高校 WEB 站点的上传漏洞分析及防范

白兴瑞，刘耀炎
(龙岩学院 现代教育技术中心，福建 龙岩 364000)

摘 要：由于高校和高校院系部门的 WEB 代码大量使用未经严格检测的开源代码以及开发人员算法上的出错，使高校 WEB 站点的上传漏洞逐渐突出．通过对站点中的上传漏洞研究，分析了一些上传漏洞的原因，提出了相应的防范措施，能有效防止上传漏洞．

关键词：WEB；上传漏洞；网站安全
中图分类号：TP393 **文献标识码**：A **文章编号**：1673-2065(2011)04-0034-03

文件上传是高校 WEB 站点平台一个不可少的功能，因为其信息发布系统中一般有一个文件上传的模块．高校 WEB 站点的上传漏洞特别突出，其根源是由于经费和其它原因．高校院系部门的 WEB 站点代码大多是网上下载没有经过严格测试的开源代码，特别是文件上传部分的控制不足或者处理缺陷，导致非法用户可以越过其本身权限向服务器上传可执行的动态脚本文件．上传漏洞本身的攻击性不强，但如果上传漏洞与 webshell 结合起来，那么这个网站服务器就成肉鸡了；另外高校院系部门的信息发布系统重复利用这些带有漏洞的源代码使问题更严重，攻击者对高校 WEB 站点的攻击也往往从上传漏洞开始的．

实现文件数据的上传保存，一般有 2 种方法：一是将被上传的文件当作记录形式，将其保存进入 WEB 站点的数据库中．由于是作为记录形式存放在数据库中，增大数据库文件的容量，造成 WEB 服务器的数据负担；二是采用文件方式保存，利用编程语言中的一些文件函数或系统控件实现文件在 WEB 服务器上的存储．这种方法不会增加数据库负但，而对上传文件格式、大小通常也没有限制(asp.net 中的文件上传大小可以通过 web.config 文件来自定义其大小)，因此在高校 WEB 站点的上传文件大多采用第二种方法．

1 上传漏洞分析

著名的安全网站 security-assessment.com 上安全研究员 Brett Moore 发表了一篇《0x00 vs asp file uploads》论文，论文指出在 ASP 编写的代码中，这些 ASP 程序充许将 Null Byte 作为文件名称的，这样会导至 NULL Byte 后面的字符不被读取到，在某些性况下会避开程序对上传文件类型的限制；而对于 PHP 及 Perl 语言开发的程序也都存在这样避开程序对上传文件类型的限制．Null Byte 是处理字符串时判断字符串结束的标志，在 Windows 系统中，Null Byte 是 0x00，在 Brett Moore 论文中解释了如何引发上传漏洞：ASP 程序中，文件上传是利用 Form 表单进行的，当 input 对象的属性为 file 并且数据编码方式为 multipart/form-data 时，就可以进行文件上传，这种编码方式可以将二进制、甚至那些非 ASCII 码范围的数据上传到网站上，因此也可将 NULL Byet 上传递给程序．对于上传漏洞，其攻击目标是文件路径或者是文件类型，其最终目标上传非法代码文件．

1.1 文件路径

先分析以下代码：

```
<% formPath=upload.form("filepath")
if right(formPath,1)<> "/" then formPath=formPath&"/"
filename=formPath&year(now)&month(now)&day(now)&hour(now)&minute(now)&second(now)&ranNum&
fileExt%>
```

在第一句代码中，从变量 filepath 中获取文件的保存路径，然后在第二句中，用路径变量 formPath 加随机生成的数字及经过判断的扩展名合成为一个新的变量，这个变量 Filename 就是上传文件保存的路径及名称．如选择“111.jpg”文件上传，在上传过程中，随文件一起上传的还有一个 FilePath 变量，假设其值为

收稿日期：2010-12-25
作者简介：白兴瑞(1971-)，男，福建龙岩人，龙岩学院现代教育技术中心网络部实验师；
刘耀炎(1979-)，男，福建龙岩人，龙岩学院现代教育技术中心技术部助理实验师．

