through modifications to the multipart post data using a hex editor, or by using a web proxy.

Multipart Form Post

```
POST /upload.asp HTTP/1.0
Content-Type: multipart/form-data; boundary=--------------------------
7d4cb161b009c
Host: localhost
Content-Length: 359
Pragma: no-cache
Cookie: ASPSESSIONIDSAADRCRS=LAKNNAKAGMIBJCOOLBIFEHIK

-----------------------------7d4cb161b009c
Content-Disposition: form-data; name="YourFile"; filename="c:\nc.exe .bmp"
Content-Type: text/plain

Proof Of Upload Test File
brett.moore@security-assessment.com
-----------------------------7d4cb161b009c
Content-Disposition: form-data; name="submit"

Upload
-----------------------------7d4cb161b009c
```

The filename parameter of the above post has been changed as such;

N	C	.	E	X	E	**(null)**	.	B	M	P
4E	43	2E	45	58	45	**00**	2E	42	4D	50

Note that an actual NULL byte (0x00) has been inserted between the .exe and the .bmp.

Copyright Security-Assessment.com Ltd 2004

Script Tests

The following two file save scripts shown below are examples of upload scripts where the extension of the written file can be arbitrarily set.

In both cases tFile is the filename parameter passed through the post.

Example One (File Extension Appending)

```
Public Sub Save(Path)
 Dim objFSO, objFSOFile
 Dim lngLoop

 Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
 Set objFSOFile =
       objFSO.CreateTextFile(objFSO.BuildPath(Path, tFile + ".bmp"))

 ‘ Write the file contents
 For lngLoop = 1 to LenB(m_Blob)
      objFSOFile.Write Chr(AscB(MidB(m_Blob, lngLoop, 1)))
 Next

 objFSOFile.Close
End Sub
```

Example Two (File Extension Checking)

```
Public Sub Save(Path)
 Dim objFSO, objFSOFile
 Dim lngLoop


 ‘ Check the file extension
 if right(tFile,4) <>  “.bmp”  then exit sub

 Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
 Set objFSOFile=
       objFSO.CreateTextFile(objFSO.BuildPath(Path, tFile))

 ‘ Write the file contents
 For lngLoop = 1 to LenB(m_Blob)
   objFSOFile.Write Chr(AscB(MidB(m_Blob, lngLoop, 1)))
 Next

 objFSOFile.Close
End Sub
```


Copyright Security-Assessment.com Ltd 2004

Final Summary

It has commonly been thought that web applications written in ASP are safe from the problems associated with NULL bytes. While in most instances this is true, it can be seen here that applications that make use of objects external to the native ASP scripting language, can be affected by NULL bytes.

It is probable that other objects and areas can also be manipulated to some extent when their data is collected through a multipart/form-data post.

As in other areas, proper validation of user input is paramount to the security of web applications. It is therefore important to check input not only for common attack strings used for folder traversal, but also for NULL bytes before using the input in the creation of files.

References

Perl CGI problems - rain.forest.puppy
http://www.phrack.org/show.php?p=55&a=7

Bugtraq Post Regarding PHP and null bytes
http://seclists.org/lists/bugtraq/2003/Jan/0159.html

OWASP HTML Version
http://www.cgisecurity.com/owasp/html/guide.html#id2846281

Forms in HTML documents
http://www.w3.org/TR/REC-html40/interact/forms.html#h-17.13.4

Security-Assessment.com
www.security-assessment.com

Copyright Security-Assessment.com Ltd 2004

About Security-Assessment.com

Security-Assessment.com is an established team of Information Security consultants specialising in providing high quality Information Security Assurance services to clients throughout the UK, Europe and Australasia. We provide independent advice, in-depth knowledge and high level technical expertise to clients who range from small businesses to some of the worlds largest companies

Using proven security principles and practices combined with leading software and proprietary solutions we work with our clients to provide simple and appropriate assurance solutions to Information security challenges that are easy to understand and use for their clients.

Security-Assessment.com provides security solutions that enable developers, government and enterprises to add strong security to their businesses, devices, networks and applications. We lead the market in on-line security compliance applications with the SA-ISO Security Compliance Management system which enables companies to ensure that they are effective and in line with accepted best practice for Information Security Management.

Copyright Information

These articles are free to view in electronic form, however, Security-Assessment.com and the publications that originally published these articles maintain their copyrights. You are entitled to copy or republish them or store them in your computer on the provisions that the document is not changed, edited, or altered in any form, and if stored on a local system, you must maintain the original copyrights and credits to the author(s), except where otherwise explicitly agreed by Security-Assessment.com Ltd.

Copyright Security-Assessment.com Ltd 2004