“image”，当这些值传到 upfile.asp 中，filename 就变成了：“image/20101130321944973.jpg”，上传成功后，该 111.jpg 就被保存到 image 文件夹内，文件名字也被改成了：“20101130321944973.jpg”．如果将其 FilePath 值改为“image/aa.asp0x00”，由于 asp 语言认为 null byte 代表字符串结束，服务器在读取这段变量时，null byte 后面的字符也就被忽略掉了，这样一来，Filename 就成了：“image/aa.asp”，程序再用 file.SaveAs 进行保存的话，这个文件就保存成了 aa.asp 文件．这对于 web 程序通常是不允许的，因为这个 asp 代码如果是一个 webshell 代码，这可以成功上传 webshell 的权限．对于文件路径，可以通过把文件路径变量改为常量的方式来避免上述漏洞：

if info_name=“bbs” then FilePath = “/bbs/upload/”　　else FilePath = “/uploadpic/”　　end if

1.2　文件类型

在网络上提供的开源代码中，有些考虑到实际使用及网站服务器安全，对于文件上传作了只允许上传某些类型的文件，对符合要求的，WEB 应用程序就会让上传文件在服务器上保存，但对于 Null Byte 上传漏洞却无法防止．下面是一段通用的上传文件代码：

```
If right(filename,4) <> “.jpg”  then exit
Path=server.mappath(“/uploads/”)
Set objf=server.createobject(“scripting.filesystemobject”)
Set objfile=objf.createtextfile(path + “\” + filename)
Objfile.write <file contents>
Objfile.close
```

如果上传的文件不是 jpg 则认为上传文件格式非法，退出不再执行程序；如果合法，就将 filename 与 path 组合成一个文件名，上传到服务器保存．这段程序看似没问题，但如果上传的文件名是 example.asp0x00.jpg 呢？由于 asp 语言认为 null byte 代表字符串结束，这时 filename 与 path 组合成的文件就变成“path/example.asp”，这对于 web 程序也是不允许的．

2　发掘上传漏洞

如何来发掘上传漏洞呢？如果无法查看源代码的情况下，可以通过重新构造带 null byte 文件名的数据包来发掘．首先利用 winsock expert 截获网站正常流程方式下上传文件的数据包．winsock expert 会对程序自动记录下来，可以获得一个正常的文件上传数据包．把截取的数据包复制到一个文本文件：test.txt，因为 null byte 是不可见字符，所以这里采用 winhex 打开该文件，找到其完整的上传文件名称．上传漏洞是由于文件的完整文件名造成的，所以这时把这个正常上传的文件完整名修改成一个“非法”的测试语句，比如把“example.txt”文件名中的 “ampl”改成“.asp”，“e”改成“0x00”，采用替换的方式可以避免再去修改数据包大小的 content-length 属性．这样就把“example.txt”文件名改成“ex.asp.txt”文件了．接着利用 nc 把这修改后的数据包文件提交给上传程序．如：nc.exe　http://xxx.xxx.xx 80<test.txt，如果在目标地址上创建了上传的文件或存在该文件的 url 地址，则存在上传漏洞．这种在发掘上传漏洞时，根据上传涵洞的原理，截获正常上传文件时的数据包，分析里面所有可能被利用的变量时，还可以测试修改文件名、文件路径等方面来发掘上传漏洞．

如果可以查看脚本代码，那么通过耐心的分析找漏洞，不仅能找到因为 Null Byte 产生的上传文件漏洞，而且还可以找到算法上出错的漏洞．例如 WEB 网站通常会对上传的文件类型作限制，通过文件的扩展名来防止一些木马代码的上传．通过对文件后缀名的判断，这策略本身是安全的，但如果算法出了问题呢？如代码：If instr(filename , “asp”) then replace (filename, “asp”, “ ”)，如果攻击者把上传的文件扩展名改为“.aspasp”，那么上述的代码就会把其文件扩展名改为“.asp”，这容易上传木马代码了，上传漏洞也就产生了．另外，也可以通过分析进行上传文件的表单网页，查看其源代码中 input 标签的名字，input 标签中的属性控制用户传递数据给 web 应用程序，有时候关健的变量会被表单的 hidden 属性隐藏．

3　上传漏洞的防范方法

针对 Null Byte 漏洞，可以通过以下代码完成：

```
function Trueimg(filetrue)
    str_len=len(filename)
    pos=Instr(filename,chr(0))
```

```
    if pos=0 or pos=str_len then   Trueimg=true   else   Trueimg=false end if
end function
if Trueimg(filename)=false then    response.write "非法文件"
  response.end   end if
file.SaveAs Server.mappath(filename)
```

对于web的上传漏洞，最好的策略仍然是对上传文件类型的过滤限制，上述的上传漏洞是因为算法上的出错，其实如果换一个角度的算法：在保存上传文件时，只充许合法的文件名保存，其它文件都禁止，这样也可以成功地防止非法文件类型的上传发生，具体代码如下：

```
If fileextend = "bmp" then upload = allrow
Else if fileextend = "jpg" then upload = allrow
Else if fileextend = "gif" then upload = allrow
Else if fileextend = "png" then upload = allrow
Else upload = forbidden    End if
```

通过文件的扩展名限制了脚本代码的上传，但如果将上传的ASP文件后缀改为JPG或其它允许的类型，仍然是可以上传的，如何来识别上传的不是非法的代码呢？一种是判断通过解析二进制文件，判断文件头是否是图片标志可能；二是通过代码本身的行为来控制，如以下代码：

```
sFile=server.mappath(FileName)
set MyFile=server.CreateObject("Scripting.FileSystemObject")
set MyText=MyFile.OpenTextFile(sFile, 1)
sTextAll=lcase(MyText.ReadAll)
MyText.close
```

判断用户文件中的危险操作

```
sStr=".getfolder .createfolder .deletefolder .createdirectory .deletedirectory .saveas wscript.shell script.encode. 重命名 修改 属性 文件浏览器 新建 复制 成功 参数错误 服务器 空间 下载"
sNoString=split(sStr, " ")
for n=0 to ubound(sNoString)
  if instr(sTextAll,sNoString(n))   then   set filedel=server.CreateObject ("Scripting.FileSystemObject")
   filedel.deletefile server.mappath(FileName)
Response.end    end if    next
```

对于通过更改文件的扩展名来绕过文件类型的检测，还有一种更有效的方法．文件在创建时，文件的前8个字节内容是确定的，因此，通过检测文件的前面字节就可以确定文件类型，如asp的前面字节是6 037(10进制)，exe文件的前面字节是7 790(10进制)从而判断是否在允许之列．

上传漏洞对高校的WEB网站具有很强的攻击力，因为这种漏洞可以允许攻击者上传任意文件到服务器上，甚至包括webshell这样的木马程序也能成功被上传．根据笔者从事WEB安全的实践经验，提出了一些相应的防范措施．另外，如果从服务器的架构设置方面，还可以对上传的文件存放文件夹设置读、写权限，取消执行权限，从权限上禁止上传的恶意代码运行．

Analysis and Prevention of Upload Vulnerability of the WEB in University

BAI Xing-rui, LIU Yao-yan
(Center for Modern Education Technology, Longyan College, Longyan, Fujian 364000, China)

Abstract: Due to the extensive use of the non-strictly examined open-source codes in university and the algorithm mistakes made by the developers, the upload vulnerability of the WEB in university is increasing. Through the research on the upload vulnerability of the WEB, this paper analyzes the reasons of it and then puts forward some prevention measures, which can prevent the upload vulnerability effectively.

Key words: WEB; upload vulnerability; security of web

(责任编校：李建明　　英文校对：李玉玲)